

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“Implementación de algoritmos de planificación
y control en la plataforma Segway RMP-100 ”**

POR

Alfredo Delgado Spíndola

TESIS

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL
GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

DIRECTOR DE TESIS

Dr. Ricardo Emmanuel Campa Cocom

CODIRECTOR DE TESIS

Dr. Eusebio Bugarín Carlos

ISSN: 0188-9060



RIITEC: (06)-TMCIE-2016

Torreón, Coahuila, México
Junio 2016

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de La Laguna

Torreón, Coah., 6/Junio/2016
Dependencia: DEP./CPCIE
Oficio: DEPI/CPCE/136/2016
Asunto: Autorización de impresión
de tesis.

C. Alfredo Delgado Spíndola
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.
PRESENTE

Después de haber sometido a revisión su trabajo de tesis titulado:

**"Implementación de algoritmos de planificación y control en la plataforma
Segway RMP-100"**

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro RIITEC: 1061-TMCIE-2016, para que proceda a la impresión del mismo.

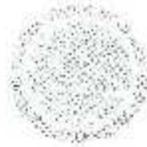
ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

DR. JOSÉ LUIS MEZA MEDINA
Jefe de la División de Estudios de Posgrado e Investigación
del Instituto Tecnológico de la Laguna

SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
de la Laguna
División de Estudios de Posgrado
e Investigación

JLMV/RFSR





Torrón, Coah., 16/Junio/2016

DR. JOSE LUIS MEZA MEDINA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:

**"Implementación de algoritmos de planificación y control en la plataforma
Segway RMP-100"**

Desarrollado por el **C. Alfredo Delgado Spindola**, con número de control **M1413005** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

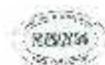
ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN


Dr. Ricardo E. Campa Cocom
Asesor/Director de Tesis


Dr. Eusebio Bugarin Carlos
Codirector de Tesis


Dr. Alfonso Pámanes García
Comité Tutorial


Dr. Alejandro E. Dzul López
Comité Tutorial



Dedicada a mi familia, y principalmente a mis padres, que me brindaron todo su apoyo.

Agradecimientos

Al CONACYT y al Instituto Tecnológico de la Laguna, por los apoyos dados para que yo pudiera realizar estos estudios.

A los revisores miembros del Comité de Tesis, **Dr. Eusebio Bugarín**, **Dr. Alejandro Dzul** y **Dr. Alfonso Pámanes**, por ayudarme a mejorar el presente trabajo.

A todos mis compañeros del posgrado, los cuales hicieron más amenas las muchas horas pasadas en el laboratorio, y con los cuales además pude contar cuando no entendía algún tema; principalmente a mis compañeros de generación **Marco Rodríguez**, **Isaac Gandarilla** y **Ricardo Ovalle**, quienes con su apoyo y su forma de ser me ayudaron a dar lo mejor de mí durante este tiempo; sobre todo, gracias por ser mis nuevos amigos.

A mi asesor de tesis, **Dr. Ricardo Campa**, quien con su forma de trabajar y con su admirable ejemplo ayudó a que yo pudiera obtener mejores resultados, los cuales ni yo mismo esperé.

A mis amigos, los cuales tal vez nunca lean esto, pero sin los cuales no hubiera podido aguantar el estrés de estar en un posgrado; gracias amigos.

A mis hermanos, para los cuales intento ser el mejor ejemplo posible y me ayudan a ser una mejor persona; gracias por aguantarme.

Y sobre todo a mis padres, **Norma R. Spíndola Félix** y **Ramón A. Delgado González**, sin los cuales simplemente yo no sería nada; gracias por darme todo en cuanto han podido y hacerme lo que soy hoy; esto es para ustedes.

Implementación de algoritmos de planificación y control en la plataforma Segway RMP-100

Alfredo Delgado Spíndola

Resumen

La plataforma robótica móvil RMP-100 de Segway Robotics es una plataforma de transporte diseñada para su integración en sistemas que contengan unidades de control. La unidad de control deberá generar comandos de velocidad lineal y angular adecuados para satisfacer una cierta tarea previamente establecida. Una característica relevante de la RMP-100 es que cuenta únicamente con dos ruedas coaxiales con actuadores independientes, de manera que constituyen un robot móvil diferencial o tipo uniceño y, para mantener el equilibrio, la plataforma funciona como un péndulo invertido sobre ruedas. La plataforma cuenta con dos modos de operación: el modo de autobalanceo y el modo de tractor; en el modo de autobalanceo, como su nombre lo dice, la plataforma usa un control interno para mantener el péndulo balanceado en su configuración vertical; en el modo de tractor, por otro lado, no se aplica ningún control al péndulo, por lo que se recomienda el uso de un apoyo extra al usar este modo, para evitar que el robot caiga. El presente trabajo se centra primeramente en el modelado de la plataforma, y diferentes métodos de modelado cinemático/dinámico son estudiados y aplicados. Posteriormente se presentan algunos controladores que se pueden aplicar a plataformas que presentan este tipo de características. Previo a la implementación de dichos controladores, se desarrolló una nueva interfaz de software para el control en tiempo real de la plataforma, lográndose con esto un mejor desempeño. Al final, se implementaron los controladores estudiados, empleando la interfaz desarrollada, lo cual permitió evaluar tanto el desempeño de los controladores, como el de la misma interfaz.

Implementation of planning and control algorithms in the Segway RMP-100 platform

Alfredo Delgado Spíndola

Abstract

The robotic mobility platform RMP-100 from Segway Robotics is a transport platform designed for its integration into systems that contain control units. The control unit must generate the linear and angular velocity commands required to meet a certain task previously defined. One of the most relevant features of the RMP-100 platform is that it has only two coaxial wheels, independently actuated, so that it constitutes a differential or unicycle-like mobile robot, and in order to keep its equilibrium, the platform works as a wheeled inverted pendulum. The RMP-100 platform has two operation modes: the self-balance mode and the tractor mode. In the self-balance mode the platform uses an inner controller to keep the pendulum balanced in its vertical configuration; the tractor mode does not apply any controller to the pendulum, so the use of an extra support when using this mode is recommended to prevent the robot from falling. This work first focuses on the modeling of the platform and different kinematics/dynamics modeling methods are studied and applied. After that some controllers which can be applied to this kind of platforms are presented. In order to implement these controllers, a new software interface for real-time control was developed. At the end, the studied controllers were implemented using the developed interface, thus allowing to evaluate the performance of both the controllers and the interface.

Índice general

1. Introducción	1
1.1. Conceptos básicos	1
1.2. Sistemas robóticos	4
1.3. Robótica móvil	6
1.3.1. Robots móviles con ruedas	8
1.4. Péndulos subactuados	12
1.5. Objetivos de la tesis	15
1.6. Organización del documento	15
2. Modelado de los RMR	17
2.1. Fundamentos	18
2.1.1. Tipos de ruedas	19
2.1.2. Matrices jacobianas de las ruedas	22
2.1.3. Clasificación de los RMR	24
2.2. Modelado cinemático de velocidad	28
2.2.1. Método de Muir-Neuman	30
2.2.2. Método de Campion <i>et al.</i>	32
2.3. Modelado dinámico	34
2.3.1. Método de Euler-Lagrange	35
2.3.2. Método de Kane	37
3. Planificación y control	40
3.1. Planificación de movimientos de un RMR	41
3.1.1. Planificación bajo restricciones diferenciales	42
3.1.2. Planificación simple	42

3.2. Control de movimiento de un RMR	43
3.2.1. El problema de estabilización en RMR	45
3.2.2. Control de regulación	46
3.2.3. Control de seguimiento	49
3.3. Control de balanceo de un PIR	50
3.3.1. El controlador LQR	50
4. La plataforma Segway RMP-100	52
4.1. Descripción y datos técnicos	53
4.2. Arquitectura de control	55
4.2.1. Actuadores y sensores	56
4.2.2. Estructura de control jerárquica	58
4.3. Modelado de la plataforma	59
4.3.1. Modelo cinemático de velocidad	60
4.3.2. Modelo dinámico	61
5. Interfaz de software	64
5.1. Interfaz original	64
5.2. Control en tiempo real en Windows	65
5.2.1. RTX: Una extensión de tiempo real para MS-Windows	66
5.2.2. Programación con RTX en MS-Visual Studio	68
5.3. Comunicación vía bus CAN	70
5.3.1. Protocolo de comunicación bus CAN	70
5.3.2. Programación de la tarjeta CAN-AC2-PCI de Softing	72
5.3.3. Mensajes CAN de la RMP-100	73
5.4. Interfaz desarrollada	78
6. Implementación de algoritmos en modo de auto balanceo	81
6.1. Regulación de postura	82
6.1.1. Controlador de Oriolo <i>et al.</i>	82
6.1.2. Controlador de Tanner y Kyriakopoulos	88
6.2. Control de seguimiento	95
6.2.1. Controlador de Oriolo <i>et al.</i>	95

7. Implementación de algoritmos en modo de tractor	101
7.1. Control LQR	102
7.1.1. Control de regulación	103
7.1.2. Control de seguimiento	112
8. Conclusiones	119
8.1. Aportaciones de la tesis	119
8.2. Trabajo futuro	120
APÉNDICES	126
A. RMP-100: Modelado cinemático de velocidad	126
A.1. Método de Muir-Neuman	126
A.1.1. Modelo cinemático de velocidad en espacio de postura	126
A.1.2. Modelo cinemático de velocidad en espacio de configuración	129
A.2. Método de Campion <i>et al.</i>	130
A.2.1. Modelo cinemático de velocidad de en espacio de postura	131
A.2.2. Modelo cinemático de velocidad en espacio de configuración	132
B. RMP-100: Modelado dinámico	133
B.1. Método de Euler-Lagrange	134
B.2. Método de Kane	143

Índice de figuras

1.1.	Diagrama de bloques de un sistema robótico.	5
1.2.	Centro instantáneo de rotación de un robot móvil con ruedas.	10
1.3.	Diagrama de un RMR diferencial (vista superior).	11
1.4.	Diferentes tipos de péndulos invertidos: (a) carro-péndulo; (b) péndulo de Füzuta; (c) péndulo invertido esférico; (d) péndulo con rueda inercial; (e) acrobot; (f) pendubot; (g) péndulo invertido sobre ruedas.	13
2.1.	Marcos coordinados en un RMR: (a) marco inercial y marco del robot; (b) asignación de marcos en las ruedas.	18
2.2.	Rueda convencional: (a) vista lateral; (b) vista frontal; (c) vista superior; (d) fotografía; (e) rueda de castor.	21
2.3.	Rueda omnidireccional o succa: (a) vista lateral; (b) vista superior; (c) fotografía.	22
3.1.	Estructura jerárquica de control de un RMR.	44
4.1.	Segway RMP-100: (a) fotografía del robot; (b) diagrama cinemático.	52
4.2.	Unidad de control: (a) interfaz externa; (b) interior.	53
4.3.	Botones de la interfaz: (a) inicio, (b) activación de motores, (c) modo de tractor y (d) modo de auto balanceo.	55
4.4.	Fotografía del motor de la plataforma.	57
4.5.	Fotografía de la tarjeta de manejadores de la plataforma.	57
4.6.	Sensores de orientación.	58
4.7.	Diagrama de bloques de la estructura de control de la plataforma.	58
4.8.	Diagrama de comunicación/control de la plataforma.	59
5.1.	Interfaz original.	65

5.2. Arquitectura de MS-Windows con RTX.	67
5.3. Programación de procesos WIN32.	69
5.4. Programación de procesos RTSS.	70
5.5. Conector CAN definido por CIA (DS-102-1).	71
5.6. Contenido de un mensaje CAN2.0a.	72
5.7. Comunicación entre procesos de la nueva interfaz para el Segway RMP-100.	79
5.8. Interfaz inicial (derecha) y final (izquierda).	80
6.1. Regulador de Oriolo <i>et al.</i> : Acciones de control (simulación).	83
6.2. Regulador de Oriolo <i>et al.</i> : Variables de postura (simulación).	84
6.3. Regulador de Oriolo <i>et al.</i> : Trayectoria recorrida (simulación).	85
6.4. Regulador de Oriolo <i>et al.</i> : Acciones de control (experimento).	86
6.5. Regulador de Oriolo <i>et al.</i> : Variables de postura (experimento).	87
6.6. Regulador de Oriolo <i>et al.</i> : Trayectoria recorrida (experimento).	88
6.7. Regulador de Tanner-Kyriakopoulos: Acciones de control (simulación).	89
6.8. Regulador de Tanner-Kyriakopoulos: Variables de postura (simulación).	90
6.9. Regulador de Tanner-Kyriakopoulos: Trayectoria recorrida (simulación).	91
6.10. Regulador de Tanner-Kyriakopoulos: Acciones de control (experimento).	92
6.11. Regulador de Tanner-Kyriakopoulos: Variables de postura (experimento).	93
6.12. Regulador de Tanner-Kyriakopoulos: Trayectoria recorrida (experimento).	94
6.13. Movimiento de la plataforma hasta el origen.	94
6.14. Seguidor de Oriolo <i>et al.</i> : Acciones de control (simulación).	96
6.15. Seguidor de Oriolo <i>et al.</i> : Errores de postura (simulación).	97
6.16. Seguidor de Oriolo <i>et al.</i> : Trayectoria recorrida (simulación).	97
6.17. Seguidor de Oriolo <i>et al.</i> : Acciones de control (experimento).	98
6.18. Seguidor de Oriolo <i>et al.</i> : Errores de postura (experimento).	99
6.19. Seguidor de Oriolo <i>et al.</i> : Trayectoria recorrida (experimento).	99
6.20. Movimiento de la plataforma a lo largo de la trayectoria.	100
7.1. Regulador LQR: Acciones de control (simulación).	105
7.2. Regulador LQR: Variables de postura (simulación).	106
7.3. Regulador LQR: Acciones de control (experimento 1).	108
7.4. Regulador LQR: Variables de postura (experimento 1).	109
7.5. Regulador LQR: Acciones de control (experimento 2).	110
7.6. Regulador LQR: Variables de postura (experimento 2).	111

7.7. Seguidor LQR: Acciones de control (simulación).	113
7.8. Seguidor LQR: Variables de postura (simulación).	114
7.9. Seguidor LQR: Trayectoria recorrida (simulación).	115
7.10. Seguidor LQR: Acciones de control (experimento).	116
7.11. Seguidor LQR: Variables de postura (experimento).	117
7.12. Seguidor LQR: Trayectoria recorrida (experimento).	118
A.1. Ubicación de los marcos coordenados en la plataforma Segway RMP-100	127
B.1. Diagrama esquemático.	133

Índice de tablas

2.2.	Definición de la matriz W y el vector u para diferentes tipos de RMR.	33
4.1.	Tabla de especificaciones.	56
4.2.	Parámetros del modelo dinámico de la plataforma Segway RMP-100.	63
5.1.	Distribución de pines del conector del bus CAN.	71
5.2.	Comandos de configuración.	75
5.3.	Factor de conversión de las variables enviadas por la plataforma.	77
5.4.	Comparación entre la interfaz original y la desarrollada.	80

1 Introducción

El presente trabajo de tesis se dedica a la implementación de ciertos algoritmos representativos de planificación y control en la plataforma robótica móvil (modelo RMP-100 de Segway Robotics) que se encuentra en el Laboratorio de Mecatrónica y Control de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de la Laguna. Esta plataforma presenta varias características que la hacen interesante desde el punto de vista del control, las cuales se mencionan a detalle en el capítulo 4. En este capítulo, en la sección 1.1, se comenzará definiendo algunos conceptos básicos utilizados en robótica; en la sección 1.2 se presentan los tipos de controladores que se abordarán en el presente trabajo; las secciones 1.3 y 1.4 revisan las características de los robots móviles con ruedas y los péndulos subactuados, respectivamente. En la sección 1.5 se definen los objetivos del presente trabajo, y por último, en la sección 1.6 se describe la organización del resto del documento.

1.1. Conceptos básicos

La mecánica es la rama de la ciencia que tiene que ver con el comportamiento de los cuerpos físicos cuando están sujetos a fuerzas y/o desplazamientos. Un cuerpo físico (u objeto físico) es una colección de materia con un contorno identificable que puede moverse en un espacio tridimensional. En particular un cuerpo físico rígido (o simplemente cuerpo rígido) es un objeto sólido no deformable de modo que al moverse en el espacio solo puede presentar movimientos de traslación y/o rotación.

Por otra parte se le llama *configuración* a la ubicación en el espacio tridimensional de los elementos de un sistema mecánico, y *espacio de configuración* al conjunto de todas las posibles configuraciones que puede tener tal sistema. Además, se les llama *coordenadas generalizadas* a los parámetros que describen a un sistema mecánico respecto a alguna

configuración de referencia. Si se tienen m coordenadas generalizadas $\rho_1, \rho_2, \dots, \rho_m$ es posible agruparlas en un vector de forma que

$$\boldsymbol{\rho} = [\rho_1 \quad \rho_2 \quad \dots \quad \rho_m]^T \in \mathbb{R}^m.$$

Y si se toma la derivada con respecto al tiempo de cada coordenada generalizada entonces

$$\dot{\boldsymbol{\rho}} = [\dot{\rho}_1 \quad \dot{\rho}_2 \quad \dots \quad \dot{\rho}_m]^T \in \mathbb{R}^m$$

es el vector de velocidades generalizadas donde $\dot{\rho}_i$ es la i -ésima velocidad generalizada.

Es importante señalar que las m coordenadas generalizadas no necesariamente tienen que ser independientes, en el sentido en el que se pueden establecer relaciones entre ellas y/o sus derivadas.

Sea n el número mínimo de coordenadas necesarias para definir de manera única la configuración de un sistema mecánico, entonces $n \leq m$ y se puede definir un conjunto de n coordenadas generalizadas independientes, que es comúnmente denotado como q_1, q_2, \dots, q_n o en forma de vector:

$$\mathbf{q} = [q_1 \quad q_2 \quad \dots \quad q_n]^T \in \mathbb{R}^n$$

Y aunque pueden emplearse diferentes conjuntos de coordenadas mínimas el número n es constante y se le conoce como el *número de grados de libertad* (g.d.l) del sistema.

Se le llama sistema mecánico restringido (o con restricciones) es aquel que es modelado utilizando coordenadas generalizadas no mínimas, las cuales están sujetas a restricciones que pueden ser holonómicas o no holonómicas.

Un sistema mecánico con restricciones holonómicas es aquel que es modelado m coordenadas generalizadas no mínimas que están sujetas a $\nu = m - n$ restricciones de la forma

$$\lambda_i(\boldsymbol{\rho}) = 0, \quad i = 1, 2, \dots, \nu. \quad (1.1)$$

o en forma de vector

$$\boldsymbol{\lambda}(\boldsymbol{\rho}) = \mathbf{0} \in \mathbb{R}^\nu$$

Se debe hacer notar que cualquier restricción holonómica al ser derivada implica también una restricción en términos de las velocidades generalizadas, dadas por el vector $\dot{\lambda}$, esto es

$$\dot{\lambda}(\rho, \dot{\rho}) = \dot{c} \in \mathbb{R}^r$$

Las restricciones holonómicas se presentan siempre en robots de cadena cerrada, como los robots paralelos.

Por otra parte una restricción no holonómica, es una restricción en la que intervienen las velocidades generalizadas, pero de la cual no se puede extraer una restricción que dependa solo de las coordenadas generalizadas. En otras palabras, si

$$\mu_i(\rho, \dot{\rho}) = 0, \quad i = 1, 2, \dots, \nu, \quad (1.2)$$

es un conjunto de restricciones no holonómicas significa que tales restricciones no pueden ser integradas para obtener restricciones de la forma (1.1). En otras palabras, una restricción no holonómica no reduce el espacio de configuración del sistema, sino que sólo reduce el espacio de velocidades instantáneas. Este tipo de restricciones aparecen en sistemas mecánicos con puntos de contacto de rodamiento (ruedas) o puntos de contacto de deslizamiento (Bloch, 2003). Lo que se traduce en que los robots móviles con ruedas tienen este tipo de restricciones.

Se define como *vector de variables de estado* z al conjunto de variables de un sistema que pueden representar el estado completo del sistema en un determinado instante (Palm, 2013). En un sistema mecánico comúnmente se seleccionan las coordenadas generalizadas y las velocidades generalizadas como variables de estado, es decir

$$z = \begin{bmatrix} \rho \\ \dot{\rho} \end{bmatrix} \in \mathbb{R}^{2m}$$

Se le llama *postura* de un cuerpo rígido a la combinación de la posición y la orientación del mismo en el espacio. La parte de posición queda generalmente descrita por medio de un vector $p \in \mathbb{R}^3$; para la parte de orientación no existe una representación única debido a que la orientación pertenece a una variedad de dimensión 3 y existen varias parametrizaciones de la misma tales como ángulos de Euler o matrices de rotación (Caupa y de la Torre, 2009).

Así que en el caso general de movimiento en 3D, la dimensión del espacio de configuración de la postura es seis, aunque si el movimiento del cuerpo se restringe a un plano (2D) entonces la dimensión del espacio de postura se reduce a tres, ya que para la posición se requieren solo dos coordenadas (generalmente X y Y) mientras que para la orientación se requiere conocer sólo un ángulo (θ) que da la rotación con respecto a un eje perpendicular al plano de movimiento.

1.2. Sistemas robóticos

En la década de 1980 la robótica fue definida como la ciencia que estudia la conexión inteligente entre la percepción y la acción (Siciliano y Khatib, 2008). El Instituto de Robótica de América define robot industrial como: "manipulador programable, multifuncional y diseñado para mover material, herramientas o dispositivos especializados a través de movimientos variables, programados para desarrollar diversas tareas".

Cabe mencionar que algunos robots se encuentran fijos a un área de trabajo con al menos un eslabón fijo (la base) al mundo; tal es el caso de los robots manipuladores seriales y paralelos cuya base está fija en algún lugar del ambiente, espacio de trabajo y son los que se encuentran normalmente en la industria; por otro lado, los robots que son capaces de moverse en su ambiente (aire, agua o por una superficie), cambiando la posición de todo el cuerpo en el ambiente, se conocen en la literatura como *robots móviles*.

En el caso particular de los robots manipuladores seriales, el número de articulaciones es igual al número de grados de libertad. Entonces, en estos casos se puede decir que se requieren 6 grados de libertad para describir la postura del efector final de un robot en un espacio tridimensional, utilizando 3 grados de libertad para la posición y 3 grados de libertad para la orientación. Existen casos en los que los sistemas cuentan con más de 6 g.d.l., estos tipos de sistemas son capaces de modificar su configuración sin cambiar la postura del efector final, a estos tipos de sistemas se lo conoce como robots redundantes.

Un sistema robótico (ver figura 1.1) consta de los elementos indispensables para cumplir con una tarea especificada, los cuales pueden ser agrupados así:

- *Planta:* Es el sistema mecánico que se encarga de realizar la tarea especificada.
- *Planificador de tareas:* Se encarga de generar un vector de señales de referencia deseadas ρ_d que deberá seguir la planta para realizar una tarea específica. Esta

señal de referencia puede ser fija (constante) o variable (una función del tiempo), dependiendo de la tarea a realizar.

- *Controlador:* Se encarga de comparar el vector de coordenadas generalizadas ρ (las cuales son obtenidas de la planta a través de sensores u observadores) con el vector ρ_d (la salida del planificador), para generar acciones de control (definidas por el vector u) que serán la entrada a los actuadores. Estas acciones de control modificarán el estado de la planta, para lograr que la salida de la planta sea igual a la referencia deseada, es decir, $\rho = \rho_d$. Estas acciones del controlador pueden ser de diferente tipo y dependerán del tipo de actuadores con los que cuente la planta.
- *Sensores:* Estos se encargan de monitorear las variables de interés en el sistema (como el desplazamiento en cada una de las articulaciones del robot), así como del ambiente en el que se mueve (la ubicación del robot en el ambiente). Las lecturas de los sensores pueden ser enviadas al controlador para que cambie la configuración del sistema con el fin de completar cierta tarea o pueden ser enviados al planificador con el fin de generar nuevas señales de referencia.
- *Actuadores:* Estos recibirán las señales del controlador, para producir los movimientos en la planta que ayuden a cumplir con la tarea especificada.

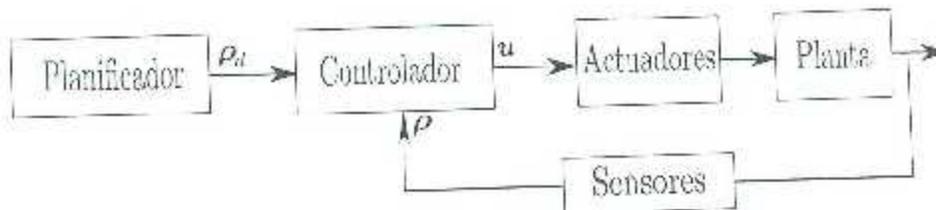


Figura 1.1: Diagrama de bloques de un sistema robótico.

En el presente trabajo se da por hecho que los bloques de sensores y actuadores funcionan de forma correcta, por lo que los problemas a resolver ahora son: la planificación de tareas, el estudio de algoritmos de control y el estudio de la planta.

Debido a que la mayoría de los métodos para planificación y control de robots necesitan información de la planta para poderlos aplicar, entonces el primer paso consiste en identificar los modelos matemáticos de la planta a utilizar. Los modelos que son necesarios son: el modelo cinemático y el modelo dinámico.

La *cinemática* es la rama de la física que estudia el movimiento de los cuerpos sin tomar en cuenta las fuerzas que lo causan. La cinemática describe la geometría

del sistema y las relaciones iniciales que dan los valores de la posición, velocidad y/o aceleración de los puntos de importancia del sistema. La *dinámica*, por otro lado, es la rama de la física que describe la evolución en el tiempo de un sistema físico, tomando en cuenta los movimientos en el sistema y las fuerzas que los producen.

Una vez obtenido el modelo requerido para el diseño del controlador, es necesario definir qué tipo de controlador se va a diseñar. Basados en Kelly y Santibáñez (2003), la manera más sencilla de especificar el movimiento de un robot es la denominada punto-a-punto. Esta metodología consiste en determinar una serie de puntos en el espacio de trabajo del robot por donde se desea que éste o el extremo final pase, con el fin de satisfacer cierta aplicación. El problema de control consiste, en resumen, en hacer pasar el extremo del manipulador (en el caso de robots industriales) o el mismo robot (en el caso de móviles) por dichos puntos.

Una forma más general para especificar el movimiento de un manipulador es la llamada planificación de trayectorias. En este caso, se determina una curva o trayectoria parametrizada temporalmente en el espacio de trabajo y el problema de control consiste en hacer pasar ya sea el extremo del manipulador o el robot en sí por dicha trayectoria tan cerca como sea posible. Este problema de control recibe el nombre de *control de seguimiento*.

Un planteamiento particularmente sencillo para control, y que es un caso particular del control de seguimiento, es el denominado *control de regulación*. En este planteamiento, la trayectoria especificada en el espacio de trabajo del robot es simplemente un punto en dicho espacio. La problemática de control de regulación consiste en llevar el extremo del manipulador (en el caso de robots industriales) o el mismo robot (en el caso de móviles) a dicho punto, independiente de su posición inicial.

Ya que se tiene conocimiento sobre la planta, se ha diseñado el controlador adecuado para cumplir con la tarea, se conocen los movimientos que debe seguir el robot para cumplir con ésta y se tienen las lecturas de los sensores del robot, es posible acoplar todos los bloques para lograr que el robot cumpla con la tarea especificada.

1.3. Robótica móvil

Un robot móvil es un vehículo que cuenta con un control automático, lo que le da cierto grado de autonomía. Para realizar ciertas tareas específicas pueden moverse

en diferentes medios. La historia de los primeros robots móviles data de la segunda guerra mundial, donde éstos emergieron como resultado de los avances tecnológicos en las nuevas áreas de investigación, como las ciencias de la computación y la cibernética. Los primeros robots móviles fueron bombas automatizadas que contaban con sistemas de control que los dirigía hacia un objetivo que tenían que destruir. De esta forma se podría decir que los primeros robots móviles fueron los misiles V1 y V2.

Una vez acabado el conflicto bélico, la idea de utilizar sistemas de control en mecanismos móviles se empezó a ampliar en campos diferentes al de las armas. En 1948 W. G. Walter desarrolló dos robots móviles autónomos los cuales servían para explorar su entorno. Estos robots móviles estaban provistos de un sensor que detectaba fuentes de luz; si encontraban una, el sistema de control los dirigía hacia ella, teniendo la capacidad de evadir obstáculos o incluso moverlos. A partir de esto se dedicaron más recursos y más tiempo a la investigación y al desarrollo de diferentes robots móviles.

Las agencias espaciales, en su afán de conocer y explorar sitios a los que es imposible enviar personas, han desarrollado un sinnúmero de robots móviles para alcanzar este objetivo; las primeras misiones fueron lanzadas a la luna, donde los robots móviles únicamente iban equipados con cámaras y equipos de comunicación que les permitían tomar imágenes de la superficie lunar y enviarlas de regreso a la Tierra, pero con el paso de los años se han desarrollado robots móviles cada vez más complejos, que no sólo se encargan de tomar imágenes sino que ya son capaces hasta de manipular objetos, tomar muestras de la atmósfera en la que se encuentran, e incluso unos cuantos robots móviles están provistos de laboratorios portátiles que se encargan de analizar las muestras tomadas en la misma plataforma.

Además de estos avances en el tema espacial, la robótica móvil encontró aplicaciones una vez más en el campo militar, lográndose el desarrollo de robots móviles con capacidad de volar, los cuales pueden ser utilizados como simples exploradores, utilizando cámaras de alta resolución, o también para destruir objetivos enemigos; esto con el desarrollo de robots móviles de mayor tamaño, con capacidad de cargar y manipular armas.

La exploración ha sido una de las áreas que más ha impulsado el desarrollo de diferentes robots móviles; esto debido principalmente a que este tipo de robots puede diseñarse para lograr tener acceso a lugares donde sería imposible o de gran riesgo el acceso humano, como en la exploración submarina, donde en muchos casos se requiere maniobrar en espacios muy pequeños a los que no pueden acceder los submarinos tripulados, o como en la exploración de áreas con algún tipo de toxicidad.

En resumen, los robots móviles pueden desplazarse en diferentes ambientes y pueden

ser clasificados en cuatro categorías, dependiendo del ambiente en el que se mueven, estas categorías son:

- *Robots móviles espaciales:* Son robots que se encuentran fuera de la atmósfera terrestre (como los satélites).
- *Robots móviles aéreos:* Estos robots son conocidos comúnmente como UAV (vehículos aéreos no tripulados, por sus siglas en inglés), se desplazan en el aire y suelen usar hélices (como en el caso de los helicópteros y aviones pequeños) o turbinas (para el caso de los UAV de mayor tamaño).
- *Robots móviles acuáticas:* Se desplazan en la superficie de un medio acuático o bien, sumergidos en éste; son conocidos como AUV (del inglés "autonomous underwater vehicle").
- *Robots móviles terrestres:* Estos robots se desplazan por tierra, y a su vez pueden ser subclasificados en diferentes categorías, dependiendo del tipo de actuador con el que logran el movimiento, la clasificación es la siguiente
 - Robots móviles con piernas: Generalmente inspirados en algún ser vivo terrestre.
 - Robots móviles con orugas: Son utilizados principalmente para terrenos extremos donde hay poco agarre, como en la nieve.
 - Robots móviles con ruedas: Tal vez son los robots móviles más usados, debido a que pueden lograr grandes velocidades con mayor facilidad.

Debido a que la plataforma con la que se va a trabajar en este trabajo de tesis es un robot móvil terrestre con ruedas, a continuación se presentan las características más importantes de este tipo de sistemas.

1.3.1. Robots móviles con ruedas

Los robots móviles con ruedas, a diferencia de los robots con piernas, alcanzan velocidades de desplazamiento más altas, lo que ha hecho que muchos prefieran este tipo de locomoción para el diseño de un robot móvil.

En los robots móviles con ruedas (RMR) se pueden variar las características de movilidad del robot a través de la distribución y tipo de ruedas montadas en el robot

y que están en contacto con la superficie. Una rueda es un dispositivo que provee o permite movimiento relativo entre su montura y una superficie en la que se supone que hace únicamente un punto de contacto o *punto de rodamiento*.

Un RMR está formado (en la mayoría de los casos) por un cuerpo rígido y un número n de ruedas, las cuales son las que se encargan de darle capacidad de movimiento al robot. Estas ruedas pueden ser de diferentes tipos dependiendo de las características de movilidad que requiera el robot.

Los RMR son también los más utilizados en la industria debido a que comúnmente se cuenta con superficies duras y lisas en el ambiente industrial, lo que hace posible la utilización de ruedas de manera eficiente. Algunas de las aplicaciones que tienen los RMR en la industria son: seguimiento automático de caminos trazados con líneas de colores o cintas reflectoras, y el transporte de materiales a lo largo de la trayectoria que siguen, reduciendo los tiempos de operación de algunos procesos.

En el caso de los RMR las restricciones del tipo (1.2) pueden ser agrupadas de la siguiente manera:

$$A(\rho)\dot{\rho} = 0 \quad (1.3)$$

donde $A(\rho) \in \mathbb{R}^{n \times m}$ es conocida como la matriz de restricciones no holonómicas planificadas.

Clasificación de los RMR

Los RMR pueden ser clasificados tomando en cuenta la disposición de las ruedas con los que cuenta, las configuraciones más comunes son:

- *RMR omnidireccional*: Como su nombre lo dice, este robot es capaz de lograr movimiento instantáneo en cualquier dirección sin necesidad de cambiar su orientación, generalmente esto se logra con un arreglo de ruedas especiales que, en conjunto, producen movimientos en cualquier dirección.
- *RMR diferencial*: Este tipo de robot cuenta con un arreglo de dos ruedas independientes con un eje en común.
- *RMR tipo carro*: Este tipo de robot tiene una configuración similar a la de los vehículos comunes (automóviles); pueden contar con una o más ruedas fijas en

un eje, y en otro eje cuentan con una o más ruedas que pueden ser disecionadas para cambiar el sentido del movimiento del robot.

Centro instantáneo de rotación

Un concepto geométrico de suma importancia en los RMR es el conocido como el *centro instantáneo de rotación* (Siegwart *et al.*, 2011). Para explicar este concepto, considere una rueda girando y produciendo un movimiento en la dirección de giro de la rueda y con su movimiento lateral restringido. Esta restricción puede ser mostrada dibujando una *línea de cero rascamiento (LCM)* a través de su eje horizontal, perpendicular al plano de la rueda. En cualquier instante dado, el movimiento de la rueda a lo largo de la LCM debe de ser cero. En otras palabras, la rueda deberá moverse a lo largo de un círculo de radio R de forma que el centro de ese círculo está localizado en la intersección de las LCM. Este punto de intersección, llamado *centro instantáneo de rotación* (ICR por sus siglas en inglés) se puede encontrar en cualquier punto a lo largo de LCM (cuando R es infinito, la rueda se mueve en una línea recta).

Para robots que cuentan con dos o más ruedas, el ICR se encuentra trazando la LCM de cada una de las ruedas y entonces el punto en el que las líneas se intersectan será el ICR del robot; (ver figura 1.2). La movilidad de los RMR queda restringida al lugar dónde pueda ser localizado el ICR, el cual puede ser ajustado cambiando la orientación de algunas de sus ruedas.

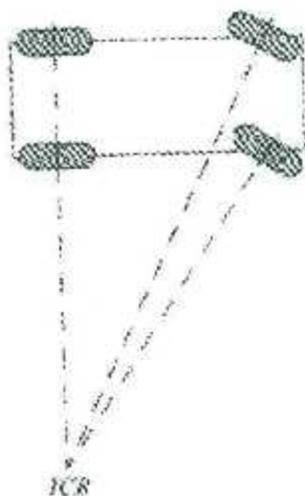


Figura 1.2: Centro instantáneo de rotación de un robot móvil con ruedas.

Robot móvil diferencial y tipo uniciclo

Un caso especial es el caso del RMR diferencial, donde el ICR sigue restringido a encontrarse a lo largo de la LCM pero no en un punto en específico, sino que la posición de éste se mueve dependiendo de la diferencia de las velocidades de las dos ruedas con las que cuenta. Por ejemplo, si ambas ruedas giran con la misma rapidez y en el mismo sentido, entonces el robot diferencial se moverá en línea recta, y si giran con la misma rapidez en sentidos opuestos, el ICR se encontrará en la distancia media entre ambas ruedas a lo largo del eje en común, causando que el robot gire en su propio eje.

Un robot móvil diferencial es uno de los tipos más básicos pero más utilizados en el área de robótica móvil; esto se debe en gran medida a que el robot ocupa una área muy pequeña en comparación con otro tipo de robots móviles con más ruedas, y a que puede girar en su propio eje vertical, como ya se mencionó anteriormente. En la figura 1.3 se muestra el diagrama de un RMR diferencial y algunos de sus parámetros. Además de las dos ruedas actuadas, fijas e independientes con las que cuenta el robot, es común que éste cuente con una tercera rueda, por lo general situada debajo del cuerpo del robot; esta rueda no está actuada y solamente sirve como punto de apoyo para que el robot permanezca en posición vertical. Las entradas de control para este tipo de robots son generalmente los pares o velocidades aplicados a cada una de las dos ruedas actuadas.

Otro tipo de robot móvil importante es el robot uniciclo, que cuenta únicamente con una rueda sobre la cual se mueve, dicha rueda produce una rapidez lineal v en la dirección de orientación de la rueda. Además, el uniciclo cuenta con un mecanismo que produce una rapidez angular ω que permite al robot girar sobre su propio eje.

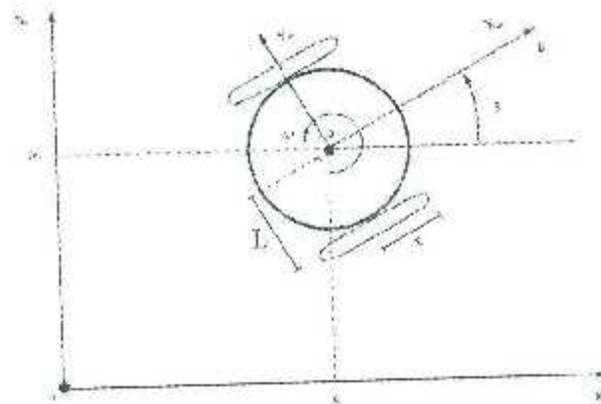


Figura 1.3: Diagrama de un RMR diferencial (vista superior)

cambiando con esto la orientación de la rueda. Debido a que la mayoría de los algoritmos de planificación de movimientos entregan una rapidez lineal y una rapidez angular, es más fácil trabajar con un modelo que emplee este tipo de entradas en el sistema.

Nótese que, se puede transformar las rapidezes angulares de cada rueda de un robot diferencial en la rapidez lineal y angular del robot v y ω usando la siguiente relación:

$$v = \frac{r}{2} (\dot{\varphi}_r + \dot{\varphi}_l) \quad (1.4)$$

$$\omega = \frac{r}{2L} (\dot{\varphi}_r - \dot{\varphi}_l), \quad (1.5)$$

donde $\dot{\varphi}_r$ y $\dot{\varphi}_l$ son las rapidezes angulares de la rueda izquierda y derecha respectivamente, r es el radio de las ruedas y L es la longitud del punto medio del robot a cada una de las ruedas, medida a lo largo del eje común. Una vez hecha esta transformación de velocidades, es común llamar a estos sistemas *robots tipo unicycle*, ya que aunque sigue siendo el mismo sistema que cuenta con dos ruedas independientes, su modelo cinemático es idéntico al de un unicycle.

1.4. Péndulos subactuados

Un sistema mecánico subactuado es un sistema con menos actuadores que grados de libertad. La mayoría de estos sistemas son subactuados por diseño: los actuadores pueden llegar a ser caros y/o pesados y por esto algunas veces son evitados en el diseño del sistema; otros sistemas pueden llegar a ser subactuados debido a fallas en los actuadores, y en tal caso se pueden emplear leyes de control diseñadas especialmente para este tipo de sistemas, que entrarían en funcionamiento únicamente cuando se detecte una falla.

