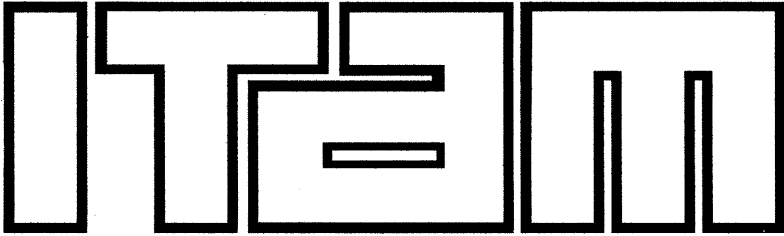


INSTITUTO TECNOLOGICO AUTONOMO DE MEXICO



**DISEÑO E IMPLEMENTACION DE UN ROBOT MOVIL
OMNIDIRECCIONAL PARA APLICACIONES COLABORATIVAS**

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

P R E S E N T A

EDGAR ALEJANDRO GRANADOS OSEGUEDA

ASESOR: DR. MARCO ANTONIO MORALES AGUIRRE

CIUDAD DE MEXICO

2017

“Con fundamento en los artículos 21 y 27 de la Ley Federal de Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada **“Diseño e Implementación de un Robot Móvil Omnidireccional para Aplicaciones Colaborativas”**, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la biblioteca Raúl Baillères Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una prestación”

Edgar Alejandro Granados Osegueda

Fecha

Firma

Abstract

En este trabajo se presenta el diseño e implementación de un robot móvil omnidireccional para aplicaciones colaborativas, tomando como marco de referencia la competencia Robocup Small Size League (SSL) utilizando la metodología “Diseño y Desarrollo de Producto”. El modelo cinemático que describe el movimiento omnidireccional del robot es implementado con dos lazos cerrados de control a bordo del robot en una tarjeta de desarrollo MOJO la cual cuenta con una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4. Se presenta un experimento realizado sin sistema externo, uno utilizando un sistema externo para generación de comandos de velocidad y un tercer experimento para generación de trayectorias dinámicas. Los experimentos realizados ilustran el buen comportamiento del sistema diseñado. Por último, se realiza una comparación con cinco equipos participantes en Robocup SSL 2016.

Índice general

1. INTRODUCCIÓN	1
1.1. Objetivo	2
1.2. Alcance	2
1.3. Organización del documento	3
2. MARCO TEÓRICO	4
2.1. RoboCup Small Size League	4
2.2. Estándares y Códigos Relevantes	7
2.3. Robot Móvil Omnidireccional	8
2.3.1. Modelo Cinemático	9
2.4. Modelado por Deposición Fundida	12
2.5. Hardware del Robot	13
2.5.1. Tarjeta de Desarrollo Mojo	13
2.5.2. Motor Eléctrico	14
2.5.3. Electronic Speed Controller	15
2.5.4. Xbee	15
2.6. Comunicación	16
2.7. Control	17
2.8. Estrategia Metodológica	17

ÍNDICE GENERAL

3. DISEÑO Y DESARROLLO DE UN ROBOT SSL	19
3.1. Planeación	19
3.2. Desarrollo de Concepto	20
3.3. Diseño a Nivel de Sistema	22
3.3.1. Análisis de los Componentes del Robot	22
3.3.2. Interacciones entre Sistemas	25
3.3.3. Arquitectura Propuesta	25
3.4. Diseño a Detalle	28
3.4.1. Chasis	28
3.4.2. Energía	29
3.4.3. Movimiento	30
3.4.4. Aditamentos de Apoyo	32
3.4.5. Drivers	34
3.4.6. Cómputo	35
4. RESULTADOS	43
4.1. Resultados por Sistema	43
4.1.1. Chasis	43
4.1.2. Energía	45
4.1.3. Movimiento	45
4.1.4. Aditamentos de Apoyo	45
4.1.5. Cómputo	47
4.1.6. Drivers	47
4.2. Resultados de la Integración de los Sistemas	48
4.2.1. Pruebas a Nivel de Motor	48
4.2.2. Pruebas de Integración con el Sistema Externo	50
4.2.3. Pruebas con Retroalimentación de Visión y Trayectoria Dinámica	51

ÍNDICE GENERAL

4.3. Comparación con Otros Robots SSL	56
5. CONCLUSIONES Y TRABAJO FUTURO	60
GLOSARIO	62
REFERENCIAS	64

Índice de figuras

2.1. Standard Pattern (Robocup-SSL, 2015)	5
2.2. Arquitectura de un equipo SSL. (Robocup-SSL, 2015)	6
2.3. Ejemplo de Rueda Omnidireccional (Rojas, 2005)	8
2.4. Ejemplo de Rueda Orientable (GmbH, 2016)	9
2.5. Kuka Omnidireccional Robot (Bischoff, 2011)	10
2.6. Distribución de Ángulos y Fuerzas	10
3.1. Prototipo del Robot	21
3.2. Prototipo del Kicker	22
3.3. Esquemático: Elementos del Robot Omnidireccional	23
3.4. Interacciones Incidentales	26
3.5. Interacciones Deseadas	26
3.6. Arquitectura del Robot	27
3.7. Chasis: Modelo CAD	29
3.8. Motor y Transmisión: Modelo CAD	31
3.9. Rueda Omnidireccional Utilizada	33
3.10. Kicker: Modelo CAD	33
3.11. Dribbler	34
3.12. Circuito para el Kicker	36
3.13. Modelo Implementado en la FPGA	37

ÍNDICE DE FIGURAS

3.14. Diagrama del modelo implementado	40
3.15. Ciclos de Control implementados en el AVR	41
3.16. Secuencia de Funciones en el microcontrolador	42
4.1. Robot Real vs Modelo CAD	44
4.2. Ensamble de la Rueda, Motor y Transmisión	46
4.3. Velocidades Deseadas vs Velocidades Reales	49
4.4. Trayectorias sin Retroalimentación de Visión	52
4.5. Error en la Orientación del Robot en las Pruebas sin Retroali- mentación de Visión	53
4.6. Arquitectura del Sistema Utilizado	54
4.7. Movimiento del Robot con Retroalimentación de Visión y Tra- yectorias Dinámicas	55
4.8. Velocidades Deseadas vs Velocidades Reales	57

Índice de tablas

3.1. Secuencia de los Sensores Hall por Motor	39
4.1. Secuencia para las pruebas a Nivel de Motor	50
4.2. Norma cuadrada de los errores e_X [mm], e_Y [mm] y e_θ [grados] .	56

Capítulo 1

INTRODUCCIÓN

Los robots móviles son utilizados en un amplio rango de actividades, como exploración (e.g., DEPTHX en exploración acuática, y MER de la Nasa en la exploración de Marte) y transporte. Existen diversos tipos de robots móviles incluyendo los humanoides (e.g., el Nao), los voladores y los que se mueven con ruedas. Estos últimos tienen una estructura mecánica y modelos relativamente sencillos (Corke, 2011). Por esto, son una muy buena opción para realizar tareas sobre pisos planos. Un caso específico de este tipo de robots es el robot omnidireccional que puede alcanzar cualquier pose en el plano sin realizar maniobras auxiliares. Esto lo hace muy útil en ambientes dinámicos con poco espacio para maniobrar o que requieren cambios abruptos de dirección como fábricas, hospitales y almacenes. Asimismo, en diversas tareas el trabajo colaborativo de múltiples robots tiene beneficios tales como la distribución de objetos en inventarios o la supervisión o vigilancia en entornos grandes.

En este trabajo se presenta el diseño e implementación de un robot con el que se pueden implementar soluciones para aplicaciones colaborativas tanto para investigación como para enseñanza. Dado que el diseño requiere ser versátil para adaptarse fácilmente, se implementa un diseño modular. Una de las aplicaciones específicas del robot es en la RoboCup Small Size League, por lo que se toman las especificaciones físicas de ésta liga como marco de referencia para el diseño del robot.

Se describe a detalle la arquitectura del sistema propuesto. Se presenta la

CAPÍTULO 1: INTRODUCCIÓN

estructura del robot, incluyendo los mecanismos para movimiento y las características del cómputo interno. El robot forma parte de un sistema que incluye cómputo y percepción externo. El sistema externo es usado para validar la implementación pero su diseño y funcionamiento están fuera del alcance de éste trabajo. El cómputo interno utiliza una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4. También, se describen el modelo cinemático del robot y el algoritmo de control utilizado. Para validar el modelo cinemático y el algoritmo de control, se realizaron experimentos de seguimiento de trayectorias que mostraron un desempeño adecuado.

1.1. Objetivo

Dotar al laboratorio de robótica del Instituto Tecnológico Autónomo de México (ITAM) de un robot móvil omnidireccional que pueda ser utilizado en diversos escenarios de investigación y enseñanza. En específico, la solución deberá cumplir con las especificaciones de RoboCup Small Size League (SSL) para poder participar en la competencia, además de poder ser utilizado en otros desarrollos independientes a Robocup.

1.2. Alcance

Realizar el diseño e implementación de un robot móvil omnidireccional para distintos escenarios de investigación. La implementación es independiente del sistema utilizado para controlar el robot, además de ser robusta. Se cumplen las especificaciones de Robocup SSL además de incluir los aditamentos usuales de un robot para Robocup SSL. El robot recibe un perfil de velocidad y es capaz de seguirlo con teniendo poco error en la trayectoria. Adicionalmente, es capaz de seguir una trayectoria dinámica respondiendo a cambios en el perfil de velocidad a la frecuencia de un sistema externo.