Una clase de sistemas subactuados que ha captado la atención de los investigadores son los llamados péndulos invertidos; esto debido en gran medida a la baja complejidad de su modelado y control en comparación con otros sistemas subactuados, pero que a su vez permiten estudiar el problema de control en sistemas subactuados.

El péndulo subactuado, en su forma más sencilla, contiene una varilla rígida que gira en torno a un pivote, el cual no está actuado. Un péndulo tiene dos equilibrios: uno estable cuando la varilla se encuentra en posición vertical hacia abajo (esto debido a la fuerza de gravedad que lo lleva hasta este punto), y uno inestable; cuando la varilla apunta hacia arriba. Cuando la tarea de control es mantener la varilla cerca del equilibrio

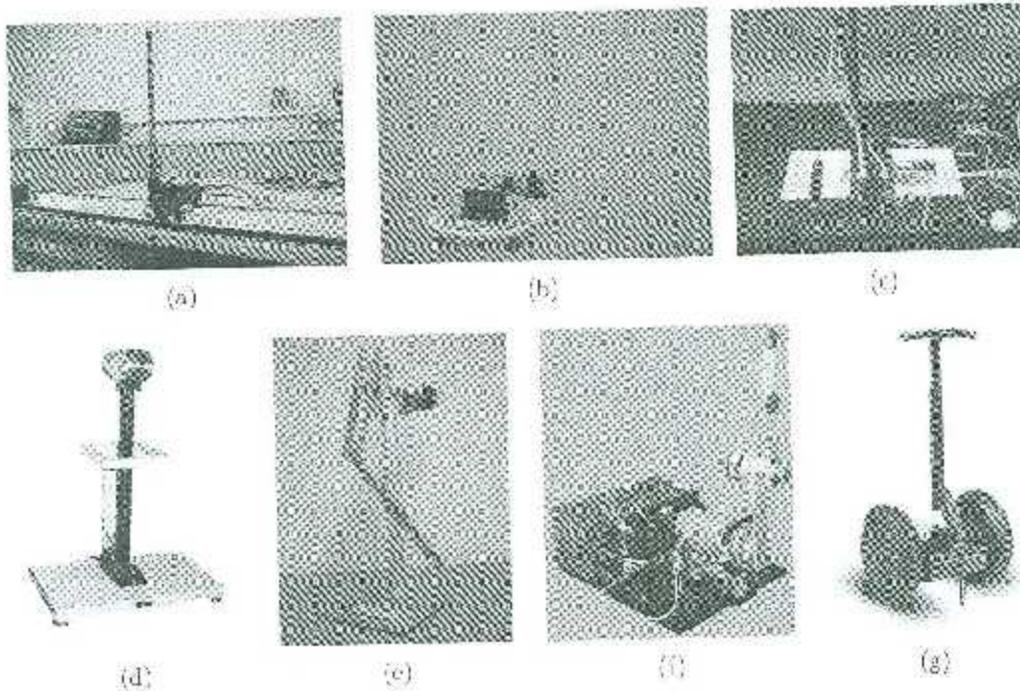


Figura 1.4: Diferentes tipos de péndulos invertidos: (a) carro-péndulo; (b) péndulo de Furuta; (c) péndulo invertido esférico; (d) péndulo con rueda inercial; (e) acrobot; (f) pendubot; (g) péndulo invertido sobre ruedas.

inestable, se dice que se está controlando un péndulo invertido.

Existen diferentes tipos de sistemas que se basan en el péndulo invertido; el objetivo de todos estos es el de mantener el péndulo en el equilibrio inestable a pesar de las perturbaciones que se puedan presentar. Como ya se mencionó, el pivote respecto al cual gira el péndulo no está actuado, por lo que se requiere que a giras de las fuerzas aplicadas al sistema sean transferidas indirectamente al péndulo para lograr el objetivo de control deseado.

Existen diferentes sistemas tipo péndulo invertido (ver Figura 1.4), siendo los más comunes los siguientes (Alvarado, 2010):

- *Sistema carro-péndulo:* Es un sistema que consiste en una varilla rígida unida mediante un pivote a un carro, el cual restringe su movimiento a lo largo de un riel. Las fuerzas externas al pivote son producidas por los movimientos horizontales del carro.
- *Péndulo de Furuta:* Es un sistema que consiste en un péndulo invertido sobre

un brazo rotatorio que se mueve en un plano horizontal. Las fuerzas aplicadas al péndulo son transmitidas por la fuerza centrífuga de la articulación actuada.

- *Péndulo con rueda inercial:* Es un sistema que consiste en un péndulo con una rueda que produce momentos de inercia lo suficientemente grandes para mover el péndulo hasta la posición vertical a través del giro de la rueda.
- *Acrobot:* Este sistema es un robot planar de dos eslabones que se mueve en un plano vertical, este sistema es subactuado debido a que sólo tiene un actuador en la segunda articulación.
- *Pendubot:* Este sistema es muy parecido al acrobot, la única diferencia es que la articulación actuada en este caso es la primera.
- *Péndulo invertido esférico:* Este sistema está formado por una varilla delgada montada en una articulación universal que puede moverse libremente en un plano horizontal.
- *Péndulo invertido sobre ruedas (PIR):* Es un mecanismo parecido al carro-péndulo, con la diferencia de que ahora el péndulo está acoplado a un mecanismo con ruedas, el cual ya no tiene limitado su movimiento sobre un riel sino que puede moverse libremente en el plano mientras mantiene la configuración vertical del péndulo.

Los péndulos invertidos sobre ruedas son mecanismos que presentan muchos retos al momento de diseñar controladores que logren cumplir con alguna tarea útil. Esto debido a que presentan un grado de complejidad más elevado al momento de diseñar controladores, a diferencia de los sistemas totalmente actuados; pero además, las ruedas que producen las fuerzas para mantener el péndulo en posición vertical, también agregan complejidad al momento de diseñar controladores.

La plataforma en la que se implementaron los algoritmos abordados en este trabajo de tesis es un péndulo invertido sobre ruedas. El sistema utilizado fue construido por la compañía Segway, la cual fue fundada en 1999, y desde entonces se ha dedicado al diseño y fabricación de plataformas móviles, principalmente de tipo PIR. En un principio la compañía únicamente producía plataformas que servían como transporte para personas; este tipo de plataformas es conocido como *Segway PT (Personal Transporter)* y ganó gran popularidad al tratarse de vehículos capaces de alcanzar altas velocidades y al ser fáciles de controlar, utilizando únicamente la inclinación del cuerpo de la persona. Años después, debido al interés de agencias privadas de desarrollar investigación con este

tipo de plataformas, nació otra línea de plataformas móviles conocidas como *Segway RMP* (*Robotic Mobility Platform*), las cuales ya están especializadas para investigación, y además, permiten la experimentación con algoritmos de control y planificación, utilizando una computadora externa. Una de las principales ventajas que presenta las plataformas Segway RMP, con respecto a otras en el mercado es su gran capacidad de carga, lo que permite que puedan ser equipadas con sensores y/o actuadores extra sin perjudicar el desempeño de la plataforma.

1.5. Objetivos de la tesis

El objetivo general de la tesis consiste en estudiar e implementar algoritmos de planificación y control de movimiento en la plataforma Segway RMP-100.

Los objetivos específicos de la tesis son los siguientes:

- Estudiar la teoría general de modelado, planificación de tareas y control de robots móviles con ruedas tipo unicycle.
- Estudiar algoritmos de control para balanceo de un PIR, así como de seguimiento de postura en un RMR.
- Desarrollar un sistema computacional para el control en tiempo real de la plataforma RMP-100.
- Implementar y evaluar el desempeño de los controladores de seguimiento y de balanceo estudiados en la plataforma RMP-100, utilizando el sistema desarrollado.

1.6. Organización del documento

El resto de este documento de tesis se organiza de la siguiente manera:

El capítulo 2 presenta primero las restricciones a las que están sujetos los RMR debido a los diferentes tipos de ruedas con las que puede contar. Posteriormente se describen dos métodos utilizados para la obtención del modelo cinemático de velocidad de un RMR. En este mismo capítulo se presentan dos métodos para la obtención del modelo dinámico de robots con restricciones no holonómicas: el método de Euler Lagrange y el método de Kane.

En el capítulo 3 se abordan conceptos básicos sobre planificación de tareas, especialmente aplicada a RMR tipo unicycle. En cuanto al control, primero se explica el teorema de Brockett, el cual da las condiciones necesarias para poder aplicar controladores lineales clásicos a una planta; y debido a que los robots móviles con ruedas no cumplen con el teorema, en este mismo capítulo se presentan algunas alternativas para el control de este tipo de plataformas.

En el capítulo 4 se describe detalladamente la plataforma con la que se va a trabajar, desde aspectos técnicos hasta los modelos cinemático y dinámico obtenidos como parte de esta tesis.

En el capítulo 5 se detallan las mejoras realizadas al sistema computacional para el control de la plataforma (la forma en que éstas se realizaron y la manera en que opera el sistema). En este capítulo también se mencionan datos importantes sobre la interfaz de control, el protocolo de comunicación y el periodo de muestreo con el que funciona la interfaz.

En el capítulo 6 se presentan los resultados de la simulación y experimentación de controladores diseñados para robots móviles con ruedas tipo unicycle, mientras que en el capítulo 7 se presentan los resultados de la simulación y experimentación de los controladores de balanceo diseñados para péndulos invertidos sobre ruedas. Ambos tipos se pueden implementar en la plataforma RMP-100 cambiando su modo de operación.

Finalmente, en el capítulo 8, se dan las conclusiones de este trabajo de tesis, así como un resumen de las aportaciones de la misma y el trabajo a futuro que podría ser realizado en la plataforma Segway RMP-100.

Además se incluyen dos apéndices: en el primero se describe a detalle la obtención del modelo cinemático de velocidad de la plataforma, utilizando los métodos estudiados; en el segundo apéndice se aplican el método de Euler-Lagrange y de Kane para obtener el modelo dinámico de la plataforma.

2 Modelado de los RMR

Un robot móvil con ruedas es modelado como un cuerpo rígido con un número de ruedas arbitrario. Además de los que se abordan en esta tesis, algunos trabajos sobre modelado cinemático se pueden encontrar en (Maddocks, 1990), (Shin y Park, 2001), (Whockuk *et al.*, 2004), (Low y Leow, 2005). Mientras que para la obtención del modelo dinámico se encuentran los trabajos publicados en (Campion *et al.*, 1996), donde se toma como base la obtención del modelo utilizando el método de Euler-Lagrange para cuerpos sin restricciones, y después los multiplicadores de Lagrange para tomar en consideración las restricciones no holonómicas; en (Zhao y BeMent, 1992) se utilizan las ecuaciones de Newton-Euler para obtener todas las ecuaciones de movimiento y se toman en cuenta sólo las de importancia, otros trabajos donde se obtiene el modelo dinámico de robots móviles pueden ser encontrados en (Boyden y Velinsky, 1994) y (Albagul y Wahyudi, 2004). En el presente trabajo se utilizaron los métodos para modelado cinemático publicados en (Campion *et al.*, 1996) y (Muir y Neuman, 1987) y para la obtención del modelo dinámico se usó el método empleado en (Soto, 2014) y se comparó con el obtenido con el método de Kane, el cual fue publicado por primera vez en (Kane, 1961) y después fue aplicado al área de robótica y robots móviles con ruedas en (Kane y Levinson, 1983) y (Thanjavur y Rajagopalan, 1997).

El estudio del comportamiento de robots móviles con ruedas generalmente se limita al análisis de su cinemática de velocidad, esto es porque se considera que el robot se mueve a velocidades relativamente bajas y con cargas pequeñas; de esta forma las aceleraciones e inercias del robot pueden ser despreciadas. Cuando se considera que el RMR opera a velocidades grandes o bajo cargas de gran tamaño (en donde las masas e inercias del sistema ya no son despreciables), es necesario obtener las ecuaciones dinámicas del sistema.

En este capítulo primero se presentan algunos conceptos fundamentales para el modelado de los RMR. Luego se describen los dos métodos para el modelado cinemático

de velocidad; el primero propuesto en (Muir y Neuman, 1987) será denominado como método de Muir-Neuman mientras que el segundo, que se propone en (Campion *et al.*, 1996) y es descrito también en (de Wit *et al.*, 1996) se denominará como método de Campion *et al.* Al final se describen los dos métodos para el modelado dinámico: el de Euler Lagrange y el de Kane.

2.1. Fundamentos

En esta sección se abordan algunos fundamentos necesarios para entender la estructura de un RMR y poder aplicar los métodos de modelado descritos en las siguientes secciones. Para empezar considérese la figura 2.1 en la que se presentan algunos marcos coordenados que resultan útiles. En la figura 2.1(a) se ha dibujado el RMR con el marco $\Sigma_r(X_r, Y_r, Z_r)$ asociado a él, dicho marco se coloca en el centro del RMR con el eje X_r apuntando hacia la dirección de avance, la misma figura muestra el marco fijo (inercial) $\Sigma_0(X_0, Y_0, Z_0)$; el ángulo θ es el que da la orientación del marco Σ_r respecto al marco Σ_0 . Luego en la figura 2.1(b) se muestran cuatro marcos:

- El marco $\Sigma_r(X_r, Y_r, Z_r)$ asociado al robot.
- El marco $\Sigma_H(X_H, Y_H, Z_H)$ el cual tiene la misma orientación que el marco Σ_r .

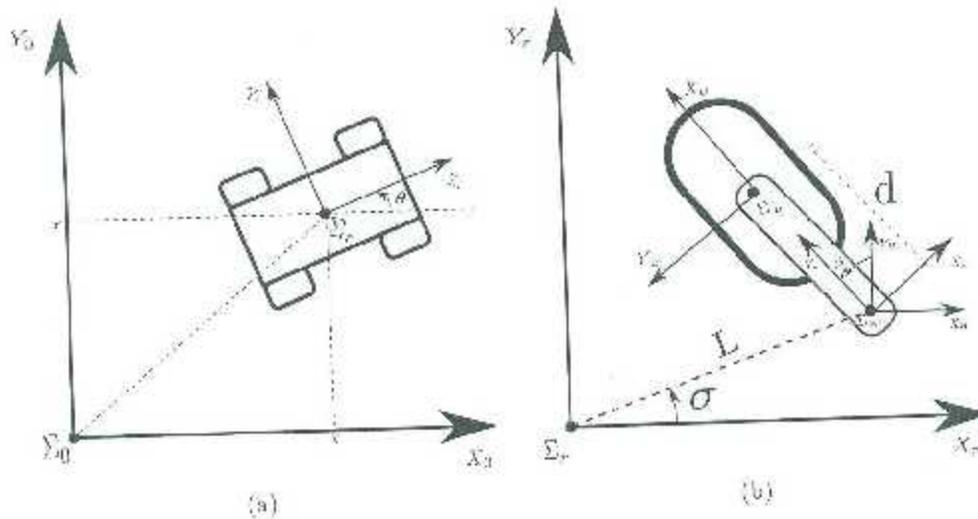


Figura 2.1: Marcos coordenados en un RMR: (a) marco inercial y marco del robot; (b) asignación de marcos en las ruedas.

pero está trasladado al eje de rotación de la rueda. L y σ dan las coordenadas polares del origen de Σ_H con respecto a Σ_r .

- El marco $\Sigma_s(X_s, Y_s, Z_s)$ tiene el mismo origen que Σ_H pero está orientado de manera que X_s sea perpendicular a la dirección de avance de la rueda.
- El marco $\Sigma_w(X_w, Y_w, Z_w)$ se encuentra en el centro de la rueda y está ligado a ésta de manera que X_w está en la dirección de avance. Nótese que α y β representan las coordenadas polares del origen de Σ_w con respecto a Σ_H .

Los marcos mencionados (Σ_H , Σ_s y Σ_w) son conocidos como los marcos de la cadera, de direccionamiento y de la rueda, respectivamente.

Ahora es posible definir la postura del robot por medio del vector

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \in \mathbb{R}^3, \quad (2.1)$$

donde (x, y) son las coordenadas cartesianas de Σ_r con respecto a Σ_0 . Además, la matriz de rotación que nos permite expresar la orientación de Σ_r con respecto a Σ_0 está definida por:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \text{SO}(3) \quad (2.2)$$

2.1.1. Tipos de ruedas

En la referencia Muir y Neuman (1987), se clasifican las ruedas en tres tipos: las ruedas convencionales, las ruedas omnidireccionales (succas) y las ruedas esféricas. Debido a que las más usadas son las dos primeras, estas son las que detallan a continuación.

Ruedas convencionales

La *rueda convencional* es la que tiene el diseño más simple y por eso es la más común (ver figura 2.2). En general puede tener hasta dos grados de libertad: una rotación

respecto al eje Y_w dada por la rapidez angular de la rueda, que permite un movimiento lineal a lo largo de la superficie de contacto en dirección perpendicular al eje X_w ; y una rotación alrededor del eje Z_w dada por la variación de los ángulos β y σ .

Cuando el ángulo β es fijo se dice que se tiene una *rueda fija*. Cuando el eslabón cuenta con un actuador para controlar el ángulo β se considera que es una *rueda direccionable*; pero si este eslabón no es actuado, se considera que es una *rueda de castor*. Las ruedas de castor al no ser actuadas pueden girar 360° libremente, una característica importante de este tipo de ruedas es la distancia d entre el origen del marco Σ_H y el origen del marco de la rueda Σ_w , en las ruedas convencionales se considera que la fuerza para producir un deslizamiento rotacional es muy pequeña, por lo que este tipo de deslizamiento es considerado como un grado de libertad. Por otro lado, la fuerza necesaria para que exista un deslizamiento transversal (en la dirección del eje de la rueda) es demasiado grande por lo que no se considera permisible.

En las ruedas fijas se presentan dos tipos de restricciones no holonómicas:

- *Restricción de rodamiento puro:* Toda velocidad angular de la rueda, con radio r , respecto a su eje horizontal (Y_w) produce una velocidad lineal en el eje X_w igual a:

$$[-\sin(\sigma + \beta) \quad \cos(\sigma + \beta) \quad L \cos(\beta)] R(\theta)^T \dot{\xi} + r\dot{\psi} = 0 \quad (2.3)$$

- *Restricción de no deslizamiento lateral:* La fuerza requerida para tener un deslizamiento lateral es demasiado grande por lo que se considera no permisible, así que se tiene:

$$[\cos(\sigma + \beta) \quad \sin(\sigma - \beta) \quad L \sin(\beta)] R(\theta)^T \dot{\xi} = 0 \quad (2.4)$$

Cabe mencionar que para el caso de las ruedas de castor, la restricción de no deslizamiento lateral queda de la siguiente manera:

$$[\cos(\sigma + \beta) \quad \sin(\sigma - \beta) \quad d + L \sin(\beta)] R(\theta)^T \dot{\xi} + d\dot{\beta} = 0 \quad (2.5)$$

donde $\dot{\psi}$ es la velocidad angular de la rueda con respecto al eje de rotación de la misma. Estas restricciones ayudarán más adelante a identificar las velocidades actuadas en una rueda empleando el método de modelado de Muir-Neuman.

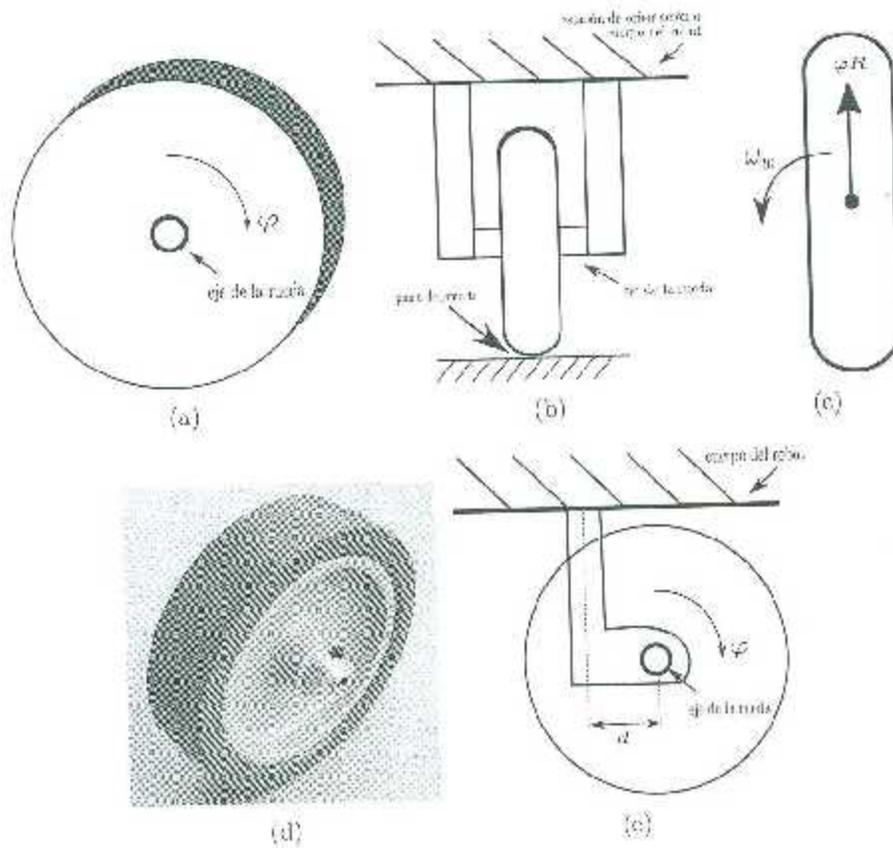


Figura 2.2: Rueda convencional: (a) vista lateral; (b) vista frontal; (c) vista superior; (d) fotografía; (e) rueda de castor.

Ruedas omnidireccionales

Este tipo de ruedas también conocidas como *ruedas mescano* o *ruedas succas*, se basan en el diseño de las ruedas convencionales, pero se modificaron agregando rodillos a lo largo de su circunferencia, lo cual se puede apreciar en la figura 2.3. Los rodillos, de radio r , comúnmente cuentan con un ángulo de orientación fijo (η) de 45° ó 90° con respecto al plano de la rueda. Esta modificación a la rueda convencional le da un grado de libertad más a la rueda; este grado extra, proporcionado por el movimiento de los rodillos, elimina la restricción de no deslizamiento. Estas ruedas sólo cuentan con la restricción de rodamiento puro que está dada por:

$$[-\sin(\sigma + \beta - \eta) \quad \cos(\sigma + \beta + \eta) \quad L \cos(\beta + \eta)] R(\theta)^T \dot{\xi} - r \cos(\eta) \dot{\phi} = 0 \quad (2.6)$$

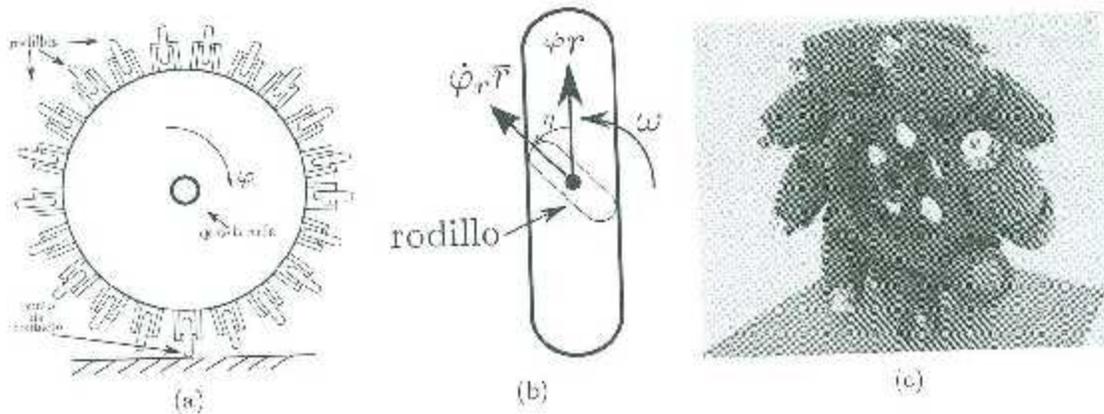


Figura 2.3: Rueda omnidireccional o succa: (a) vista lateral; (b) vista superior; (c) fotografía.

2.1.2. Matrices jacobianas de las ruedas

De acuerdo con (Muir y Neuman, 1987) el movimiento de cada rueda puede ser descrito en términos de cuatro variables que son definidas como sigue:

- ω_{x_w} : Componente en x de la velocidad angular de la rueda respecto al marco Σ_w .
- ω_{y_w} : Componente en y de la velocidad angular de la rueda respecto al marco Σ_w .
- ω_{z_w} : Componente en z de la velocidad angular de la rueda respecto al marco Σ_w .
- ω_{z_s} : Componente en z de la velocidad angular de la rueda respecto al marco Σ_s .

En la misma referencia se explica como determinar las matrices de transformación entre las variables y las velocidades de postura del RMR (es decir, \dot{x} , \dot{y} y $\dot{\theta}$), para cada tipo de rueda. A esas matrices se les conoce como jacobianas de las ruedas.

A continuación se presentan las matrices jacobianas de las ruedas para los tipos de ruedas más comunes.

Ruedas convencionales fijas

En este tipo de ruedas no se permite un giro en la dirección del eje X_w . Además, al ser rueda fija tampoco hay giro en el eje Z_w , de modo que las únicas variables de interés son ω_{y_w} que corresponde a $\dot{\varphi}$ y ω_{z_w} que sería $\dot{\theta}$. En resumen, se tiene

$$\omega_{x_w} = 0 \quad \omega_{y_w} = \dot{\varphi} \quad \omega_{z_w} = \dot{\theta} \quad \omega_{z_x} = 0$$

y la matriz jacobiana de la rueda sería

$$J_f = \begin{bmatrix} r \cos(\theta_w) & L \sin(\sigma) \\ r \sin(\theta_w) & -L \cos(\sigma) \\ 0 & 1 \end{bmatrix} \quad (2.7)$$

donde $\theta_w = \sigma + \beta - \frac{\pi}{2}$, de modo que

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r \cos(\theta_w) & L \sin(\sigma) \\ r \sin(\theta_w) & -L \cos(\sigma) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{y_w} \\ \omega_{z_w} \end{bmatrix}$$

Ruedas convencionales direccionales y castor

En este tipo de ruedas se cumple que

$$\omega_{x_w} = 0 \quad \omega_{y_w} = \dot{\varphi} \quad \omega_{z_w} = \dot{\theta} + \dot{\beta} \quad \omega_{z_x} = \dot{\beta}$$

y la matriz jacobiana de estas ruedas sería

$$J_d = \begin{bmatrix} r \cos(\theta_w) & L \sin(\sigma) + d \sin(\theta_w) & -L \sin(\sigma) \\ r \sin(\theta_w) & -[L \cos(\sigma) + d \cos(\theta_w)] & L \cos(\sigma) \\ 0 & 1 & -1 \end{bmatrix} \quad (2.8)$$

de modo que

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r \cos(\theta_w) & L \sin(\sigma) + d \sin(\theta_w) & L \sin(\sigma) \\ r \sin(\theta_w) & -[L \cos(\sigma) + d \sin(\theta_w)] & L \cos(\sigma) \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_{y_w} \\ \omega_{x_w} \\ \omega_{z_w} \end{bmatrix}$$

Rueda omnidireccional

En este caso $\omega_{z_w} = 0$, pero si hay giros posibles en todos los ejes del marco Σ_w de forma que la jacobiana de estas ruedas es

$$J_{om} = \begin{bmatrix} r \cos(\theta_w) & -r \cos(\theta_w) \cos(\eta) - \bar{r} \sin(\theta_w) \sin(\eta) & L \sin(\sigma) + d \sin(\theta_w) \\ r \sin(\theta_w) & \bar{r} \cos(\theta_w) \sin(\eta) - r \sin(\theta_w) \cos(\eta) & -[L \cos(\sigma) + d \sin(\theta_w)] \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

de modo que

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r \cos(\theta_w) & -r \cos(\theta_w) \cos(\eta) - \bar{r} \sin(\theta_w) \sin(\eta) & L \sin(\sigma) + d \sin(\theta_w) \\ r \sin(\theta_w) & \bar{r} \cos(\theta_w) \sin(\eta) - r \sin(\theta_w) \cos(\eta) & -[L \cos(\sigma) + d \sin(\theta_w)] \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{x_w} \\ \omega_{y_w} \\ \omega_{z_w} \end{bmatrix}$$

2.1.3. Clasificación de los RMR

Considérese ahora el caso de un robot móvil de tipo general, equipado con n ruedas de alguno de los tipos descritos anteriormente (rueda fija, rueda direccional, rueda de castor y rueda omnidireccional). Para mayor claridad, se usarán los siguientes subíndices para identificar las cantidades relativas a estos cuatro tipos: f para ruedas fijas, s para ruedas direccionables, c para ruedas de castor y om para ruedas sueltas. El número de ruedas de cada tipo se denota por N_f , N_s , N_c y N_{om} , con $N_f + N_s + N_c + N_{om} = N$. La configuración de tal robot queda completamente descrita por los siguientes vectores de coordenadas:

- Coordenadas de la postura del robot:

$$\xi(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \in \mathbb{R}^3$$

- Coordenadas de los ángulos de orientación de las ruedas direccionables y de castor:

$$\beta(t) = \begin{bmatrix} \beta_s(t) \\ \beta_c(t) \end{bmatrix} \in \mathbb{R}^{N_s+N_c}$$

donde $\beta_s \in \mathbb{R}^{N_s}$ y $\beta_c \in \mathbb{R}^{N_c}$.

- Coordenadas de los ángulos de rotación de las ruedas

$$\varphi(t) = \begin{bmatrix} \varphi_s(t) \\ \varphi_d(t) \\ \varphi_c(t) \\ \varphi_{\text{cast}}(t) \end{bmatrix} \in \mathbb{R}^{N_f+N_d-N_c+N_{\text{cast}}}$$

donde $\varphi_s \in \mathbb{R}^{N_f}$.

Al conjunto completo de las coordenadas anteriores se le conoce como *coordenadas de configuración* del RMR y se les agrupa en el vector ρ , de modo que

$$\rho(t) = \begin{bmatrix} \xi(t) \\ \beta(t) \\ \varphi(t) \end{bmatrix} \in \mathbb{R}^{N_f+2N_s+2N_d+N_{\text{cast}}-N_c} \quad (2.10)$$

Las ecuaciones de restricción de todas las ruedas del RMR se pueden escribir de manera compacta como sigue:

$$J_1(\beta_s, \beta_c)R(\theta)\dot{\xi} + J_2\dot{\varphi} = 0, \quad (2.11)$$

$$C_1(\beta_s, \beta_c)R(\theta)\dot{\xi} + C_2\dot{\beta}_c = 0 \quad (2.12)$$

En (2.11) se tiene que

$$J_1(\beta_s, \beta_c) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \\ J_{1c}(\beta_c) \\ J_{1om} \end{bmatrix} \in \mathbb{R}^{N \times 3} \quad (2.13)$$

donde J_{1f} , J_{1s} , J_{1c} y J_{1om} son matrices de N_f , N_s , N_c y N_{om} renglones, respectivamente, que se extraen directamente de las restricciones (2.3) y (2.6). En particular, J_{1f} y J_{1om} son constantes, mientras que J_{1s} y J_{1c} son variantes en el tiempo, debido a $\beta_s(t)$ y $\beta_c(t)$, respectivamente. Además, $J_2 \in \mathbb{R}^{N \times N}$ es una matriz diagonal constante, cuyos elementos en la diagonal son los radios de las ruedas, excepto para los radios de las ruedas suecas, que están multiplicados por $\cos(\eta)$.

Por otro lado, en (2.12) se encuentran

$$C_1(\beta_s, \beta_c) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \\ C_{1c}(\beta_c) \end{bmatrix} \in \mathbb{R}^{(N_f + N_s + N_c) \times 3}, \quad C_2 = \begin{bmatrix} 0 \\ 0 \\ C_{2s} \end{bmatrix} \in \mathbb{R}^{(N_f + N_s + N_c) \times N_s}, \quad (2.14)$$

donde C_{1f} , C_{1s} y C_{1c} son matrices de N_f , N_s y N_c renglones, respectivamente, que se extraen de las restricciones de no deslizamiento (2.4) y (2.5). En particular, C_{1f} es constante, mientras que C_{1s} y C_{1c} son variantes en el tiempo. Además, $C_{2s} \in \mathbb{R}^{N_s \times N_s}$ es una matriz diagonal, cuyos elementos en la diagonal son iguales a d , para las N_s ruedas de castor.

Considérense ahora las primeras $N_f + N_s$ restricciones de no deslizamiento lateral de (2.12), que pueden ser escritas explícitamente como

$$C_1^*(\beta_s) R(\theta) \dot{\xi} = 0, \quad (2.15)$$

donde

$$C_1^*(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s d) \end{bmatrix} \in \mathbb{R}^{(N_f + N_s) \times 3}, \quad (2.16)$$

Ahora bien, nótese que (2.15) implica que el vector $R(\theta) \dot{\xi}$ pertenece al espacio nulo de $C_1^*(\beta_s)$, es decir $R(\theta) \dot{\xi} \in \mathcal{N}\{C_1^*(\beta_s)\}$, y que $\text{rank}\{C_1^*(\beta_s)\} \leq 3$ (donde $\text{rank}\{A\}$ representa el rango de la matriz A). Así que, si $\text{rank}\{C_1^*(\beta_s)\} = 3$ entonces $R(\theta) \dot{\xi} =$

0 y cualquier movimiento del RMR en el plano sería imposible. Por otra parte, si $\text{rank}\{C_1^*(\beta_s)\} \leq 2$ entonces debe existir un espacio nulo de $C_1^*(\beta_s)$, de dimensión $3 - \text{rank}\{C_1^*(\beta_s)\}$ que permita que el RMR se mueva en el plano, sin violar las restricciones de no deslizamiento.

Grado de movilidad

El grado de movilidad de un robot está definido como:

$$\delta_m = 3 - \text{rank}(C_1^*(\beta_s)) \quad (2.17)$$

donde

$$C_1^*(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}$$

Por ejemplo, el caso en el que el que $\text{rank}(C_1^*(\beta_s)) = 2$ implica que el robot tiene al menos 2 ruedas fijas y, si hay más de 2, que sus ejes se intersectan en el "ICR" cuya posición respecto al carro es fijo. En dado caso, esto nos dice que el único movimiento posible sería una rotación alrededor del "ICR" fijo. Esta limitación no es aceptable en la práctica por lo que se busca por diseño que $\text{rank}(C_1^*(\beta_s)) \leq 1$.

Así que de los diferentes tipos de RMR los que resultan de interés son aquellos en los que

$$1 \leq \delta_m \leq 3.$$

Grado de direccionabilidad

El grado de direccionabilidad δ_d es el número de ruedas direccionables que pueden ser orientadas independientemente con el fin de orientar el robot, y está dado por

$$\delta_d = \text{rank}(C_d(\beta_s))$$

El rango aceptable de δ_s es

$$0 \leq \delta_s \leq 2.$$

Clasificación

Una vez obtenidos δ_m y δ_s , es posible clasificar los RMR en cinco tipos, donde el nombre asignado al tipo es simplemente (δ_m, δ_s) :

- *Tipo (3,0)*: Este tipo de robot sólo cuenta con ruedas omnidireccionales y de castor, son llamados robots omnidireccionales.
- *Tipo (2,0)*: Este tipo de robot cuenta con una o varias ruedas fijas pero todas en un sólo eje común, son llamados robots diferenciales.
- *Tipo (2,1)*: Este tipo de robot no cuenta con ruedas fijas, por lo general es una rueda fija y las demás son tipo castor para apoyo.
- *Tipo (1,1)*: Cuenta con ruedas fijas en un eje en común y ruedas direccionables en un eje común pero diferente al de las ruedas fijas. Los robots tipo carro pertenecen a esta clasificación.
- *Tipo (1,2)*: Cuenta con al menos dos ruedas direccionables y una rueda tipo castor para el balanceo.

2.2. Modelado cinemático de velocidad

En este documento se distinguirán entre dos tipos de modelos cinemáticos de velocidad:

- *Modelo cinemático de velocidad en espacio de postura*: Es el modelo más simple y capaz de dar una descripción global del RMR, sin considerar los desplazamientos angulares de las ruedas.
- *Modelo cinemático de velocidad en espacio de configuración*: Este modelo incluye ecuaciones que describen la variación del desplazamiento de las ruedas actuadas.

Este último modelo es el que se empleará más adelante para obtener el modelo dinámico, ya que permite incluir las restricciones no holonómicas en el análisis del robot. A pesar de que algunas metodologías utilizadas en robots seriales pueden ser aplicadas también en los RMR, existen diferencias importantes entre estos dos tipos de sistemas que no pueden ser ignoradas; algunos ejemplos son:

- Los RMR contienen múltiples cadenas cerradas, mientras que los robots seriales sólo forman cadenas cerradas cuando el órgano terminal entra en contacto con objetos fijos.
- El contacto entre las ruedas y el piso constituye una articulación de tipo “par superior”, es decir, una articulación en la que sólo existe un punto de contacto entre los cuerpos y éste no es fijo (Muir y Neuman, 1987); mientras que los robots seriales cuentan con articulaciones de tipo “par inferior” (donde se restringe la superficie de contacto entre los cuerpos a un área específica).
- No todos los grados de libertad de las ruedas en un RMR cuentan con sensores de velocidad o posición.

Además, se deben tomar las siguientes consideraciones para hacer del problema de modelado de los RMR una tarea más fácil:

- Los RMR no contienen partes flexibles.
- Tienen cero o sólo un actuador de dirección por rueda.
- El ángulo de dirección se encuentra en el plano $X - Y$.
- Los RMR se mueven en una superficie plana (la superficie de movimiento).
- La fricción traslacional (perpendicular al plano de la rueda) al punto de contacto entre la rueda y la superficie de movimiento es lo suficientemente grande para impedir deslizamiento traslacional.
- La fricción rotacional (fricción de giro) entre el punto de contacto de la superficie de movimiento y la rueda es lo suficientemente pequeño para que exista un movimiento rotacional.

2.2.1. Método de Muir-Neuman

Este método aplica la convención de Sheth-Uicker (Sheth y Uicker, 1971) para asignar marcos coordenadas. A partir de esto, es posible aplicar matrices de transformación homogéneas para obtener el modelo cinemático de posición del RMR. Para el modelo cinemático de velocidad se utilizan las matrices jacobianas de las ruedas que ya se presentaron en la sección 2.1. Por último, si es necesario, se calcula el modelo cinemático de aceleración.

La aplicación de la convención de Sheth-Uicker para asignar los marcos de coordenadas elimina las ambigüedades en las matrices de transformación que podrían existir utilizando otro tipo de convenciones.

Debido a que los controladores estudiados en este trabajo requieren únicamente del modelo cinemático de velocidad, a continuación se presentan los pasos necesarios para obtener el modelo cinemático de velocidad en espacio de postura y el modelo cinemático de velocidad en espacio de configuración. Para más detalle acerca de la obtención de los modelos cinemáticos de posición y aceleración referirse a (Muir y Neuman, 1987).

Pasos preliminares

Para el desarrollo y aplicación del método de Muir-Neuman para el modelado cinemático de los RMR se sugiere seguir los siguientes pasos:

- I: Identificar el número y tipo de ruedas.
- II: Colocar los marcos Σ_r del robot y Σ_{R_i} , Σ_{S_i} y Σ_{W_i} de la i -ésima rueda.
- III: Obtener las matrices jacobianas J_i de cada rueda con $i = 1, 2, \dots, N$, las cuales fueron mencionadas en la sección 2.1.1.
- IV: Obtener la matriz jacobiana J_w , la cual está dada por:

$$J_w = \begin{bmatrix} [{}^r J_{1a}^T \Delta(J_{1a}) J_{1a}]^{-1} J_{1a}^T \Delta(J_{1a}) \\ [{}^r J_{2a}^T \Delta(J_{2a}) J_{2a}]^{-1} J_{2a}^T \Delta(J_{2a}) \\ \vdots \\ [{}^r J_{Na}^T \Delta(J_{Na}) J_{Na}]^{-1} J_{Na}^T \Delta(J_{Na}) \end{bmatrix} \quad (2.18)$$

donde J_{ia} es la parte del jacobiano de la i -ésima rueda que se encuentra actuada, mientras que J_{ib} es la parte no actuada del jacobiano de la i -ésima rueda.

$$\Delta(U) = \begin{cases} -I & \text{si } U \text{ es una matriz nula} \\ U(U^T U)^{-1} U^T - I & \text{en cualquier otro caso} \end{cases}$$

V: De la matriz J_a extraer únicamente los renglones que se multipliquen por las velocidades actuadas de cada una de las ruedas para formar la matriz jacobiana J_E , que relaciona el vector de velocidades actuadas de cada una de las ruedas $\dot{\varphi}$ con el vector de velocidades de control del RMR u , y de los renglones que multiplican velocidades no actuadas se forma la matriz jacobiana J_D , la cual relaciona el vector de velocidades de las ruedas no actuadas (ruedas de castor) $\dot{\beta}_c$ con el vector de velocidades del RMR u . Los vectores $\dot{\varphi}$ y $\dot{\beta}_c$ quedan así definidos como:

$$\dot{\varphi} = J_E u \quad (2.19)$$

$$\dot{\beta}_c = J_D u \quad (2.20)$$

Modelo cinemático de velocidad en espacio de postura

Una vez obtenidos los vectores de velocidades de control del robot u es posible conocer el modelo cinemático de velocidad en espacio de postura, el cual está dado por:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\dot{\xi}} = \underbrace{\begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(\theta)} \underbrace{\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix}}_u \quad (2.21)$$

Cuando se cuenta con ruedas direccionables, el modelo cinemático de velocidad en espacio de postura queda definido por

$$\dot{\xi} = R(\theta) u \quad (2.22)$$

$$\dot{\beta}_c = \zeta \quad (2.23)$$

donde $\dot{\beta}_c$ es el vector de velocidades de dirección de las ruedas direccionables y ζ es el vector de entrada de control de estas ruedas.

Modelo cinemático de velocidad en espacio de configuración

Este contiene el modelo cinemático de velocidad en espacio de postura dado por (2.22) y (2.23), pero además se incluyen los vectores de velocidades de las ruedas actuadas $\dot{\varphi}$ y de velocidades de giro en las ruedas direccionables ($\dot{\beta}_s$) y de castor ($\dot{\beta}_c$). Entonces se puede expresar el modelo cinemático de velocidad en espacio de configuración como:

$$\dot{\rho} = S(\rho)\bar{u} \quad (2.24)$$

donde

$$\dot{\rho} = \begin{bmatrix} \dot{\xi}^T & \dot{\beta}_s^T & \dot{\beta}_c^T & \dot{\varphi}^T \end{bmatrix}^T, \quad S(\rho) = \begin{bmatrix} R(\theta) & 0 \\ 0 & I \\ J_D & 0 \\ J_E & 0 \end{bmatrix}, \quad \bar{u} = \begin{bmatrix} u \\ \zeta \end{bmatrix}. \quad (2.25)$$

2.2.2. Método de Campion *et al.*

En este método primeramente se obtienen los grados de movilidad y de direccionabilidad para identificar el tipo de robot, y a partir de esto empezar el modelado.

Modelo cinemático en espacio de postura

El modelo cinemático en espacio de postura, para cuando el grado de movilidad $\delta_s > 1$, está dado por la siguiente ecuación:

$$\dot{\xi} = R(\theta)W(\beta_s)u \quad (2.26)$$

$$\dot{\beta}_s = \zeta \quad (2.27)$$

Cuando el robot no cuenta con ruedas direccionables, es decir $\delta_s = 0$, la matriz $W(\beta_s)$ es una constante y $\zeta = 0$ por lo tanto la ecuación (2.26) puede ser reescrita como:

$$\dot{\xi} = R(\theta)Wu. \quad (2.28)$$

En la tabla 2.1 se muestran los parámetros usados en cada tipo de robot, donde

Tabla 2.1: Definición de la matriz W y el vector \mathbf{u} para diferentes tipos de RMR

Robot tipo (3,0):	$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_\theta \end{bmatrix}$
Robot tipo (2,0):	$W = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$
Robot tipo (2,1):	$W = \begin{bmatrix} \cos(\beta_{s1}) & 0 \\ -\text{sen}(\beta_{s1}) & 0 \\ 0 & 1 \end{bmatrix}$	$\mathbf{u} = \begin{bmatrix} u_x \\ u_\theta \end{bmatrix}$
Robot tipo (1,1):	$W = \begin{bmatrix} L\text{sen}(\beta_{s1}) \\ 0 \\ \cos(\beta_{s1}) \end{bmatrix}$	$\mathbf{u} = [u_x]$
Robot tipo (1,2):	$W = \begin{bmatrix} L\text{sen}(\beta_{s1} + \beta_{s2}) \\ -2L\text{sen}(\beta_{s1})\text{sen}(\beta_{s2}) \\ \text{sen}(\beta_{s2} - \beta_{s1}) \end{bmatrix}$	$\mathbf{u} = [u_x]$

W es la matriz que restringe las velocidades robot y \mathbf{u} es el vector de velocidades de entrada que se pueden controlar.

Modelo cinemático en espacio de configuración

Para obtener este modelo es necesario ahora obtener el vector de velocidades angulares de las ruedas $\dot{\varphi}$ y las velocidades de giro en las ruedas direccionables $\dot{\beta}_s$, el cual estará dada únicamente por el valor de la entrada ζ mientras que las demás velocidades estén dadas por:

$$\dot{\beta}_c = D(\beta_c)W(\beta_s)\mathbf{u} \quad (2.29)$$

$$\dot{\varphi} = E(\beta_s, \beta_c)W(\beta_s)\mathbf{u} \quad (2.30)$$

donde

$$\begin{aligned} D(\beta_c) &= -C_{2c}^{-1}C_{1c}(\beta_s) \\ E(\beta_s, \beta_c) &= -J_2^{-1}J_1(\beta_s, \beta_c). \end{aligned}$$

Ahora si es posible obtener el modelo cinemático de velocidad en espacio de configuración:

$$\dot{\rho} = S(\rho)\bar{u}$$

donde

$$\dot{\rho} = [\dot{\xi}^T \ \dot{\beta}_s^T \ \dot{\beta}_c^T \ \dot{\varphi}^T]^T, \quad S(\rho) = \begin{bmatrix} R(\theta)W(\beta_s) & 0 \\ 0 & I \\ D(\beta_c)W(\beta_s) & 0 \\ E(\beta_s, \beta_c)W(\beta_s) & 0 \end{bmatrix}, \quad \bar{u} = \begin{bmatrix} u \\ \zeta \end{bmatrix} \quad (2.31)$$

2.3. Modelado dinámico

Los RMR son mecanismos que están sujetos a restricciones no holonómicas, por lo que las técnicas convencionales para modelado dinámico de robots sin restricciones no pueden ser utilizadas directamente. Para solucionar este problema se tienen muchas alternativas; en el presente trabajo se aplican dos métodos para obtener el modelo dinámico. En el primer método se usan las ecuaciones de Euler-Lagrange para obtener un modelo en términos de variables no mínimas, y después se aplica el llamado método de proyección (Soto, 2014) para obtener el modelo con variables mínimas. El segundo método es el propuesto por Kane (Kane, 1961) para mecanismos con restricciones, el cual directamente da el modelo mínimo.

2.3.1. Método de Euler-Lagrange

El modelo dinámico de un mecanismo con restricciones pfaffianas, (como las que tiene un RMR) está dado por:

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})}{\partial \dot{\boldsymbol{\rho}}} \right\} - \frac{\partial \mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})}{\partial \boldsymbol{\rho}} = \boldsymbol{\tau}_e + A(\boldsymbol{\rho})^T \boldsymbol{\lambda} \quad (2.32)$$

donde el lagrangiano $\mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ se define como:

$$\mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = K(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) - U(\boldsymbol{\rho}) \quad (2.33)$$

siendo $K(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ la energía cinética total de sistema y $U(\boldsymbol{\rho})$ la energía potencial de éste; $\boldsymbol{\tau}_e \in \mathbb{R}^m$ es el vector de fuerzas generalizadas externas y $\boldsymbol{\lambda} \in \mathbb{R}^r$ es el vector de multiplicadores de Lagrange, los cuales aseguran el cumplimiento de las restricciones no holonómicas dadas por

$$A(\boldsymbol{\rho})\dot{\boldsymbol{\rho}} = \mathbf{0}. \quad (2.34)$$

La ecuación (2.32) puede ser reescrita como:

$$M(\boldsymbol{\rho})\ddot{\boldsymbol{\rho}} + C(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})\dot{\boldsymbol{\rho}} + \boldsymbol{g}(\boldsymbol{\rho}) = \boldsymbol{\tau}_e + A(\boldsymbol{\rho})^T \boldsymbol{\lambda} \quad (2.35)$$

donde $M(\boldsymbol{\rho}) \in \mathbb{R}^{m \times m}$ es una matriz simétrica y definida positiva conocida como matriz de inercias, $C(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) \in \mathbb{R}^{m \times m}$ es la matriz de términos generados por fuerzas centrífugas y de Coriolis, $\boldsymbol{g}(\boldsymbol{\rho}) \in \mathbb{R}^m$ es el vector de fuerzas debidas a la gravedad. Además, como se explica en (Sciavicco y Siciliano, 2000), $M(\boldsymbol{\rho})$ y $\boldsymbol{g}(\boldsymbol{\rho})$ satisfacen las siguientes propiedades:

$$K(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \frac{1}{2} \dot{\boldsymbol{\rho}}^T M(\boldsymbol{\rho}) \dot{\boldsymbol{\rho}} \quad (2.36)$$

$$\boldsymbol{g}(\boldsymbol{\rho}) = \frac{\partial U(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \quad (2.37)$$

Por otro lado, se sabe que, dada la matriz $M(\boldsymbol{\rho})$, es posible calcular la matriz $C(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$

empleando los símbolos de Christoffel, los cuales están dados por:

$$c_{2js} = \frac{1}{2} \left(\frac{\partial M_{ij}}{\partial \rho_h} - \frac{\partial M_{ih}}{\partial \rho_j} - \frac{\partial M_{jh}}{\partial \rho_i} \right) \quad (2.38)$$

donde $M_{ij}(\rho)$ denota el ij -ésimo elemento de la matriz de inercia $M(\rho)$. Ciertamente, el kj -ésimo elemento $C_{kj}(\rho, \dot{\rho})$ de la matriz $C(\rho, \dot{\rho})$ está dado por

$$C_{kj}(\rho, \dot{\rho}) = [c_{1ka}(\rho) \quad c_{2ka}(\rho) \quad \dots \quad c_{nka}(\rho)] \dot{\rho} \quad (2.39)$$

Ahora bien, con el fin de proyectar la dinámica (2.35) al espacio tangente de la variedad de restricción (2.34) se necesita encontrar una matriz $S(\rho) \in \mathbb{R}^{n \times n}$ cuyo espacio columna pertenezca al espacio nulo de $A(\rho)$, i.e., $A(\rho)S(\rho) = 0$ (Soto, 2014). Entonces todas las velocidades dependientes factibles $\dot{\rho}$ del sistema restringido pertenecen al espacio generado por las columnas de $S(\rho)$, el cual está parametrizado por un vector de velocidades independientes $\dot{q} \in \mathbb{R}^n$, lo que lleva a la siguiente relación

$$\dot{\rho} = S(\rho)\dot{q}. \quad (2.40)$$

Nótese que para un sistema particular, la matriz $S(\rho)$ depende de la selección de los vectores de coordenadas generalizadas $\rho \in \mathbb{R}^n$ y de las coordenadas independientes $q \in \mathbb{R}^n$. Utilizando la matriz $S(\rho) \in \mathbb{R}^{n \times n}$ es posible reducir el sistema (2.35) a un nuevo sistema restringido dado de la forma:

$$M_r(\rho)\ddot{q} - C_r(\rho, \dot{q})\dot{q} + g_r(\rho) = \tau_r \quad (2.41)$$

donde

$$\begin{aligned} M_r(\rho) &= S(\rho)^T M(\rho) S(\rho) \\ C_r(\rho, \dot{q}) &= S(\rho)^T C(\rho, \dot{\rho}) S(\rho) + S(\rho)^T \dot{M}(\rho) \dot{S}(\rho) \\ g_r(\rho) &= S(\rho)^T g(\rho) \\ \tau_r &= S(\rho)^T \tau_\rho \end{aligned}$$

2.3.2. Método de Kane

Las dos formulaciones más frecuentemente usadas en el área de robótica para obtener el modelo dinámico son la de Euler-Lagrange (vista en la sección anterior) y la de Newton-Euler, pero éstas presentan ciertas desventajas, al momento de lidiar con sistemas con restricciones no holonómicas, como en el caso de los RMR. La formulación de Kane, por otro lado, permite eliminar fuerzas y pares que no son de interés, reduciendo el número de operaciones que se tienen que realizar para obtener el modelo dinámico del robot.

Para obtener el modelo dinámico de un sistema con restricciones, utilizando el método de Kane, es necesario en primer lugar reducir las variables con las que se va a trabajar, es decir, encontrar el vector de las velocidades mínimas $\dot{\mathbf{q}}$ en función del vector de las velocidades generalizadas (no mínimas) $\dot{\boldsymbol{\rho}}$ esto se logra utilizando la matriz $S(\boldsymbol{\rho})$ definida en (2.4C), y despejando el vector de velocidades mínimas de forma que:

$$\dot{\mathbf{q}} = S(\boldsymbol{\rho})^+ \dot{\boldsymbol{\rho}}$$

donde $S(\boldsymbol{\rho})^+$ es la pseudo-inversa izquierda de $S(\boldsymbol{\rho})$. Hay que recordar que la matriz $S(\boldsymbol{\rho})$ puede ser obtenida encontrando el espacio nulo de la matriz de restricciones no holonómicas $A(\boldsymbol{\rho})$.

Ahora, sabiendo que $\mathbf{q} \in \mathbb{R}^n$, donde n es el número de grados de libertad del sistema restringido, es necesario encontrar las velocidades lineales y angulares de los b eslabones o cuerpos con los que cuenta el sistema. La velocidad lineal de un cuerpo k , se denota aquí por \mathbf{v}_k , está dada por la derivada temporal del vector de posición del centro de masa del cuerpo, o sea:

$$\mathbf{v}_k = \dot{\mathbf{p}}_k = \frac{d}{dt} \mathbf{p}_k.$$

Por otro lado, cuando se utilizan matrices de rotación, se puede demostrar que la velocidad angular del cuerpo K está dada por (Campa y de la Torre, 2009)

$$\boldsymbol{\omega}_k = \begin{bmatrix} r_{21}\dot{r}_{31} + r_{22}\dot{r}_{32} + r_{23}\dot{r}_{33} \\ r_{31}\dot{r}_{11} + r_{32}\dot{r}_{12} + r_{33}\dot{r}_{13} \\ r_{11}\dot{r}_{21} + r_{12}\dot{r}_{22} + r_{13}\dot{r}_{23} \end{bmatrix} \quad (2.42)$$

donde r_{ij} y \dot{r}_{ij} , respectivamente, son los elementos i, j de la matriz de rotación R_k que

establece la orientación relativa del cuerpo con respecto a un marco de referencia y de su derivada \hat{R}_k . Una vez obtenidas las velocidades lineales y angulares v_k y ω_k en función de las variables generalizadas mínimas es posible obtener las matrices jacobianas de posición J_p y de orientación J_o , de la forma:

$$J_p = \left[\frac{\partial v_1(q, \dot{q})^T}{\partial \dot{q}} \quad \frac{\partial v_2(q, \dot{q})^T}{\partial \dot{q}} \quad \dots \quad \frac{\partial v_b(q, \dot{q})^T}{\partial \dot{q}} \right]^T \in \mathbb{R}^{n \times 3b} \quad (2.43)$$

$$J_o = \left[\frac{\partial \omega_1(q, \dot{q})^T}{\partial \dot{q}} \quad \frac{\partial \omega_2(q, \dot{q})^T}{\partial \dot{q}} \quad \dots \quad \frac{\partial \omega_b(q, \dot{q})^T}{\partial \dot{q}} \right]^T \in \mathbb{R}^{n \times 3b} \quad (2.44)$$

donde b es el número de eslabones o cuerpos del sistema.

Ahora es necesario calcular las fuerzas inerciales generalizadas que actúan en cada eslabón k , dichas fuerzas están dadas por:

$$\mu_k = \frac{d}{dt} (I_k \omega_k) \quad (2.45)$$

$$f_k = \frac{d}{dt} (m_k v_k) \quad (2.46)$$

donde I_k es el tensor de inercia del centro de masa del k -ésimo eslabón respecto al marco Σ_0 , y está dado por:

$$I_k = R_k {}^k I_k R_k^T$$

donde R_k es la matriz de rotación del cuerpo k , e ${}^k I_k$ es el tensor de inercia del centro de masa del k -ésimo eslabón con respecto al marco del mismo Σ_k , orientado de tal forma que solo tiene componentes en la diagonal, es decir

$${}^k I_k = \begin{bmatrix} I_{k_x} & 0 & 0 \\ 0 & I_{k_y} & 0 \\ 0 & 0 & I_{k_z} \end{bmatrix}$$

y m_k es la masa correspondiente al eslabón k .

Una vez obtenidas las fuerzas inerciales generalizadas μ_k y f_k , y las jacobianas de posición y orientación, J_v y J_ω , es posible aplicar la formulación de Kane para obtener la matriz de inercias $M(\mathbf{q})$ y la matriz centrífuga y de Coriolis $C(\mathbf{q}, \dot{\mathbf{q}})$ el modelo dinámico. Al hacer esto se llega a la siguiente expresión:

$$\begin{aligned}
 M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} &= J_v^T \mathbf{f} + J_\omega^T \boldsymbol{\mu} \\
 &= \sum_{k=1}^n \left[\left(\frac{\partial \mathbf{v}_k(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^T \frac{d}{dt} (m_k \mathbf{v}_k(\mathbf{q}, \dot{\mathbf{q}})) + \left(\frac{\partial \boldsymbol{\omega}_k(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^T \frac{d}{dt} (I_k(\mathbf{q}) \boldsymbol{\omega}_k(\mathbf{q}, \dot{\mathbf{q}})) \right] \\
 &= \sum_{k=1}^n \left[\left(\frac{\partial \mathbf{v}_k(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^T (m_k \dot{\mathbf{v}}_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})) + \left(\frac{\partial \boldsymbol{\omega}_k(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^T (I_k(\mathbf{q}) \dot{\boldsymbol{\omega}}_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \right. \\
 &\quad \left. + S(\boldsymbol{\omega}_k) I_k(\mathbf{q}) \boldsymbol{\omega}_k(\mathbf{q}, \dot{\mathbf{q}})) \right] \quad (2.47)
 \end{aligned}$$

donde

$$\begin{aligned}
 \mathbf{f} &= [f_1^T \quad f_2^T \quad \dots \quad f_n^T]^T \in \mathbb{R}^{3n} \\
 \boldsymbol{\mu} &= [\mu_1^T \quad \mu_2^T \quad \dots \quad \mu_n^T]^T \in \mathbb{R}^{3n}
 \end{aligned}$$

Resolviendo la ecuación (2.47) se obtienen los elementos de la matriz $M(\mathbf{q})$ la cual cumple con $M(\mathbf{q}) = M(\mathbf{q})^T > 0$. Aunque es posible obtener la matriz centrífuga y de Coriolis $C(\mathbf{q}, \dot{\mathbf{q}})$ de igual manera usando (2.47), generalmente es preferible hacerlo a partir de los símbolos de Christoffel (2.38), tal como se hizo con las ecuaciones de Euler-Lagrange. En el presente trabajo el vector de fuerzas gravitacionales $\mathbf{g}(\mathbf{q})$ se encuentra obteniendo la derivada parcial de la energía potencial $U(\mathbf{q})$ respecto a los estados, de forma que

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}$$

3 Planificación y control

Considerérese nuevamente el diagrama de la figura 1.1. Si la referencia deseada ρ_d es constante se dice que se tiene un objetivo de *control de regulación*, dado por

$$\lim_{t \rightarrow \infty} \rho(t) = \rho_d.$$

Pero si ρ_d es variante en el tiempo (es decir $\rho_d = \rho_d(t)$) entonces se habla de un objetivo de *control de seguimiento*:

$$\lim_{t \rightarrow \infty} \rho(t) = \rho_d(t).$$

En los robots manipuladores el control de regulación es más fácil que el controlador de seguimiento. Sin embargo en los RMR, debido a las restricciones no holonómicas que estos presentan el control de regulación no puede ser resuelto utilizando métodos clásicos de control; debido a esto en los RMR es más fácil diseñar un controlador de seguimiento, donde se especifique el camino que debe seguir el robot para moverse, que diseñar un controlador de regulación.

Es por esto que en el caso de robots móviles con ruedas, el problema de planificación de trayectorias, que tengan en cuenta las restricciones no holonómicas del sistema, cobra gran importancia.

Para lidiar con este tipo de problemas se han desarrollado diferentes clases de controladores dentro de los cuales los mas importantes son:

- *Controladores de realimentación continua variantes en el tiempo*: Estos controladores, además de depender de los estados del sistema, dependen también de la variable temporal t , es decir $u(x, t)$, en lugar de la clásica ley de control $u(x)$.
- *Controladores discontinuos*: Estos controladores tienen la forma clásica de la ley

de control $u(x)$ con la excepción de que la función u no es continua en el equilibrio que se desea estabilizar.

Por lo tanto, una de las alternativas más usadas para resolver el problema de control de los RMR ha sido la de diseñar controladores de seguimiento, los cuales son más fáciles de diseñar a comparación de algunas técnicas no lineales. Entonces el problema de control se convierte en uno de planificación de trayectorias que tengan en cuenta las restricciones no holonómicas del sistema.

3.1. Planificación de movimientos de un RMR

Al hablar de planificación, en el caso de los RMR, es común referirse ya sea a la planificación de movimientos o a la planificación de trayectorias. Para entender el problema de la planificación, considérese el llamado “problema de mover un piano”, en el cual un algoritmo tiene que determinar los movimientos que se tienen que realizar para mover el piano de un cuarto a otro. La planificación comúnmente ignora la dinámica del sistema y se centra principalmente en la traslación y rotación requeridas para lograr el movimiento del piano. Existen diferentes tipos de algoritmos de planificación que abordan diferentes tipos de problemas, los más importantes son (LaValla, 2006):

- *Planificación de movimientos:* Se refiere a determinar el camino que deberá recorrer el robot para llevarlo de un punto inicial P_0 a un punto final deseado P_f .
- *Planificación de trayectorias:* Usualmente se refiere al problema de tomar la solución de un algoritmo de planificación de movimientos y determinar cómo moverse a lo largo de la solución de forma que se cumplan las limitaciones mecánicas del robot, es decir, especificar en qué tiempos se tiene que pasar por cada punto del camino.
- *Planificación bajo restricciones diferenciales:* Es una variación de las planificaciones anteriores que se aplica a sistemas que tienen restricciones cinemáticas en la velocidad, como es el caso de las restricciones no holonómicas, que se encuentran en los RMR; estas restricciones son tomadas en cuenta al momento de planificar.

El problema de planificación bajo restricciones diferenciales puede ser visto como una extensión del problema más sencillo de planificación de movimientos. Y como los RMR son sistemas sujetos a restricciones diferenciales, a continuación se aborda con mayor profundidad este tema.

3.1.1. Planificación bajo restricciones diferenciales

En el área de robótica, las restricciones diferenciales (restricciones no holonómicas en nuestro caso) suelen surgir de la dinámica o la cinemática del robot; por ejemplo, se sabe que los robots móviles están sujetos a restricciones no holonómicas, que son un caso particular de las restricciones diferenciales, por lo tanto la planificación de movimientos de los RMR tiene que hacerse bajo el marco teórico de la planificación bajo restricciones diferenciales.

Una alternativa para resolver el problema de planificación de movimientos de un robot es ignorar las restricciones al momento de planificar y dejar que el controlador sea quien se encargue de producir las señales de control que no violen dichas restricciones; pero lo mejor es abordar este problema desde la planificación, lo que lleva a algoritmos de planificación que de manera directa cumplen con los movimientos naturales del sistema.

Como ya se vio en el capítulo 2, las ruedas tienen restricciones no holonómicas debido a restricciones en su velocidad; dichas restricciones en las ruedas se trasladan al robot. Este tipo de restricciones son uno de los muchos tipos de restricciones diferenciales. La planificación de movimientos bajo restricciones diferenciales puede verse como una variante del problema del plano, donde se da un estado inicial y un estado final. La tarea ahora entonces es generar un camino en el espacio de estado que conecte el punto inicial con el final y que satisfaga las restricciones diferenciales. Se supone entonces que las restricciones diferenciales están expresadas por la ecuación de transición de estado $\dot{x} = f(x, u)$. Entonces un camino que satisfaga las restricciones de movimiento deberá ser generado utilizando una trayectoria de acción u . Existen diversos métodos computacionales que permiten generar la trayectoria de acción que cumpla todas las restricciones. Tales métodos se tratan a detalle en la referencia (LaValle, 2006).

3.1.2. Planificación simple

Como se mencionó anteriormente, una alternativa a la planificación es proponer trayectorias sin tomar en cuenta las restricciones de movimiento del sistema. En el caso del unicycle esta alternativa es muy conveniente debido a su capacidad de girar en su propio eje, por lo cual es posible seguir cualquier trayectoria deseada continua sin ningún problema.

El modelo cinemático de velocidad en espacio de postura, de un robot unicycle requiere como comandos de entrada, una rapidez lineal v y una rapidez angular ω ,

entonces el planificador debe generar consignas de rapidez deseadas v_d y ω_d . En este trabajo se toma en cuenta la forma más sencilla de generar las consignas de rapidez deseadas a partir de una trayectoria deseada $\mathbf{p}_d(t)$. Sea $\mathbf{p}_d(t) = [x_d(t) \ y_d(t)]^T$, una trayectoria continua y doblemente diferenciable cualquiera definida para todo t , por lo que $\dot{\mathbf{p}}_d(t) = [\dot{x}_d(t) \ \dot{y}_d(t)]^T$ y $\ddot{\mathbf{p}}_d(t) = [\ddot{x}_d(t) \ \ddot{y}_d(t)]^T$, entonces es posible generar una consigna de rapidez lineal deseada, la cual está dada por:

$$v_d(t) = \sqrt{\dot{x}_d(t)^2 + \dot{y}_d(t)^2}. \quad (3.1)$$

Se sabe que el movimiento de un robot unicycle está restringido al ángulo de orientación de las ruedas, por lo que su desplazamiento a lo largo de la trayectoria debe ser tangencial a la misma. Esto permite definir el ángulo de orientación de robots tipo unicycle como:

$$\theta_d(t) = \tan^{-1} \left(\frac{\dot{y}_d(t)}{\dot{x}_d(t)} \right). \quad (3.2)$$

Obteniendo la derivada temporal de (3.2) es posible obtener la rapidez angular deseada que queda como:

$$\dot{\theta}_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \dot{x}_d(t)\ddot{y}_d(t)}{\dot{x}_d(t)^2 + \dot{y}_d(t)^2} \quad (3.3)$$

para cuando $v_d(t) \neq 0$.

Así que utilizando (3.1) y (3.3), es posible generar las referencias deseadas que deberá seguir el robot tipo unicycle para garantizar la trayectoria deseada definida por $\mathbf{p}_d(t)$.

3.2. Control de movimiento de un RMR

En el marco teórico del control automático, el sistema mecánico a controlar requiere como entradas los pares en cada una de sus articulaciones actuadas, y las salidas serán las variables de configuración que se quieren supervisar. Sea τ_i el par aplicado a la i -ésima articulación y $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \dots \ \tau_n]^T$ el vector de pares aplicados. Por otro lado, como ya se explicó al tratar el tema de modelado (capítulo 2) en los RMR generalmente

se considera que las señales de entrada al robot son la rapidez de avance en dirección de X_r y Y_r , así como la rapidez de giro al rededor del eje Z , las cuales se agrupan en el vector u .

Además, se supone que el robot cuenta con controladores de velocidad que dan como salida el par en las ruedas. En la figura 3.1 se muestra esta estructura de control:

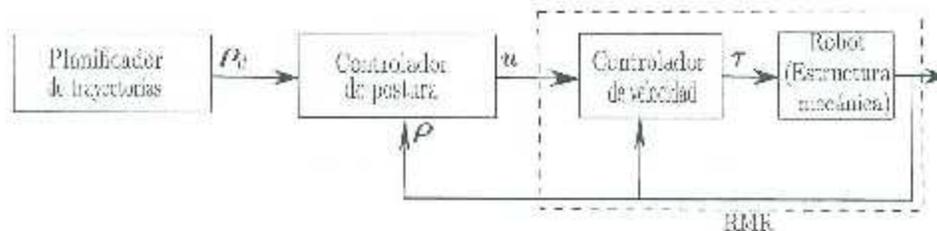


Figura 3.1: Estructura jerárquica de control de un RMR.

El objetivo del controlador de postura es generar las señales de entrada al robot (u) que se aseguren que el vector ρ converja asintóticamente a ρ_d . Cuando la salida del controlador de postura es un vector de velocidades se le conoce como un controlador cinemático y requiere que el controlador de velocidad interno (conocido como controlador dinámico) funcione correctamente.

Existe otro esquema de control que supone que la salida del controlador son los pares aplicados a la estructura mecánica; este esquema es apropiado para sistemas en los que se puede aplicar directamente los pares al robot, sin embargo en el caso de la mayoría de los RMR, este no es el caso y en caso de que se quiera aplicar un controlador dinámico es necesario transformar los pares de salida del mismo a señales de velocidad (En otras palabras, es necesario estimar u a partir de τ).

Teniendo en cuenta que la plataforma robótica PIR con la que se va a trabajar funciona como un robot móvil, pero al mismo tiempo es un sistema subactuado, se consideran dos problemas diferentes. El primero consiste en generar comandos de velocidad para un robot móvil tipo unicycle y el segundo se encarga de generar comandos de velocidad para el control de balanceo de un péndulo invertido sobre ruedas.

Es sabido que las restricciones no holonómicas con las que cuentan los RMR impiden estabilizar asintóticamente el origen utilizando una ley de control u lineal que dependa únicamente de las variables de estado del sistema, debido a que no se cumplen las condiciones del teorema de Brockett (Brockett et al., 1983), que se explica a continuación.

3.2.1. El problema de estabilización en RMR

El contenido de esta sección es tomado de (Luca y Oriolo, 1995). Considérese un sistema de control no lineal en la forma

$$\dot{x} = f(x) + G(x)u \quad (3.4)$$

El vector de estados x pertenece a \mathbb{R}^n y el vector de entradas de control a \mathbb{R}^m . Además, se supone que la matriz $G(x)$ es de rango completo.

El problema de estabilización para el sistema (3.4) consiste en encontrar una ley de control por realimentación de la forma

$$u = \alpha(x) \quad (3.5)$$

donde $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^m$, que haga que el punto de equilibrio (o una trayectoria admisible x_e) del sistema en lazo cerrado sea asintóticamente estable.

En el caso de los RMR, su modelo cinemático de velocidad tiene la forma

$$\dot{x} = G(x)u \quad (3.6)$$

que es un caso particular de (3.4). El siguiente teorema da condiciones necesarias para la existencia de una función estabilizante continua de la forma (3.5).

Teorema (Brockett)

Si el sistema

$$\dot{x} = \phi(x, u)$$

admite una realimentación continua $u = \alpha(x)$ que haga x_e asintóticamente estable, entonces la imagen del mapa

$$\phi : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$$

contiene alguna vecindad de x_e .

Quando se aplica a sistemas de la forma (3.6), dado que $G(x)$ es de rango completo, el teorema de Brockett implica que $m = n$ es una condición necesaria y suficiente para la estabilizabilidad usando una realimentación $u = \alpha(x)$ continua.

Sin embargo, salvo en el caso de RMR omnidireccionales, la condición $m = n$ no se cumple para la mayoría de los tipos de RMR. Esto significa que para este tipo de sistemas no es encontrar una ley de control estabilizante de la forma (3.5).

Para resolver el problema de estabilización en RMR es necesario entonces emplear otro tipo de señales de control, siendo las mas comunes las siguientes:

- Señales de realimentación de la forma $u = \alpha(x)$ discontinuas.
- Señales de realimentación variantes en el tiempo, es decir de la forma $u = \alpha(x, t)$.

3.2.2. Control de regulación

Para resolver el problema de regulación, el cual se encarga de llevar el sistema de una configuración inicial $\xi(0)$ a una configuración final ξ_d , se pueden consultar diversos trabajos en la literatura. En este documento sólo se abordan dos trabajos importantes.

El primero (Oriolo *et al.*, 2002), es un trabajo donde se diseña un controlador empleando la técnica de linealización por realimentación dinámica para controlar robots móviles. Se parte de un sistema de la forma

$$\dot{\xi} = R(\theta)u \quad \xi \in \mathbb{R}^n, \quad u \in \mathbb{R}^k \quad (3.7)$$

donde n es el número de variables de estado y k el número de entradas de control. De manera que en dicho trabajo se propone el siguiente compensador dinámico

$$\begin{aligned} \dot{\varepsilon} &= u_1 \cos \theta + u_2 \sin \theta \\ v_e &= \varepsilon \\ \omega_c &= \frac{u_2 \cos \theta - u_1 \sin \theta}{\varepsilon} \end{aligned} \quad (3.8)$$

donde θ es el ángulo de orientación, el compensador dinámico ε permitirá acoplar las leyes de control u_1 y u_2 , y transformarlas en comandos de velocidad v_e y ω_c que serán la entrada al robot. Por último, se eligen acciones de control tipo *PD* para u_1 y u_2 con lo

que queda:

$$\begin{aligned} u_1 &= -k_{p1}x - k_{d1}\dot{x} \\ u_2 &= -k_{p2}y - k_{d2}\dot{y} \end{aligned} \quad (3.9)$$

donde x y y son las posiciones del robot medidas con respecto a los ejes X y Y de un marco de referencia, k_{pi} son las ganancias proporcionales y k_{di} son las ganancias derivativas en cada eje, las cuales producen convergencia exponencial desde cualquier configuración inicial $\xi(0)$ al origen, bajo las siguientes hipótesis:

A1.- Las ganancias $k_{pi} > 0$, $k_{di} > 0$ ($i = 1, 2$) satisfacen las siguientes condiciones

$$k_{d1}^2 - 4k_{p1} - k_{d2}^2 - 4k_{p2} > 0 \quad (3.10)$$

$$k_{d2} - k_{d1} > 2\sqrt{k_{d2}^2 - 4k_{p2}} \quad (3.11)$$

A2.- El estado inicial de la variable auxiliar del compensador dinámico (3.8) $\varepsilon(0)$ será elegido como

$$\begin{aligned} \varepsilon(0) < 0 & \quad (\text{movimiento en reversa}) & \quad \text{si } x(0) \geq 0 \\ \varepsilon(0) > 0 & \quad (\text{movimiento hacia adelante}) & \quad \text{si } x(0) < 0 \end{aligned}$$

con $x(0)$ como la condición inicial en del robot en el eje X .

Un segundo trabajo donde se utilizan controladores no lineales para el control de regulación fue propuesto en (Tanner y Kyriakopoulos, 2003). En este método se utiliza la técnica de "backstepping" en conjunto con un controlador cinemático discontinuo que produce convergencia asintótica global del sistema al origen. En este trabajo se supone que se tiene el modelo dinámico del robot en la forma

$$\dot{x} = f(x) + g(u) \quad (3.12)$$

entonces se propone el siguiente controlador cinemático:

$$\dot{v}_d = \tanh(x)k_v[(y^2 - x^2)\cos(\theta) - 2xy\sin(\theta)] \quad (3.13)$$

$$\dot{\omega}_d = k_\omega \left[\tan^{-1} \left(\frac{2xy}{x^2 - y^2} \right) - \theta \right] \quad (3.14)$$

donde k_v y k_ω son ganancias positivas.

Esta ley de control genera una trayectoria que deberá seguir el robot para llegar al origen. Para asegurarse que el robot siga dicha trayectoria se aplica el método de backstepping el cual produce la aceleración lineal y angular del robot, w_1 y w_2 respectivamente, como acciones de control. Para transformar estas señales en los comandos que serán la entrada al robot es necesario integrar ambas señales, por lo que entonces se tiene:

$$v_c = \int w_1 dt \quad (3.15)$$

$$\omega_c = \int w_2 dt \quad (3.16)$$

estas aceleraciones están dadas por:

$$w_1 = -\tanh(x) \left[\frac{e^{\frac{-x}{v}} \tilde{v}^2 \tilde{\omega}^2 (x^2 - y^2) C_\theta}{(x^2 - y^2)^2 [\tilde{v}^2 + \tilde{\omega}^2]} + \frac{2e^{\frac{-x}{v}} \tilde{\omega} \tilde{v} xy S_\theta}{(x^2 + y^2)^2 [\tilde{v}^2 + \tilde{\omega}^2]} \right] - K_v \tilde{v}, \quad (3.17)$$

$$w_2 = -\tanh(x) \left[\frac{2e^{\frac{-x}{v}} \tilde{\omega}^2 xy S_\theta}{(x^2 + y^2)^2 [\tilde{v}^2 + \tilde{\omega}^2]} + \frac{e^{\frac{-x}{v}} \tilde{v} \tilde{\omega} (x^2 - y^2) C_\theta}{(x^2 + y^2)^2 [\tilde{v}^2 + \tilde{\omega}^2]} \right] - K_\omega \tilde{\omega}, \quad (3.18)$$

con $S_\theta = \sin\theta$, $C_\theta = \cos\theta$, $\tilde{v} = v - v_d$ y $\tilde{\omega} = \omega_d$.

Otras alternativas al control. Además de las ya mencionadas, existen otras técnicas usadas en la literatura tales como:

- *Bases de Philip Hall:* La idea es aplicar movimientos constantes a tramos para moverse en direcciones dadas por los bases de Philip Hall. Trabajos que utilizan este método pueden ser revisados en (Murray y Sastry, 1993), (Salerno y Angeles, 2003) y (Lamond *et al.*, 1994).
- *Acciones de control sinusoidales:* Este método se basa en el trabajo inicialmente hecho por Brockett. El método se emplea en un para gran número de sistemas en

los que la utilización de las bases de Philip Hall sería muy compleja. El método utiliza señales sinusoidales de diferentes frecuencias (múltiplos enteros) aplicando cada señal a una variable diferente, lo que permite modificar solo las variables de estado de interés. Trabajos que aplican este método pueden ser revisados en (Murray y Sastry (1993), Brockett (1982) y Tilbury *et al.* (1992)).

3.2.3. Control de seguimiento

Como ya se dijo anteriormente, es más fácil diseñar controladores de seguimiento para mover un RMR a través del espacio de configuración, por lo tanto, la manera más simple es proponer un controlador de seguimiento que se encargue de reducir el error entre las señales de velocidad deseadas v_d y ω_d , (las cuales pueden ser propuestas utilizando (3.1) y (3.3)) y las velocidades reales del robot v y ω . Esto puede ser logrado simplemente proponiendo controladores lineales PD, PI o PID, uno para cada una de las acciones de control.

Además de la aplicación de controladores lineales en la literatura, se encuentran controladores no lineales que ofrecen una mejor respuesta así como robustez ante perturbaciones. Uno de los trabajos más citados en la literatura, una vez más es el propuesto en (Oriolo *et al.*, 2002), el cual se describió anteriormente para el problema de regulación. En el caso de seguimiento primero se propone una trayectoria deseada dada $p_d(t)$ y se sabe que el ángulo de orientación deseado está definido por (3.2). De esta forma es posible utilizar el compensador dinámico (3.8) pero haciendo antes una modificación a las acciones de control PD, u_1 y u_2 , que ahora quedan definidas como

$$\begin{aligned} u_1 &= \ddot{x} + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y} + k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y}) \end{aligned} \quad (3.19)$$

con las ganancias PD elegidas únicamente bajo la restricción de que sean definidas positivas, es decir $k_{pi} > 0$ y $k_{di} > 0$, para $i = 1, 2$.