1.3. Organización del documento

El documento consta de cinco capítulos. En el capítulo 2: Marco Teórico se describen los antecedentes de éste trabajo, desarrollos similares y las tecnologías utilizadas. En el capítulo 3 se utiliza la metodología *Diseño y Desarrollo de Productos* (Ulrich y Eppinger, 2009) estableciendo la *Planeación*, el *Desarrollo de Concepto*, *Diseño a Nivel Sistema* y *Diseño a Nivel Detalle*. En el capítulo 4 se presentan los resultados obtenidos y en el capítulo 5 se presentan las conclusiones del proyecto y trabajo futuro propuesto.

Capítulo 2

MARCO TEÓRICO

2.1. RoboCup Small Size League

RoboCup es una iniciativa científica internacional con el objetivo de avanzar en el estado del arte de robots inteligentes. En 1997 se realizó la primera RoboCup, teniendo como objetivo incentivar la investigación de la robótica y la inteligencia artificial al establecer una competencia con visión de largo plazo. La meta de RoboCup es “Para 2050, un equipo de robots humanoides autónomos ganará un partido de fútbol soccer con reglas FIFA contra el entonces campeón de la Copa del Mundo” (Ferrein y Steinbauer, 2016).

Una categoría de RoboCup, la SSL se enfoca en la cooperación inteligente de agentes además del control en un ambiente altamente dinámico con un sistema híbrido centralizado-distribuido. Un juego de SSL de RoboCup, consiste en dos equipos de seis robots cada uno que tienen acceso a un sistema de visión global así como un árbitro. Cada robot debe cumplir con las características especificadas en las reglas F180, entre las cuales se encuentran:

- No puede tener objetos que sean peligrosos para otros robots o para humanos.
- Debe caber dentro de un cilindro de 180 mm de diámetro y 150 mm de alto.

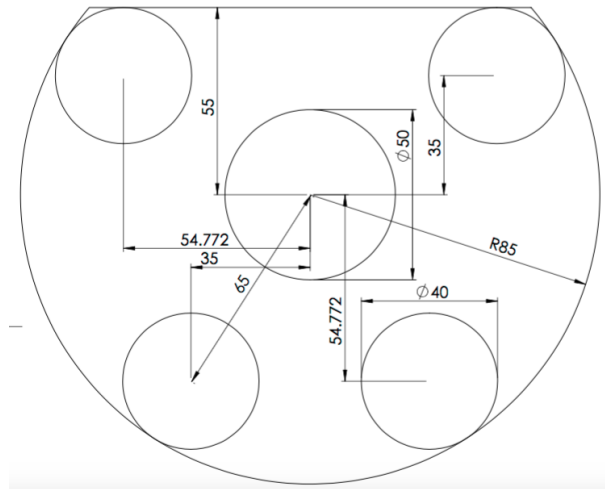


Figura 2.1: Standard Pattern (Robocup-SSL, 2015)

- El área superior debe cumplir con el Standard Pattern (Fig. 2.1).
- El área externa debe ser oscura y con acabado mate.
- No debe dañar la cancha al moverse.
- Puede utilizar comunicación inalámbrica con computadoras fuera de la cancha.
- Debe ser completamente autónomo durante el partido.

Como se muestra en la Fig. 2.2, un equipo de RoboCup SSL está conformado por al menos seis robots móviles y una computadora. La computadora debe poder conectarse y procesar la información proporcionada por los sistemas globales de SSL. Igualmente, determinar las estrategias necesarias y comunicarlas a los robots.

Los jugadores en SSL son robots omnidireccionales como jugadores de fútbol. La pose de los robots se obtiene con un sistema de visión integrado por cámaras en la parte superior de la cancha además del software provisto por la liga para el procesamiento de imágenes que detecta el Standard Pattern de cada robot.

Varios equipos se han destacado a lo largo de la historia de la competencia. *Cornell Big Red* fue uno de los primeros ganadores de la liga, explorando di-

CAPÍTULO 2: MARCO TEÓRICO

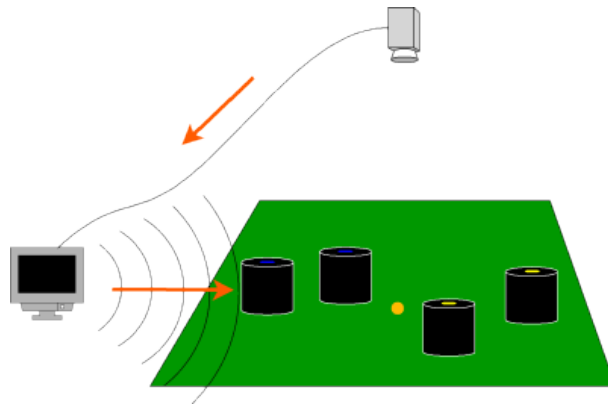


Figura 2.2: Arquitectura de un equipo SSL. (Robocup-SSL, 2015)

versos tipos de ruedas omnidireccionales como la ortogonal (D’Andrea, Kalmar-Nagy, Ganguly, y Babish, 2000) y posteriormente ruedas universales (Purwin y D’Andrea, 2003) con un control basado en torque para evitar sobre calentamiento en las escobillas de los motores y un segundo lazo para la orientación del robot utilizando un giroscopio. *FU-Fighters* fue uno de los primeros en utilizar ruedas con rodillos perpendiculares (Egorova y cols., 2003). *Skuba* también ha utilizado control PI basado en torque, con motores sin escobillas (Chaiso y Sukvichai, s.f.).

Debido al tamaño de los robots, el consumo de energía es una de las características más importantes de los robots. Por esto, se han explorado diversas opciones de cómputo a bordo del robot. Por ejemplo, los equipos del ITAM utilizaban Digital Signal Processor (DSP) (Torres, 2009) mientras *TIGERS Mannheim*, utiliza un microcontrolador (Ryll, Ommer, Geiger, Jauer, y Theis, 2014). Otra opción popular es una FPGA como la usada por *MRL* (Poudeh, Esmael-pourfard, Hosseinikia, Tafti, y Adhami-Mirhosseini, 2016). *ZJUNlict*, en cambio, usa una FPGA donde implementan un microcontrolador (Zhao, Xiong, Tong, Li, y Fang, 2013). Por último, equipos como *Skuba* utilizan tanto una Field-Programmable Gate Array (FPGA) como un microcontrolador.

El equipo Eagle Knights del ITAM ha participado en la RoboCup SSL desde 2006 en cinco ocasiones. Como consecuencia de la obsolescencia de la implementación anterior además de diversos problemas con la misma, se determinó la necesidad de realizar una nueva implementación para poder competir de manera

CAPÍTULO 2: MARCO TEÓRICO

satisfactoria en futuras RoboCups. Es necesario la incorporación de tecnologías que antes no eran accesibles ya fuera por costo o por ser de gran tamaño respecto al robot.

Algunos de los trabajos realizados en el ITAM relacionados con RoboCup SSL incluyen (Torres, 2009) donde se muestra una implementación de un sistema de inteligencia para controlar robots de SSL. En (Ruiz, 2007) se muestra el diseño del sistema para controlar la pelota de versiones anteriores del robot. Un control de trayectorias para los robots SSL se detalla en (Rodríguez, 2009).

2.2. Estándares y Códigos Relevantes

Se toman las reglas de la SSL (Committee, 2016) como el estándar que el robot debe cumplir. Durante el desarrollo se seleccionaron estándares y códigos específicos de acuerdo a la funcionalidad deseada.

Los materiales utilizados para manufacturar las piezas propias son Acrilonitrilo Butadieno Estireno (ABS) y High Impact Polystyrene (HIPS). Específicamente, se utilizan los productos de la marca *Zortrax*: Z-ABS (“Z-ABS Material Data Sheet”, 2014) y Z-HIPS (“Z-HIPS Material Data Sheet”, 2014). Para ensamblar éstas piezas se utilizan tornillos M3x5 o M3x12. Adicionalmente, se utilizan tornillos 6-32 UNC para acoplar dos piezas comerciales debido a las características de éstas. Un tornillo $\frac{1}{4}'' - 20 \times 2''$ como eje de cada rueda. Por último, se utilizan engranes de *pitch* 32 con 20 grados de ángulo de presión.

Para la alimentación eléctrica, se utilizan baterías *LiPo* 2S y 3S (Salt, 2012). Se utilizan componentes TTL para los circuitos propios. La tarjeta de desarrollo utilizada opera con LVTTL.

Para la comunicación entre componentes se utiliza Serial Peripheral Interface (SPI) 0 (Davis, Huynh, y Peck Jr, 2013), Universal Asynchronous Receiver/Transmitter (UART) o Pulse Width Modulation (PWM) (Valvano, 2012). Para la comunicación inalámbrica se utiliza XBee (International, 2015), específicamente la implementación del protocolo IEEE 802.11n (Tanenbaum, 2003); ésta comunicación se realiza mediante User Datagram Protocol (UDP) (Tanenbaum, 2003).



Figura 2.3: Ejemplo de Rueda Omnidireccional (Rojas, 2005)

2.3. Robot Móvil Omnidireccional

Un robot móvil omnidireccional se puede mover en un plano entre cualesquiera dos poses arbitrarias A y B trasladándose sobre una línea recta mientras gira sobre su propio eje (Rojas, 2005). Las poses A y B se caracterizan por una posición (x, y) y una orientación θ con respecto a un sistema de referencia. Se puede lograr movimiento omnidireccional con al menos 3 ruedas omnidireccionales concéntricas colocadas con distintos ángulos de separación (Rojas y Förster, 2006). Ruedas adicionales proporcionan redundancia que permite cierta tolerancia a fallas por lo que el robot de 4 ruedas es muy popular.

La principal ventaja de los robots omnidireccionales es su maniobrabilidad. El OmniBot de la NASA fue desarrollado como un robot altamente maniobrable para ambientes peligrosos (Houshangi y Lippitt, 1999). El Airtrax ATX-3000 es un levantacargas para trabajo en almacenes (Adăscăliței y Doroftei, 2011). El *OMNI-chair* es una silla de ruedas omnidireccional diseñada para darle mayor capacidad de movimiento a su usuario (Borgolte, Hoyer, Bühler, Heck, y Hoelper, 1998).

Existen diversos diseños de rueda omnidireccional, dos tipos son ortogonales y universales (Ashmore y Barnes, 2002). Las ruedas ortogonales consisten en dos esferas cortadas por dos planos paralelos; el eje de la rueda es perpendicular a las superficies recortadas. Las ruedas universales consisten en una rueda con rodillos en un ángulo en la periferia, un diseño muy popular es la rueda Mecanum, Fig. 2.5. Sus principales limitaciones son el diseño complejo (conceptual y físico) así como el gran espacio que ocupan los rodillos. Otro diseño popular de rueda



Figura 2.4: Ejemplo de Rueda Orientable (GmbH, 2016)

universal por su diseño compacto cuenta con rodillos alineados al eje principal de la rueda. También existen las ruedas orientables (Fig. 2.4) que necesitan de un motor extra por rueda para orientar la rueda al ángulo deseado. La rueda HOG (Cardán semiesférica omnidireccional por sus siglas en inglés) es una superficie giratoria semiesférica la cual se encuentra montada en una suspensión Cardán. Esta rueda se mueve por motores, de manera inversa al mecanismo de un *mouse* mecánico.

Lograr un movimiento correcto en un robot omnidireccional presenta diversos retos. Al utilizar por lo menos tres motores se necesita implementar un algoritmo de control eficiente. Este tipo de robots suelen requerir de una suspensión para que la rueda esté en contacto con el suelo permanentemente. Solamente si se garantiza que la superficie donde se utilizará será plana, se puede evitar el uso de la suspensión. Dependiendo del posicionamiento de las ruedas, la velocidad máxima posible no es igual en todas direcciones.

2.3.1. Modelo Cinemático

La velocidad deseada del robot es expresada mediante el vector $V_d = \begin{pmatrix} v_x & v_y & \omega \end{pmatrix}^T$. Es necesario transformar éste vector a uno de velocidades deseadas de motores $V_m^d = \begin{pmatrix} m_1 & m_2 & m_3 & m_4 \end{pmatrix}^T$.

A partir de la ecuación general de fuerza $F = Ma$ se puede derivar una

CAPÍTULO 2: MARCO TEÓRICO



Figura 2.5: Kuka Omnidirectional Robot (Bischoff, 2011)

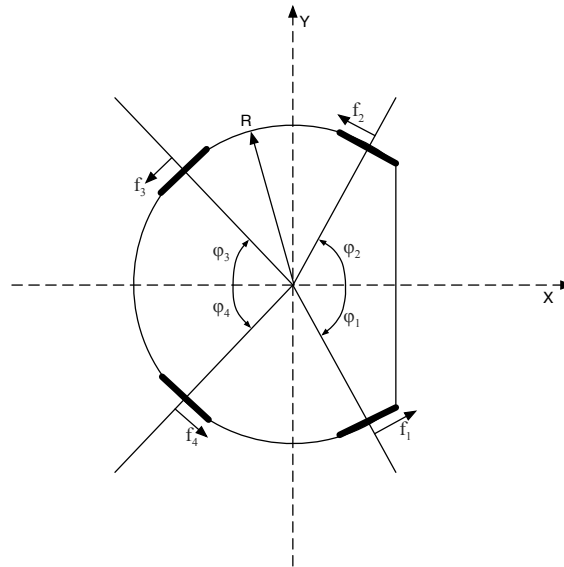


Figura 2.6: Distribución de Ángulos y Fuerzas

CAPÍTULO 2: MARCO TEÓRICO

ecuación para obtener la aceleración $a = \frac{1}{M} (F_1 + F_2 + F_3 + F_4)$ específica para el caso de cuatro motores. Al utilizar ruedas omnidireccionales concéntricas, cada motor aporta tanto en los componentes de X y Y como en la rotación del robot. Utilizando el diagrama de la Fig. 2.6, es posible derivar las ecuaciones (2.1).

$$\begin{aligned} Ma_{1x} &= |f_1| \sin(\varphi_1); & Ma_{1y} &= |f_1| \cos(\varphi_1) \\ Ma_{2x} &= -|f_2| \sin(\varphi_2); & Ma_{2y} &= |f_2| \cos(\varphi_2) \\ Ma_{3x} &= -|f_3| \sin(\varphi_3); & Ma_{3y} &= |f_3| \cos(\varphi_3) \\ Ma_{4x} &= |f_4| \sin(\varphi_4); & Ma_{4y} &= |f_4| \cos(\varphi_4) \end{aligned} \quad (2.1)$$

Dado R el radio de rueda utilizada, a partir de la ecuación general se puede derivar la ecuación (2.2).

$$\dot{w} = \frac{R}{I} (f_1 + f_2 + f_3 + f_4) \quad (2.2)$$

Sustituyendo el momento de inercia $I = \alpha MR^2$ para un cilindro de masa desconocida en (2.2) se obtiene (2.3).

$$R\dot{w} = \frac{1}{M\alpha} (f_1 + f_2 + f_3 + f_4) \quad (2.3)$$

Estas ecuaciones se pueden expresar matricialmente como se muestra en la ecuación (2.4) cuya forma reducida se muestra en la ecuación (2.5), donde C_α se conoce como la Matriz de Acoplamiento de Fuerzas.

$$\begin{pmatrix} a_x \\ a_y \\ R\dot{\omega} \end{pmatrix} = \frac{1}{M} \begin{bmatrix} \sin \varphi_1 & -\sin \varphi_2 & -\sin \varphi_3 & \sin \varphi_4 \\ \cos \varphi_1 & \cos \varphi_2 & -\cos \varphi_3 & -\cos \varphi_4 \\ \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} \quad (2.4)$$

$$a = C_\alpha F \quad (2.5)$$

Los componentes que aporta cada motor se obtiene de los ángulos detallados en la Fig. 2.6. Para considerar tanto la reducción por un tren de engranes e , como el perímetro de la rueda r se utiliza la ecuación 2.3.1 para cada componente.

$$v_x' = \frac{v_x \cdot e}{2\pi r}$$

La ecuación matricial (2.6) se puede derivar de la ecuación (2.5) para obtener el vector V_m^d . La forma reducida se muestra en la ecuación (2.7) donde a D se le conoce como la matriz de acoplamiento de velocidades.

$$v_m^d = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{bmatrix} \sin \varphi_1 & \cos \varphi_1 & 1 \\ -\sin \varphi_2 & \cos \varphi_2 & 1 \\ -\sin \varphi_3 & -\cos \varphi_3 & 1 \\ \sin \varphi_4 & -\cos \varphi_4 & 1 \end{bmatrix} \begin{pmatrix} v_x' \\ v_y' \\ R\omega' \end{pmatrix} \quad (2.6)$$

$$v_d = DV_d' \quad (2.7)$$

2.4. Modelado por Deposición Fundida

El método de Modelado por Deposición Fundida permite la fabricación de sólidos de forma libre (Solid Freeform Fabrication). Fue creado por Stratasys Inc. (Wright, 2001). El material, en forma de filamento, se funde y mediante un

CAPÍTULO 2: MARCO TEÓRICO

extrusor se deposita en una plancha, creando capas de acuerdo al modelo creado en un programa de Computer Aided Design (CAD). El proceso de extrusión presenta algunos problemas al tratarse de piezas grandes ya que la diferencia de temperaturas en diferentes zonas de la pieza puede acarrear deformaciones durante el proceso; esto depende de las características del material como la dilatación por la temperatura. El formato usado para crear las piezas es *STL* (*STereoLitografía*), el cual se basa en representar el objeto a partir de triángulos (la resolución suele ser un parámetro modificable). El objeto es *rebanado* para crear capas y se determina la trayectoria necesaria para realizar el objeto. Aunque la forma de operación de una máquina de Modelado por Deposición Fundida se parece a un *router CNC* al basarse en la adición de material, permite crear piezas de formas complejas. Algunas piezas pueden requerir de algún soporte que no forme parte del producto final pero es fácilmente removible, requiriendo un mínimo trabajo de terminado.

Las piezas realizadas mediante Modelado por Deposición Fundida son anisótropas (dependen de la dirección), además de estar discretizadas. En (Ahn, Montero, Odell, Roundy, y Wright, 2002) se examinan algunas características de las piezas así construidas a través de pruebas de concentración de esfuerzos, de tensión y de compresión; se determina que las características que más efecto tienen en la pieza final son: dirección de extrusión y espesor de las capas.