3.3. Control de balanceo de un PIR

En la sección anterior se abordó el problema de control de movimiento de un RMR. Cuando la plataforma Segway RMP-100 funciona en modo de autobalanceo es posible suponer que el controlador interno de balanceo funciona perfectamente y la plataforma se podría ver como un RMR tipo unicycle. En esta sección se considera el caso en el que se trabaja la plataforma en modo tractor y es necesario diseñar un controlador para el balanceo del péndulo invertido. En la presente tesis se trabajó con un controlador LQR, además de este método existe un gran número de estudios donde se aplican diferentes tipos de controladores, como la linealización parcial (Pathak *et al.*, 2005), el uso de redes neuronales (Anderson, 1989), control por modos deslizantes (Nawawi *et al.*, 2006), entre otros, que también pueden ser consultados en libros especializados en este tipo de plataformas (Li *et al.*, 2013).

3.3.1. El controlador LQR

Para aplicar este controlador, se parte de que se ha obtenido el modelo linealizado del sistema, el cual tiene la forma

$$\dot{z} = Az + B\bar{\tau} \in \mathbb{R}^{2n} \quad (3.20)$$

donde $z = [\mathbf{q} \quad \dot{\mathbf{q}}]^T \in \mathbb{R}^{2n}$ es el vector de variables de estado, $\bar{\tau} \in \mathbb{R}^k$ es el vector de fuerzas de entrada, $B \in \mathbb{R}^{n \times k}$ es la matriz que relaciona las fuerzas de entrada con el sistema y $A \in \mathbb{R}^{2n \times 2n}$ es la matriz de la dinámica interna del sistema. Es bien sabido por la literatura que el sistema (3.20) puede ser estabilizado por un control de realimentación de estados lineal dado por

$$\bar{\tau} = -Kz \in \mathbb{R}^k \quad (3.21)$$

donde $K \in \mathbb{R}^{k \times 2n}$ es la matriz de ganancias que está definida como

$$K = R^{-1}B^T P$$

donde P se calcula resolviendo la ecuación algebraica de Riccati dada por

$$A^T P + PA - PBR^{-1}B^T P - Q = 0.$$

Utilizando matrices diagonales definidas positivas Q y R con las dimensiones apropiadas, las cuales son ajustadas para modificar el desempeño del sistema, se asegura que la función costo

$$J = \int_0^{\infty} (z^T Q z + \bar{\tau}^T R \bar{\tau}) \quad (3.22)$$

sea minimizada. A esto se le conoce como controlador LQR.

Por lo tanto que para diseñar este control es necesario proponer unas matrices Q y R adecuadas, las cuales por simplicidad se consideran matrices diagonales.

4 La plataforma Segway RMP-100

En el laboratorio de Mecatrónica y Control de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de la Laguna se cuenta desde hace algunos años con un sistema robótico en el cual se pueden aplicar los algoritmos abordados en este trabajo; este sistema es la plataforma Segway RMP-100, la cual se muestra en la figura 4.1(a).

En este capítulo se dará primero una descripción de tal sistema, así como algunos datos que permitan entender las capacidades del mismo. Posteriormente, se presentan los modelos cinemático y dinámico del robot, que son de utilidad para el diseño de sus controladores.

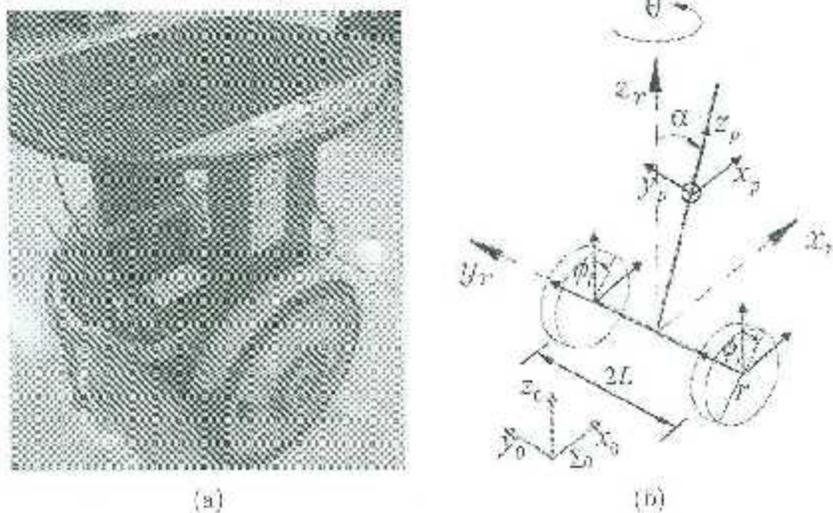


Figura 4.1: Segway RMP-100: (a) fotografía del robot; (b) diagrama cinemático.

4.1. Descripción y datos técnicos

La plataforma Segway RMP-100 es una versión dedicada para investigación de la versión comercial Segway PT (Personal Transporter). Las principales diferencias entre ambas versiones es que la RMP-100 no cuenta con un manubrio para sujetarse sino que en lugar de éste, cuenta con una plataforma circular diseñada para transporte de material o para que pueda incorporarse equipo extra al robot de manera fácil, ya que tiene agujeros con rosca para colocar tornillos y con esto asegurar el equipo a la plataforma; otra diferencia importante es la integración de una unidad de control, la cual se muestra en la figura 4.2(a). Esta unidad de control funciona como una interfaz desde la cual el usuario puede activar y desactivar la operación de la plataforma, así como otras funciones; además cuenta con un paro de emergencia.

Otra función de la unidad de control, es realizar la comunicación entre la computadora de control (externa) y el robot. De manera interna, la unidad de control trabaja utilizando un protocolo de comunicación bus CAN a una frecuencia de 100Hz. La plataforma Segway RMP-100 tiene dos modos de operación:

- *Modo autobalanceo:* En este modo se activa un controlador interno para mantener balanceada la plataforma de forma automática, este modo de operación permite diseñar controladores de movimiento tratando al robot como si fuera un robot

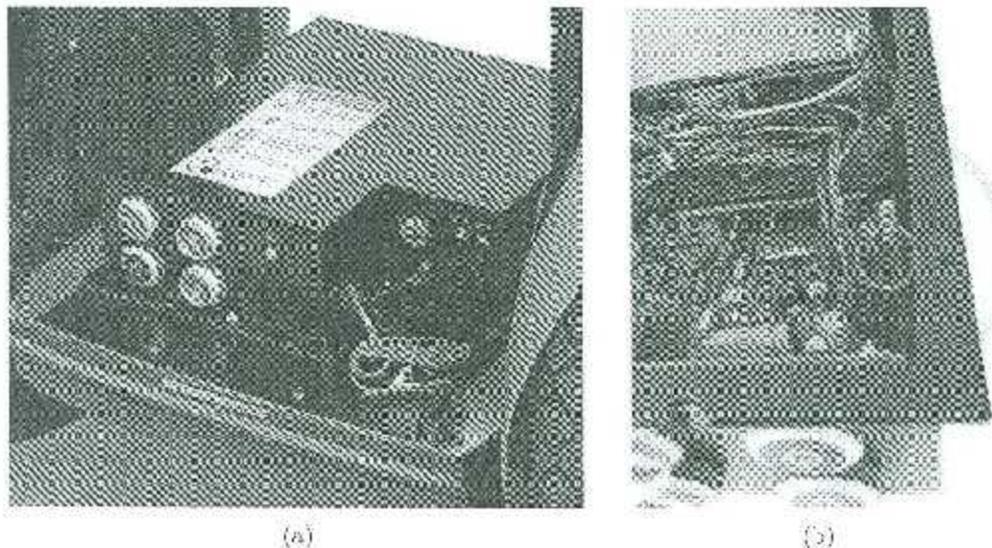


Figura 4.2: Unidad de control: (a) interfaz externa; (b) interior.

tipo unicycle.

- *Modo tractor:* En este modo de operación la plataforma no se mantendrá balanceada, por lo que es necesario agregar un tercer punto de apoyo para que la plataforma no caiga. Este modo de operación, al no contar con ningún tipo de controlador para el balanceo, permite probar algoritmos de control de diseño propio.

Para enviar los comandos indicados al robot, así como obtener las lecturas de los estados del robot, será necesario establecer una comunicación con la unidad de control a través de una computadora de control externa al robot. Para lograr esta comunicación, la plataforma puede utilizar alguno de los dos protocolos siguientes:

- *USB:* Este protocolo de comunicación es el que viene por omisión en la plataforma; para emplearlo se utiliza un cable USB estándar, el cual se conecta de la unidad de control a la computadora externa.
- *bus CAN:* Para usar este modo de comunicación es necesario desmontar la unidad de control de la base del robot; enseguida hay que localizar el conector "CAN A" el cual se señala en la figura 4.2(b). Se deberá usar un conector estándar para bus CAN. La computadora de control deberá tener un puerto de bus CAN compatible con el protocolo de comunicación bus CAN 2.0 para lograr la lectura correcta de la información del robot, así como para mandar los comandos necesarios. Para mayor información sobre el protocolo bus CAN, ver la sección 5.4, en el capítulo 5.

En la parte externa de la unidad de control se pueden encontrar cuatro botones, cada uno con una función específica, como se detalla a continuación:

- *Botón de inicio:* En la figura 4.3(a) se muestra un botón cuya función consiste en activar la comunicación y la mayoría de los componentes del sistema, con excepción de los motores del robot. Para poder realizar cualquier tarea con el robot este es el primer botón que se debe de activar.
- *Botón de activación de motores:* En la figura 4.3(b) se muestra un botón cuya función es activar los motores del robot; cuando se activa este botón, el robot empieza a moverse si recibe los comandos de velocidad indicados.
- *Botón para activación del modo tractor:* En la figura 4.3(c) se muestra un botón cuya función es activar el modo tractor del robot; cuando se activan los motores del robot, este modo es el que queda activado por omisión.

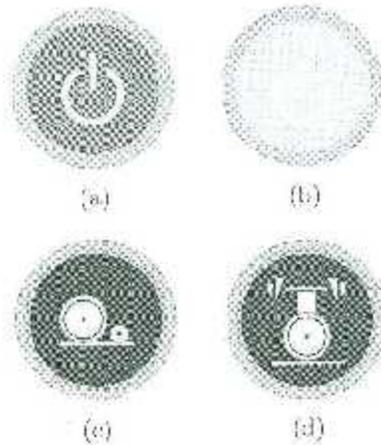


Figura 4.3: Botones de la interfaz: (a) inicio, (b) activación de motores, (c) modo de tractor y (d) modo de auto balanceo.

- *Botón para activación del modo autobalanceo:* En la figura 4.3(d) se muestra un botón cuya función es activar el controlador interno de balanceo con el que cuenta la plataforma, dicho controlador no entrará en funcionamiento si la plataforma no está en posición vertical en un inicio, con un error aproximado de $\pm 5^\circ$; esto por cuestiones de seguridad.

En la tabla 4.1 se presentan las características importantes del robot. Una de las características que más llaman la atención, es la gran capacidad de carga que posee, ya que, como se mencionó anteriormente, esta plataforma está diseñada para que sea fácil fijar a ella sensores láser, cámaras, brazos manipuladores, pantallas, etc. La capacidad de carga extra nominal es de 45 Kg.

4.2. Arquitectura de control

En esta sección se describir primero los elementos de actuación (motores) y de realimentación (sensores) con los que cuenta la plataforma; posteriormente se explica la forma en que las señales de control llegan hasta los motores y cómo las señales de realimentación provenientes de los sensores regresan hasta la computadora externa, conformando un esquema de control jerárquico con varios niveles de abstracción.

Tabla 4.1: Tabla de especificaciones.

Descripción	RMP-100
Máxima Velocidad	10 Km/h
Máximo comando de entrada de rapidez lineal v_c	2.8 m/s
Máximo comando de entrada de rapidez angular ω_c	2.5 rad/s
Capacidad de carga	45 Kg
Tipo de Batería	NiMH
Voltaje de entrada para el cargador	90 - 260 VCA; 50-60 Hz
Vida de la batería	300 a 500 ciclos de carga.
Temperatura de operación	0°C a 50°C
Rango de operación de carga completa	\approx 13 Km
Ancho total	56 cm.
Altura total	69 cm.
Largo total	41 cm.
Peso	57 Kg
Constante de par del motor	0.054 Nm/A.
Pico de corriente de cada motor	70 A
Corriente nominal (operación continua)	24 amp
Relación de engranaje en las ruedas	24:1

4.2.1. Actuadores y sensores

La plataforma cuenta con dos servo motores del tipo de CD sin escobillas (brushless DC) que se encuentran instalados uno en cada rueda. Estos motores están en una configuración redundante, es decir, cuentan con un embobinado doble que ayuda a prevenir accidentes en el robot debidos a fallas en los motores; es decir, si se detecta una falla en algún embobinado del motor, el otro embobinado compensa la falta del primero, evitando mayores problemas. En la figura 4.4 se muestra una fotografía de uno de los motores. Para el funcionamiento de los motores se emplean cuatro manejadores (drivers) usando dos para cada motor, es decir uno para cada embobinado. En la figura 4.5 se muestra la tarjeta electrónica que contiene los manejadores de un motor. Bajo condiciones normales ambos embobinados trabajan en paralelo, compartiendo la carga. Estos motores son capaces de mantener una potencia de 2 Hp.

Por otra parte, la plataforma Segway RMP-100 cuenta con sensores para medir los diferentes estados de la plataforma; básicamente son:

- *Encoders de efecto Hall:* Se encuentran en las ruedas y se encargan de monitorear la posición y velocidad de cada una de éstas.

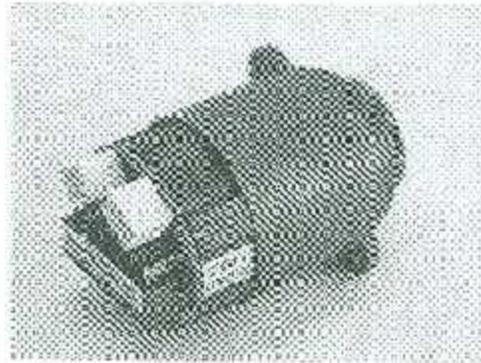


Figura 4.4: Fotografía del motor de la plataforma.

- *Sensores de orientación:* La plataforma cuenta con un conjunto de sensores de balanceo; se trata de cinco giroscopios de estado sólido capaces de medir las velocidades angulares mediante el uso del efecto de Coriolis. Estos sensores se encuentran en un arreglo especial para medir los ángulos y velocidades de elevación (del péndulo respecto al eje Z) e inclinación (respecto al eje X). Este arreglo de sensores se encuentra instalado en la base de la plataforma. Una característica importante de este arreglo de sensores es la detección absoluta del ángulo de inclinación mediante el uso de sensores con fluidos electrolíticos que ayudan a detectar la dirección de la fuerza de gravedad, por lo que no es necesario inicializar la plataforma en 0° para su correcto funcionamiento. En la figura 4.6 se muestra dicho arreglo de sensores.
- *Medidores de corriente:* Se encargan de medir la corriente consumida por cada motor; en conjunto con la constante del motor y la relación de engranaje permiten

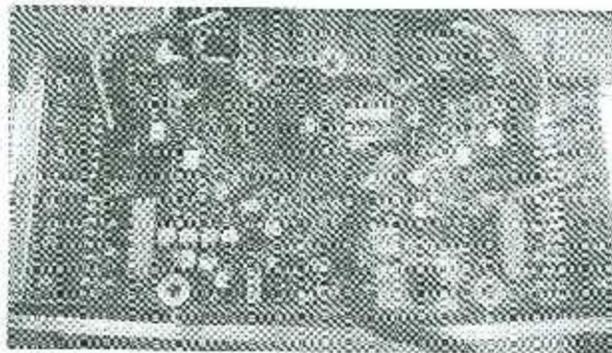


Figura 4.5: Fotografía de la tarjeta de manejadores de la plataforma.



Figura 4.6: Sensores de orientación.

tener la medida del par aplicado en cada rueda. Estos sensores se encuentran en los manejadores de los motores.

4.2.2. Estructura de control jerárquica

La RMP-100 entra en la categoría de RMR tipo (2,3) (ver sección 2.1), por lo que el vector de entradas de control está dado por $\mathbf{u} = [v_c \quad \omega_c]^T$. Se renombrarán las consignas de control como $v_c = u_x$ y $\omega_c = u_\theta$, por lo que ahora el vector de entradas está dado por:

$$\mathbf{u} = \begin{bmatrix} v_c & \omega_c \end{bmatrix}^T, \quad (4.1)$$

Como se explicó en la sección 4.1 la computadora de control envía a la plataforma los comandos de rapidez de desplazamiento (v_c) y la rapidez de giro (ω_c). Esas señales de control son recibidas (ya sea por USB o bus CAN) por la que se denomina interfaz de control (ver figura 4.7), que se encarga no solo de procesar las señales del operador (selección de modos de operación, arranque, paro de emergencia, etc.) sino también de comunicarse con las unidades de control de cada rueda que son las que finalmente se encargan de implementar un control de corriente para cada rueda, así como también de regresar a la interfaz de control las señales leídas de los sensores. En la figura 4.8 se muestra un diagrama que ilustra los diferentes bloques de comunicación y control en la

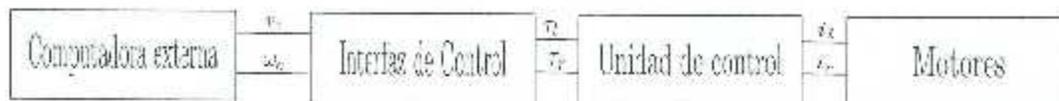


Figura 4.7: Diagrama de bloques de la estructura de control de la plataforma.

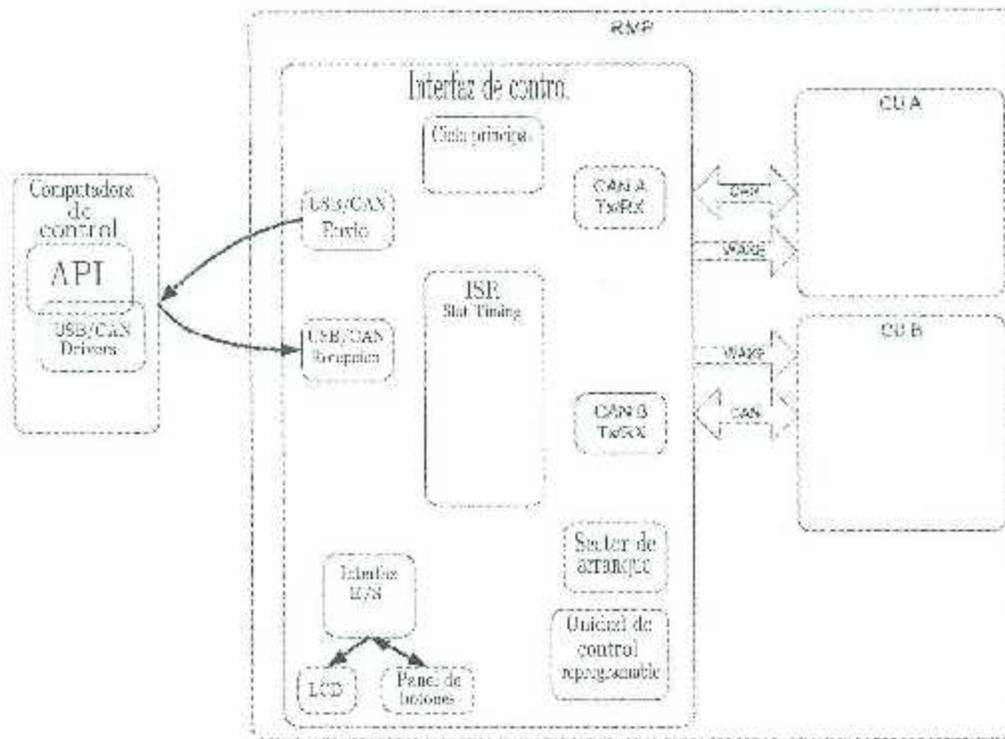


Figura 4.8: Diagrama de comunicación/control de la plataforma.

plataforma RMP-100.

4.3. Modelado de la plataforma

Para el diseño de la mayoría de los controladores es necesario obtener algún modelo de la plataforma (cinemático o dinámico); algunos controladores sólo necesitan del modelo cinemático pero otros, como por ejemplo los controladores de balanceo, necesitan además el modelo dinámico para poder calcular las leyes de control. En esta sección sólo se presentan los resultados obtenidos de la aplicación de los métodos de modelado presentados en el capítulo 2; el procedimiento detallado para llegar a estos modelos se deja en los apéndices A y B.

4.3.1. Modelo cinemático de velocidad

Siguiendo los dos métodos descritos en el capítulo 2 (Muir y Neuman (1987) y Campion *et al.* (1996)), se obtuvieron el modelo cinemático de velocidad en espacio de postura y el modelo cinemático de velocidad en espacio de configuración.

Modelo cinemático de velocidad en espacio de postura

Está dado por la siguiente ecuación:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega_c \end{bmatrix} \quad (4.2)$$

donde

- \dot{x} : Es la componente de la velocidad lineal en la dirección x del marco de referencia.
- \dot{y} : Es la componente de la velocidad lineal en la dirección y del marco de referencia.
- $\dot{\theta}$: Es la componente de la velocidad angular del robot medida alrededor del eje z del marco de referencia.
- θ : Es el ángulo de orientación del marco del robot medido con respecto al marco de referencia.
- v : Es la rapidez de avance del robot, perpendicular al eje de las ruedas.
- ω : Es la rapidez de giro del robot.

Modelo cinemático de velocidad en espacio de configuración

Está dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_l \\ \dot{\varphi}_r \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{r} & \frac{-L}{r} & 0 \\ \frac{1}{r} & \frac{L}{r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \\ \dot{\alpha} \end{bmatrix} \quad (4.3)$$

donde, además de las variables descritas anteriormente, se tiene que

- α : Es la componente de la velocidad angular del péndulo del robot medida respecto al eje y del marco montado en el robot.
- φ_l : Es la velocidad angular de la rueda izquierda del robot.
- φ_r : Es la velocidad angular de la rueda derecha del motor.
- L : Es la distancia que existe del punto medio del robot hacia las ruedas medido a lo largo del eje de las ruedas.
- r : Es el radio de las ruedas.

4.3.2. Modelo dinámico

Si se selecciona como vector de coordenadas de configuración mínimas

$$\mathbf{q} = [x, \theta, \alpha]^T \in \mathbb{R}^3,$$

donde x , indica el desplazamiento de avance del robot, entonces el modelo dinámico queda

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \boldsymbol{\tau}, \quad (4.4)$$

donde la matriz de inercias $M(\mathbf{q})$ tiene la forma

$$M = \begin{bmatrix} m_{11} & 0 & m_{13} \\ 0 & m_{22} & 0 \\ m_{13} & 0 & m_{33} \end{bmatrix}$$

y cuyos elementos están dados por:

$$\begin{aligned} m_{11} &= \frac{1}{r^2}(I_{iy} + I_{ry}) + (m_l + m_p + m_r) \\ m_{13} &= m_p l_p \cos(\alpha) \\ m_{22} &= I_{ix} + I_{px} + I_{rx} + (m_l + m_r) L^2 \\ &\quad + (I_{rx} - I_{rx}) \sin^2(\alpha) + (I_{iy} - I_{ry}) \frac{L^2}{r^2} + m_p l_p^2 \sin^2(\alpha) \\ m_{33} &= m_p l_p^2 + I_{py} \end{aligned}$$

La matriz centrífuga y de Coriolis $C(\mathbf{q}, \dot{\mathbf{q}})$ está dada por:

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & -c_2 \dot{\alpha} \\ 0 & c_1 \dot{\alpha} & \dot{c} \\ 0 & -c_1 \dot{\theta} & 0 \end{bmatrix}$$

con

$$\begin{aligned} c_1 &= (m_p l_p^2 + (I_{px} - I_{pz})) \cos(\alpha) \sin(\alpha) \\ c_2 &= m_p l_p \sin(\alpha) \end{aligned}$$

y el vector de gravedad es

$$g(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ -g m_p l_p \sin(\alpha) \end{bmatrix}$$

En la tabla 4.2 se muestran los valores y la descripción de los parámetros que aparecen en el modelo dinámico. Algunos de estos parámetros fueron tomados de las especificaciones del robot; otros fueron estimados numéricamente a partir del modelo CAD del robot.

Tabla 4.2: Parámetros del modelo dinámico de la plataforma Segway RMP-100.

Abreviatura	Descripción	Valor
r	radio de las ruedas	0.2000 m
L	distancia media entre ruedas	0.2340 m
l_p	distancia al centro de masa del péndulo	0.1513 m
m_p	masa del péndulo	53.6825 kg
m_r	masa de la rueda derecha	2.3500 kg
m_l	masa de la rueda izquierda	2.3500 kg
I_{p_y}	momento de inercia del péndulo en el eje Y	3.5920 kgm ²
I_{p_z}	momento de inercia del péndulo en el eje Z	1.8394 kgm ²
I_{p_x}	momento de inercia del péndulo en el eje X	0.0000 kgm ²
I_{l_y}	momento de inercia de la rueda izquierda en el eje Y	0.0550 kgm ²
I_{l_z}	momento de inercia de la rueda izquierda en el eje Z	0.0282 kgm ²
I_{r_y}	momento de inercia de la rueda derecha en el eje Y	0.0550 kgm ²
I_{r_z}	momento de inercia de la rueda derecha en el eje Z	0.0282 kgm ²

5 Interfaz de software

Una interfaz es la conexión funcional entre dos sistemas o dispositivos de cualquier tipo que produce una comunicación a diferentes niveles. En este capítulo se describe la interfaz de software desarrollada para comunicar la computadora de control con la plataforma RMP-100. Básicamente esta interfaz se encarga de recibir los datos enviados por la plataforma y traducirlos en información útil que se empleará para el cálculo de las señales de control en la computadora de control. Una vez obtenidas las leyes de control, la interfaz deberá convertir esta información en mensajes que pueda entender la plataforma Segway RMP-100.

Como ya se mencionó anteriormente, la comunicación entre la plataforma y la computadora de control se logra a través de comunicación USB ó bus CAN.

5.1. Interfaz original

Cuando se adquirió la plataforma, ésta venía con un programa denominado RMI_Demo (o simplemente Demo), que mostraba en pantalla una interfaz gráfica de usuario (GUI) con la cual se podía establecer comunicación con el robot. El código del programa Demo está disponible así como todas las bibliotecas de funciones que se requieren para su compilación. En la figura 5.1 se muestra la GUI del programa Demo.

La principal desventaja de la interfaz Demo es que no funciona en tiempo real, y eso limita su aplicación en tareas de control que requieren un alto desempeño.

En la sección siguiente se describe más a detalle qué es un sistema de control en tiempo real, y cómo se puede implementar un sistema de este tipo en una computadora con el sistema operativo Windows.

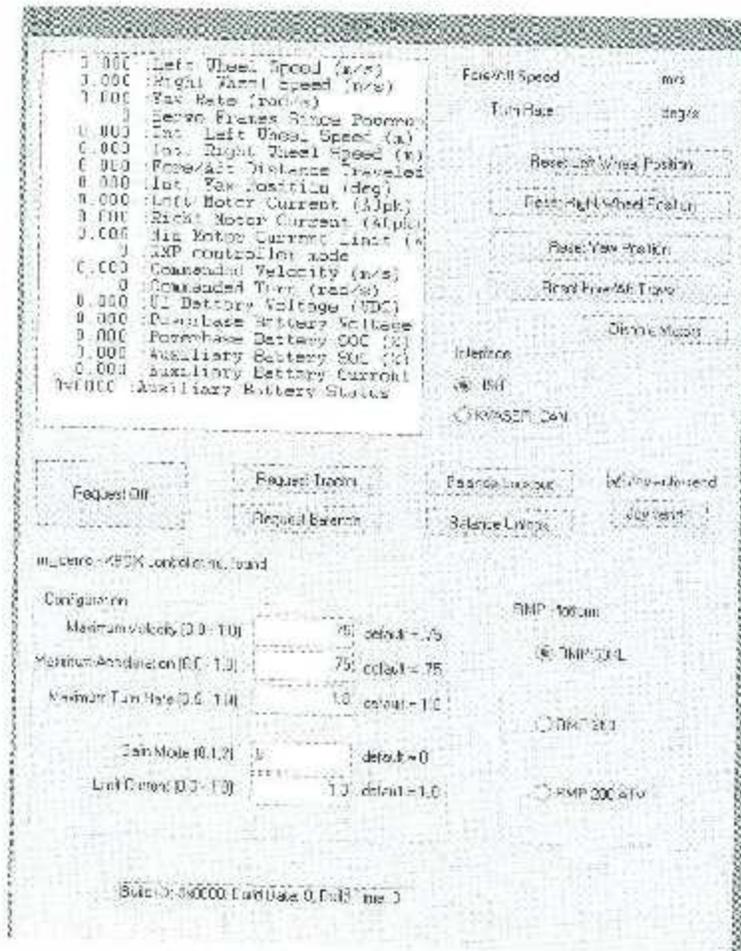


Figura 5.1: Interfaz original.

5.2. Control en tiempo real en Windows

Como se explica en (Campa, 2005), la implementación de sistemas de control para sistemas electromecánicos, realizando tareas a altas velocidades, requiere ya sea de hardware dedicado o software de control con características de tiempo real. Un sistema de tiempo real es aquel donde es imperativo que la respuesta ocurra dentro del intervalo de tiempo requerido.

Existen típicamente dos arquitecturas de hardware para implementar sistemas de control de tiempo real (Costescu *et al.* (1999); Monroy *et al.* (2001)):

- El esquema tradicional consiste de dos computadoras trabajando en paralelo: una computadora de propósito general (típicamente una PC), y una computadora

dedicada, tal como un procesador digital de señales (DSP), equipada con una tarjeta adquisidora de datos. El programa de control es desarrollado en la computadora maestra y luego es transferido al DSP, desde donde se ejecuta. Los datos obtenidos durante la ejecución son transferidos a la computadora maestra para tareas de monitoreo.

- El segundo esquema consta de una única computadora personal (con un solo procesador) equipada con una tarjeta adquisidora de datos. Para que este esquema funcione, es necesario que la computadora cuente con un sistema operativo de tiempo real que permita cumplir con los requerimientos de tiempo del algoritmo de control.

Un sistema operativo de tiempo real (RTOS, por sus siglas en inglés) es un sistema diseñado para trabajar con aplicaciones de tiempo real, es decir, aplicaciones que procesan datos sin retrasos. Una característica importante de los RTOS es el nivel de su consistencia relativa a la cantidad de tiempo que se necesita para aceptar y completar tareas de una aplicación, esta cualidad se llama *jitter*. Un sistema de tiempo real “duro” tiene menor *jitter* que un sistema de tiempo real “suave”, esto quiere decir que los RTOS duros aceptan y completan las tareas de una forma más exacta respecto al tiempo en el que fueron requeridas, es decir el retraso entre el tiempo en el que se llama una función y el tiempo en que se ejecuta es casi inexistente.

5.2.1. RTX: Una extensión de tiempo real para MS-Windows

Los sistemas operativos de la familia MS-Windows no son sistemas operativos de tiempo real y por ello sus aplicaciones no son empleadas en tareas de control de procesos.

Para superar esta limitante, algunas compañías de software han desarrollado módulos comerciales que se instalan sobre el sistema operativo Windows y le dan características de sistema operativo de tiempo real. Uno de estos módulos es el denominado RTX (de Real-Time eXtension) de Interval Zero. RTX permite que Windows funcione a la vez como un sistema de propósito general y un sistema operativo del tiempo real de alto desempeño. Esto es posible gracias a la característica de manejar dos diferentes clases de procesos de tiempo real:

- *Procesos Win32*: Usan un subconjunto de las funciones estándar de Windows (Win32 API) para conservar la ventaja de usar interfaces gráficas (GUIs) y otras

funciones de la API de Windows, pero también se pueden usar algunas funciones de tiempo real de RTX. Al ser manejadas por el calendarizador de Win32, estos procesos no son usados para tareas de tiempo real duro.

- *Procesos RTSS:* Son ejecutados sobre el subsistema de tiempo real (RTSS) que es instalado por el RTX. Usan su propio calendarizador para procesos RTSS, con una prioridad mayor que el calendarizador de Win32, de manera que estos procesos pueden lograr tareas de tiempo real duro.

Utilizar procesos de tiempo real tiene algunas ventajas, como por ejemplo la utilización de servicios de reloj más rápidos, y con esto los procesos pueden ser ejecutados a altas velocidades (hasta 10KHz en procesos RTSS), se tiene un rango completo de prioridades para hilos (de 0 hasta 127 en procesos RTSS) y la comunicación entre procesos a través del uso de memoria compartida. En la figura 5.2 se muestra cómo queda la arquitectura de Windows una vez que ha sido instalada la extensión de RTX. Desde el punto de vista de procesos de Windows, un proceso RTSS es equivalente a ir y volver de una interrupción, ya que ésta sólo realiza cierta tarea cada que un temporizador alcanza cierto nivel de activación, y mientras este temporizador no se active, el proceso RTSS está simplemente en estado de espera, liberando el procesador para que realice otras tareas.

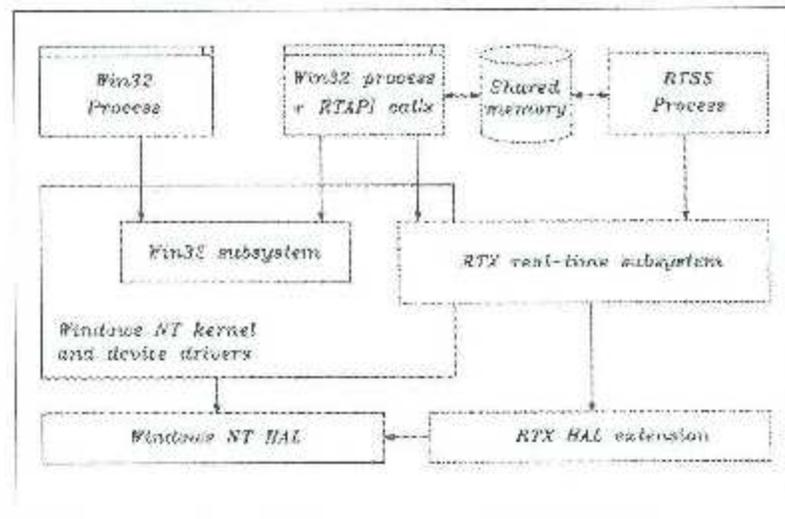


Figura 5.2: Arquitectura de MS-Windows con RTX.

5.2.2. Programación con RTX en MS-Visual Studio

Para la realización de este proyecto se utilizó la versión 7.0 de RTX, en la cual se desarrollaron los programas a utilizar, y los cuales tienen las siguientes características:

- La programación se realiza utilizando MS-Visual Studio C++ 2005 (al instalar RTX se agrega un asistente para la creación de proyectos en tiempo real).
- Existen cuatro tipos de proyectos para RTX:
 - Win32 (Debug/Release) generan ejecutables .EXE
 - RTSS (Debug/Release) generan ejecutables .RTSS
- Los procesos .RTSS se ejecutan utilizando el programa RTSSRUN.EXE, el cual se instala junto con el RTX.

La biblioteca de funciones API para tiempo real de RTX incluye funciones para manejo de:

- Relojes y temporizadores.
- Excepciones.
- Interrupciones.
- Comunicación entre procesos.
- Memoria compartida.
- Puertos E/S.
- Procesos e hilos.

Siendo las funciones para uso de temporizadores y de memoria compartida los más importantes utilizadas para el desarrollo de la interfaz de la plataforma Segway RMP-100. El término memoria compartida se refiere a una región de memoria física a la cual diferentes procesos pueden tener acceso de manera simultánea, esto permite compartir información entre los diferentes procesos que tengan acceso a dicha memoria compartida.

En (Campa *et al.*, 2004) se describe el software WinMechlab que fue desarrollado para el control en tiempo real de un robot manipulador de dos grados de libertad. El WinMechlab hace uso de un archivo de cabecera (.h) en el cual se definen 6 funciones con las cuales se puede desarrollar un programa en tiempo real con acceso a memoria compartida. Las funciones son las siguientes:

- INICIALIZAR_PROGRAMA(long, int, RTFCONDCT)
- EJECUTAR_RUTINA_PERIODICA(int, long)
- TERMINAR_PROGRAMA()
- ABRIR_MEMORIA_COMPARTIDA(type*,char*)
- type LEER_MEMORIA_COMPARTIDA(type*)
- ESCRIBIR_MEMOBIA_COMPARTIDA(type,type*).

Para generar un proceso de tiempo real del tipo Win32 se debe hacer uso del manejador de eventos de Windows. Se parte de un proyecto gráfico tipo Win32 que utilice la biblioteca MFC de Windows. Suponiendo que tal proyecto se nombra display, al generarse el proyecto se crean varios archivos entre los cuales se encuentra display.cpp que contiene las funciones que manejan las instrucciones internas de la aplicación, y displayDlg que incluye todas las funciones para manejo de eventos en la interfaz gráfica. En el constructor de la aplicación (en display.cpp) se deben poner las funciones para abrir la memoria compartida y crear un temporizador que cada vez que se activa llama a la función RUTINA_PERIODICA. En esta función es donde se colocan las instrucciones que se requieran ejecutar periódicamente. Además si se quiere que la interfaz actualice información en la ventana, se debe incluir en la rutina periódica una función que genere el evento WM_PAINT que llama a la función OnPaint en displayDlg.cpp (ver figura 5.3).

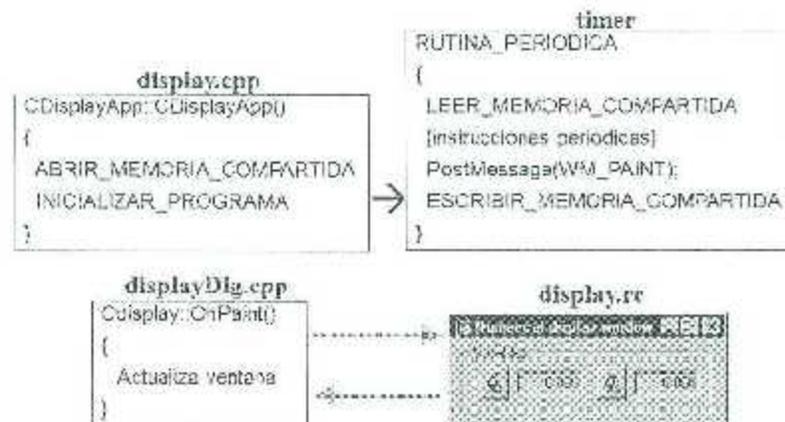


Figura 5.3: Programación de procesos WIN32.

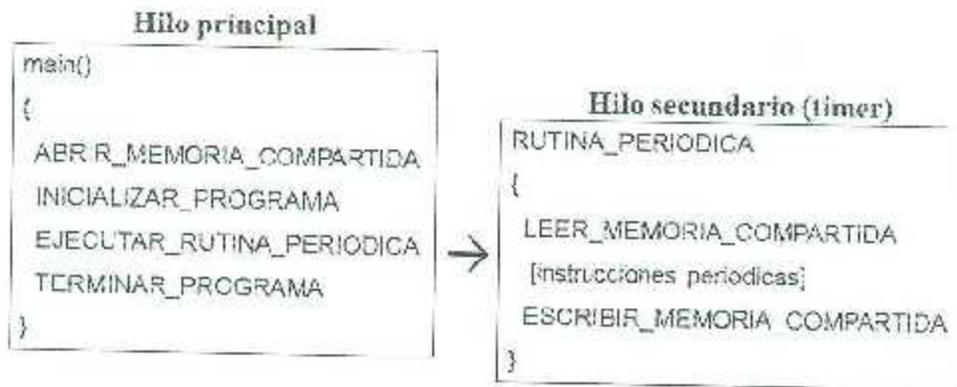


Figura 5.4: Programación de procesos RTSS.