2.5. Hardware del Robot

La función del hardware consiste en la captura de señales, su procesamiento y envío a los actuadores. En éste trabajo se utilizan componentes digitales por la facilidad de reprogramación que proporcionan.

2.5.1. Tarjeta de Desarrollo Mojo

La tarjeta de desarrollo Mojo integra una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4 comunicados mediante SPI.

Una FPGA es un dispositivo lógico programable de propósito general de programación multinivel (Trimberger, 1994). La configuración del FPGA se rea-

CAPÍTULO 2: MARCO TEÓRICO

liza mediante un lenguaje de descripción de Hardware (HDL por sus siglas en inglés). Al ser circuitos lógicos, las operaciones implementadas en un FPGA no son secuenciales como en un lenguaje de programación, pudiendo tener tantas operaciones lógicas (con entradas y salidas) paralelas como lo permita cada FPGA.

Un microcontrolador es una computadora en un solo chip, con interfaces que facilitan su uso en el control de dispositivos (Mano y Kime, 2007). ATME- AVR es un microcontrolador programable que utiliza una arquitectura RISC. Es posible cambiar la programación a través de un lenguaje de programación de alto nivel (usualmente se utiliza *C*) que facilita la implementación de operaciones aritméticas.

2.5.2. Motor Eléctrico

Un motor eléctrico es un tipo de actuador que convierte energía eléctrica a energía mecánica (Maxon, 2014). En el contexto de éste trabajo se utilizan dos tipos de motores eléctricos: *Con Escobillas* y *Sin Escobillas*. El funcionamiento de los motores se basa en convertir energía eléctrica a magnética a mecánica. Se utilizan dos fuentes de magnetismo: imanes permanentes y bobinas. La generación de torque se basa en controlar la corriente de cada devanado de la fase del motor (Hanselman, 2003).

I. Motor Con Escobillas

Los motores con escobillas utilizan corriente directa para su operación. Las escobillas sirven para alimentar la corriente al devanado. De forma mecánica, las escobillas convierten la corriente de las bobinas a torque (Cox, 2006). Para controlar éste tipo de motores, se suele utilizar un *Puente H* (Instruments, 2005) activado con una señal PWM que se puede generar con un microcontrolador o FPGA. Uno de los principales problemas asociados a los motores con escobillas es el desgaste que éstas presentan por el uso.

II. Motor Sin Escobillas

La conmutación de los motores sin escobillas es electrónica usualmente trifásica, basada en saber la posición del rotor. Para controlar un motor sin escobillas es necesario un circuito que realice los cambios en la fase de acuerdo a la velocidad deseada, éste tipo de circuitos se conoce como Electronic Speed Controller. Los motores sin escobillas pueden tener sensores tipo *Hall* para determinar la velocidad real del motor a través de detectar el campo magnético de las bobinas.

III. Pulse Width Modulation

PWM se utiliza para codificar datos en una señal binaria al variar el *duty cycle* con frecuencia fija. La potencia requerida del motor corresponde al *duty cycle* generado. PWM es muy utilizado para el control de motores debido a que la respuesta del motor ante variaciones de voltaje no es lineal. Generalmente, PWM es usado si requiere que un microcontrolador determine la potencia deseada del motor, utilizando una salida digital variando el *duty cycle* conforme se requiere.

2.5.3. Electronic Speed Controller

Un Electronic Speed Controller sirve como interfaz entre un motor sin escobillas y el dispositivo que determina la velocidad deseada del motor. Por lo general, el Electronic Speed Controller recibe la velocidad deseada mediante una señal como PWM y genera las señales correspondientes para cada fase del motor.

2.5.4. Xbee

Xbee es un producto de la empresa Digi que ofrece una solución de reducido tamaño que permite convertir comunicaciones inalámbricas a seriales fácilmente, originalmente enfocado a sensores y otras aplicaciones en locaciones remotas (“Xbee Wi-Fi”, 2015). Actualmente, Xbee ofrece soporte a diversos protocolos de comunicación inalámbrica como: 802.15.4, 802.11 y LTE. La comunicación entre Xbee y el dispositivo anfitrión se puede realizar mediante UART o SPI.

2.6. Comunicación

Dentro del robot es necesario contar con distintos tipos de comunicación. Un tipo de comunicación inalámbrica es necesaria para recibir los comandos de velocidad. Igualmente, es necesario comunicar los distintos componentes que implementan el cómputo a bordo así como sus interfaces con los actuadores.

Wi-Fi es utilizado para la comunicación inalámbrica. Es un estándar para Wireless Local Area Network (WLAN) formalmente llamado IEEE 802.11. Opera en la banda ISM definida por la ITU-R. Una red 802.11 está generalmente compuesta por dispositivos terminales y un Punto de Acceso el cual se encarga de direccionar las comunicaciones entre los dispositivos terminales y otras redes. Al utilizar un medio compartido, las comunicaciones Wi-Fi son propensas a pérdida de información por colisiones (Tanenbaum, 2003).

Para la comunicación entre componentes, se utiliza tanto SPI, UART y PWM. SPI es un protocolo de comunicación *Full-Duplex* no simétrico. Un *bus* SPI debe tener solamente un dispositivo maestro y uno o más esclavos. El maestro es capaz de comunicarse con cualquier esclavo pero los esclavos solamente se pueden comunicar con el maestro cuando éste inicia la comunicación. Consiste de cuatro señales: Master In Slave Out (MISO), Master Out Slave In (MOSI), Serial Clock (SCK) y Slave Select (SS). UART es un protocolo de comunicación serial que consiste en transmitir un bit cada tiempo constante t . UART transmite en secuencias de Bytes pudiendo ser *Full-Duplex* (Valvano, 2012).

UDP es uno de los dos protocolos principales utilizados en Internet en la capa de transporte. UDP provee la capacidad de enviar información sin la necesidad de establecer una conexión. Ésto implica que no existe garantía de entrega de los paquetes ni del orden de entrega. En UDP no existen retransmisiones de mensaje de *acknowledge* por lo que el dispositivo que envía no puede saber si el paquete fue recibido correctamente (Tanenbaum, 2003). Aunque ésto presenta limitantes para su uso en muchas aplicaciones, UDP ha sido ampliamente usado en aplicaciones de tiempo real ya que en éstas aplicaciones solamente se requiere el último paquete disponible.

2.7. Control

Un sistema de control consiste en subsistemas y procesos (o plantas) ensambladas para obtener una salida deseada con el rendimiento deseado ante una entrada específica. De acuerdo a (Nise, 2014), un sistema de control tiene 4 objetivos principales:

- Amplificar potencia
- Control remoto
- Adaptar la señal de entrada
- Compensar por perturbaciones

Un *Lazo Cerrado* es una configuración de control en la que sensores en el sistema miden la salida y determinan el *error*. Si el error es distinto de cero, el sistema actúa la planta, haciendo la corrección necesaria para llevar el error a cero. Un tipo de control a lazo cerrado es el control *PI* (ecuación (2.8)) que alimenta a la planta una proporción P de la señal más su integral I . La parte proporcional determina la salida a partir de la magnitud del error, actuando solamente con la información instantánea. En contraste, la parte integral obtiene información del comportamiento del sistema a lo largo del proceso, con lo que determina la salida. Las variables k_p y k_i son conocidas como las ganancias proporcional e integral respectivamente.

$$v_c = k_p e(t) + k_i \int_0^t e(\tau) d\tau \quad (2.8)$$

2.8. Estrategia Metodológica

Como estrategia metodológica para guiar el desarrollo del proyecto se utiliza el método *Diseño y Desarrollo de Producto* (Ulrich y Eppinger, 2009). Éste es un proceso bien definido que proporciona aseguramiento de calidad al especificar fases y puntos de inspección. Especifica métodos de administración del proyecto

CAPÍTULO 2: MARCO TEÓRICO

para evaluar la operación contra los objetivos establecidos identificando áreas problemáticas. Al documentar el proceso, se facilita la identificación de mejoras para futuras iteraciones del producto. El proceso consta de seis fases: Planeación, Desarrollo del Concepto, Diseño a Nivel Sistema, Diseño a Nivel Detalle, Pruebas e Inicio de Producción.

Se utiliza por ser una metodología general para desarrollo de todo tipo de productos. Por su carácter general, la metodología contempla desarrollo en áreas que no son de interés para éste trabajo como mercadotecnia. Igualmente, especifica tareas para el área de manufactura al considerar que se realizará una producción grande lo cual escapa a los objetivos del trabajo y la solución presentada.

En el capítulo 3 se presentan las primeras cuatro fases de la metodología. Para mantener breve el documento, no se presentan todos los análisis ni iteraciones realizadas. Solamente se presentan los resultados más relevantes de cada fase. En el capítulo 4 se presenta la quinta fase de la metodología. La última fase no se realizó por no requerir de una producción en serie del producto.

Capítulo 3

DISEÑO Y DESARROLLO DE UN ROBOT SSL

En éste capitulo se presentan los resultados del diseño de un robot SSL de acuerdo a la metodología de *Diseño y Desarrollo de Producto* (Ulrich y Eppinger, 2009). En el primer paso, *Planeación*, se establecen las restricciones del producto y los objetivos que debe cumplir. En *Desarrollo de Concepto* se prueba la viabilidad del uso de ciertas tecnologías mediante prototipos, para validar el concepto. En el tercer paso, *Diseño a Nivel Sistema* se establece la arquitectura del producto definiendo los sistemas y componentes que lo conforman además de identificar interacciones y las interfaces entre los sistemas. El cuarto paso *Diseño a Detalle* presenta el diseño final de los componentes de cada sistema y sus características. El quinto paso, *Pruebas*, se presenta como parte del capítulo 4.

3.1. Planeación

Se hace la planeación considerando las características de versiones previas y las especificaciones de la SSL. Las características de versiones anteriores del Robot para Robocup SSL que se toman en cuenta en este diseño son:

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

- Alta dependencia de manufactura externa, por no contar con el equipo necesario para realizar manufactura interna.
- Piezas frágiles, requiriendo ser reemplazadas frecuentemente durante pruebas y juegos. La rueda era uno de los componentes con mayores problemas.
- Tiempo de ensamblado elevado debido al número de piezas, impactando en los costos.
- Cómputo basado en un DSP volviendo la programación compleja, difícil de parametrizar y modificar.
- No se tenía un diseño modular, haciendo difícil realizar modificaciones.
- La carcasa no tenía el Standard Pattern integrado en su diseño, presentando problemas con la visión de la liga y siendo complejo modificar el *id* de cada robot.
- Problemas en el sistema de energía, afectando particularmente a los motores. Estos requerían ser reemplazados frecuentemente tras descomponerse. Además de elevar costos, el tiempo de reemplazo era alto.

En la sección 2.1 se detallan las reglas y especificaciones que un robot debe cumplir para poder participar en Robocup SSL. La solución debe cumplir con dichas especificaciones además de atacar los problemas detectados en versiones anteriores. Específicamente, el robot debe ser fácilmente modificable y de bajo costo, priorizando manufactura local.

Para poder mantener bajos costos y asumiendo que se tendrá una producción de pocas unidades, el método de Modelado por Deposición Fundida resulta muy atractivo al contar en el laboratorio con el equipo adecuado. A pesar de esto, no se pueden descartar otros métodos de manufactura ni utilizar piezas comerciales al proporcionar mayor calidad.

3.2. Desarrollo de Concepto

Para validar la viabilidad de utilizar Modelado por Deposición Fundida como principal técnica de manufactura, se crearon dos prototipos utilizando piezas de versiones anteriores.

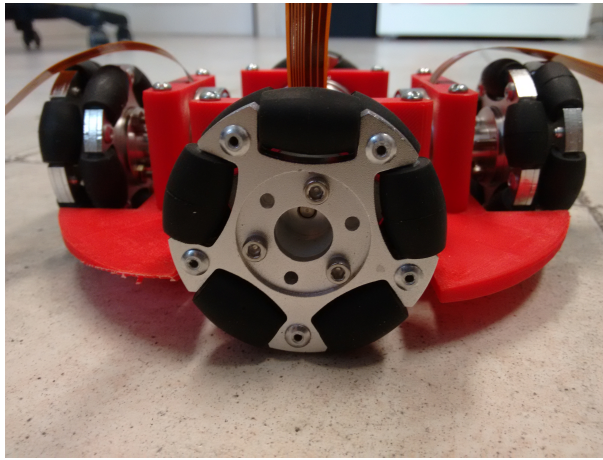


Figura 3.1: Prototipo del Robot

El primer prototipo se muestra en la Fig. 3.1). Tanto la base como la carcasa de cada motor se manufacturaron mediante Modelado por Deposición Fundida. Mediante éste prototipo se validó la posibilidad de imprimir piezas de dimensiones similares a la base (150 mm de radio por 5mm de ancho). Aunque la base salió ligeramente dañada del proceso de Modelado por Deposición Fundida por su tamaño, no afectó gravemente su estructura. Adicionalmente, con éste prototipo se pudo comenzar a probar el funcionamiento de los motores y el movimiento omnidireccional. Las ruedas utilizadas en el prototipo ocupan mucho espacio por lo que es necesario utilizar otras ruedas que ocupen menos espacio así como implementar un reductor de velocidad ya que se requiere menor velocidad de los motores y mayor torque. Por último, el motor debe ir elevado ya que de quedar cerca del suelo puede dañarse por la pelusa de la cancha.

El segundo prototipo se muestra en la Fig. 3.2. El objetivo de éste prototipo fue validar si las piezas manufacturadas mediante Modelado por Deposición Fundida se podían utilizar para el Kicker sin que se dañasen rápidamente. Adicionalmente, se probó una primera versión del circuito de control. Las piezas funcionaron adecuadamente aunque la sujeción de las piezas presentó problemas debido a la fuerza ejercida por el selenoide.

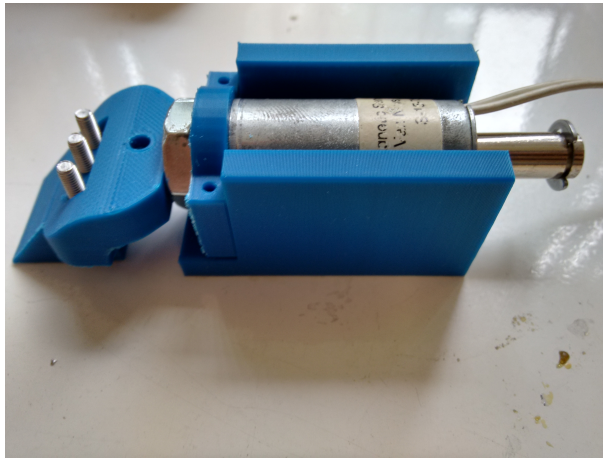


Figura 3.2: Prototipo del Kicker

3.3. Diseño a Nivel de Sistema

Para realizar el Diseño a Nivel de Sistema, se identifican componentes funcionales que requiere la solución y se identifican las relaciones entre ellos para definir sistemas. Se presentan las interacciones incidentales entre los sistemas así como las interfaces entre sistemas. La arquitectura propuesta es modular, compuesta de seis sistemas.

3.3.1. Análisis de los Componentes del Robot

Para obtener una arquitectura modular, se identificaron 20 componentes necesarios de acuerdo a su función y se agruparon en 6 sistemas como se muestra en la Fig. 3.3, de los cuales dos son considerados críticos: Cómputo y Mecánica.

I. Chasis

Componentes cuya función principal es proporcionar soporte estructural, protegiendo al resto de los sistemas ante colisiones. Está integrado por tres componentes: Base, Carcasa y Tapa. La función de la base es unir los sistemas mientras que la carcasa protege a los sistemas ante colisiones. La tapa implementa el Standard Pattern para ser compatible con los requerimientos de Robocup

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

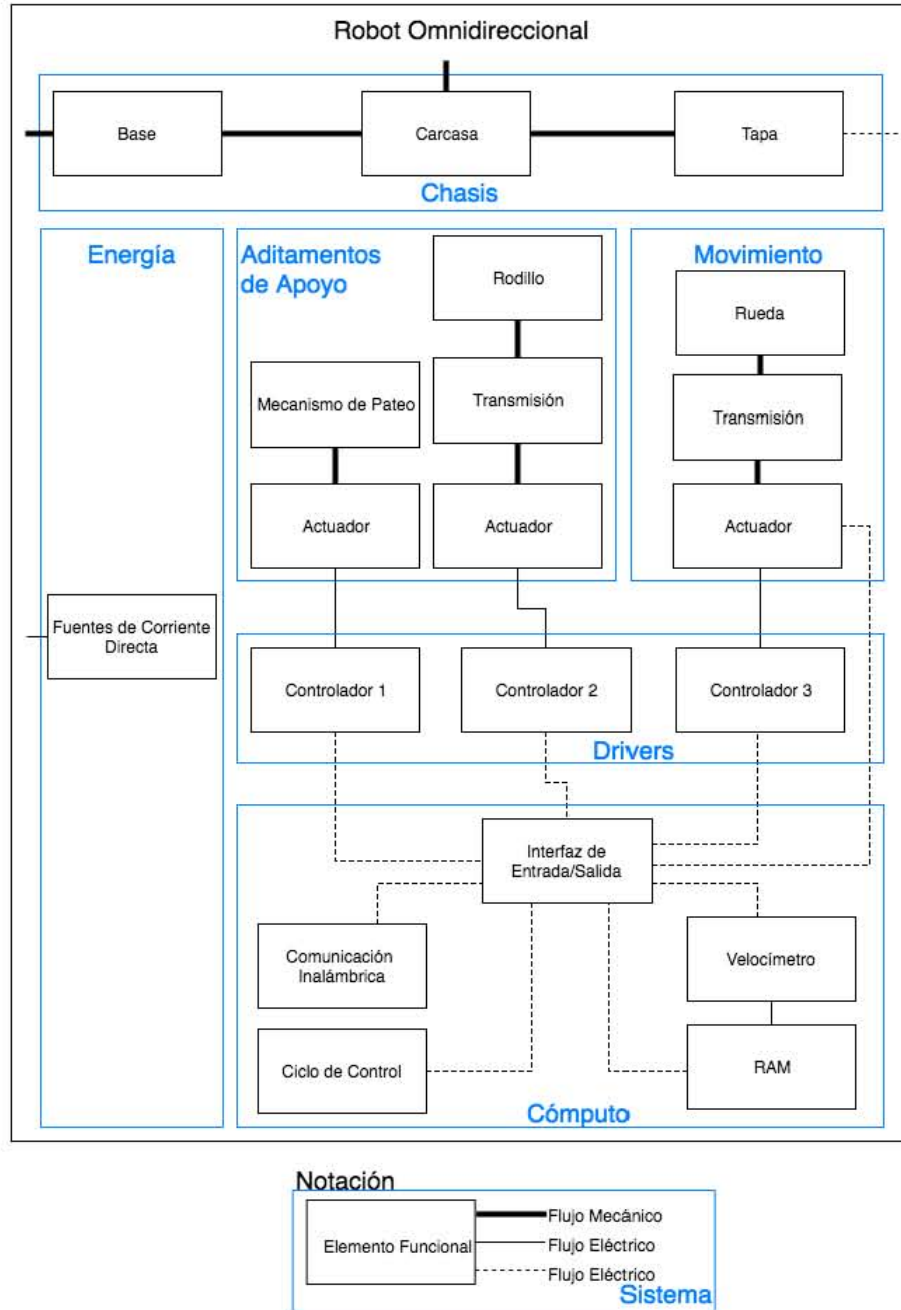


Figura 3.3: Esquemático: Elementos del Robot Omnidireccional

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

SSL.