Para generar un proceso de tiempo real duro, es decir RTSS, se debe hacer uso del manejador de eventos de RTX. Con la instalación de RTX se puede crear un nuevo tipo de proyectos para MS-Visual Studio, los cuales incluyen las bibliotecas de funciones y las configuraciones necesarias para generar un proceso RTSS. Un proyecto de este tipo es más simple que un proceso WIN32 ya que funciona en modo consola. Dentro de la función `main()` se debe abrir la memoria compartida y activar el temporizador que llama a la rutina periódica como un hilo secundario del proceso (ver figura 5.4).

5.3. Comunicación vía bus CAN

En esta sección se dan detalles del protocolo de comunicación bus CAN que se usó en la nueva interfaz de software desarrollado. Se eligió este protocolo debido a que no fue posible obtener un tiempo de operación menor a 30ms, cuando se usaba la comunicación por USB. Así que fue necesario implementar la comunicación entre la computadora y la unidad de control a través del protocolo de comunicación bus CAN.

5.3.1. Protocolo de comunicación bus CAN

Como se explica en (Zúñiga, 2014), el protocolo CAN (Controller Area Network) es un concepto de bus serie multi-maestro desarrollado por la firma alemana Robert Bosch GmbH, en 1986. Este protocolo fue desarrollado para llevar a cabo el enlace entre contro-

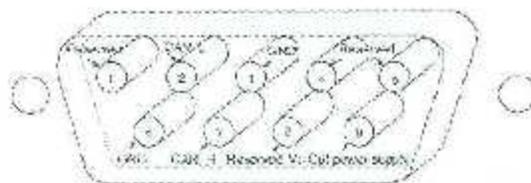


Figura 5.5: Conector CAN definido por CIA (DS-102-1).

ladores, actuadores y sensores, y está provisto de una elevada velocidad de transmisión combinada con una importante inmunidad frente a interferencias electromagnéticas.

El protocolo de comunicación bus CAN tiene dos tipos de estándares, la versión de altas velocidades (hasta 1 Mbps) y la versión para aplicaciones de baja velocidad (hasta 120kbps). La comunicación con la plataforma Segway RMP-100 requiere del uso del estándar de alta velocidad. Los tipos de conectores a utilizar para las líneas de transmisión no están definidos por el estándar original pero si fueron definidos por otros estándares. En la figura 5.5 se muestra la distribución de pines de un conector de 9 pines utilizado para la tarjeta CAN establecido por el estándar CIA DS 102 1, y en la tabla 5.1 se detalla la distribución de los pines del conector bus CAN que tiene la interfaz de control mostrado en la figura 4.2(b). Otro punto importante sobre el protocolo es la trama de datos, es decir, cómo son generados los mensajes CAN. Las tramas de datos son usadas para transmitir información entre un nodo emisor y uno o más nodos receptores. Las tramas de datos no usan explícitamente direcciones para identificar los receptores del mensajes. En su lugar, cada nodo receptor establece los mensajes que serán recibidos según el contenido de la información, la cual está codificada en el campo identificador de la trama. Existen dos formatos de tramas de datos definidos en la especificación CAN 2.0: la normal (CAN 2.0a) y la extendida (CAN 2.0b). La trama normal consta de 11 bits para la identificación del mensaje (ID) y es la utilizada en los mensajes de la plataforma Segway RMP-100. La figura 5.6 muestra el formato que debe seguir cada mensaje CAN enviado. Los dos primeros bytes del mensaje se

Tabla 5.1: Distribución de pines del conector del bus CAN.

PIN	SEÑAL
1	CAN H
2	CAN L
3	Tierra

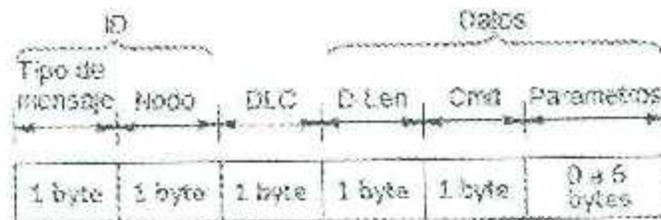


Figura 5.6: Contenido de un mensaje CAN2.0a.

utilizan para especificar el identificador (ID) del mensaje en el formato estándar, lo cual significa que únicamente se emplean 11 bits de los 2 bytes.

5.3.2. Programación de la tarjeta CAN-AC2-PCI de Softing

Para realizar la comunicación con la plataforma Sogway RMP-100, se utilizó una tarjeta bus CAN con la que ya se contaba en el laboratorio de Mecatrónica y Control. La tarjeta CAN-AC2-PCI de Softing es una tarjeta compatible con Windows 7, XP o 2000. La cual puede ser utilizada con la API *CAN Layer 2* utilizando C++.

Las tarjetas interfaz CAN de Softing pueden ser utilizadas en dos modos de operación: En modo FIFO y en modo buffer de objetos CAN:

- *Modo FIFO*: La comunicación entre el bus CAN y la aplicación de la computadora es procesada secuencialmente mediante el uso de FIFOs. El mensaje que llega primero al buffer es el primero en ser procesado.
- *Modo buffer de objetos*: En este modo los mensajes y sus datos son guardados en dos listas de objetos, una para transmisión y otra para recepción, de esta forma puede guardar hasta 2048 mensajes en cada buffer.

La ventaja de utilizar el modo de buffer de objetos es que todos los mensajes están disponibles para su procesamiento en cualquier instante, ya que se puede tener acceso a cualquier mensaje sin importar el orden en el que fueron recibidos. Para la programación de la tarjeta se utilizó el modo buffer de objetos ya que permite tener acceso de forma más rápida a los mensajes enviados por la plataforma.

5.3.3. Mensajes CAN de la RMP-100

En el caso de la RMP-100, se envían 8 mensajes diferentes con diferentes identificadores (ID's 0x401-0x408), a una frecuencia de 100Hz, con la información sobre el estado del robot. La computadora debe responder con los comandos de velocidad a una frecuencia no menor a 5Hz y no mayor a 100Hz. Además de los comandos de velocidad es posible mandar otro tipo de instrucciones al robot, como el cambio de modo de operación, reiniciar el estado de las variables o apagar motores.

En esta sección se desglosa el contenido de los mensajes más importantes que son enviados y recibidos por la interfaz de control. Para describir los mensajes se utilizan los siguientes identificadores:

- *Source*: Será el encargado de producir y enviar el mensaje.
- *Destination*: Se encarga de recibir el mensaje.
- *Header*: Identificador del mensaje.
- *Byte X*: Número de byte enviado (del 1 al 8), además, se incluyen algunas características importantes con este dato.

Mensaje de Heartbeat de la plataforma

Este mensaje es usado para mantener el procesador del controlador interno activo. Debe ser enviado por la interfaz de control cada 10ms. La información importante de este mensaje está dada por:

- *Source*: Interfaz de control.
- *Destination*: Controlador interno.
- *Header*: 0xC688.
- *Byte 1*: No se usa.
- *Byte 2*: Identificador del CAN, 0 = CAN A y 1 = CAN B (tipo char).
- *Byte 3-4*: Estado de la batería [0,1,2] (tipo int).
- *Byte 5-6*: Voltaje de la batería [0-500] (tipo int).
- *Byte 7-8*: No se usan.

Mensaje de apagado

Este mensaje apaga inmediatamente los motores de la plataforma. Usar este comando podría causar que la plataforma caiga debido a que se desactiva cualquier controlador de balanceo. La información importante del mensaje es:

- *Source:* Computadora de control.
- *Destination:* Interfaz de control.
- *Header:* 0x0412.
- *Bytes 1-8:* El contenido es ignorado.

En este caso el contenido del mensaje no importa, cuando llega un mensaje a la interfaz de control con ese identificador, ésta desactiva los motores.

Mensaje de control

Este mensaje es utilizado para mandar las velocidades de control a la plataforma o cambiar la configuración de la misma. Los comandos son leídos por la interfaz de control a una velocidad de 100Hz. En el caso de que se interrumpa la comunicación, el último mensaje se mantiene 0.4 segundos, y después los comandos de velocidad son cambiados a 0 por cuestiones de seguridad. Para mantener los comandos de velocidad deseados, estos deben de ser enviados a una frecuencia mínima de 2.5Hz. Las características del mensaje están dadas por:

- *Source:* Computadora de control.
- *Destination:* Interfaz de control.
- *Header:* 0x0413.
- *Bytes 1-2:* Velocidad lineal de control (401cnts = 1 m/seg)
- *Bytes 3-4:* Velocidad angular de control (413.6755cnts = 1 rad/seg)
- *Bytes 5-6:* Comandos de configuración los cuales se muestran en la tabla 5.2
- *Bytes 7-8:* Parámetros de configuración los cuales se muestran en la tabla 5.2

Tabla 5.2: Comandos de configuración.

Comando	Valor	Rango
Ninguno	0	No importa
Factor de escala de velocidad	10	0-16 \rightarrow (0 -1.0)
Factor de escala de aceleración	11	0-16 \rightarrow (0 -1.0)
Factor de escalamiento de velocidad angular	12	0-16 \rightarrow (0 -1.0)
Tipo de ganancia*	13	0,1,2
Factor de escalamiento de corriente	14	0-256 \rightarrow (0-1.0)
Bloqueo de modo de autobalanceo	15	0 = Balanceo permitido 1 = Balanceo bloqueado
Modo de operación	16	1 = Modo tractor 2 = Modo autobalanceo
Reiniciar integradores	50	1 = posición de rueda derecha 2 = posición rueda izquierda 4 = giro de θ 8 = movimiento frontal/reversa

*El tipo de ganancia es utilizada para configurar el control interno de balanceo y se elige dependiendo de la carga que se agregará al sistema.

Monitoreo de mensajes

Cada mensaje de monitoreo es generado a una frecuencia de 100 Hz por la interfaz de control. Estos mensajes contiene información de los sensores así como de otras variables del robot. A continuación se presenta el contenido de los siete mensajes más importantes, en este caso los mensajes son transformados en datos de 2 bytes, los cuales se definirán como Data Slot, de esta manera:

- Data Slot 0 = Byte 1 + Byte 2.
- Data Slot 1 = Byte 3 + Byte 4.
- Data Slot 2 = Byte 5 + Byte 6.
- Data Slot 3 = Byte 7 + Byte 8.

Ahora el contenido de cada mensaje, así como su apuntador, están definidos como:

- Mensaje 1

Header: 0x0401.

Data Slot 0: Ángulo de elevación.

Data Slot 1: Velocidad de elevación.

Data Slot 2: Ángulo de alabeo.

Data Slot 3: Velocidad de alabeo.

- **Mensaje 2**

Header: 0x3402.

Data Slot 0: Velocidad de la rueda izquierda.

Data Slot 1: Velocidad de la rueda derecha.

Data Slot 2: Velocidad de guiñada.

Data Slot 3: Servo frames.

- **Mensaje 3**

Header: 0x0403.

Data Slot 0: Posición integrada de la rueda izquierda (Low).

Data Slot 1: Posición integrada de la rueda izquierda (High).

Data Slot 2: Posición integrada de la rueda derecha (Low).

Data Slot 3: Posición integrada de la rueda derecha (High).

- **Mensaje 4**

Header: 0x0404.

Data Slot 0: Integral de la posición directa/reversa (Low).

Data Slot 1: Integral de la posición directa/reversa (High).

Data Slot 2: Integral de la posición de guiñada (Low).

Data Slot 3: Integral de la posición de guiñada (High).

- **Mensaje 5**

Header: 0x0405.

Data Slot 0: Par del motor izquierdo.

Data Slot 1: Par del motor derecho.

Data Slot 2: N/A.

Data Slot 3: N/A.

- Mensaje 6

Header: 0x0406.

Data Slot 0: Modo de operación.

Data Slot 1: Tipo de ganancia.

Data Slot 2: Batería de la interfaz de control.

Data Slot 3: Batería del control interno.

- Mensaje 7

Header: 0x0407.

Data Slot 0: Comando de velocidad recibido.

Data Slot 1: Comando de velocidad angular recibido.

Data Slot 2: N/A.

Data Slot 3: N/A.

Para poder traducir estos mensajes recibidos a valores que puedan ser de utilidad, se utilizan los factores de conversión mostrados en la tabla 5.3.

Tabla 5.3: Factor de conversión de las variables enviadas por la plataforma.

Variable	Factor de conversión
Elevación	7.8 cuentas/grado
Velocidad de elevación	7.8 cuentas/grado/seg
Inclinación	7.8 cuentas/grado
Velocidad de inclinación	7.8 cuentas/grado/seg
Velocidad de la rueda izquierda	401 cuentas/m/seg
Velocidad de la rueda derecha	401 cuentas/m/seg
Velocidad de giro	7.8 cuentas/grado/seg
Cuentas del servo	0.01 seg/cuenta
Integral del desplazamiento de la rueda izquierda	40181 cuentas/m
Integral del desplazamiento de la rueda derecha	40181 cuentas/m
Integral del desplazamiento de avance/reversa	40181 cuentas/m
Posición de giro (integral de la velocidad de giro)	117031 cuentas/rev
Voltaje del controlador interno	4 cuentas/volt
Voltaje de la interfaz de control	1.4 + cuentas*0.0125
Par del motor izquierdo	1463 cuentas/Nm
Par del motor derecho	1463 cuentas/Nm

5.4. Interfaz desarrollada

Una vez estudiadas las instrucciones en la API de la tarjeta CAN, las instrucciones en la API del RTX, y desglosados los mensajes para la comunicación utilizando el protocolo bus CAN con el robot, se diseñó una interfaz de comunicación totalmente nueva, la cual reduce drásticamente funciones innecesarias y emplea funciones de tiempo real del RTX. Dicha interfaz se divide en cuatro procesos: dos procesos Win32 (.EXE) que funcionan principalmente como interfaces gráficas de usuario, y otros dos procesos de tiempo real duro (.RTSS).

Específicamente los procesos de la interfaz de control son los siguientes

- **CONTROLADOR.RTSS:** Es el proceso principal, se encarga de realizar la comunicación con la plataforma a través del puerto CAN y de calcular las leyes de control que después serán enviadas por el mismo puerto. De forma adicional tiene acceso a la variable del tiempo generada por **TIMER.RTSS** a través de la memoria compartida; crea un archivo de texto con las variables monitorizadas y es capaz de guardar estas variables en memoria compartida. Este proceso funciona a un tiempo de muestreo de 10ms.
- **TIMER.RTSS:** Este proceso se encarga únicamente de actualizar la variable de tiempo real y guardarla en memoria compartida; esta variable se refresca a un frecuencia de 10Khz, que corresponde a un periodo de muestreo de $T=0.1ms$.
- **DISPLAY.EXE:** Este proceso tiene acceso a memoria compartida para poder visualizar en pantalla alguna de las variables de interés (por omisión el tiempo) y además cuenta con un botón que funciona como botón de paro, ya que cierra los demás ejecutables, y deteniendo con esto la plataforma Segway RMP-100.
- **MANAGER.EXE:** Este proceso funciona como administrador de los demás. Su función es ejecutar los tres procesos antes mencionados, para que funcionen de manera simultánea y modificar variables en memoria compartida que ayudan a la correcta sincronización de los ejecutables.

En la figura 5.7 se muestra la forma en que se relacionan estos procesos entre sí, las líneas punteadas indican que son llamados por el **MANAGER.EXE**. Los procesos se comunican entre sí utilizando memoria compartida, y el único proceso con comunicación directa al robot es el **CONTROLADOR.RTSS**. Debido a su importancia, este proceso será detallado con mayor profundidad a continuación.

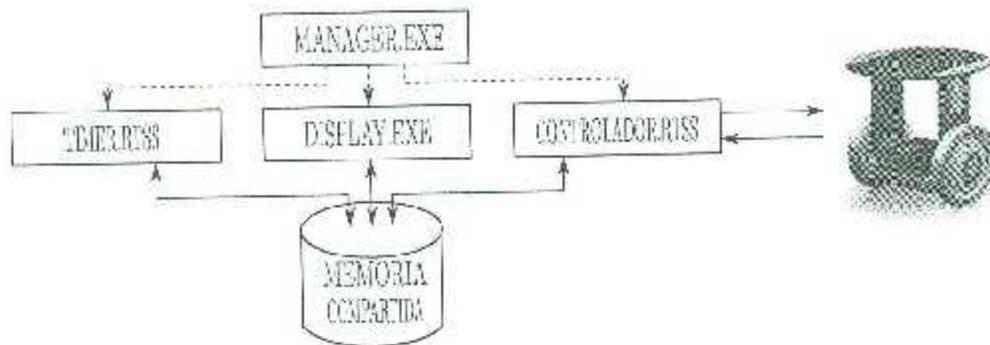


Figura 5.7: Comunicación entre procesos de la nueva interfaz para el Segway RMP-100.

Controlador.RTSS

Como ya se vio anteriormente, este proceso es el encargado de las tareas más importantes para lograr el control de la plataforma, los demás procesos sirven sólo como interfaces de usuario y para actualizar el contador de tiempo real. Este proceso que utiliza funciones API de tiempo real compatibles con la tarjeta utilizada, está compuesto por una rutina principal *main()* en la cual se pueden encontrar los siguientes elementos de código:

- *Inicialización de la tarjeta Softing:* En esta parte del código se configura la tarjeta CAN para ser utilizada, se elige el modo de operación y velocidad de transferencia. Una vez hecho esto ya es posible establecer comunicación con la plataforma.
- *Inicialización de la plataforma:* En esta sección se mandan comandos en formato CAN para configurar la plataforma con respecto a la tabla 5.2 y se reinician todos los integradores del sistema.
- *Funciones de tiempo real:* Aquí se utilizan las funciones de tiempo real de RTX, siendo la más importante de ellas la llamada RUTINA_PERIODICA, ya que en ésta se ejecutarán los comandos para leer las variables del sistema, se transformarán a datos entendibles, se calcularán las leyes de control, y por último se enviarán las mismas a la plataforma.

En la figura 5.8 se muestra cómo quedó la interfaz final y se compara con la interfaz inicial. Por último, en la tabla 5.4 se comparan algunas de las características más importantes de la interfaz inicial (*rmf_demo*) y la interfaz final.

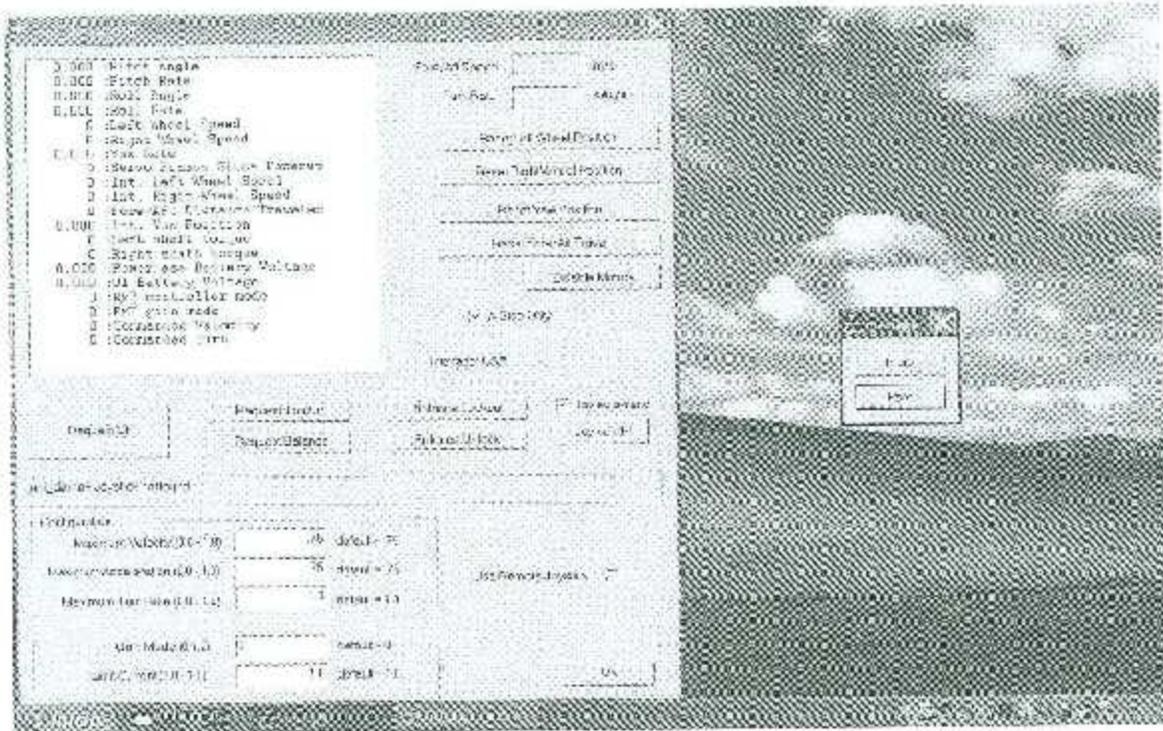


Figura 5.8: Interfaz inicial (derecha) y final (izquierda).

Tabla 5.4: Comparación entre la interfaz original y la desarrollada.

Característica	rmi_demo	Interfaz desarrollada
Comunicación	USB	Bus CAN
Sistema Operativo	Windows Xp - 7	Depende de la tarjeta CAN
Compatibilidad con RTX	Funciones de tiempo real	Tiempo real duro
Tipo de proceso	.EXE	.RTSS
Periodo de muestreo	30ms	10ms

6 Implementación de algoritmos en modo de auto balanceo

Cuando la plataforma Segway RMP-100 se configura en modo de autobalanceo se habilita un controlador que se encarga de mantener el péndulo en posición vertical. Y aunque este controlador de balanceo no es ideal, para fines de análisis se supondrá que el balanceo del péndulo no interfiere con el movimiento del robot móvil tipo unicycle conformado por la base y las ruedas. En otras palabras, se despreciará la dinámica del péndulo invertido y se considerará la plataforma simplemente como un robot tipo unicycle.

Considérese el vector de coordenadas de postura ξ dado por:

$$\xi = [x \quad y \quad \theta]^T.$$

Los controladores implementados en este capítulo entregan la rapidez de desplazamiento v_c y la rapidez de desplazamiento angular ω_c , es decir

$$u = [v_c \quad \omega_c]^T.$$

Se implementaron dos tipos de controladores de movimiento: controladores de regulación de postura y controladores de seguimiento de trayectorias.

En cada caso se obtuvieron resultados en simulación así como resultados de los experimentos realizados en el laboratorio en esta plataforma. Para los resultados en simulación se utilizó el paquete *Matlab* versión *2013b* y la herramienta *Simulink* del mismo.

6.1. Regulación de postura

El objetivo de control de regulación de postura es lograr que la postura de la plataforma real dada por ξ coincida asintóticamente con una postura deseada constante ξ_d . Se evaluaron dos controladores de este tipo: el controlador propuesto de Oriolo *et al.* (2002) y el controlador de Turner y Kyriakopoulos (2003).

En las subsecciones siguientes se presentan los resultados obtenidos, tanto en simulación como en experimento.

6.1.1. Controlador de Oriolo *et al.*

Para la implementación de este controlador se utilizaron las leyes de control PD (3.9) y el compensador dinámico (3.8), el cual se encarga de producir las consignas que serán las señales de control.

Para la implementación, se consideró como condición inicial del vector de coordenadas de postura la siguiente:

$$\xi(0) = [-1 \quad 1 \quad 0]^T,$$

y se buscó estabilizar los estados en el origen, es decir

$$\xi_d = [0 \quad 0 \quad 0]^T.$$

Para la implementación de este controlador, los valores de las ganancias tipo PD utilizadas fueron: $k_{p1} = 0.1385[\text{m}]$, $k_{v1} = 0.7447[\frac{\text{m}}{\text{s}}]$, $k_{p2} = 0.1805[\text{m}]$ y $k_{v2} = 0.85[\frac{\text{m}}{\text{s}}]$, que se seleccionaron mediante un proceso empírico de sintonización, cuidando cumplir con las hipótesis A1 y A2 mencionadas en la sección 3.2.2 y una condición inicial para el compensador dinámico de $\varepsilon(0) = 0.062$; primero se implementó el sistema en Simulink y se sintonizaron las ganancias hasta lograr un desempeño aceptable en simulación; posteriormente se usaron esos mismos valores de ganancias en los experimentos y se inició un nuevo proceso de sintonización. Por último, para fines de comparación los valores de ganancias que mostraron mejor desempeño en los experimentos, se tomaron para realizar nuevamente las simulaciones y obtener las gráficas.

Resultados de simulación

En la figura 6.1 se muestran las consignas de control enviadas a la planta. Nótese que los valores de las señales de control v_c y ω_c están dentro de los valores permitidos para la planta.

En la figura 6.2 se observa cómo los elementos del vector de variables de postura ξ , son llevados desde la postura inicial $\xi(0)$ hasta la postura deseada $\xi_d = 0$. Se puede apreciar como la evolución de las variables de posición x y y es mucho más rápida que para la variable de orientación y se estabilizan en el origen en aproximadamente 20 segundos, mientras que θ lo hace en aproximadamente 80 segundos.

En la figura 6.3 se muestra la trayectoria recorrida por el robot en el plano X - Y . En este controlador no se puede especificar directamente la trayectoria que sigue el robot, pero es posible variarla ajustando los valores de las ganancias de los controladores PD (3.9).

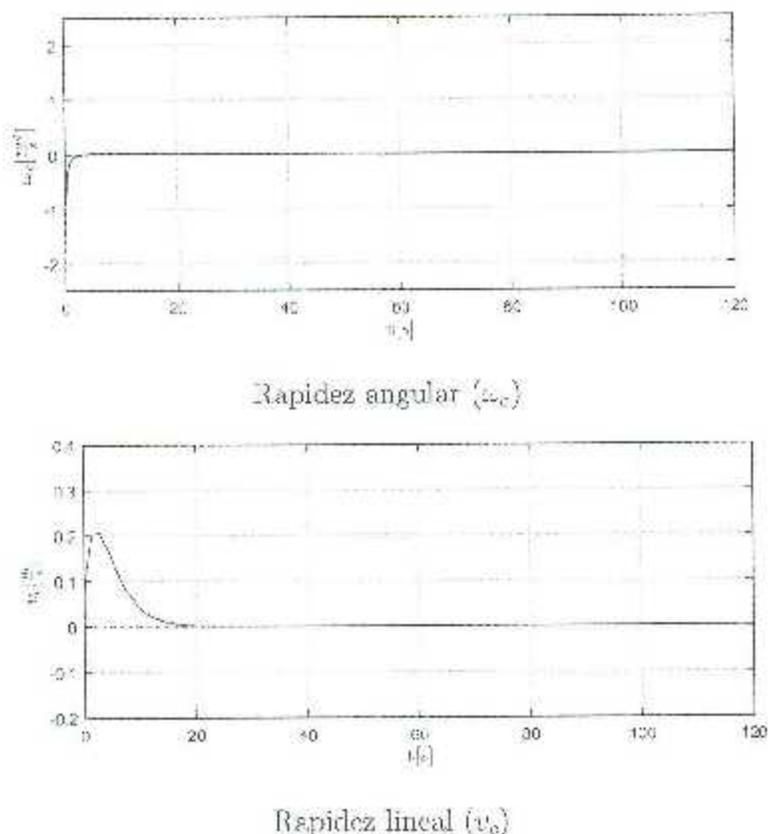
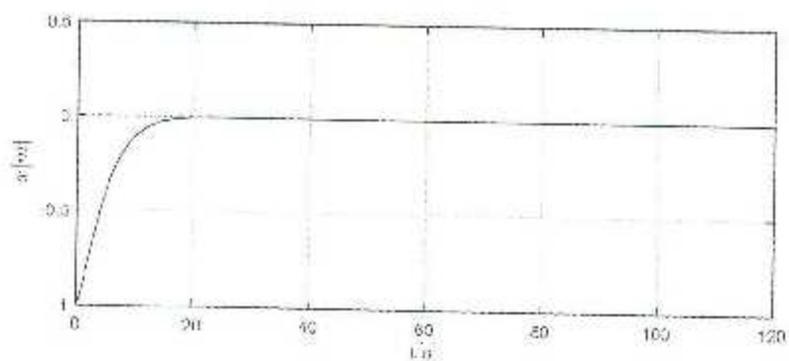
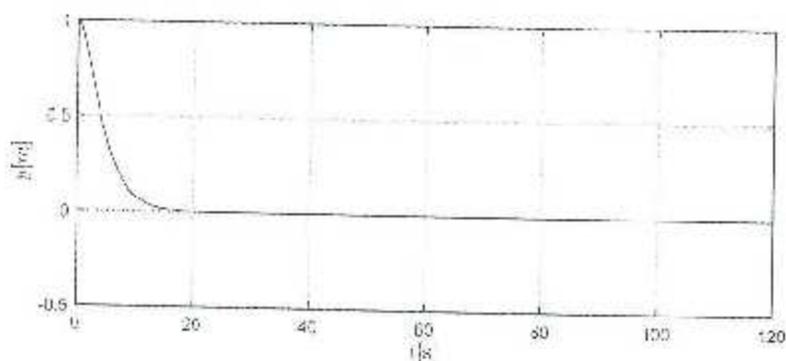


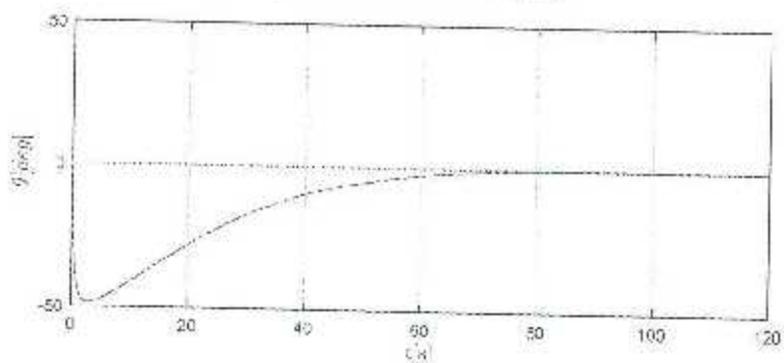
Figura 6.1: Regulador de Oriolo *et al.*: Acciones de control (simulación).



Desplazamiento en el eje x



Desplazamiento en el eje y



Variación de la orientación del robot

Figura 6.2: Regulador de Oriolo *et al.*: Variables de postura (simulación).

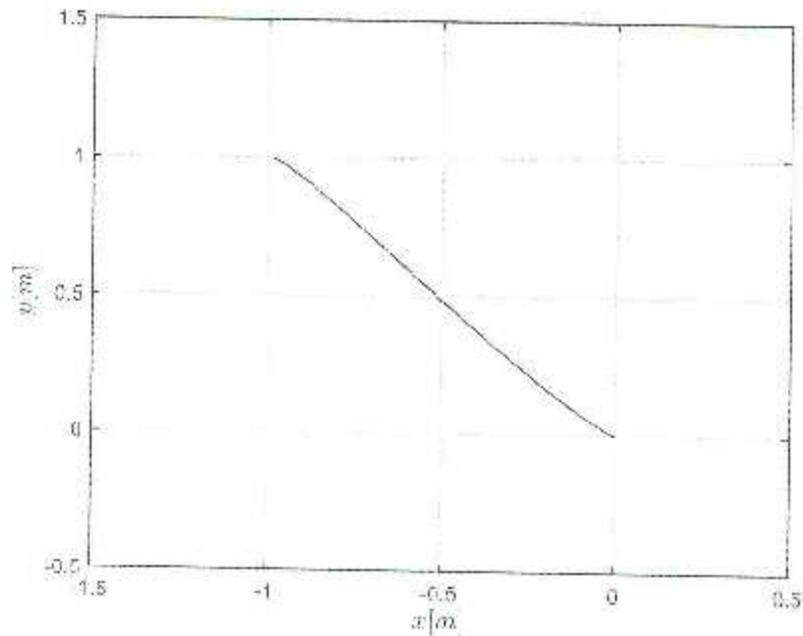
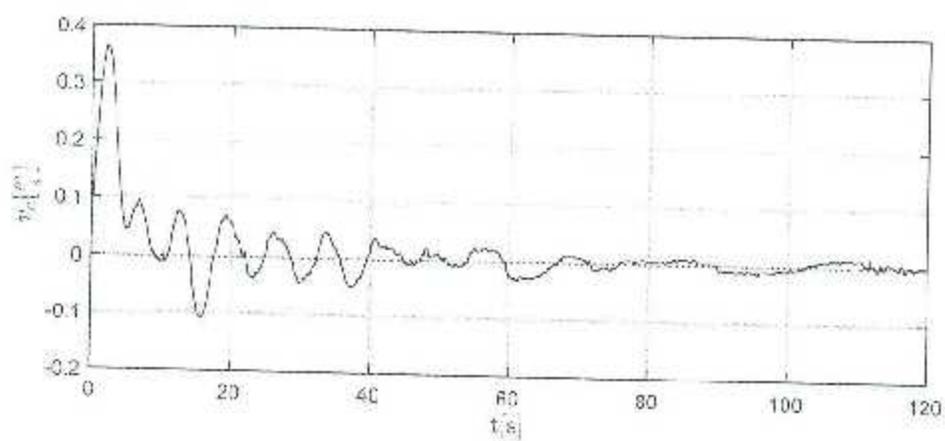


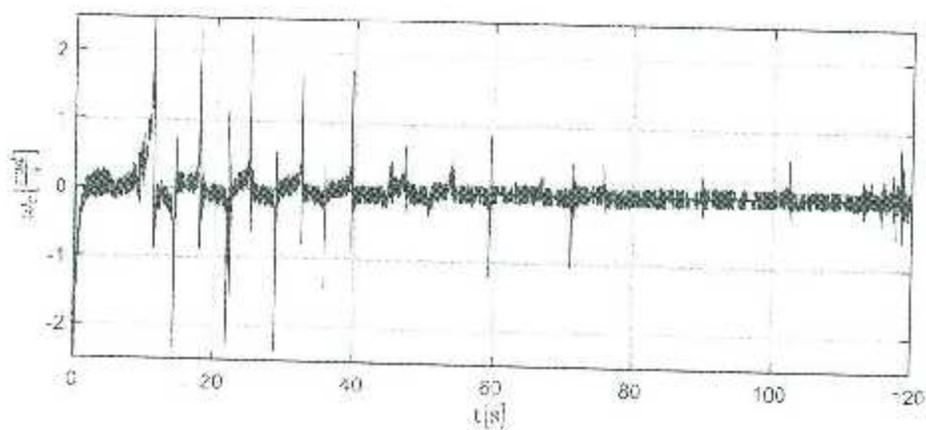
Figura 6.3: Regulador de Oriolo *et al.*. Trayectoria recorrida (simulación)

Resultados experimentales

La diferencia más importante entre los resultados de simulación y los experimentales se muestra en la figura 6.4, donde se puede notar que la rapidez de desplazamiento lineal del vehículo v_c cambia de signo, esto implica que $\varepsilon = 0$ ya que $\dot{\zeta} = v_c$, esto produce una singularidad en la rapidez de desplazamiento angular ω_c como puede apreciarse en la respuesta del controlador. Estos cambios del sentido del movimiento del robot se deben a que el robot oscila alrededor del origen como se puede apreciar en la figura 6.5. Estas oscilaciones son ocasionadas principalmente por perturbaciones debidas al control de balanceo interno de la plataforma, el cual produce fuerzas para mantener el péndulo en posición vertical pero que sacan al robot del equilibrio. A pesar de estas perturbaciones y la singularidad que aparece en las señales de control las señales de control se encuentran en valores permitidos y los errores de las coordenadas de postura son pequeños, en comparación con el tamaño de la plataforma, solo que esta variación en ω_c produce oscilaciones en la orientación del robot el cual oscila alrededor del origen, como se puede apreciar también en las figuras 6.5 y 6.6.

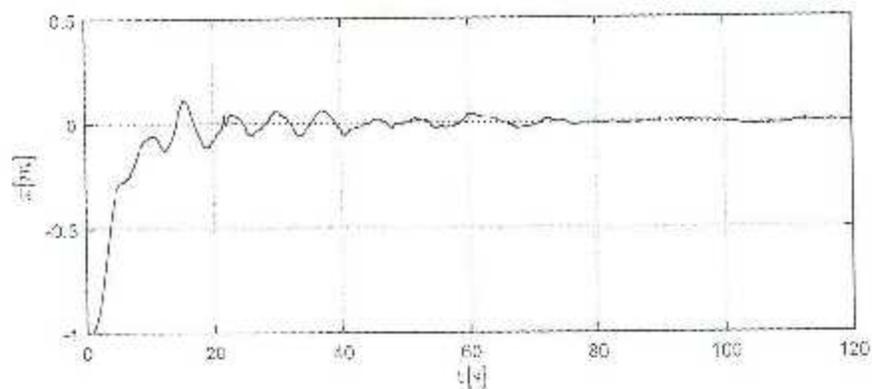
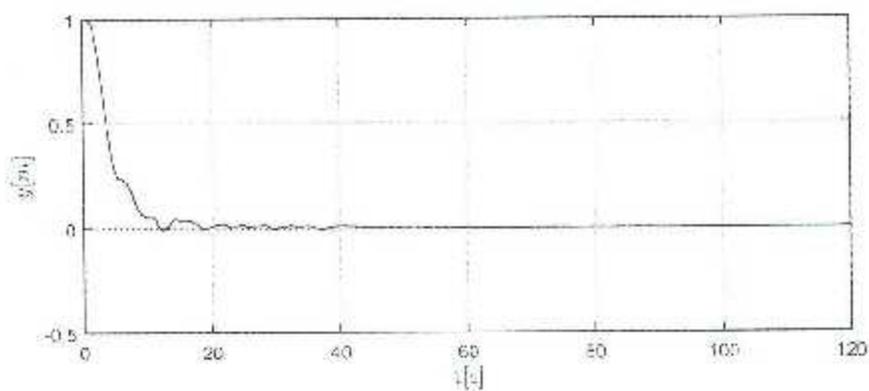
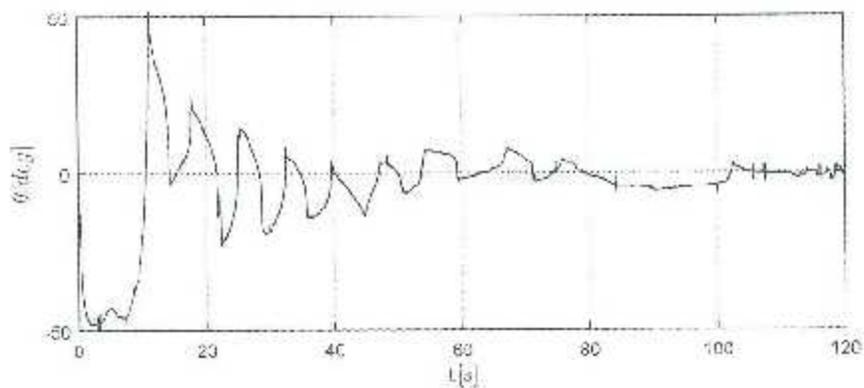


Rapidez lineal (v_c)



Rapidez angular ($\dot{\omega}_c$)

Figura 6.4: Regulador de Oriolo *et al*: Acciones de control (experimento).

Desplazamiento en el eje x Desplazamiento en el eje y 

Variación de la orientación del robot

Figura 6.5: Regulador de Oriolo *et al.*: Variables de postura (experimento).