II. Movimiento

Sistema integrado por los componentes que le proporcionan movimiento en el mundo al robot. Está integrado por tres componentes: Actuador, Transmisión y Rueda. El actuador genera movimiento que se transmite mediante la transmisión a la rueda. Para obtener movimiento omnidireccional son necesarios por lo menos tres instancias de cada componente, por redundancia se utilizan cuatro. Por la funcionalidad que éste sistema proporciona es considerado crítico.

III. Aditamentos de Apoyo

Estos componentes son específicos para participar en juegos de Robocup SSL. Un actuador, transmisión y rodillo son para el Dribbler que retiene la pelota mientras que los otros dos sirven para patear la pelota. Para utilizar el robot en un contexto distinto a SSL, éste es el sistema que se puede reemplazar.

IV. Energía

Fuentes de corriente directa que alimentan a otros sistemas. Tanto el sistema de Movimiento como el de Aditamentos de Apoyo requieren 12V. El sistema de Cómputo requiere 5V.

V. Drivers

Componentes que sirven como interfaces entre sistemas. Se tienen 3 drivers, uno para cada tipo de actuador.

VI. Cómputo

Los componentes del sistema se encargan de obtener, almacenar y procesar datos para generar y enviar señales de control a otros sistemas. Está compuesto de cinco componentes: Interfaz de Entradas/Salidas, Comunicación Inalámbrica, Función de Control, Velocímetro y Registros. El primero, sirve para comunicar

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

componentes a través de distintos tipos de señales y protocolos de comunicación. La comunicación inalámbrica recibe datos de un sistema externo. La función de control calcula a partir de la información disponible la siguiente señal de control que aplicar a los actuadores del sistema mecánico. El velocímetro determina la velocidad real del robot mientras que la Random-Access Memory (RAM) recibe datos de diversos componentes, los almacena y envía conforme son solicitados.

Se considera un sistema crítico debido a que el desempeño del robot está directamente relacionado a la frecuencia con la que éste sistema opere.

3.3.2. Interacciones entre Sistemas

En la Fig. 3.4 se muestran las tres interacciones incidentales identificadas entre los distintos sistemas. La primera se da entre el chasis y los drivers, el calor generado por los drivers no se disipa rápidamente. La segunda es entre el chasis y las vibraciones generadas por la mecánica. La tercera se da al activar los componentes para el juego con la mecánica, pudiendo introducir errores en el movimiento del robot.

Las interfaces entre los sistemas se muestran en la Fig. 3.5. El sistema de baterías alimenta directamente al cómputo y a los drivers. El cómputo recibe señales del sistema mecánico y saca señales de control a los drivers que activan tanto el sistema mecánico como los aditamentos para el juego.

3.3.3. Arquitectura Propuesta

En la Fig. 3.6 se muestra la arquitectura propuesta con los sistemas mencionados previamente así como los componentes de cada sistema y las interacciones existentes entre los componentes. A partir de los componentes de cada sistema, se puede comenzar a realizar el diseño a detalle de cada componente. Cada uno puede consistir de una o más piezas.

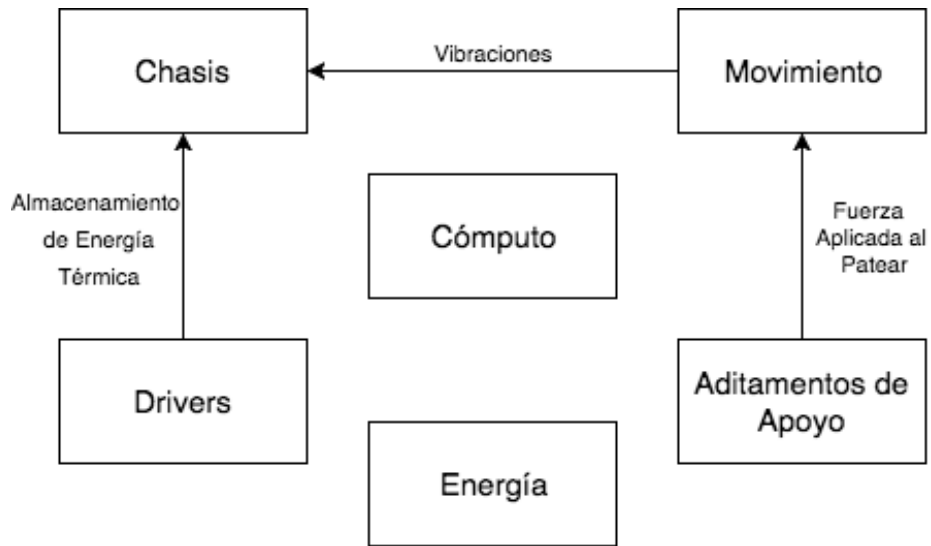


Figura 3.4: Interacciones Incidentales

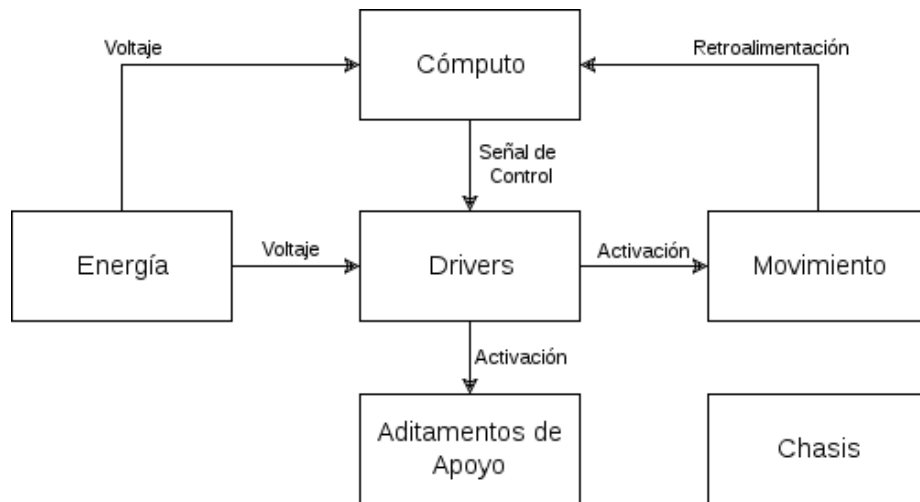


Figura 3.5: Interacciones Deseadas

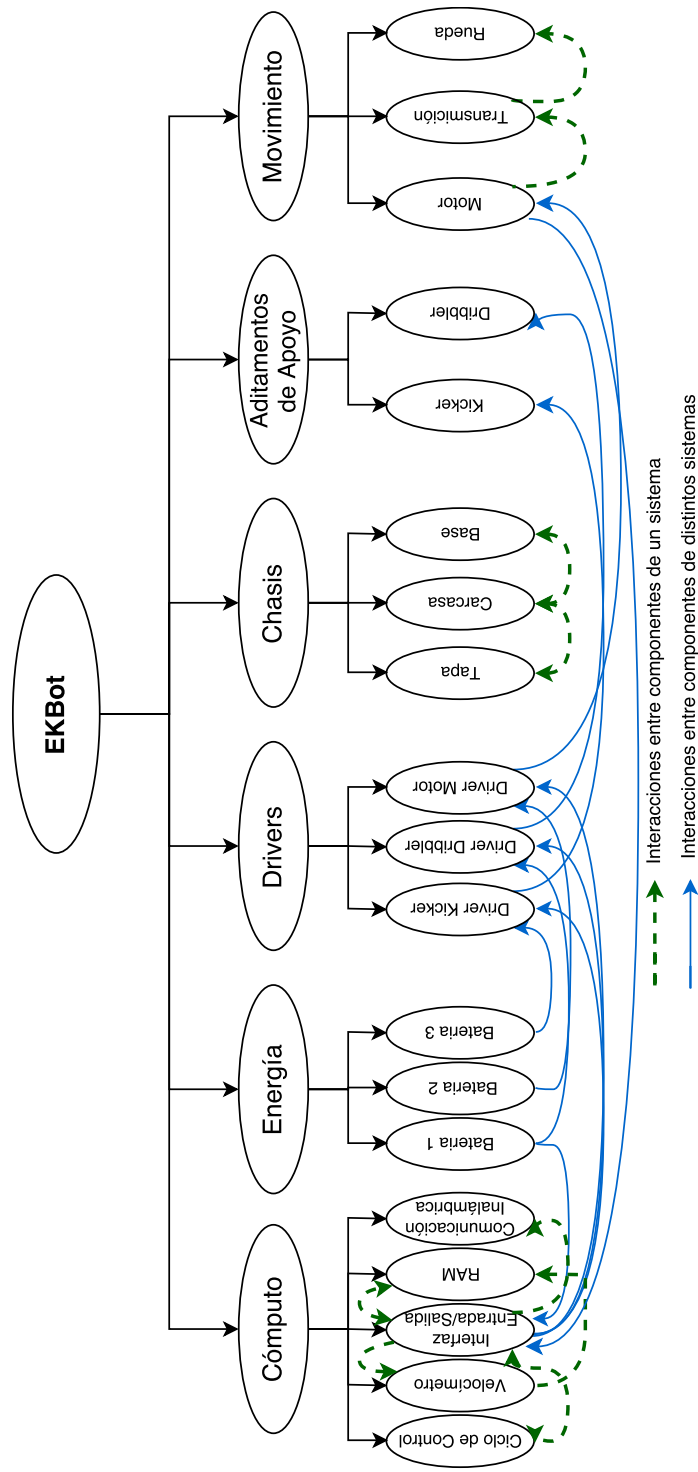


Figura 3.6: Arquitectura del Robot

3.4. Diseño a Detalle

En ésta sección se presentan los diseños finales obtenidos para cada componente de la arquitectura propuesta. Cada uno de los componentes son ensamblados con tornillos m3 de 5mm o 13mm de acuerdo a cada pieza. Cada componente puede estar compuesto por una o más piezas únicas. Con la finalidad de reducir costos, se prioriza utilizar como método de fabricación Modelado por Deposición Fundida debido a que se cuenta con el equipo necesario. Para ello, se prioriza utilizar plástico ABS por ser el de menor costo para ser usado en Modelado por Deposición Fundida aunque se utiliza plástico HIPS en las piezas de mayores dimensiones y que requieren mayor resistencia. Se utilizan piezas comerciales en los componentes donde se requiere alta precisión. El principal criterio de selección entre las alternativas de piezas comerciales es el costo, a menos que se trate de un sistema crítico, entonces se consideran otras características.

3.4.1. Chasis

Las piezas que comprenden éste sistema son las de mayores dimensiones, presentando dificultades para su manufactura por las características de dilatación del plástico ABS. Para disminuir los problemas relacionados a manufacturar piezas grandes mediante Modelado por Deposición Fundida, se utiliza plástico HIPS. Aunque el plástico HIPS tiene un costo superior al ABS, se caracteriza por ser resistente a impactos, una característica deseada en la carcasa. Adicionalmente, permite manufacturar piezas de dimensiones mayores mediante Modelado por Deposición Fundida, eliminando la necesidad de dividir las piezas en más pequeñas, facilitando el ensamblado. El modelo CAD del sistema se muestra en la Fig. 3.7.

I. Tapa

La tapa consta de tres piezas únicas: dos diseños propios y tres imanes iguales con los que se ajusta al chasis facilitando el acceso a los componentes interiores. Una pieza cuenta con el Standard Pattern, lo cual facilita intercambiar los identificadores de cada robot.

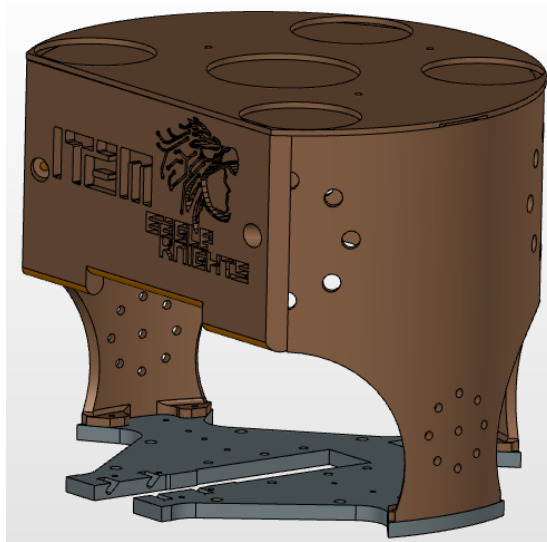


Figura 3.7: Chasis: Modelo CAD

II. Carcasa

Compuesta de dos piezas únicas, su función principal es proteger al resto de los componentes ante posibles colisiones durante el juego y las pruebas. Una pieza se coloca al frente del robot, de la otra se utilizan tres y se colocan alrededor del robot ensambladas a la base. Se cuenta con orificios en las piezas para facilitar la ventilación para disipar el calor generado por los drivers.

III. Base

A ésta pieza se acoplan los demás sistemas así como la carcasa. Estructuralmente es una de las piezas críticas porque es el soporte de todos los componentes físicos.

3.4.2. Energía

Se utilizan dos modelos de baterías *LiPo* distintas. Una batería de 7V para alimentar los componentes del Cómputo y al controlador del Dribbler. Dos baterías de 12V alimentan a un controlador cada una. Específicamente, una ali-

menta a los drivers de los motores y la otra al controlador del Kicker. El conector de la batería de 7V es distinto al de las baterías de 12V para evitar conectarlas a los sistemas erróneos. Utilizando una batería de 900mAh para la alimentación de los motores, el mínimo tiempo de uso esperado es de 6.75 minutos si los motores se encuentran a la máxima velocidad. El tiempo es el mismo para el Kicker si el consumo de éste fuera el máximo de manera continua.

3.4.3. Movimiento

El sistema mecánico está conformado por tres componentes, requiriendo cuatro *instancias* de cada componente. Todas las piezas no comerciales están manufacturadas mediante Modelado por Deposición Fundida utilizando plástico ABS para reducir costos.

I. Motor

Se utiliza un motor sin escobillas Maxon 200142 (“EC 45 flat 42.9 mm, brushless, 30 Watt”, 2012) que cuenta con tres sensores hall. El motor se acopla al chasis mediante una pieza propia. Motores de ésta marca han sido utilizados ampliamente por otros equipos de SSL ya sea éste modelo o similares; estos motores habían sido previamente adquiridos para una generación anterior de los robots pero no fueron utilizados. Con la finalidad de reducir costos, se optó por utilizar estos motores en lugar de comprar nuevos. Los motores constituyen el componente más caro del robot (por necesitar cuatro) pero a la vez son uno de los componentes críticos ya que se requiere de motores eficientes, de tamaño reducido y que cuenten con retroalimentación para realizar control a lazo cerrado. En la Fig. 3.8 se muestran acoplados éste componente y la transmisión.

II. Transmisión

Para acoplar el motor a la rueda se utiliza un tren de engranes rectos con reducción de velocidad de 3.5. Se optó por utilizar piezas comerciales debido a que se trata de un componente que requiere alta precisión para un acoplamiento adecuado. Se utilizan engranes rectos ya que, aunque éste tipo de engrane ocupa mayor espacio respecto a engranes planetarios, son más baratos. Al motor se

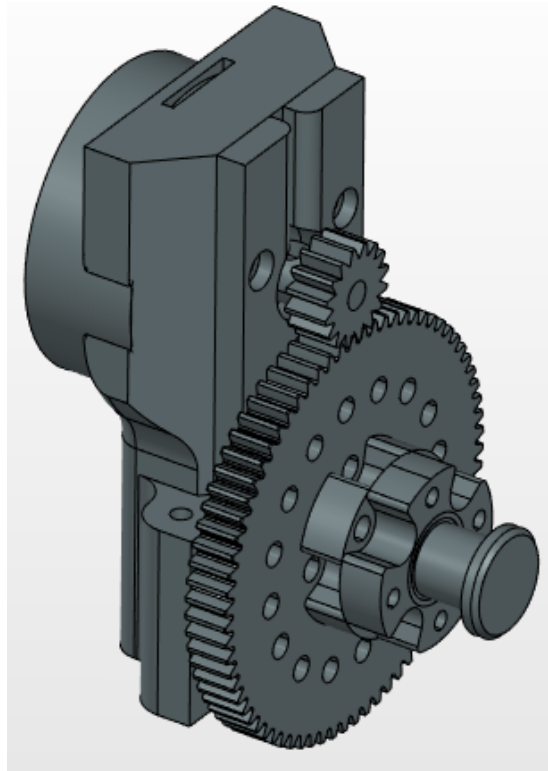


Figura 3.8: Motor y Transmisión: Modelo CAD

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

acopla mediante un prisionero el *pinion* (*4mm bore 32 pitch, 16t shaft mount pinion gear*, 2017). La rueda se acopla a un *hub* (*dual ball bearing hub*, 2017) al cual también se acopla el engrane (*1/2"bore 32 pitch aluminum hub gears*, 2017). Se utiliza como eje un tornillo de $\frac{1}{4}$ " el cual se introduce en el *hub* y en *motor.hub*. Son necesarios dos separadores para ajustar la rueda a la altura del eje adecuada.