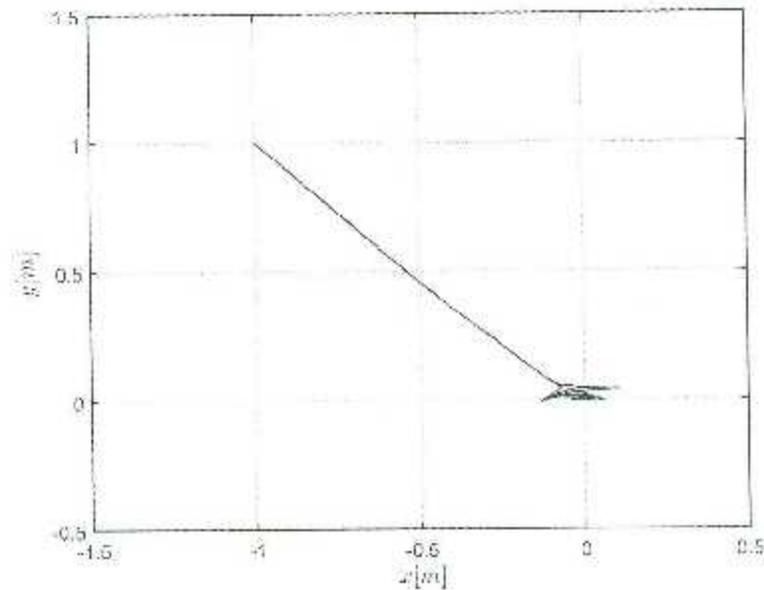


Figura 6.6: Regulador de Oriolo *et al.*: Trayectoria recorrida (experimento).

6.1.2. Controlador de Tanner y Kyriakopoulos.

Para la implementación de este controlador se utilizaron los comandos de velocidades deseadas propuestas en (3.13) y (3.14), a partir de los cuales se diseña un control que asegure que el robot siga las velocidades deseadas. Dicho controlador está dado por (3.17) y (3.18) y utilizando (3.15) y (3.16) se obtienen las señales de control v_x y ω_z . Los valores de ganancias utilizados en este controlador son: $k_v = 0.355[m]$, $k_\omega = 0.633[rad]$, $k_{xv} = 0.225[\frac{m}{s}]$ y $k_{z\omega} = 0.616[\frac{rad}{seg}]$.

Estas ganancias se utilizaron para realizar la simulación y para implementar el controlador en la plataforma.

Resultados de simulación

En las figuras 6.7 nuevamente se muestran las señales de control; se puede notar que en esta ocasión tales señales cambian de sentido, haciendo con esto que el robot gire y avance en ambas direcciones para lograr llegar a la posición deseada, ambas señales se encuentran dentro de los valores permitidos.

En la figura 6.8 se presenta la evolución de las variables de estado del robot; se puede apreciar que las coordenadas de la posición x y y pasan el transitorio en aproximadamente 40 segundos, pero la posición en el eje x no llega exactamente a cero. La diferencia más notable con respecto a la simulación del controlador de Oriolo *et al.* se aprecia en el ángulo de orientación θ el cual oscila pero logra estabilizarse en un tiempo menor. Por último en la figura 6.9 se muestra la trayectoria recorrida por el robot.

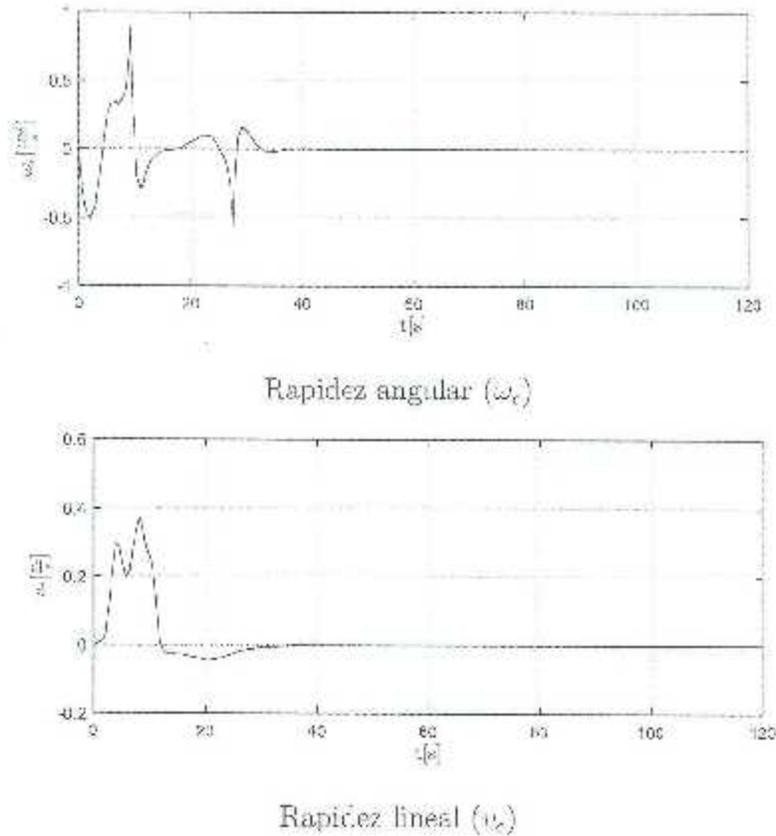
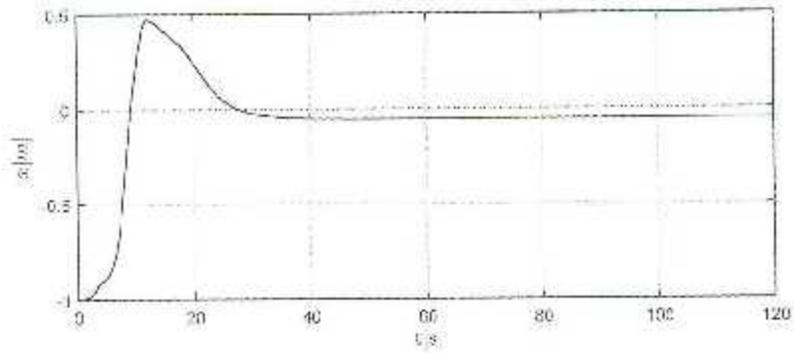
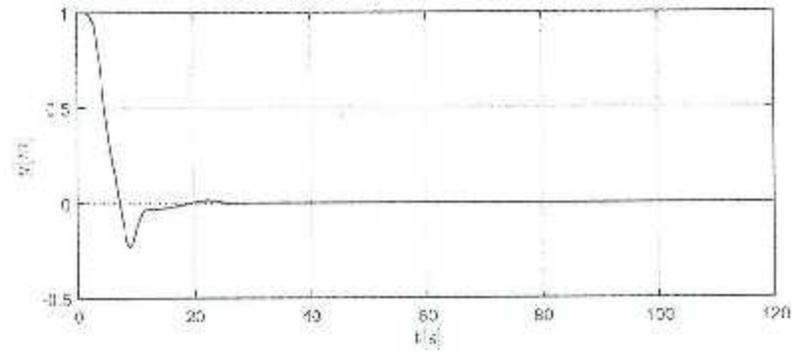
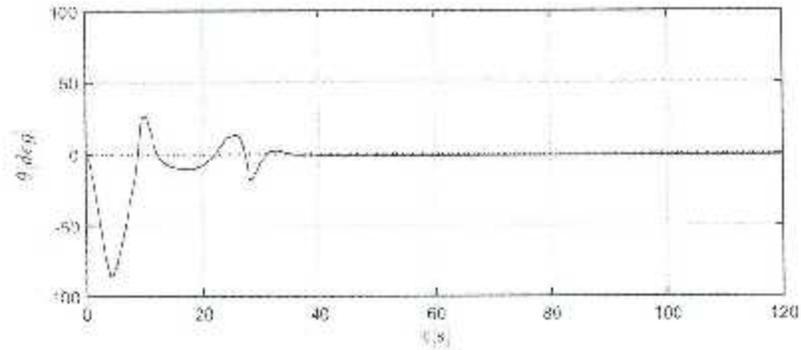


Figura 6.7: Regulador de Tanner-Kyriakopoulos: Acciones de control (simulación).

Desplazamiento en el eje x Desplazamiento en el eje y 

Variación de la orientación del robot

Figura 6.8: Regulador de Tanner-Kyriakopoulos. Variables de postura (simulación).

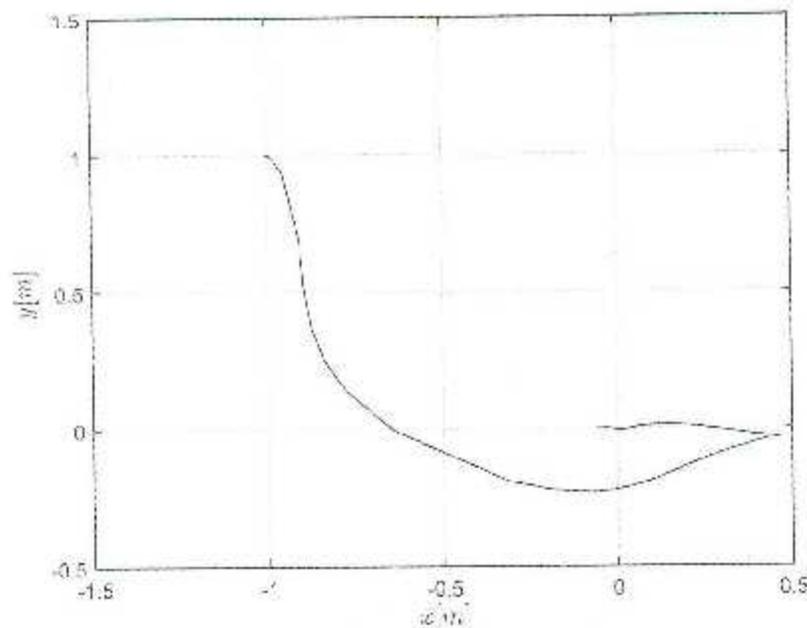


Figura 6.9: Regulador de Tanner-Kyriakopoulos: Trayectoria recorrida (simulación).

Resultados experimentales

En la figura 6.10 se presentan las señales de control resultantes durante las pruebas experimentales con el controlador de Tanner-Kyriakopoulos; el principal cambio con respecto a la simulación es la aparición de oscilaciones en ambas señales las cuales se estabilizan en aproximadamente 80 segundos, a pesar de estas oscilaciones nuevamente los valores de ambas señales de control se encuentran dentro de los valores permitidos. En la figura 6.11 se observa como la posición del robot en el eje y en aproximadamente 70 segundos, mientras que para el eje x tarda aproximadamente 100 segundos, aunque presenta un pequeño error de 0.02m probablemente debido al efecto de la fricción no modelada. Para el ángulo de orientación θ , después de presentar algunas oscilaciones la variable es llevada a prácticamente 0 grados después de 110 segundos. Por último, se muestra en la figura 6.12 la trayectoria recorrida por el robot desde su condición inicial hasta que se en el origen, la cual no es tan directa como la trayectoria recorrida utilizando el controlador de Oriolo *et al.*

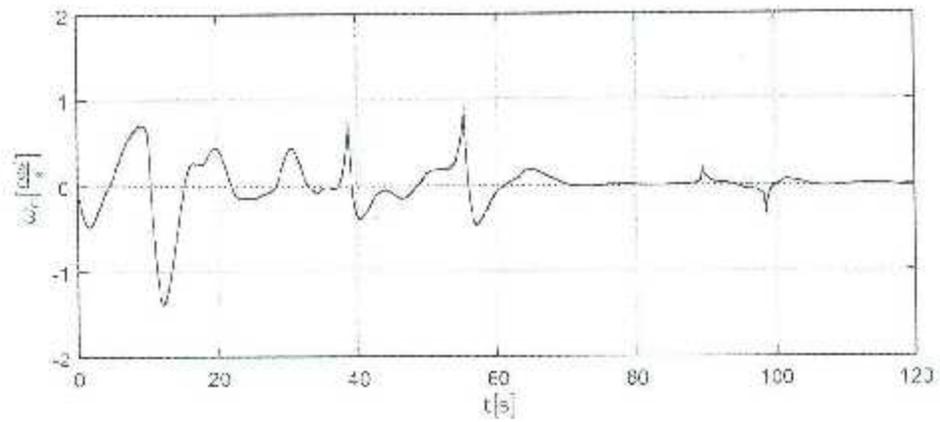
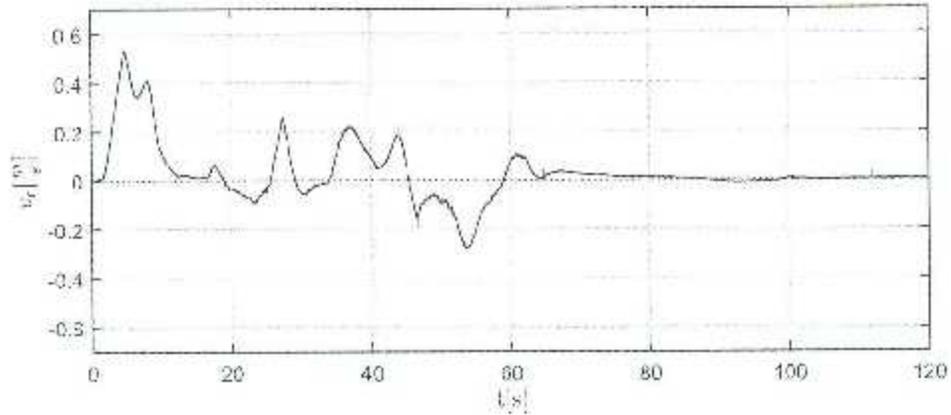
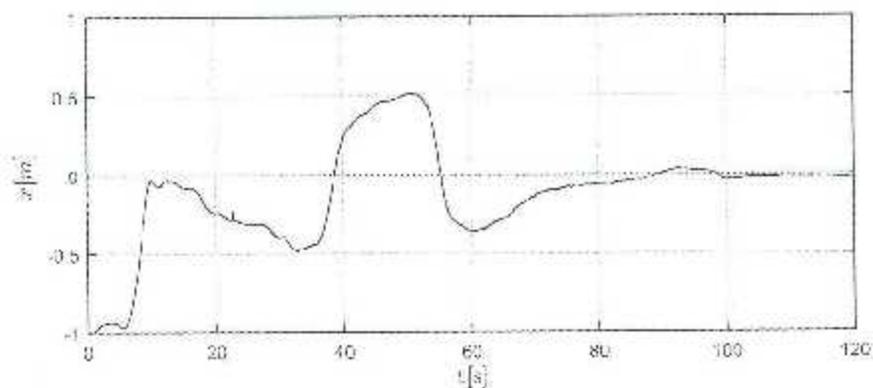
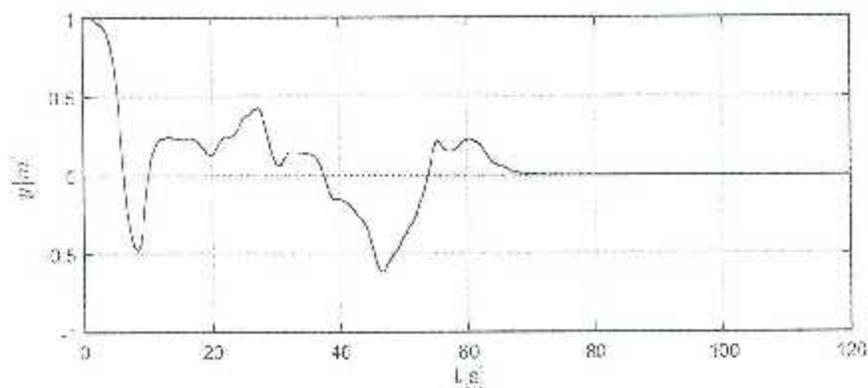
Rapidez angular (ω_e)Rapidez lineal (v_e)

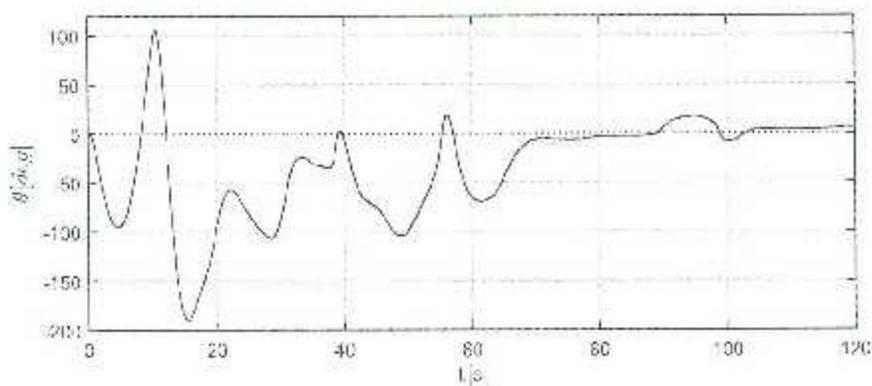
Figura 6.10: Regulador de Tanner-Kyriakopoulos: Acciones de control (experimento).



Desplazamiento en el eje x



Desplazamiento en el eje y



Variación de la orientación del robot

Figura 6.11: Regulador de Tanner-Kyriakopoulos: Variables de postura (experimento).

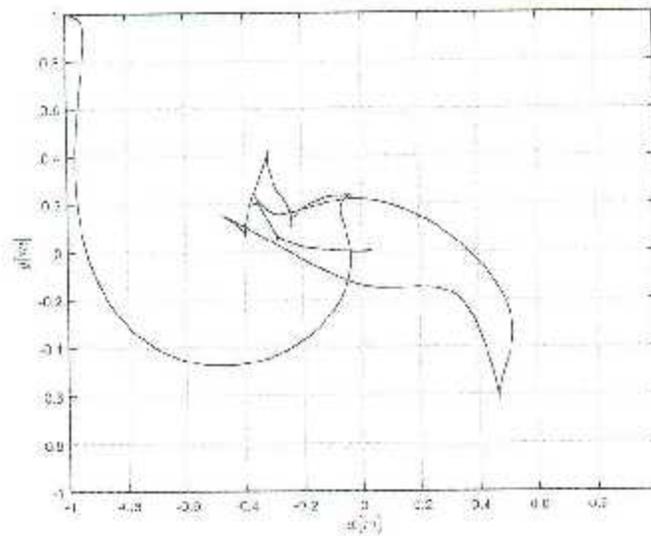


Figura 6.12: Regulador de Tanner-Kyriakopoulos: Trayectoria recorrida (experimento).

Por último, en la figura 6.20 se muestran imágenes extraídas de los experimentos realizados en el laboratorio, donde se ve el movimiento de la plataforma lo largo de la trayectoria.

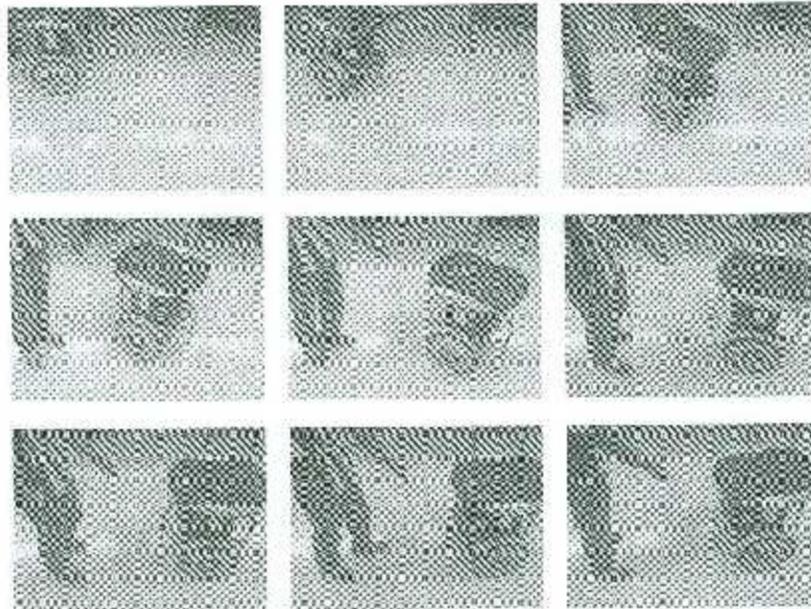


Figura 6.13: Movimiento de la plataforma hasta el origen.

6.2. Control de seguimiento

El problema de control de seguimiento de trayectorias, consiste en lograr que la postura del robot móvil descrita por el vector ξ , converja asintóticamente a una trayectoria de postura deseada variante en el tiempo $\xi_d(t)$. En esta sección se prueba el controlador propuesto en (Oriolo *et al.*, 2002) y se presentan los resultados obtenidos de la simulación, así como los resultados experimentales.

6.2.1. Controlador de Oriolo *et al.*

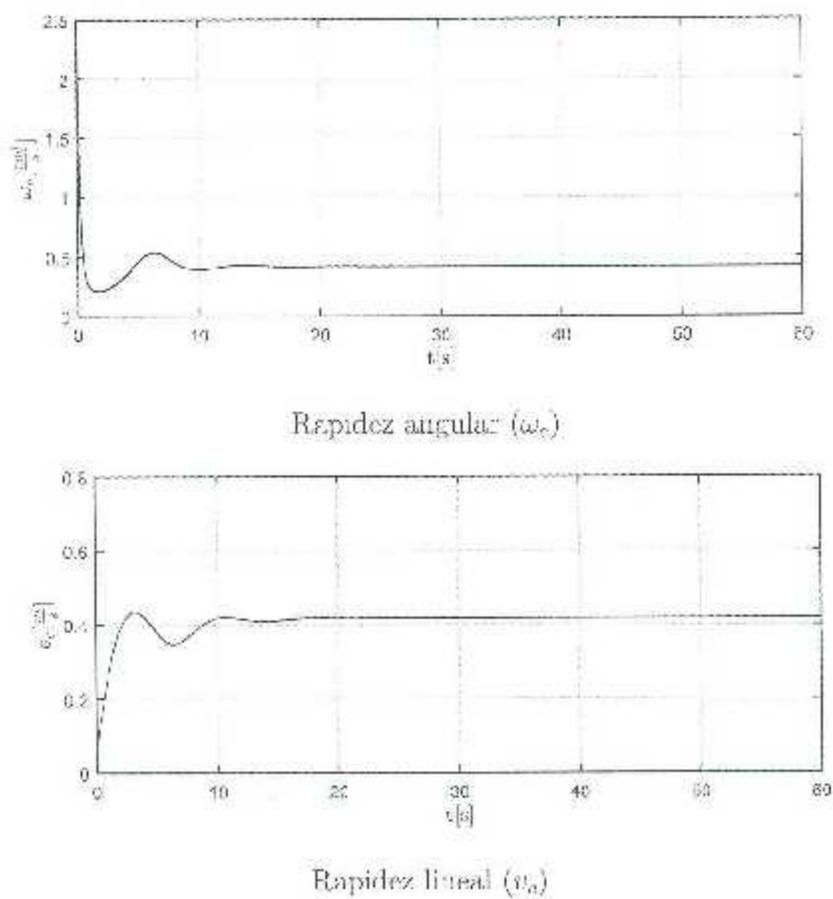
Para la implementación de este controlador se utilizaron las leyes de control de seguimiento PD vistas en (3.19) y una trayectoria deseada diseñada como

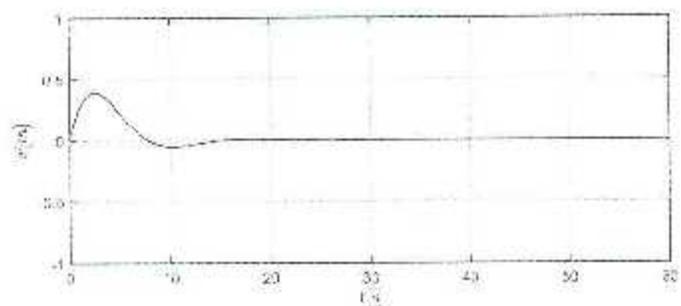
$$\begin{aligned}x_d(t) &= c \operatorname{sen}(\omega_0 t) \\y_d(t) &= c[1 - \cos(\omega_0 t)]\end{aligned}$$

con $c = 1.0[\text{m}]$ y $\omega_0 = \frac{2\pi}{15}[\frac{\text{rad}}{\text{s}}]$, por lo tanto el robot deberá seguir un círculo que deberá completar cada 15 segundos. El tiempo de ejecución, tanto de simulación como en los experimentos fue de 60 segundos. Para la implementación de este controlador, tanto en simulación como en los experimentos, se utilizaron los siguientes valores de las ganancias $k_{p,1} = 0.23 [\text{m}]$, $k_{p,2} = 0.059 [\text{m}]$, $k_{v,1} = 0.515 [\frac{\text{m}}{\text{s}}]$ y $k_{v,2} = 0.35 [\frac{\text{m}}{\text{s}}]$, las cuales fueron elegidas bajo la única restricción de que fueran positivas.

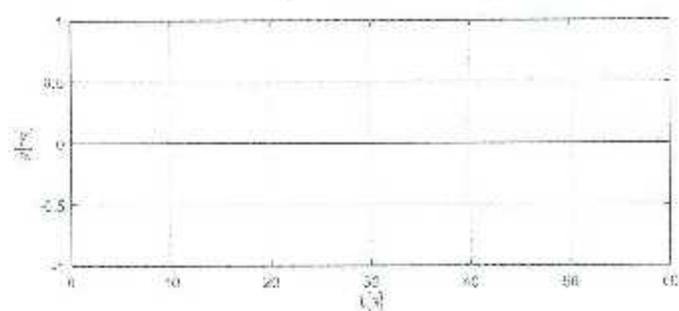
Resultados de simulación

En la figura 6.14 se muestran las consignas de control, las cuales rápidamente tienden a un valor constante (para trazar un círculo se necesita que ambas velocidades sean constantes) en aproximadamente 20 segundos. Por otro lado en la figura 6.15 se tienen los errores de seguimiento en el eje x y el eje y . En el eje x se tiene un error de hasta 0.4m, el cual se reduce a cero en aproximadamente 15 segundos. En el eje y , por otro lado, no existe error de seguimiento. En la figura 6.16 se puede ver cómo el robot tarda aproximadamente un ciclo en alcanzar la trayectoria deseada esto concuerda con el error de seguimiento \tilde{x} , y después de este tiempo sigue correctamente la trayectoria.

Figura 6.14: Seguidor de Oriolo *et al.*: Acciones de control (simulación).



Error de seguimiento en el eje x



Error de seguimiento en el eje y

Figura 6.15: Seguidor de Oriolo *et al.*: Errores de postura (simulación).

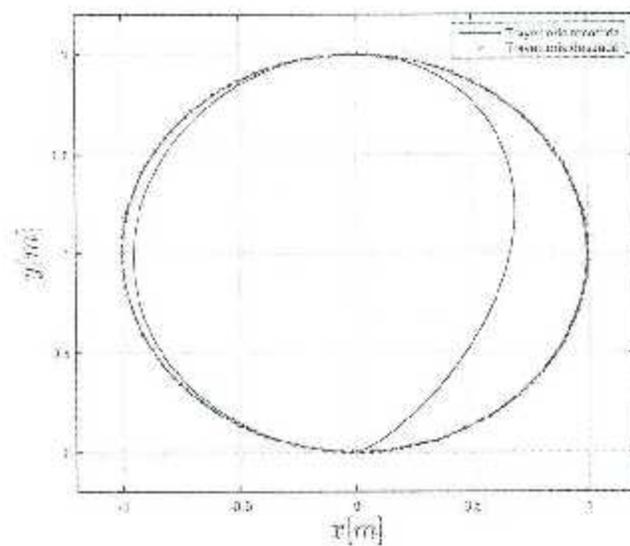


Figura 6.16: Seguidor de Oriolo *et al.*: Trayectoria recorrida (simulación).

Resultados de experimentación

En la figura 6.17 se muestran los comandos de velocidad enviados al robot; estos ya no se van a un valor constante después de cierto tiempo como en la simulación, oscilan alrededor de cierto valor, debido a perturbaciones que causan que el robot no pueda seguir de manera perfecta la trayectoria. Es importante recalcar que todos los valores de las consignas de control están dentro de los límites permitidos por la plataforma.

En el caso de los errores de seguimiento que se muestran en la figura 6.18, es posible observar que los errores están cercanos a cero, pero perturbaciones en el robot evitan que se mantengan en cero. Dichas perturbaciones se deben principalmente al control de balanceo que produce velocidades diferentes a las de control v_c para estabilizar al mismo. A pesar de estas perturbaciones los errores de seguimiento permanecen en $\pm 0.2m$ para \hat{x} y $\pm 0.1m$ para \hat{y} . En la figura 6.19 se muestra la trayectoria recorrida en el plano X-Y, el robot alcanza la trayectoria deseada en aproximadamente medio ciclo y a partir de ahí la sigue con errores pequeños.

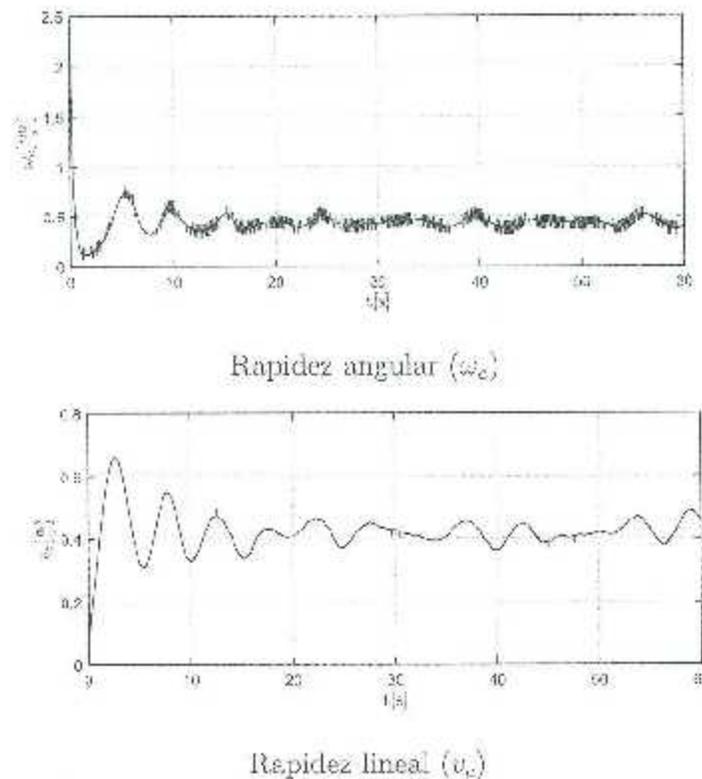
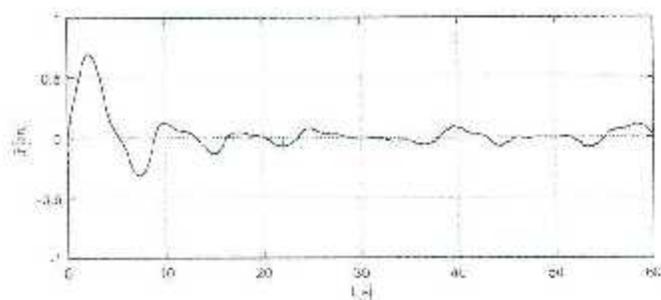
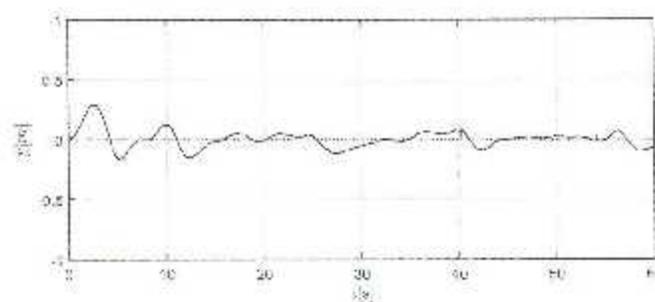
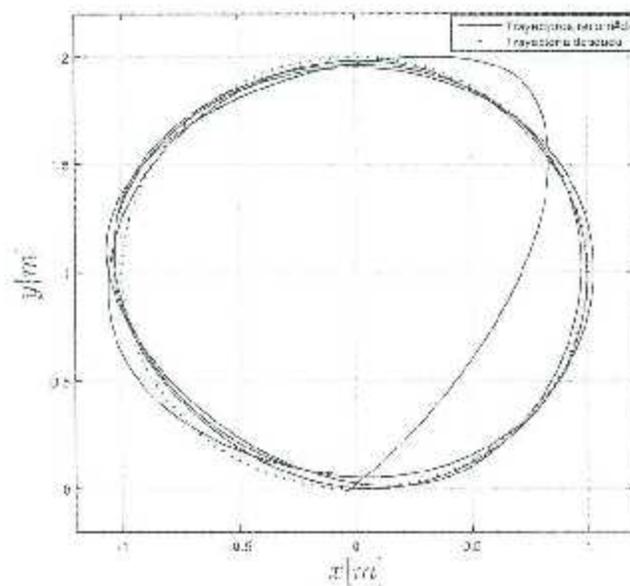


Figura 6.17: Seguidor de Oriolo *et al.*: Acciones de control (experimento).

Error de seguimiento en el eje x Error de seguimiento en el eje y Figura 6.18: Seguidor de Oriolo *et al.*: Errores de postura (experimento).Figura 6.19: Seguidor de Oriolo *et al.*: Trayectoria recorrida (experimento).

Por último, en la figura 6.20 se muestran imágenes extraídas de los experimentos realizados en el laboratorio, donde se ve el movimiento de la plataforma lo largo de la trayectoria.

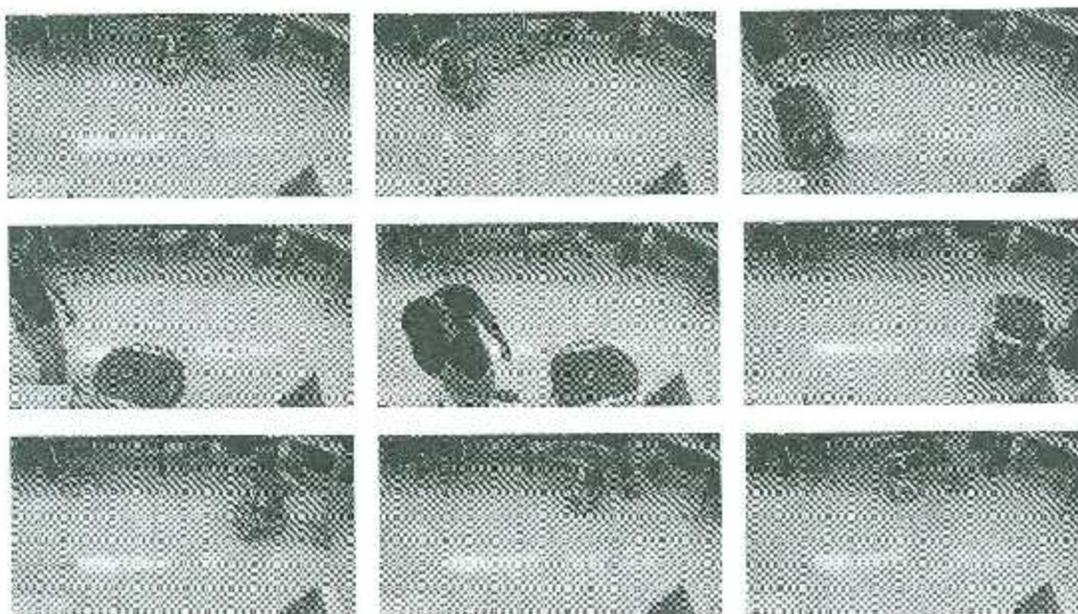


Figura 6.20: Movimiento de la plataforma a lo largo de la trayectoria.

7 Implementación de algoritmos en modo de tractor

En este capítulo se utiliza la plataforma en modo tractor, es decir, sin el uso del controlador interno encargado de balancear el péndulo invertido. Este modo de operación permite probar controladores diseñados para el control del balanceo del péndulo, es decir, se tomarán en cuenta todos los grados de libertad del robot y se presentarán resultados de la implementación de un controlador LQR.

Siguiendo con la notación del capítulo 2, se define el vector de variables de estado $\mathbf{z} = [\mathbf{q} \ \dot{\mathbf{q}}]^T$, donde:

$$\mathbf{q} = [x_r \ \theta \ c]^T$$

El controlador implementado en este capítulo entrega como salida los pares que deberán ser aplicados en cada rueda, τ_l para la rueda izquierda y τ_r para la rueda derecha; por lo tanto, el vector de fuerzas generalizadas está dado por:

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix}$$

Como las entradas al robot deben ser las velocidades v_c y ω_c , se requiere aplicar una transformación de pares a velocidad. Para esto se considera que las masas y momentos

de inercia de las ruedas se desprecian, y entonces es posible escribir:

$$\begin{aligned} F_{x_r} &= (m_b + m_w) \ddot{x}_r = \frac{1}{r} [\tau_r + \tau_l] \\ \tau_\theta &= [I_{b_z} + I_{w_z}] \ddot{\theta} = \frac{L}{r} [\tau_r - \tau_l] \end{aligned} \quad (7.1)$$

donde F_{x_r} es la fuerza de avance del robot en dirección de las ruedas y τ_θ es el par de giro del robot respecto al eje z . Entonces las velocidades de control pueden ser calculadas como:

$$\begin{aligned} v_c = \dot{x}_{rc} &= \frac{1}{r(m_b + m_w)} \int [\tau_r + \tau_l] dt \\ \omega_c = \dot{\theta}_c &= \frac{L}{r(I_{b_z} + I_{w_z})} \int [\tau_r - \tau_l] dt. \end{aligned} \quad (7.2)$$

Utilizando (7.1) y (7.2) es posible obtener el vector de entradas de control que está definido como

$$\mathbf{u} = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}.$$

Los resultados presentados en este capítulo se dividen nuevamente en dos categorías, el controlador aplicado para regulación y el mismo controlador utilizado para seguimiento.

7.1. Control LQR

Para este controlador considérese $\mathbf{z} = [\mathbf{q} \quad \dot{\mathbf{q}}]^T \in \mathbb{R}^6$ como el vector de variables de estado. En esta sección se trabajará con el modelo (4.4), el cual se linealizó alrededor del punto de equilibrio $\mathbf{z}_e = [\mathbf{0}^2 \quad \mathbf{0}^2]^T$, y utilizando los parámetros de la tabla 4.1 se llega a

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{r} \in \mathbb{R}^6$$

donde las matrices A y B están definidas como

$$A = \left. \frac{\partial f(z, \bar{\tau})}{\partial z} \right|_{z=z_0, \bar{\tau}=\bar{\tau}_0} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 0 & 0 & 0 \end{bmatrix}, B = \left. \frac{\partial f(z, \bar{\tau})}{\partial \bar{\tau}} \right|_{z=z_0, \bar{\tau}=\bar{\tau}_0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b_1 & b_1 \\ -b_2 & b_2 \\ -b_3 & -b_3 \end{bmatrix}$$

con

$$\begin{aligned} a_1 &= \frac{(g m_p^2 l_p^2)}{(m_p l_p^2 + I_{py})h - m_p^2 l_p^2} \\ a_2 &= \frac{g m_p l_p h}{(m_p l_p^2 + I_{py})h - m_p^2 l_p^2} \\ b_1 &= \frac{m_p l_p^2 + I_{py}}{r((m_p l_p^2 + I_{py})h - m_p^2 l_p^2)} \\ b_2 &= \frac{L}{r(I_{bz} + I_{pz} + I_{vz} - I_{lz} + L^2(m_l + m_r) - \frac{L^2(I_{ly} - I_{ry})}{r^2})} \\ b_3 &= \frac{m_p l_p}{r((m_p l_p^2 + I_{py})h - m_p^2 l_p^2)} \\ h &= (m_b + m_l + m_p + m_r + \frac{I_{ly}}{r^2} + \frac{I_{ry}}{r^2}) \end{aligned}$$

y la ley de control $\bar{\tau}$ definida por

$$\bar{\tau} = -Kz \in \mathbb{R}^2 \quad (7.3)$$

7.1.1. Control de regulación

Para este control se utilizó la configuración inicial:

$$q(0) = \begin{bmatrix} 0 & 0.0174 & 0.0349 \end{bmatrix}^T$$

y se buscó estabilizar los estados en el origen:

$$q_d = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T.$$

Además para la implementación de este controlador la matriz de ganancias K está dada por:

$$K = \begin{bmatrix} -7.0711 & -7.0711 & -175.501 & -14.8468 & -7.836 & -41.4221 \\ -7.0711 & 7.0711 & -175.501 & 14.8468 & 7.836 & -41.4221 \end{bmatrix}.$$

Resultados de simulación

En la figura 7.1 se presentan las consignas de control, las cuales fueron obtenidas a partir de los pares utilizando la conversión par-velocidad dada por (7.1) y (7.2). Es posible apreciar que ambas señales están dentro de los límites permitidos para la plataforma.

Por otro lado, en la figura 7.2 se tiene la evolución de los estados a través del tiempo; donde se puede ver cómo la inclinación del péndulo es llevada hasta el equilibrio en menos de 10 segundos; de igual manera es posible notar la variable de posición x_p es sacada del equilibrio, esto debido al acoplamiento que existe entre esta variable y la inclinación del péndulo α ; sin embargo, una vez que el ángulo del péndulo es aproximadamente cero la posición del robot es llevada al origen; por último, es posible la evolución de la variable de orientación del robot θ , como esta variable no está acoplada directamente con las otras dos variables de estado, su evolución no presenta perturbaciones.

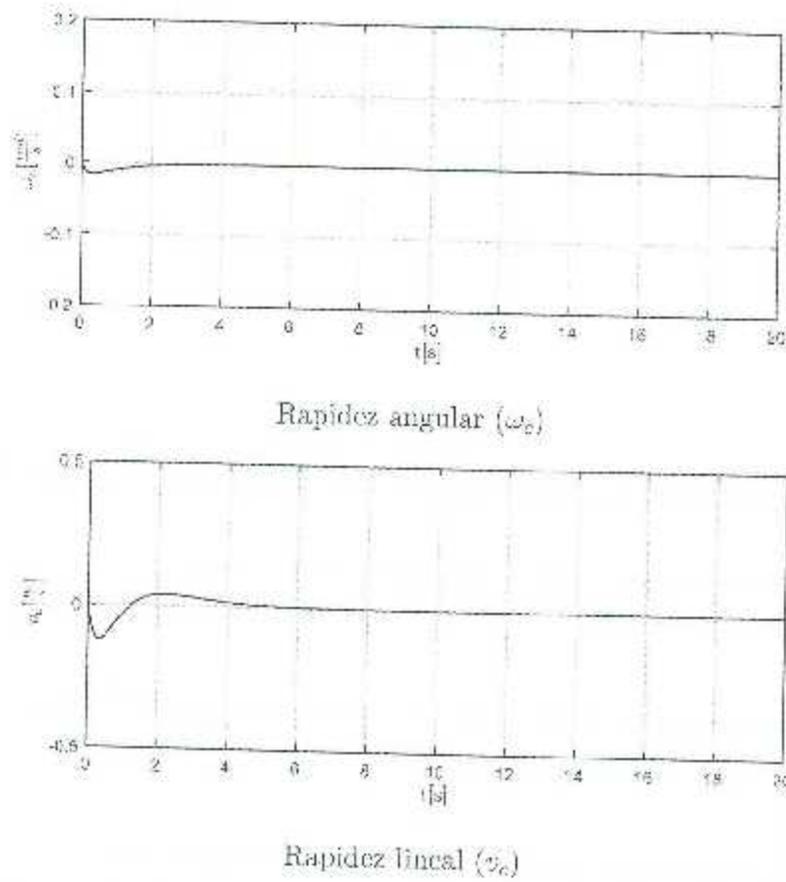
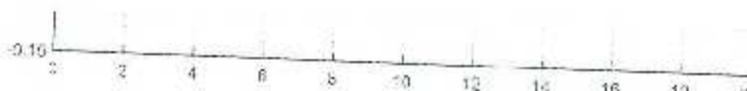


Figura 7.1: Regulador LQR: Acciones de control (simulación).



Experimentos

Para los primeros resultados experimentales se utilizaron los mismos valores de ganancias usadas para la simulaciones; en la figura 7.3 se muestra la consigna de rapidez angular enviada al robot, la cual oscila alrededor de cero y tiene una amplitud de $\pm 0.05 [\frac{rad}{s}]$; por otro lado es posible apreciar la consigna de rapidez lineal, nuevamente con oscilaciones alrededor de cero con una amplitud de $\pm 0.1 [\frac{m}{s}]$.

En la figura 7.4 se encuentra la evolución de los estados del robot desde su condición inicial. Para α , se puede notar cómo la variable es llevada desde la condición inicial hasta el equilibrio, en el cual permanece teniendo pequeñas oscilaciones menores a ± 0.5 [grados]; es posible notar también que θ es también llevada hasta el equilibrio, en el cual oscila pero, ahora en ± 0.25 [grados]. Por último se puede apreciar cómo la variable de desplazamiento del robot x_r es sacada del equilibrio y después tiende a estabilizarse en aproximadamente 0.05 [m].

Además de estos resultados, en los cuales se utilizaron exactamente las ganancias obtenidas del método LQR, se presentan también los resultados obtenidos después de haber sintonizado de manera empírica el controlador, y además se agregó una acción integral para el estado x_r (para reducir el error en estado estacionario) donde $k_i = -0.000001$. La matriz de ganancias utilizado en estos experimentos está dada por

$$K = \begin{bmatrix} -17.7679 & -5.1 & -0.000001 & -126.37 & -15.726 & -5.2 & -49.205 \\ -17.7679 & 5.1 & -0.000001 & -126.37 & -15.726 & 5.2 & -49.205 \end{bmatrix}$$

y el vector de variables de estado se define ahora como

$$z = [x_r \quad \dot{x}_r \quad \int x_r \quad \theta \quad \dot{\theta} \quad \alpha \quad \dot{\alpha}]^T.$$

En la figura 7.5, se muestran las señales de rapidez angular y la rapidez lineal enviadas al robot, las cuales se encuentran muy por debajo de los límites permitidos. A pesar de que se empezó con un error más grande en la posición del péndulo, las consignas de control mostradas siguen estando muy por debajo de los límites permitidos por el sistema. Además, al hacer este cambio en las ganancias, se logró inicializar la inclinación con un ángulo más grande de $\alpha_0 = -10$ [grados], logrando de igual manera un buen control sobre la plataforma, la cual oscila alrededor del equilibrio con una amplitud aproximada de ± 0.5 [grados], como se puede apreciar en la figura 7.6; de igual manera en la misma

figura se tiene la evolución de la variable de orientación del robot θ donde se eligió una condición inicial en el equilibrio y el sistema permanece ahí con oscilaciones alrededor de ± 0.3 [grados]; por último se puede apreciar cómo el estado x_1 es sacado bruscamente del origen; esto debido a la acción de control necesaria para equilibrar el péndulo; nuevamente una vez que el péndulo logra llevarse al equilibrio (en 2 segundos), la posición del robot empieza a ser llevada hasta el equilibrio, donde llega en aproximadamente 5 segundos y después oscila alrededor de éste con una amplitud de ± 0.01 [m].

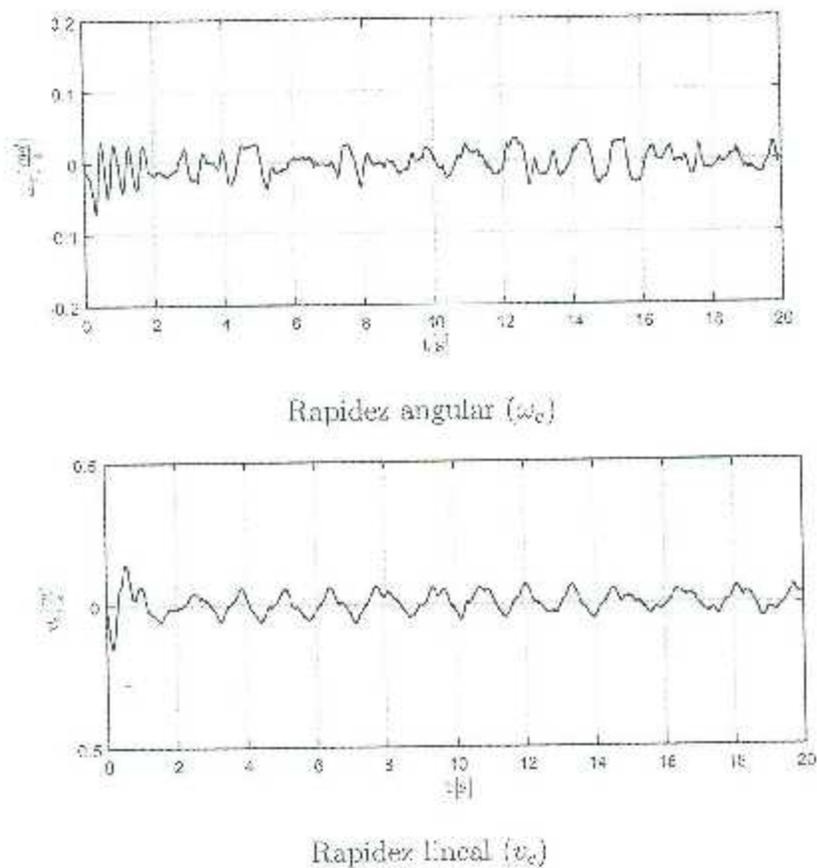
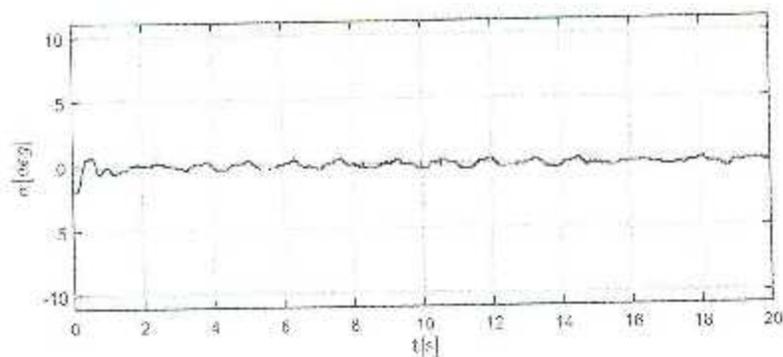
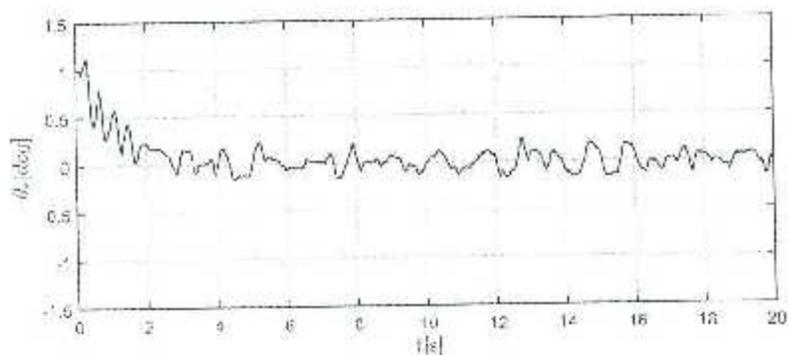


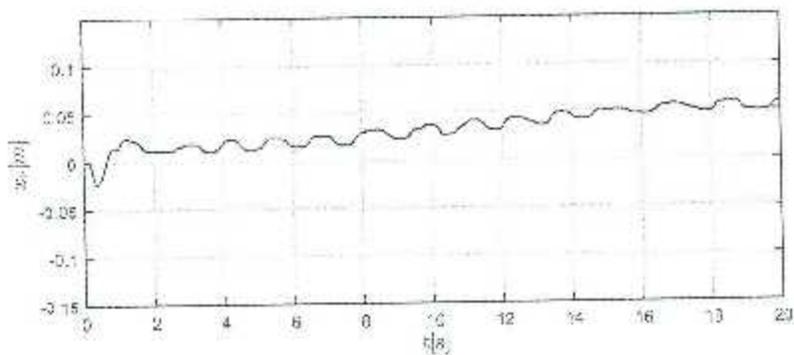
Figura 7.3: Regulador LQR: Acciones de control (experimento 1).



Angulo de inclinación del péndulo



Orientación del robot



Posición en la dirección de avance del robot

Figura 7.4: Regulador LQR: Variables de postura (experimento 1).

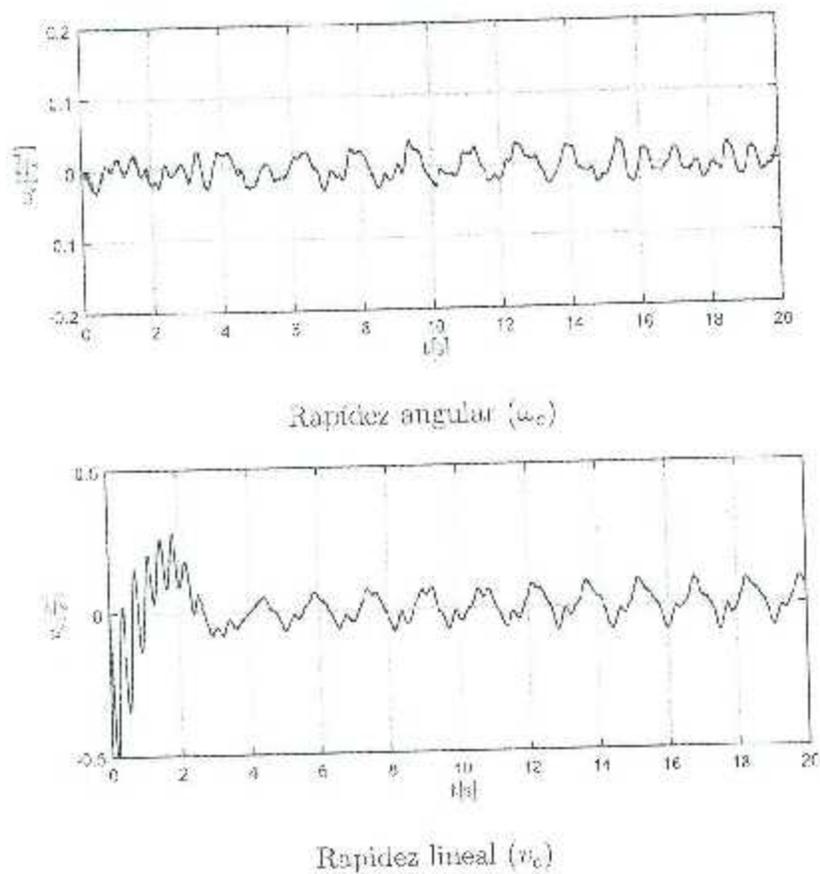
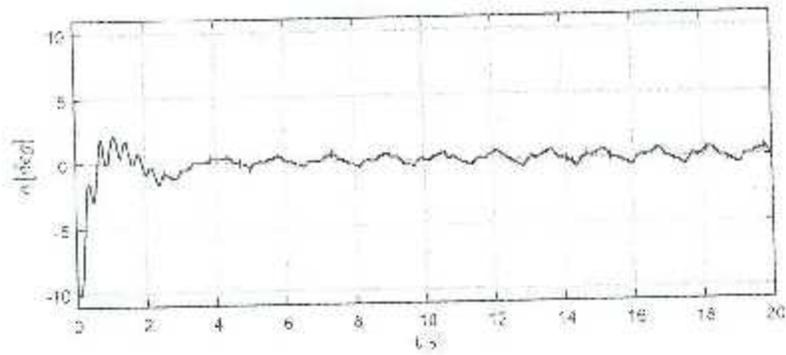
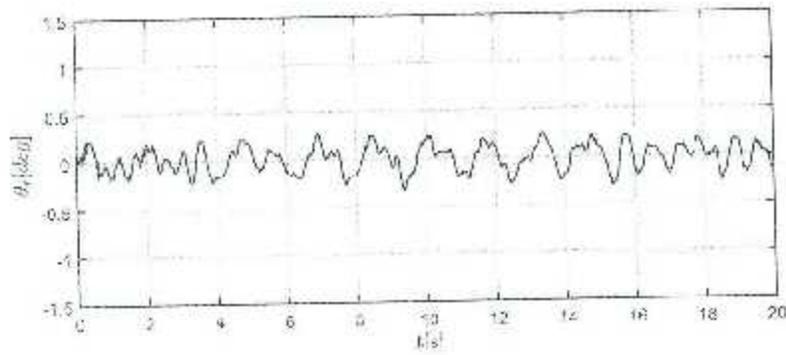


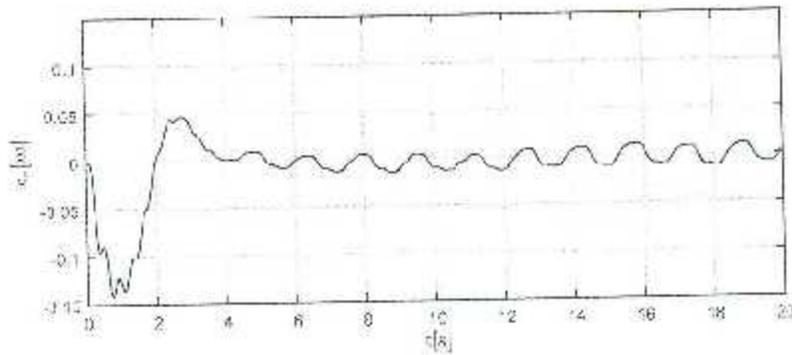
Figura 7.5: Regulador LQR: Acciones de control (experimento 2).



Inclinación del péndulo

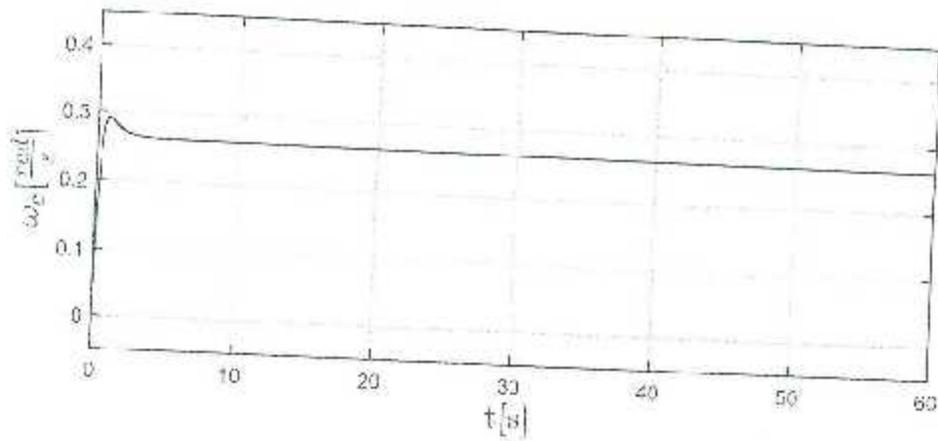


Orientación del robot

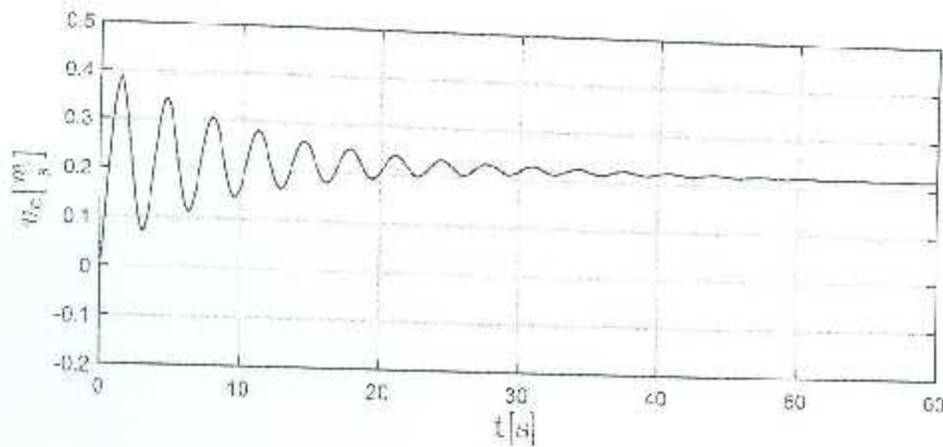


Posición en dirección de avance del robot

Figura 7.6: Regulador LQR: Variables de postura (experimento 2).

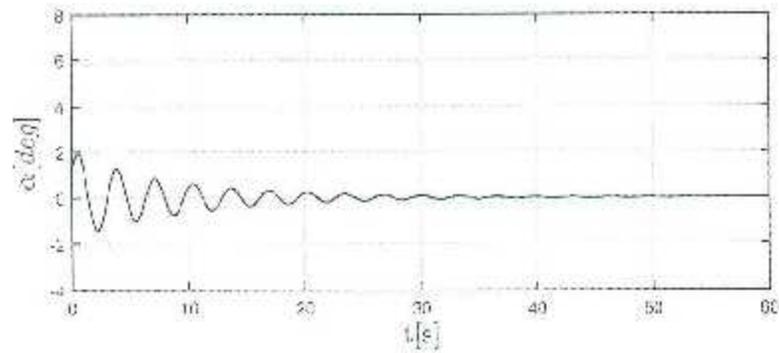


Rapidez angular (ω_c)

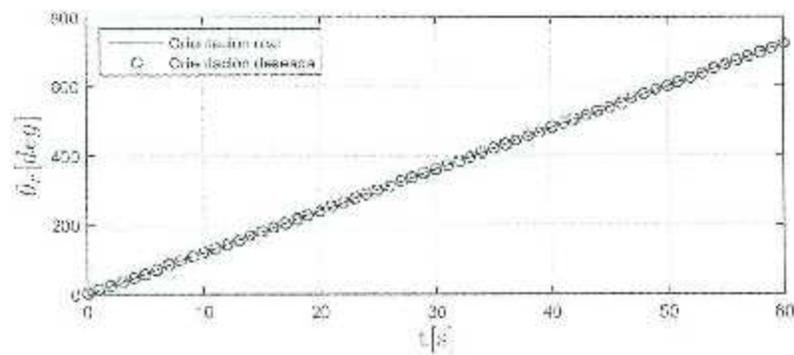


Rapidez lineal (v_c)

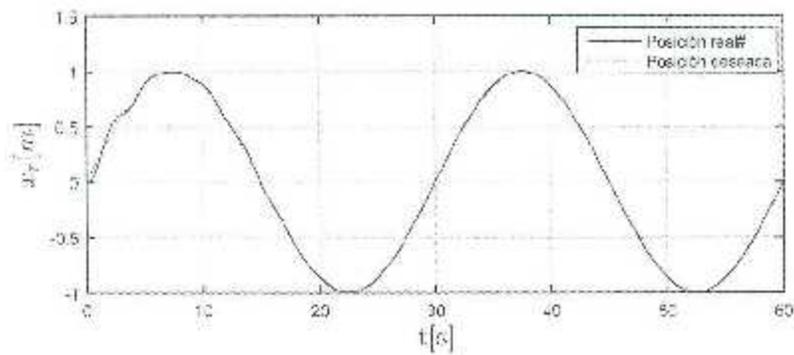
Figura 7.7: Seguidor LQR: Acciones de control (simulación).



Inclinación del péndulo



Angulo de orientación del robot



Posición en la dirección de avance del robot

Figura 7.8: Seguidor LQR: Variables de postura (simulación).

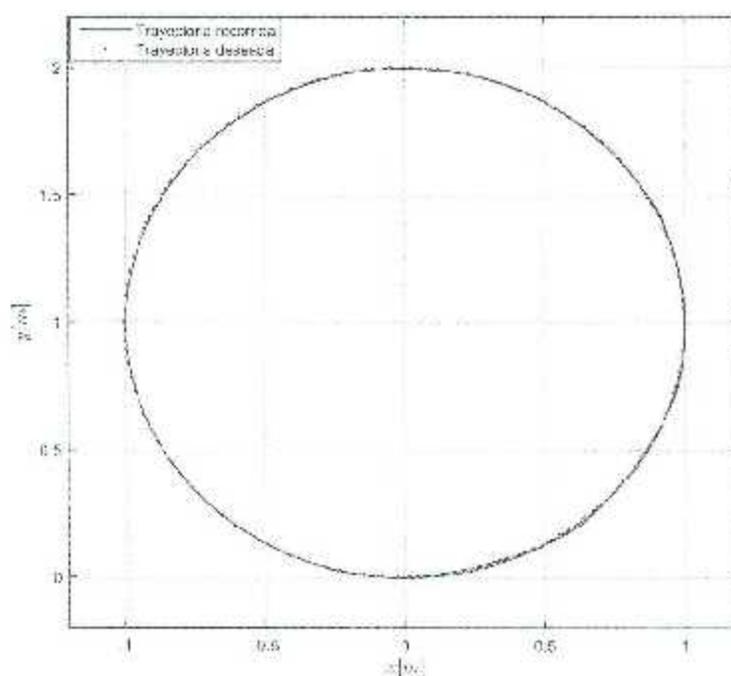
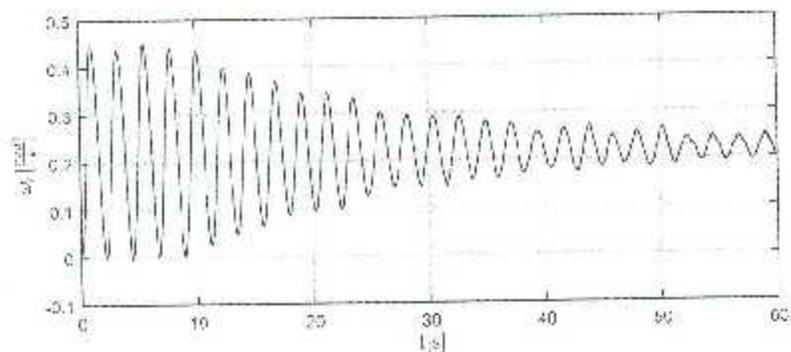


Figura 7.9: Seguidor LQR: Trayectoria recorrida (simulación).

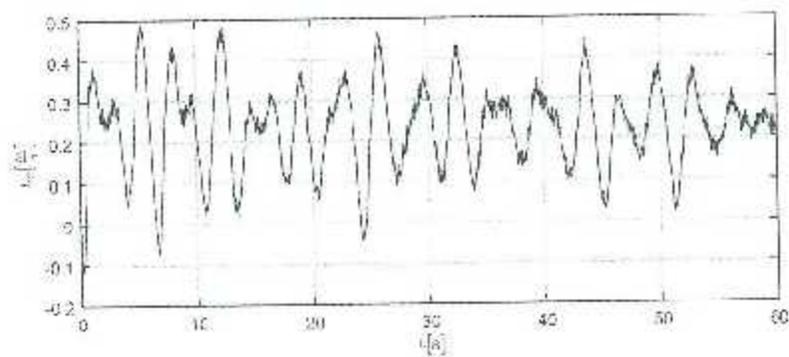
Experimentos

Las señales de control mostradas en la figura 7.10 se encuentran dentro de los valores permitidos, aunque cuentan con grandes oscilaciones. A pesar de esto las variables logran ser controladas. La evolución de las variables se muestran en la figura 7.11, para el caso del ángulo de inclinación α , se logró mantener el péndulo con oscilaciones alrededor de los 2° con una amplitud de $\pm 4^\circ$ aproximadamente, esto debido a que el movimiento continuo en el eje x_r perturba la inclinación del robot. Para la orientación del robot θ , se logró que el robot siguiera la señal de referencia con oscilaciones alrededor de ésta. Por último para el movimiento en la dirección de la orientación del robot x_r , el robot intenta seguir la trayectoria, pero debido a perturbaciones ocurridas entre el péndulo y esta variable se producen errores de seguimiento, los cuales se observan en la figura 7.12, donde es posible apreciar que el círculo que traza el robot se va moviendo. Este movimiento de la trayectoria producida por el robot se debe a que en este controlador se realimentan variables relativas al robot, sin tener en cuenta las posiciones absolutas

en el eje x y el eje y .

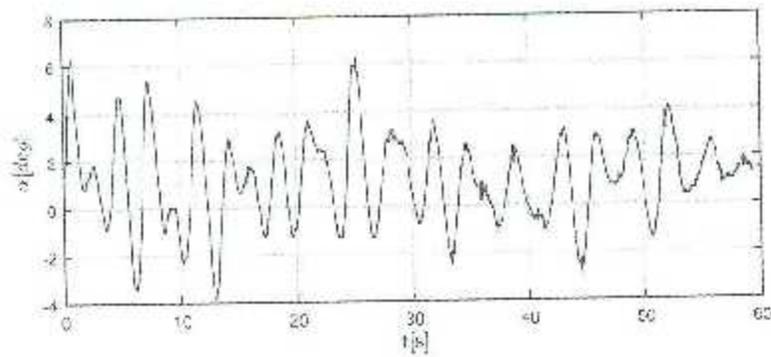


Rapidez angular (ω_c)

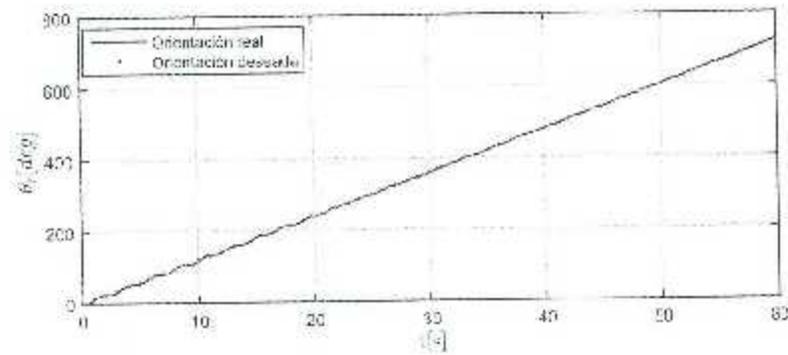


Rapidez lineal (v_c)

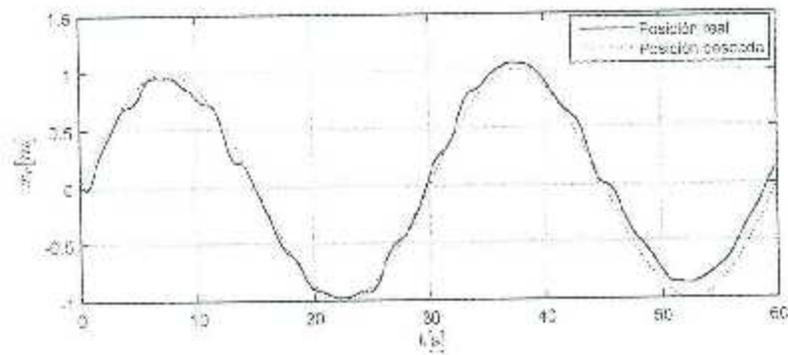
Figura 7.10: Seguidor LQR: Acciones de control (experimento).



Inclinación del péndulo



Orientación del robot



Posición en la dirección de avances del robot

Figura 7.11: Seguidor LQR: Variables de postura (experimento).

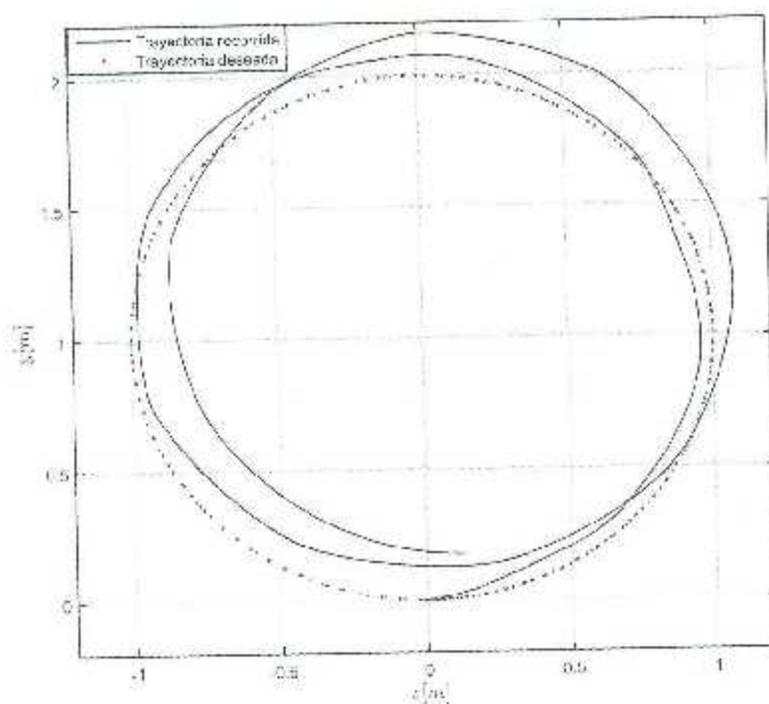


Figura 7.12: Seguidor LQR: Trayectoria recorrida (experimento).

8 Conclusiones

8.1. Aportaciones de la tesis

- Se presentan dos metodologías para el modelado cinemático de velocidad de robots móviles con ruedas y además se aborda el modelado dinámico para este tipo de sistemas (con restricciones no holonómicas), utilizando el método de Euler-Lagrange así como un método alternativo: el método de Kane. Se aplican todos estos métodos a la plataforma Segway RMP-100.
- Se desarrolló un nuevo programa para la implementación de controladores en la plataforma, ya que el programa que traía de fábrica funcionaba con periodos de muestreo muy grandes y variables (de entre 60 y 75 ms). En el presente trabajo, utilizando un protocolo de comunicación diferente (bus CAN), se logró reducir el periodo de muestreo a 13 ms, siendo este periodo ahora estricto y la cual es la máxima velocidad de comunicación que se puede lograr entre la plataforma y la computadora de control sin modificar el hardware o software interno de la plataforma. Además de lograr la mejora en la velocidad de comunicación, el nuevo programa funciona en tiempo real, lo que asegura un mejor desempeño de los controladores implementados.
- Se implementaron varios controladores en la plataforma. En el modo de autobalanceo se probaron controladores de regulación y seguimiento de trayectorias, ya que idealmente se puede considerar como si fuera un RMR tipo uniciclo. En el modo de tractor se implementó un controlador LQR para el balanceo del péndulo y se probó éste tanto para regulación como para seguimiento de trayectorias.
- Se presentó un artículo en el *Congreso Internacional de Robótica y Computación*

2016 (Soto *et al.*, 2016), que incluye tanto resultados previos como los obtenidos en esta tesis.

8.2. Trabajo futuro

Algunos temas que pueden ser abordados en el futuro con la plataforma pueden ser resumidos en los siguientes puntos:

- **Mejora de la interfaz de usuario**

Se busca modificar la interfaz para que sea más amigable al usuario. Entre las mejoras que se consideran está la generación de gráficas en línea.

- **Modelo de fricción**

Se pretende agregar al modelo dinámico del robot un modelo adecuado para la fricción existente en los actuadores así como entre las ruedas y la superficie del piso.

- **Implementación de controladores no lineales basados en el modelo**

Esto permitirá comprobar que tan bien se aproxima el modelo dinámico obtenido a la realidad, y qué tan válidas son las transformaciones de par a velocidad descritas en la sección 7.

- **Instrumentación del robot**

Sería útil para la planificación de trayectorias dotar al robot de sensores como cámaras o sensores láser que permitan conocer el entorno en el cual se encuentra.

- **Adición de un brazo manipulador a la plataforma**

Si además de sensores extras se dota a la plataforma con un brazo manipulador, esto permitiría al robot no sólo conocer su entorno sino además interactuar con él.

- **Evaluación del desempeño de controladores usando una computadora a bordo**

Sería útil comparar el desempeño de la computadora de control actual (la cual es fija y necesita de un cable muy grande para poder comunicarse con el robot) y una

computadora a bordo, la cual podría ir montada en la plataforma eliminando las restricciones de movimientos impuestas por el alcance del cable de comunicación.

- **Estudio a detalle sobre planificación de trayectorias para robots móviles**

Algunos planificadores de trayectorias emplean el modelo dinámico en la planificación, y permitirían mejorar el desempeño de controladores cinemáticos en la plataforma, ya que la mayoría de los controladores ignoran la dinámica del sistema, y ésta sería incluida ahora en la planificación de la trayectoria.

Bibliografía

- Albagul, A. y Wahyudi, M. (2004). Dynamic modeling and adaptive traction control for mobile robots. En: *Proceedings of the 30th Annual Conference of the IEEE Industrial Electronics Society*, Busan, South Korea.
- Alvarado, O. (2010). *Construcción, modelado y control de un mecanismo tipo péndulo invertido esférico*. Tesis de maestría. Instituto Tecnológico de la Laguna, Torreón, Coah, México.
- Anderson, C. W. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, vol. 9, No. 3, pp. 31–37.
- Bloch, A. M. (2003). *Nonholonomic Mechanics and Control*. Springer.
- Boyd, F. D. y Velinsky, S. A. (1994). Dynamic modeling of wheeled mobile robots for high load applications. En: *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, USA.
- Brockert, R. W. (1982). Control theory and singular riemannian geometry. En: Hilton P. J. y Young G. S. (Eds.) *New Directions in Applied Mathematics*, pp. 11–27. Springer.
- Brockett, R. W., Milman, R. S., y Sussman, H. J. (1983). Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, pp. 181–191.
- Campa, R. (2005). *Control de robots manipuladores en espacio de tarea*. Tesis doctoral, CICESE, Ensenada, B.C., México.
- Campa, R. y de la Torre, H. (2009). Pose control of robot manipulators using different orientation representations: A comparative review. En: *2009 American Control Conference*, pp. 2855–2860.

- Campa, R., Kelly, R., y Santibáñez, V. (2004). Windows-based real-time control of direct-drive mechanisms: Platform description and experiments. *Mechatronics*, vol. 14, No. 9, pp. 1021–1036.
- Campion, G., Bastin, G., y D'Andrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62.
- Costescu, N., Dawson, D., y Laffer, M. (1999). Qmotor 2.0: A real-time pc based control environment. *IEEE Control Systems Magazine*, 19(3):68–76.
- de Wit, C. C., Siciliano, B., y Bastin, G. (1996). *Theory of Robot Control*. Springer.
- Kane, T. R. (1962). Dynamics of nonholonomic systems. *Journal of Applied Mechanics*, 28:574–578.
- Kane, T. R. y Levinson, D. A. (1983). The use of Kane's dynamical equations in robotics. *The International Journal of Robotics Research*, 2:3–21.
- Kelly, R. y Santibáñez, V. (2003). *Control de movimiento de robots manipuladores*. Pearson-Prentice Hall.
- Laumond, J. P., Jacobs, P. E., Taix, M., y Murray, R. M. (1994). A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. Available at <http://planning.cs.uiuc.edu/>.
- Li, Z., Yang, C., y Fan, L. (2013). *Advanced Control of Wheeled Inverted Pendulum Systems*. Springer.
- Low, K. H. y Loow, Y. P. (2005). Kinematic modeling, mobility analysis and design of wheeled mobile robots. *Advanced Robotics*, 19:73–99.
- Luca, A. D. y Oriolo, G. (1995). *Modelling and Control of Nonholonomic Mechanical Systems*, pp. 277–342. Springer Vienna, Vienna.
- Maddocks, J. A. J. (1990). On the kinematics of wheeled mobile robots. *Autonomous Robot Vehicles*, 1:5–24.

- Monroy, C., Campa, R., y Kelly, R. (2001). An application of real-time control systems to robotics. *Robotica*, 19:323-329.
- Muir, P. F. y Neuman, C. P. (1987). Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, 4:281-340.
- Murray, R. M. y Sastry, S. S. (1993). Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, vol. 38, No. 5, pp. 700-716.
- Nawawi, S. W., Ahmad, M. N., y Osman, J. H. (2006). Control of two-wheels inverted pendulum mobile robot using full order sliding mode control. En: *Proceedings of the International Conference on Man-Machine Systems*, Langkawi, Malaysia.
- Oriolo, G., Luca, A. D., y Vendittelli, M. (2002). WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, vol. 10, No. 6, pp.835-852.
- Palm, W. (2013). *System Dynamics*. McGraw Hill.
- Pathak, K., Franch, J., y Agrawal, S. K. (2005). Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Transactions on Robotics*, vol. 21, No. 3, pp.505-513.
- Salerno, A. y Angeles, J. (2003). On the nonlinear controllability of quasiholonomic mobile robot. En: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.
- Sciavicco, L. y Siciliano, B. (2000). *Modeling and Control of Robot Manipulators*. Springer.
- Sheth, P. y Uicker, J. J. (1971). A generalized symbolic notation for mechanisms. *Journal of Engineering for Industry*, 93(1):102-112.
- Shin, D. H. y Park, K. H. (2001). Velocity kinematic modeling for wheeled mobile robots. En: *Proceedings of the IEEE Conference on Robotics and Automation*, Seoul, South Korea.
- Siciliano, B. y Khatib, O. (2008). *Handbook of Robotics*. Springer.
- Siegwart, R., Nourbakhsh, I. R., y Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT press.

- Soto, I. (2014). *Modelado y control de robots con restricciones holonómicas y no holonómicas*. Tesis doctoral, Instituto Tecnológico de la Laguna, Torreón, Coah., México.
- Soto, I., Bugarín, E., Delgado, A., y Campa, R. (2016). Implementación de algoritmos de control en tiempo real en la plataforma robótica Segway RMP-100. En *Memorias del II Congreso Internacional de Robótica y Computación*, Cabo San Lucas, B.C.S., México.
- Tanner, H. G. y Kyriakopoulos, K. J. (2003). Backstepping for nonsmooth systems. *Automatica*, vol. 39, No. 7, pp. 1259–1265.
- Thanjavur, K. y Rajagopalan, R. (1997). Ease of dynamic modelling of wheeled mobile robots (wmrs) using kane's approach. En: *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, USA.
- Tilbury, D., Laumond, J. P., Murray, R., Sastry, S., y Walsh, G. (1992). Steering car-like systems with trailers using sinusoids. En: *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France.
- Wheekuk, K., Byung-Ju, Y., y Jin, L. D. (2004). Kinematic modeling of mobile robots by transfer method of augmented generalized coordinates. *Journal of Robotic Systems*, vol. 21, No.6, pp. 301–322.
- Zhao, Y. y BeMent, S. L. (1992). Kinematics, dynamics and control of wheeled mobile robots. En: *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France.
- Zúñiga, P. (2014). *Diseño, modelado y control de una muñeca esférica con módulos powercube*. Tesis de maestría. Instituto Tecnológico de la Laguna, Torreón, Coah., México.

A RMP-100: Modelado cinemático de velocidad

En este apéndice se obtienen los modelos cinemáticos de velocidad de postura y de configuración de la plataforma Segway RMP-100, utilizando los métodos vistos en el capítulo 2.x.

A.1. Método de Muir-Neuman

Como se vio en la sección 2.2.1, es posible obtener los modelos cinemáticos de velocidad partiendo de la correcta colocación de ciertos marcos y obteniendo las matrices jacobianas de las ruedas.

A.1.1. Modelo cinemático de velocidad en espacio de postura

En la figura A.1 se muestra dos vistas de la plataforma y se han colocado los marcos Σ_0 , Σ_r , Σ_{H_1} , Σ_{H_2} y Σ_{w_1} . Se puede ver en la figura 2.2.1 que para la rueda 1 (rueda izquierda) se tiene que $\sigma_1 = \frac{\pi}{2}$ y $\beta_1 = 0$, mientras que para la rueda 2 (rueda derecha) $\sigma_2 = -\frac{\pi}{2}$ y $\beta_2 = \pi$, de modo que empleando la matriz jacobiana para ruedas fijas (2.7) se tiene que para la rueda 1:

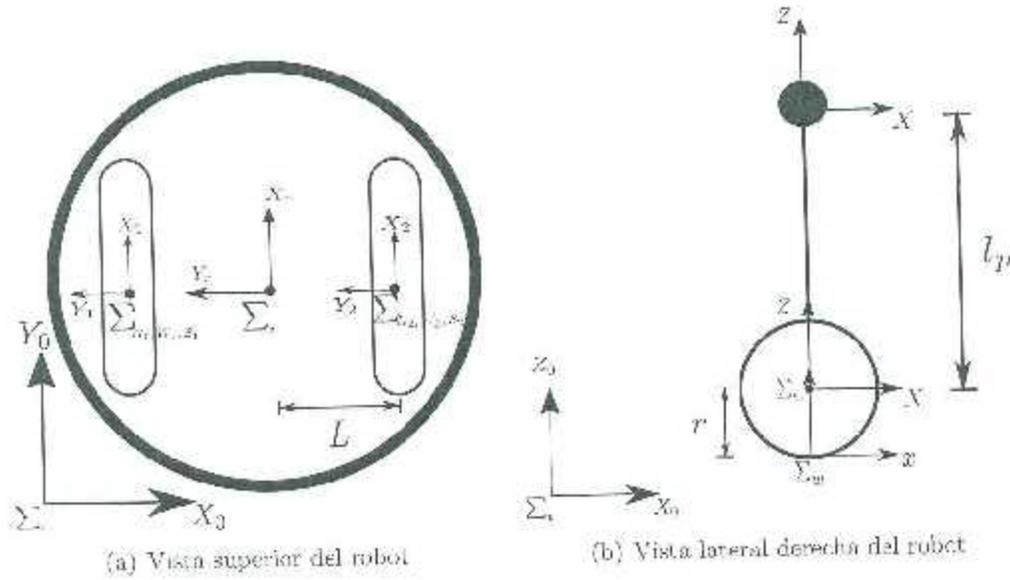


Figura A.1: Ubicación de los marcos coordenados en la plataforma Segway RMP-100

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} r & l \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\psi}_1 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} r & -L \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi}_2 \\ \dot{\psi}_2 \end{bmatrix}$$

donde $\dot{\phi}_i$ con $i = 1, 2$ es la rapidez de giro de la rueda i , respecto al eje Y_i de la misma y $\dot{\psi}_i$ es la rapidez de giro de orientación de la rueda con respecto al eje Z_i . Para combinar las dos expresiones es necesario obtener la pseudoinvertida izquierda o.e J_1^+ y J_2^+ usando:

$$J_i^+ = (J_i^T J_i)^{-1} J_i^T$$

con lo que se obtiene

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\psi}_1 \end{bmatrix} = J_1^+ \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & 0 & \frac{-l}{r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix}$$

$$\begin{bmatrix} \dot{\varphi}_2 \\ \dot{\psi}_2 \end{bmatrix} = J_2^+ \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & 0 & \frac{L}{r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix}$$

que se puede reescribir como

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\psi}_1 \\ \dot{\varphi}_2 \\ \dot{\psi}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & 0 & \frac{-L}{r} \\ 0 & 0 & 1 \\ \frac{1}{r} & 0 & \frac{L}{r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix}.$$

Pero como las ruedas de la plataforma son fijas, significa que sólo están actuadas las variables φ_1 y φ_2 ; además se observa que es posible despreciar la velocidad lineal en dirección del eje Y , del marco del robot (\dot{y}_r), así que ahora es necesario formular las matrices jacobianas para cada rueda, las cuales están dadas por:

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{-L}{r} \\ \frac{1}{r} & \frac{L}{r} \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{\theta}_r \end{bmatrix} = J_R \begin{bmatrix} v \\ \omega \end{bmatrix}$$

donde J_R es la matriz que relaciona las velocidades del RMR con las velocidades en las ruedas actuadas del mismo, $v = \dot{x}_r$ es la rapidez de avance en la dirección de las ruedas y $\omega = \dot{\theta}_r$ es la rapidez de giro alrededor del eje Z .

Despejando:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J_R^{-1} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{-r}{2L} & \frac{r}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix}$$

El modelo cinemático de velocidad en espacio de postura se obtiene usando (2.21)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\omega}_r \end{bmatrix}$$

pero como $\dot{y}_r = 0$, entonces queda simplemente

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

A.1.2. Modelo cinemático de velocidad en espacio de configuración

Para tener el modelo cinemático de velocidad en espacio de configuración, se agrega al modelo cinemático de velocidad en espacio de postura el vector $\phi = [\phi_1 \ \phi_2]^T$, es decir:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \\ \frac{1}{r} & \frac{-L}{r} \\ \frac{1}{r} & \frac{L}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

pero en el caso de la plataforma RMP-100 hay que agregar también la variable de configuración del ángulo α , de modo que queda

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{r} & \frac{-L}{r} & 0 \\ \frac{1}{r} & \frac{L}{r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ \dot{\alpha} \end{bmatrix}.$$

A.2. Método de Campion *et al.*

En este método se parte de las ecuaciones que definen las restricciones no holonómicas. Las restricciones de rodamiento puro están dadas por (ver sección 2.1.1):

$$\begin{bmatrix} -1 & 0 & L \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + r\dot{\varphi}_1 = 0 \quad (\text{A.1})$$

$$\begin{bmatrix} -1 & 0 & -L \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + r\dot{\varphi}_2 = 0 \quad (\text{A.2})$$

y debido a que ambas ruedas están en un mismo eje, la restricción de no deslizamiento lateral es la misma para ambas y está dada por:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0 \quad (\text{A.3})$$

Ahora es posible agrupar (A.1) y (A.2) de la forma (2.11) y (A.3) de la forma (2.12), entonces se tiene que las matrices $J_1(\beta_s, \beta_c)$, J_2 , $C_1(\beta_s, \beta_c)$ y C_2 están dadas por:

$$\begin{aligned} J_1(\beta_s, \beta_c) &= \begin{bmatrix} -1 & 0 & L \\ -1 & 0 & -L \end{bmatrix} & J_2 &= \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \\ C_1(\beta_s, \beta_c) &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & C_2 &= 0 \end{aligned}$$

Es importante notar que $C_1(\beta_s) = 0$ ya que el robot no cuenta con ninguna rueda direccional, lo que automáticamente hace que el grado de direccionalidad sea 0, es decir

$$\delta_s = \text{rank}(C_1(\beta_s)) = \text{rank}(0) = 0$$

y $C_1^*(\beta_s) = C_1(\beta_s, \beta_c)$ ya que sólo existen ruedas fijas. Entonces el grado de movilidad dado por (2.17) es $\delta_s = 2$. De esto se concluye que es posible definir la plataforma como un RMR tipo (2,0), y es posible utilizar las definiciones de la tabla 2.1 para este tipo de

robot, es decir

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

con $u_x = u_y = v$ y $u_\theta = \omega$.

A.2.1. Modelo cinemático de velocidad de en espacio de postura

El modelo cinemático de velocidad en espacio de postura se puede obtener con la ecuación (2.28), y está dado por:

$$\dot{\xi} = R(\vartheta)W(\beta_s)\mathbf{u},$$

entonces el modelo cinemático de velocidad en espacio de postura queda definido como

$$\dot{\xi} = \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

o equivalentemente

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\vartheta) & 0 \\ \sin(\vartheta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

A.2.2. Modelo cinemático de velocidad en espacio de configuración

Este modelo está definido según la ecuación (2.25), donde $D(\beta_c) = -C_{2c}^{-1}C_{1c}(\beta_c) = 0$,

y

$$E(\beta_s, \beta_c) = -J_2^{-1}J_1(\beta_s, \beta_c)$$

$$E(\beta_s, \beta_c) = -\begin{bmatrix} \frac{1}{r} & 0 \\ 0 & \frac{1}{r} \end{bmatrix} \begin{bmatrix} -1 & 0 & L \\ -1 & 0 & -L \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & 0 & \frac{-L}{r} \\ \frac{1}{r} & 0 & \frac{L}{r} \end{bmatrix}$$

entonces

$$S(\rho) = \begin{bmatrix} R(\theta)W(\beta_s) \\ 0 \\ 0 \\ E(\beta_s, \beta_c)W(\beta_s) \end{bmatrix}$$

sustituyendo los valores de las matrices ya obtenidas y agregando nuevamente el componente de la velocidad del péndulo $\dot{\alpha}$, se llega al mismo modelo cinemático de velocidad de configuración que ya se obtuvo por el método de Muir-Neuman:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{r} & \frac{-L}{r} & 0 \\ \frac{1}{r} & \frac{L}{r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ \dot{\alpha} \end{bmatrix}$$

B RMP-100: Modelado dinámico

En este apéndice se detalla paso a paso cómo obtener el modelo dinámico de la plataforma Segway RMP-100 utilizando los métodos vistos en el capítulo 2. Partiendo del diagrama de la plataforma presentado en la figura B.1, se sabe que se tienen tres cuerpos: el péndulo, la rueda izquierda y la rueda derecha. Y se define el vector de variables de configuración ρ como

$$\rho = [x \ y \ \theta \ \alpha \ \varphi_l \ \varphi_r]^T \quad (\text{B.1})$$

mientras que el vector de velocidades generalizadas mínimas \dot{q} está dado por

$$\dot{q} = [\dot{x} \ \dot{y} \ \dot{\alpha}]^T \quad (\text{B.2})$$

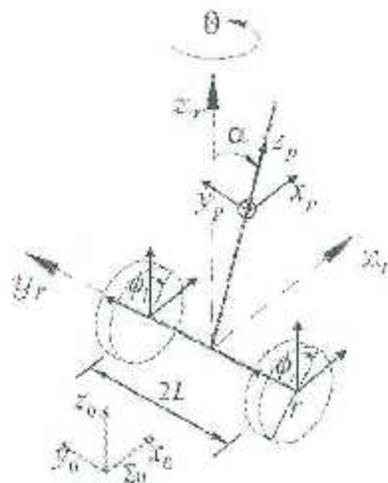


Figura B.1: Diagrama esquemático.

con $v = \dot{x}$, y $\omega = \dot{\theta}$.

B.1. Método de Euler-Lagrange

Para obtener el modelo dinámico utilizando este método, se debe obtener la energía cinética de cada cuerpo del robot; la energía total está dada por

$$K(\rho, \dot{\rho}) = K_p + K_wr + K_wl$$

donde K_p , K_wr y K_wl representan la energía del péndulo, la rueda derecha y la rueda izquierda.

La energía cinética del péndulo está dada por

$$K_p = \frac{1}{2} m_p v_p^T v_p + \frac{1}{2} \omega_p^T J_p \omega_p$$

El vector de posición del centro de masa del péndulo está definido por

$$p_p = p_b + {}^b p_p = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} l_p C_\theta S_\alpha \\ l_p S_\theta S_\alpha \\ l_p C_\alpha \end{bmatrix}$$

donde p_b es el vector de posición del marco de referencia inercial al marco de la base, ${}^b p_p$ es el vector de posición del marco de la base al marco del péndulo, $C_\alpha = \cos(\alpha)$, $S_\alpha = \sin(\alpha)$, $C_\theta = \cos(\theta)$ y $S_\theta = \sin(\theta)$; una vez más, la velocidad está dada por la derivada del vector de posición del péndulo, por lo tanto

$$v_p = \dot{p}_p = \begin{bmatrix} \dot{x} - l_p S_\theta S_\alpha \dot{\theta} + l_p C_\theta C_\alpha \dot{\alpha} \\ \dot{y} + l_p C_\theta S_\alpha \dot{\theta} + l_p S_\theta C_\alpha \dot{\alpha} \\ -l_p S_\alpha \dot{\alpha} \end{bmatrix}$$

Ahora para obtener la velocidad angular del péndulo ω_p , se considera la matriz de

rotación

$$R_p = \begin{bmatrix} C_\alpha & 0 & S_\alpha \\ 0 & 1 & 0 \\ -S_\alpha & 0 & C_\alpha \end{bmatrix}$$

y usando, (2.42) se llega a la velocidad angular del péndulo que está dada por

$$\omega_p = \begin{bmatrix} -S_\theta \dot{\alpha} \\ C_\theta \dot{\alpha} \\ \dot{\theta} \end{bmatrix};$$

con estos datos ya se puede obtener la energía cinética del péndulo, la cual queda como

$$K_p = \frac{1}{2} m_p \left(l_p^2 \dot{\alpha}^2 - \dot{x}^2 + \dot{y}^2 + l_p^2 \dot{\theta}^2 (1 - C_\alpha^2) + 2l_p C_\alpha \dot{\alpha} (C_\theta \dot{x} + S_\theta \dot{y}) + 2l_p S_\alpha \dot{\theta} (C_\theta \dot{y} - S_\theta \dot{x}) \right) + \frac{1}{2} \left(I_{py} \dot{\alpha}^2 + I_{pz} \dot{\theta}^2 + I_{px} S_\alpha^2 \dot{\theta}^2 - I_{px} S_\alpha^2 \dot{\alpha}^2 \right),$$

La energía cinética de la rueda izquierda está dada por

$$K_{wl} = \frac{1}{2} m_l v_l^T v_l + \frac{1}{2} \omega_l^T I_l \omega_l$$

donde m_l es la masa de la rueda izquierda, v_l y ω_l son las velocidades lineal y angular de la rueda izquierda, respectivamente, ambas medidas respecto marco inercial, 0I_l es el tensor de inercia del centro de masa de la rueda izquierda con respecto al marco inercial. La velocidad lineal está dada por la derivada respecto al tiempo de su vector de posición, el cual está dado por

$$p_l = p_0 + {}^0p_l = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} -L S_\theta \\ L C_\theta \\ 0 \end{bmatrix}$$

donde ahora 0p_l es el vector de posición del marco de la base al marco de la rueda izquierda. Derivando estas ecuaciones con respecto al tiempo, se puede obtener la

velocidad lineal:

$$\mathbf{v}_l = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} + \begin{bmatrix} LC_\theta \dot{\theta} \\ -LS_\theta \dot{\theta} \\ 0 \end{bmatrix}$$

La velocidad angular de la rueda izquierda se obtiene utilizando las mismas relaciones usadas para obtener la velocidad angular del péndulo, por lo tanto la velocidad está dada por

$$\boldsymbol{\omega}_l = \begin{bmatrix} -S_\theta \dot{\phi}_l \\ C_\theta \dot{\phi}_l \\ \dot{\theta} \end{bmatrix}$$

el tensor de inercia I_l está dado por

$$I_l = R_l^T I_l R_l^T$$

con

$${}^l I_l = \begin{bmatrix} I_{lx} & 0 & 0 \\ 0 & I_{ly} & 0 \\ 0 & 0 & I_{lz} \end{bmatrix},$$

de tal manera que la energía cinética de la rueda izquierda está dada por

$$K_{w_l} = \frac{1}{2} m_l (\dot{x}^2 + \dot{y}^2 - 2L\dot{\theta}(C_\theta \dot{x} - S_\theta \dot{y}) + L^2 \dot{\theta}^2) + \frac{1}{2} (I_{ly} \dot{\phi}_l^2 + I_{lz} \dot{\theta}^2 - S_\theta^2 \dot{\phi}_l^2 (I_{lx} - I_{ly})).$$

De manera similar, es posible obtener la energía cinética de la rueda derecha, la cual queda dada por

$$K_{w_r} = \frac{1}{2} m_r (\dot{x}^2 + \dot{y}^2 + 2L\dot{\theta}(C_\theta \dot{x} - S_\theta \dot{y}) + L^2 \dot{\theta}^2) + \frac{1}{2} (I_{ry} \dot{\phi}_r^2 + I_{rz} \dot{\theta}^2 + S_\theta^2 \dot{\phi}_r^2 (I_{rx} - I_{ry})).$$

Además es necesario obtener la energía potencial de cada uno de los cuerpos, pero si se considera que el robot se moverá en un plano horizontal, entonces el centro de masa de ambas ruedas, nunca cambiará de altura, por lo que no aportará energía potencial; el

donde

$$m_{11} = m_p + m_l + m_w$$

$$m_{13} = -m_p l_p S_\alpha S_\theta - m_l L C_\theta + m_w L C_\theta$$

$$m_{14} = m_p l_p C_\alpha C_\theta$$

$$m_{22} = m_p + m_l + m_w$$

$$m_{23} = m_p l_p S_\alpha C_\theta - m_l L S_\theta$$

$$m_{24} = m_p l_p C_\alpha S_\theta$$

$$m_{33} = m_p (l_p^2 - l_p^2 C_\alpha^2) + S_\alpha^2 (I_{px} - I_{py}) + L^2 (m_l + m_w) + S_\theta^2 (I_{lx} - I_{ly}) + S_\theta^2 (I_{lx} - I_{ly}) + I_{lx} - I_{ly} - I_{pz}$$

$$m_{44} = m_p l_p^2 - I_{py}$$

$$m_{55} = I_{ly}$$

$$m_{66} = I_{py}$$

La matriz centrífuga y de Coriolis $C(\rho, \dot{\rho})$ puede ser obtenida utilizando las siguientes ecuaciones

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial M_{ij}}{\partial \rho_k} + \frac{\partial M_{ik}}{\partial \rho_j} - \frac{\partial M_{jk}}{\partial \rho_i} \right),$$

y

$$C_{kj}(\rho, \dot{\rho}) = [c_{1jk}(\rho) \quad c_{2jk}(\rho) \quad \dots \quad c_{6jk}(\rho)] \dot{\rho}$$

donde $C_{kj}(\rho, \dot{\rho})$ es el kj -ésimo elemento de la matriz $C(\rho, \dot{\rho})$. Por último el vector de fuerzas debidas a la gravedad está dado por

$$\mathbf{g}(\rho) = \frac{\partial U(\rho)}{\partial \rho},$$

Ahora, es necesario introducir las restricciones del sistema. Utilizando la matriz

$$S(\rho) = \begin{bmatrix} C_\theta & 0 & 0 \\ S_\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{r} & -\frac{l}{r} & 0 \\ \frac{1}{r} & \frac{l}{r} & 0 \end{bmatrix}$$

del modelo cinemático de velocidad en espacio de configuración, entonces es posible obtener el modelo dinámico reducido, dado por (2.41), es decir

$$M_r(\rho)\ddot{q} - C_r(\rho, \dot{\rho})\dot{q} + g_r(\rho) = \tau_r$$

donde

$$M_r(\rho) = \begin{bmatrix} m_{r11} & 0 & m_{r13} \\ 0 & m_{r22} & 0 \\ m_{r13} & 0 & m_{r33} \end{bmatrix}$$

con

$$m_{r11} = \frac{1}{r^2}(I_{I2} + I_{I2}) + (m_U + m_p + m_c)$$

$$m_{r12} = 0$$

$$m_{r13} = m_p l_p C_\alpha$$

$$m_{r22} = I_{Iz} + I_{pz} + I_{Iz} + (m_U + m_c) l^2 + (I_{px} - I_{px}) S_\alpha^2 + (I_{Iy} + I_{Iy}) \frac{l^2}{r^2} + m_p l_p^2 S_\alpha^2$$

$$m_{r23} = 0$$

$$m_{r33} = m_p l_p^2 + I_{Iy}$$

La matriz centrífuga y de Coriolis reducida queda

$$C_r(\rho, \dot{\rho}) = \begin{bmatrix} 0 & 0 & -c_3 \dot{\alpha} \\ 0 & c_1 \dot{\alpha} & c_1 \dot{\theta} \\ 0 & -c_1 \dot{\theta} & 0 \end{bmatrix}$$

con

$$\begin{aligned} e_1 &= (m_p l_p^2 - (I_{px} - I_{pz})) C_\alpha S_\alpha \\ e_2 &= m_p l_p S_\alpha. \end{aligned}$$

Y por último el vector de gravedad queda definido por

$$g_i(\rho) = \begin{bmatrix} 0 \\ 0 \\ -gm_p l_p S_\alpha \end{bmatrix}.$$

B.2. Método de Kane

Para obtener el modelo dinámico utilizando este método, se parte de la matriz $S(\rho)$ que da la relación (2.40). Entonces se tiene que tal relación de velocidades está dada por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\alpha} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} C_\theta & 0 & 0 \\ S_\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{r} & \frac{-1}{r} & 0 \\ \frac{1}{r} & \frac{1}{r} & 0 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix}.$$

Ahora es necesario obtener las velocidades lineales y angulares de los cuerpos rígidos. Como ya fueron obtenidas en la sección anterior, el proceso se facilita, lo único diferente ahora es poner todas las velocidades en función de \dot{q} , donde $\dot{q} = [v \quad \dot{\theta} \quad \dot{\alpha}]$.

Para el péndulo, la velocidad lineal queda definida como

$$v_p = \begin{bmatrix} C_\theta v \\ S_\theta v \\ 0 \end{bmatrix} + \begin{bmatrix} -l_p S_\theta S_\alpha \dot{\theta} + l_p C_\theta C_\alpha \dot{\alpha} \\ l_p C_\theta S_\alpha \dot{\theta} + l_p S_\theta C_\alpha \dot{\alpha} \\ -l_p S_\alpha \dot{\alpha} \end{bmatrix}$$

y su velocidad angular

$$\omega_y = \begin{bmatrix} -S_\theta \dot{\alpha} \\ C_\theta \dot{\alpha} \\ 0 \end{bmatrix}.$$

Para la rueda izquierda la velocidad lineal queda definida como

$$v_l = \begin{bmatrix} C_\theta v \\ S_\theta v \\ 0 \end{bmatrix} + \begin{bmatrix} -LC_\theta \dot{\theta} \\ -LS_\theta \dot{\theta} \\ 0 \end{bmatrix}$$

y su velocidad angular por

$$\omega_l = \begin{bmatrix} -S_\theta \dot{\varphi}_l \\ C_\theta \dot{\varphi}_l \\ \omega \end{bmatrix} = \begin{bmatrix} -S_\theta \left(\frac{v}{r} - \frac{L}{r} \omega \right) \\ C_\theta \left(\frac{v}{r} - \frac{L}{r} \omega \right) \\ \omega \end{bmatrix}.$$

De igual manera, la velocidad lineal de la rueda derecha queda definida como

$$v_r = \begin{bmatrix} C_\theta v \\ S_\theta v \\ 0 \end{bmatrix} + \begin{bmatrix} LC_\theta \dot{\theta} \\ LS_\theta \dot{\theta} \\ 0 \end{bmatrix}$$

y su velocidad angular mediante

$$\omega_r = \begin{bmatrix} -S_\theta \dot{\varphi}_r \\ C_\theta \dot{\varphi}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -S_\theta \left(\frac{v}{r} + \frac{\delta L}{r} \right) \\ C_\theta \left(\frac{v}{r} + \frac{\delta L}{r} \right) \\ \dot{\theta} \end{bmatrix}.$$

Una vez obtenidas todas las velocidades, es posible obtener la jacobiana de posición (2.43), la cual se obtiene a través de la derivada parcial de las velocidades lineales respecto a \dot{q} :

$$J_p = \begin{bmatrix} \frac{\partial v_l}{\partial \dot{q}} & \frac{\partial v_r}{\partial \dot{q}} & \frac{\partial v_g}{\partial \dot{q}} \end{bmatrix}^T \in \mathbb{R}^{3 \times 6}$$

Desarrollando estas derivadas, la matriz J_p queda

$$J_p = \begin{bmatrix} C_\theta & -lC_\theta & 0 \\ S_\theta & -lS_\theta & 0 \\ 0 & 0 & 0 \\ C_\theta & lC_\theta & 0 \\ S_\theta & lS_\theta & 0 \\ 0 & 0 & 0 \\ C_\theta & -l_p S_\theta S_\alpha & l_p C_\theta C_\alpha \\ S_\theta & l_p C_\theta S_\alpha & l_p S_\theta C_\alpha \\ 0 & 0 & -l_p S_\alpha \end{bmatrix}.$$

La matriz jacobiana de orientación (2.44) se obtiene a través de la derivada parcial de las velocidades angulares respecto a \dot{q} :

$$J_o = \left[\frac{\partial \omega^T}{\partial \dot{q}_1} \quad \frac{\partial \omega^T}{\partial \dot{q}_2} \quad \frac{\partial \omega^T}{\partial \dot{q}_3} \right]^T \in \mathbb{R}^{3 \times 3}$$

y desarrollando nuevamente las derivadas parciales, se obtiene

$$J_o = \begin{bmatrix} -S_\theta \frac{1}{r} & S_\theta \frac{l}{r} & 0 \\ C_\theta \frac{1}{r} & -C_\theta \frac{l}{r} & 0 \\ 0 & \vdots & 0 \\ -S_\theta \frac{1}{r} & -S_\theta \frac{l}{r} & 0 \\ C_\theta \frac{1}{r} & C_\theta \frac{l}{r} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -S_\theta \\ 0 & 0 & C_\theta \\ 0 & 1 & 0 \end{bmatrix}.$$

Una vez obtenidas las velocidades del robot, es necesario encontrar las fuerzas inerciales generalizadas que actúan en cada cuerpo; esto se logra utilizando las siguientes

relaciones

$$\begin{aligned}\mu_k &= \frac{d}{dt}(I_k \omega_k) \\ f_k &= \frac{d}{dt}(m_k v_k)\end{aligned}$$

donde

$$\begin{aligned}I_k &= R_k {}^k I_k R_k^T \\ {}^k I_k &= \begin{bmatrix} I_{k_x} & 0 & 0 \\ 0 & I_{k_y} & 0 \\ 0 & 0 & I_{k_z} \end{bmatrix}\end{aligned}$$

y el subíndice k indica el cuerpo que se está analizando. Las fuerzas en el péndulo están dadas por

$$\mu_p = \frac{d}{dt}(I_p \omega_p) = \begin{bmatrix} \mu_{p1} \\ \mu_{p2} \\ \mu_{p3} \end{bmatrix}$$

donde

$$\begin{aligned}\mu_{p1} &= (I_{p_x} - I_{p_z})(\dot{\alpha}\dot{\theta}C_\theta(1 - 2C_\alpha^2) - C_\alpha S_\alpha S_\theta(\dot{\theta}^2 - \ddot{\theta})) - I_{p_z}(\ddot{\alpha}S_\theta - \dot{\alpha}\dot{\theta}C_\theta) \\ \mu_{p2} &= I_{p_y}(\ddot{\alpha}C_\theta - \dot{\theta}\dot{\alpha}S_\theta) - (I_{p_x} - I_{p_z})\left(\frac{\dot{\theta}^2 S_{2\alpha} C_\theta}{2} + \frac{\ddot{\theta} S_{2\alpha} S_\theta}{2} + \dot{\alpha}\dot{\theta}C_{2\alpha}S_\theta\right) \\ \mu_{p3} &= (I_{p_x} - I_{p_z})(\ddot{\theta}S_\alpha^2 + 2\dot{\alpha}\dot{\theta}C_\alpha S_\alpha) + I_{p_z}\ddot{\theta}\end{aligned}$$

y por

$$\begin{aligned}
 \mathbf{f}_p = \frac{d}{dt}(m_p \mathbf{v}_p) = m_p \begin{bmatrix} \ddot{v}C_\theta - vS_\theta\dot{\theta} \\ \dot{v}S_\theta + vC_\theta\dot{\theta} \\ 0 \end{bmatrix} \\
 - m_p \begin{bmatrix} -l_p S_\theta(S_\theta\ddot{\theta} + 2C_\theta\dot{\theta}\dot{\alpha}) - l_p C_\theta S_\theta(\dot{v} + \dot{\alpha}) + l_p C_\theta C_\alpha\ddot{\alpha} \\ l_p S_\alpha(C_\theta\ddot{\theta} - S_\theta(\dot{\theta}^2 + \dot{\alpha}^2)) - 2l_p C_\theta C_\alpha\dot{\theta}\dot{\alpha} + l_p S_\theta C_\alpha\ddot{\alpha} \\ -l_p(S_\alpha\ddot{\alpha} + C_\alpha\dot{\alpha}^2) \end{bmatrix}.
 \end{aligned}$$

Las fuerzas en la rueda izquierda están dadas por

$$\boldsymbol{\mu}_i = \frac{d}{dt}(I_i \boldsymbol{\omega}_i) = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \\ \mu_{i3} \end{bmatrix}$$

donde

$$\begin{aligned}
 \mu_{i1} &= \frac{\ddot{\theta}(I_{i_w} L S_\theta - I_{i_r} r C_{\varphi_i} C_\theta S_{\varphi_i} + I_{i_z} r C_{\varphi_i} C_\theta S_{\varphi_i})}{r} \\
 &\quad - \frac{\dot{\theta} \left(I_{i_w} v C_\theta - I_{i_w} L \dot{\theta} C_\theta - I_{i_x} r \dot{\theta} C_{\varphi_i} S_{\varphi_i} S_\theta + I_{i_x} r \dot{\theta} C_{\varphi_i} S_{\varphi_i} S_\theta \right)}{r} \\
 &\quad - \frac{I_{i_w} \dot{v} S_\theta + \dot{\theta} C_\theta (v - L \dot{\theta}) (I_{i_z} - I_{i_x}) (2C_{\varphi_i}^2 - 1)}{r} \\
 \mu_{i2} &= \frac{\dot{\theta} \left(2I_{i_w} v S_\theta - 2I_{i_w} L \dot{\theta} S_\theta - I_{i_w} r \dot{\theta} S_{2\varphi_i} C_\theta - I_{i_z} r \dot{\theta} S_{2\varphi_i} C_\theta \right)}{2r} \\
 &\quad - \frac{\ddot{\theta} \left(2I_{i_w} L C_\theta + I_{i_x} r S_{2\varphi_i} S_\theta - I_{i_x} r S_{2\varphi_i} S_\theta \right)}{2r} + \frac{I_{i_w} \dot{v} C_\theta}{r} \\
 &\quad - \frac{\dot{\theta} C_{2\varphi_i} S_\theta (v - L \dot{\theta}) (I_{i_z} - I_{i_x})}{r} \\
 \mu_{i3} &= \ddot{\theta} (I_{i_x} + I_{i_z} S_{\varphi_i}^2 - I_{i_x} S_{\varphi_i}^2) + \frac{\dot{\theta} S_{2\varphi_i} (v - L \dot{\theta}) (I_{i_z} - I_{i_x})}{r}
 \end{aligned}$$

Ahora

$$\mathbf{f}_i = \frac{d}{dt}(m_i \mathbf{v}_i) = m_i \begin{bmatrix} \dot{v}C_\theta - vS_\theta\dot{\theta} \\ \dot{v}S_\theta + vC_\theta\dot{\theta} \\ 0 \end{bmatrix} + m_i \begin{bmatrix} -\ddot{\theta}LC_\theta - \dot{\theta}^2LS_\theta \\ -\ddot{\theta}LS_\theta - \dot{\theta}^2LC_\theta \\ 0 \end{bmatrix}.$$

Y por último las fuerzas de la rueda derecha están dadas por

$$\boldsymbol{\mu}_r = \frac{d}{dt}(I_r \boldsymbol{\omega}_r) = \begin{bmatrix} \mu_{r1} \\ \mu_{r2} \\ \mu_{r3} \end{bmatrix}$$

$$\begin{aligned} \mu_{r1} &= -\frac{\ddot{\theta} (I_{r_1} L S_\theta + I_{r_2} r C_{\varphi_r} C_\theta S_{\varphi_r} - I_{r_2} r C_{\varphi_r} C_\theta S_{\varphi_r})}{r} \\ &\quad - \frac{\dot{\theta} (I_{r_1} v C_\theta + I_{r_2} L \dot{\theta} C_\theta - I_{r_2} r \dot{\theta} C_{\varphi_r} S_{\varphi_r} S_\theta - I_{r_2} r \dot{\theta} C_{\varphi_r} S_{\varphi_r} S_\theta)}{r} \\ &\quad - \frac{I_{r_2} \dot{v} S_\theta}{r} - \frac{\dot{\theta} C_\theta (v - L\dot{\theta}) (I_{r_1} - I_{r_2}) (2C_{\varphi_r}^2 - 1)}{r} \\ \mu_{r2} &= \frac{\ddot{\theta} (I_{r_1} L C_\theta - \frac{I_{r_2} r S_{2\varphi_r} S_\theta}{2} - \frac{I_{r_2} r S_{2\varphi_r} S_\theta}{2})}{r} + \frac{I_{r_1} \dot{v} C_\theta}{r} \\ &\quad - \frac{\dot{\theta} (I_{r_2} v S_\theta + I_{r_2} L \dot{\theta} S_\theta - \frac{I_{r_2} r \dot{\theta} S_{1\varphi_r} C_\theta}{2} - \frac{I_{r_2} r \dot{\theta} S_{1\varphi_r} C_\theta}{2})}{r} \\ &\quad - \frac{\dot{\theta} C_{2\varphi_r} S_\theta (v + L\dot{\theta}) (I_{r_1} - I_{r_2})}{r} \end{aligned}$$

$$\mu_{r3} = \dot{\theta} (I_{r_2} + I_{r_2} S_{\varphi_r}^2 - I_{r_2} S_{\varphi_r}^2) + \frac{\dot{\theta} S_{2\varphi_r} (v + L\dot{\theta}) (I_{r_1} - I_{r_2})}{r}$$

Y

$$f_r = \frac{d}{dt}(m_r {}^0\mathbf{v}_r) = m_r \begin{bmatrix} vC_\theta - vS_\theta\dot{\theta} \\ vS_\theta + vC_\theta\dot{\theta} \\ 0 \end{bmatrix} + m_r \begin{bmatrix} \dot{\theta}L C_\theta - \dot{\theta}^2 L S_\theta \\ \dot{\theta}L S_\theta + \dot{\theta}^2 L C_\theta \\ 0 \end{bmatrix}.$$

Por último es posible obtener las matrices de inercias y de Coriolis, utilizando la formulación de Kane, ver (2.47), donde

$$\begin{aligned} \mathbf{f} &= \mathbf{f}_i^T \quad \mathbf{f}_r^T \quad \mathbf{f}_p^T \\ \boldsymbol{\mu} &= \boldsymbol{\mu}_i^T \quad \boldsymbol{\mu}_r^T \quad \boldsymbol{\mu}_p^T \end{aligned}$$

Obteniéndose los elementos de la matriz $M(\mathbf{q})$, la cual teniendo en cuenta que $M(\mathbf{q}) = M(\mathbf{q})^T$, queda definida como:

$$M(\mathbf{q}) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix}$$

donde

$$\begin{aligned} m_{11} &= \frac{1}{r^2}(I_{iy} + I_{rx}) + (m_p + m_i + m_r) \\ m_{12} &= L \left(\frac{I_{rx} - I_{iy}}{r^2} + (m_r - m_i) \right) \\ m_{13} &= m_p l_p C_\alpha \\ m_{22} &= I_{iy} + I_{rx} - I_{rz} + L^2(m_r + m_i) + (I_{px} - I_{pz})S_\alpha^2 - \frac{L^2}{r^2}(I_{iy} - I_{rx}) + m_p l_p^2 S_\alpha^2 \\ &\quad + (I_{ix} - I_{iz})S_{\phi_i}^2 + (I_{rx} - I_{rz})S_{\phi_r}^2 \\ m_{23} &= 0 \\ m_{33} &= m_p l_p^2 + I_{py} \end{aligned}$$

La matriz centrífuga y de Coriolis queda definida como:

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

dónde

$$\begin{aligned} c_{11} &= c_{12} = c_{21} = c_{31} = c_{32} = 0 \\ c_{13} &= -m_p l_p S_\alpha \dot{\alpha} \\ c_{22} &= ((I_{px} - I_{py}) + m_p l_p^2)(S_\alpha C_\alpha) \dot{\alpha} \\ c_{23} &= ((I_{px} - I_{py}) + m_p l_p^2)(S_\alpha C_\alpha) \dot{\theta} \\ c_{32} &= -((I_{px} - I_{py}) + m_p l_p^2)(S_\alpha C_\alpha) \dot{\theta} \end{aligned}$$

la cual de igual manera puede ser obtenida mediante el uso de los símbolos de Christoffel a partir de la matriz $M(\mathbf{q})$. El vector de gravedad se obtiene haciendo la derivada parcial respecto a los estados de la energía potencial, que está dada por:

$$U(\mathbf{q}) = g m_p l_p C_\alpha$$

quedando

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} 0 \\ 0 \\ -g m_p l_p S_\alpha \end{bmatrix}.$$

Finalmente, para simplificar el modelo, considérese que ambas ruedas son idénticas

y por lo tanto se hacen las siguientes afirmaciones

$$m_1 = m_2$$

$$I_{r\bar{x}} = I_{i\bar{x}}$$

$$I_{r\bar{y}} = I_{i\bar{y}}$$

$$I_{r\bar{z}} = I_{i\bar{z}}$$

$$I_{i\bar{x}} = I_{i\bar{z}}$$

$$I_{i\bar{y}} = I_{r\bar{z}}$$

lo que lleva al mismo modelo dinámico obtenido utilizando el método de Euler-Lagrange.