III. Rueda

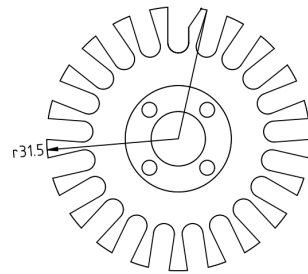
La rueda consiste en cuatro piezas únicas: la silueta, rodillo, o-ring (*Square-Profile Oil-Resistant Buna-N O-Rings*, 2017) y un cable. La silueta (Fig. 3.9 (a)) es la pieza principal de la rueda, está manufacturada mediante Modelado por Deposición Fundida. Se utilizan 21 rodillos a los cuales se les coloca un o-ring a cada uno. Mediante el cable, se insertan los rodillos a la silueta. En la Fig. 3.9 (b) se muestra el ensamble en CAD de la rueda. En la Fig. 3.9 (c) se muestran las 4 piezas únicas utilizadas en la rueda, en orden descendiente: o-ring, rodillo, cable y silueta.

3.4.4. Aditamentos de Apoyo

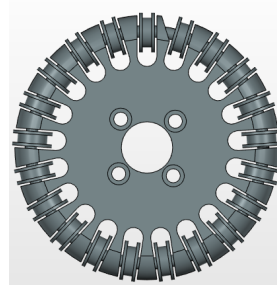
Este sistema es específico para participar en juegos de Robocup SSL. Éste sistema se puede reemplazar si el robot es utilizado para otra aplicación. El sistema está compuesto de cinco componentes, agrupados en dos subsistemas: Dribbler y Kicker. Los únicos componentes comerciales son actuadores: un motor con escobillas para el Dribbler mientras que para el Kicker es necesario utilizar solenoides comerciales. Para ambos casos el precio fue el principal factor de selección.

I. Kicker

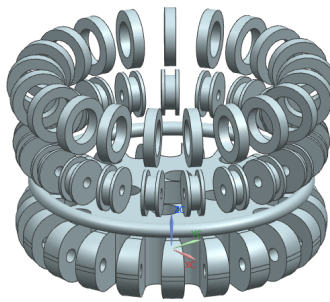
El objetivo de éste componente es patear la pelota. Se utiliza un solenoide como componente principal y un total de cuatro piezas únicas manufacturadas mediante Modelado por Deposición Fundida para acoplar el componente al chasis. Otra pieza se acopla al solenoide para realizar el pateo de la pelota. En la Fig. 3.10 se muestra el modelo CAD del kicker.



(a) Diagrama de la silueta



(b) Modelo CAD del componente



(c) Piezas que conforman la rueda

Figura 3.9: Rueda Omnidireccional Utilizada

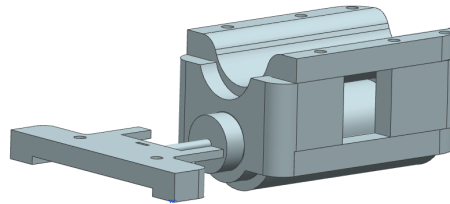


Figura 3.10: Kicker: Modelo CAD

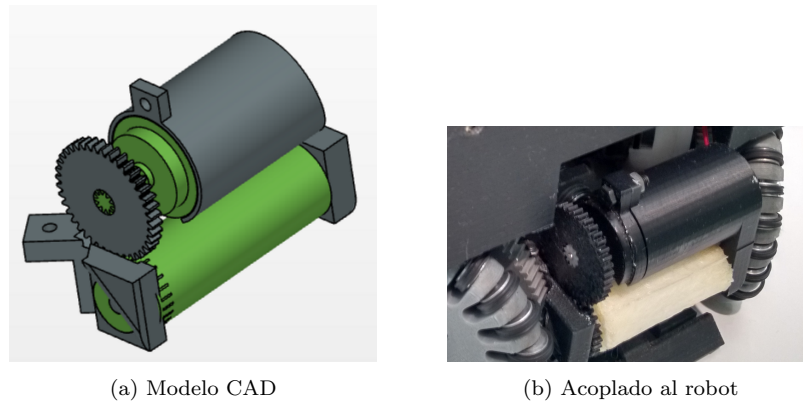


Figura 3.11: Dribbler

II. Dribbler

El dribbler permite mantener al robot mantener la posesión de la pelota. Está compuesto de tres subcomponentes: motor, transmisión y rodillo. Se utiliza un motor con escobillas (Gear y cols., 2017) debido a que no se requiere un control preciso de velocidad. Como transmisión se utiliza un tren de dos engranes manufacturados mediante Modelado por Deposición Fundida debido al bajo torque que se requiere. En la Fig. 3.11 se muestra el modelo CAD de éste componente así como el componente acoplado al robot.

3.4.5. Drivers

Éste sistema es la interfaz entre el sistema de cómputo y los actuadores. Es necesaria una interfaz debido a que el cómputo utiliza baja potencia mientras que los actuadores utilizan alta potencia. La mayoría de las piezas utilizadas en éste sistema son comerciales al tratarse de componentes eléctricos. Además de priorizar la reutilización de componentes se quiere minimizar el costo. El sistema también cuenta con piezas no comerciales manufacturadas con ABS. La función de éstas piezas es de acoplamiento.

I. Controlador de Motores

Como interfaz entre el subsistema de *Cómputo* y los motores sin escobillas que actúan las ruedas, se utilizan un Electronic Speed Controller (“Afro ESC”, s.f.) por motor. Cada Electronic Speed Controller recibe una señal PWM que codifica la velocidad deseada a 50 Hz, el ancho del pulso va de 1ms a 2ms, siendo 1.5ms la posición neutral.

II. Controlador Kicker

El circuito está compuesto de cuatro subsecciones: carga, almacenamiento, detección de la pelota y activación. La carga se encarga de incrementar 12 V de voltaje de entrada a 200 V y mantenerlos, pasándolos al almacenamiento que consiste en un capacitor de 1000 μF . La detección de la pelota consiste en un diodo infrarrojo y un fotodiodo que activan un relevador si se detecta un objeto entre ambos. La subsección de activación recibe la señal de la FPGA, estando aislada eléctricamente mediante octocopladores. El circuito se muestra en la Fig. 3.12.

III. Controlador Dribbler

El controlador del Dribbler utiliza un puente H (Instruments, 2005) para dar energía al motor además de un regulador de voltaje que entrega 5 V a la salida.

3.4.6. Cómputo

Está compuesto por un XBee WiFi y una tarjeta de desarrollo *Mojo* (“Mojo FPGA Development board”, s.f.) la cual integra una FPGA y un microcontrolador ATMEL-AVR.

En la Fig. 3.13 se muestra la arquitectura implementada en la FPGA así como los protocolos de comunicación usados entre la FPGA y los otros componentes. Los procesos implementados en la FPGA buscan aprovechar la alta frecuencia a la cual opera. Además, al no ser secuencial se pueden tener el número necesario de entradas y salidas operando de forma simultánea. En cambio, utilizar un microcontrolador ATMEL-AVR facilita la implementación de

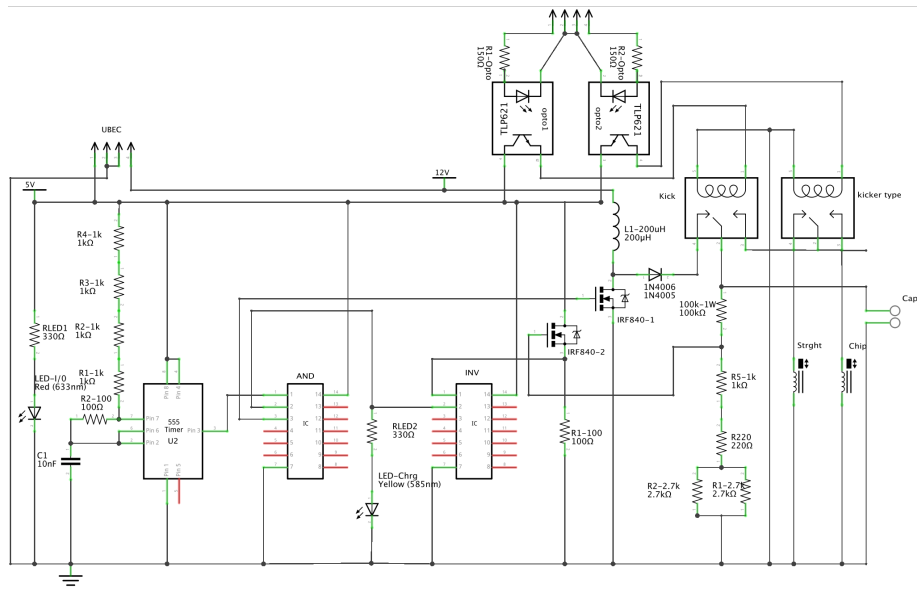


Figura 3.12: Circuito para el Kicker

las operaciones necesarias para realizar algoritmos de control así como realizar cálculos matriciales. Aunque la tarjeta *Mojo* representa un alto porcentaje del costo, al tener integradas una FPGA y un ATMEL-AVR ofrece características que son de gran utilidad como el manejo en paralelo de múltiples entradas y salidas.

I. Comunicación Inalámbrica

Para la comunicación inalámbrica se utiliza un XBee Wi-Fi el cual recibe el vector (X, Y, θ) de velocidad deseada del robot además de comandos para los aditamentos. Se utiliza direccionamiento estático y mensajes UDP para evitar retransmisiones. Cada mensaje utiliza 12 bytes que codifican el vector velocidad del robot (X, Y, θ) y la acción deseada para los aditamentos de apoyo.

CAPÍTULO 3: DISEÑO Y DESARROLLO DE UN ROBOT SSL

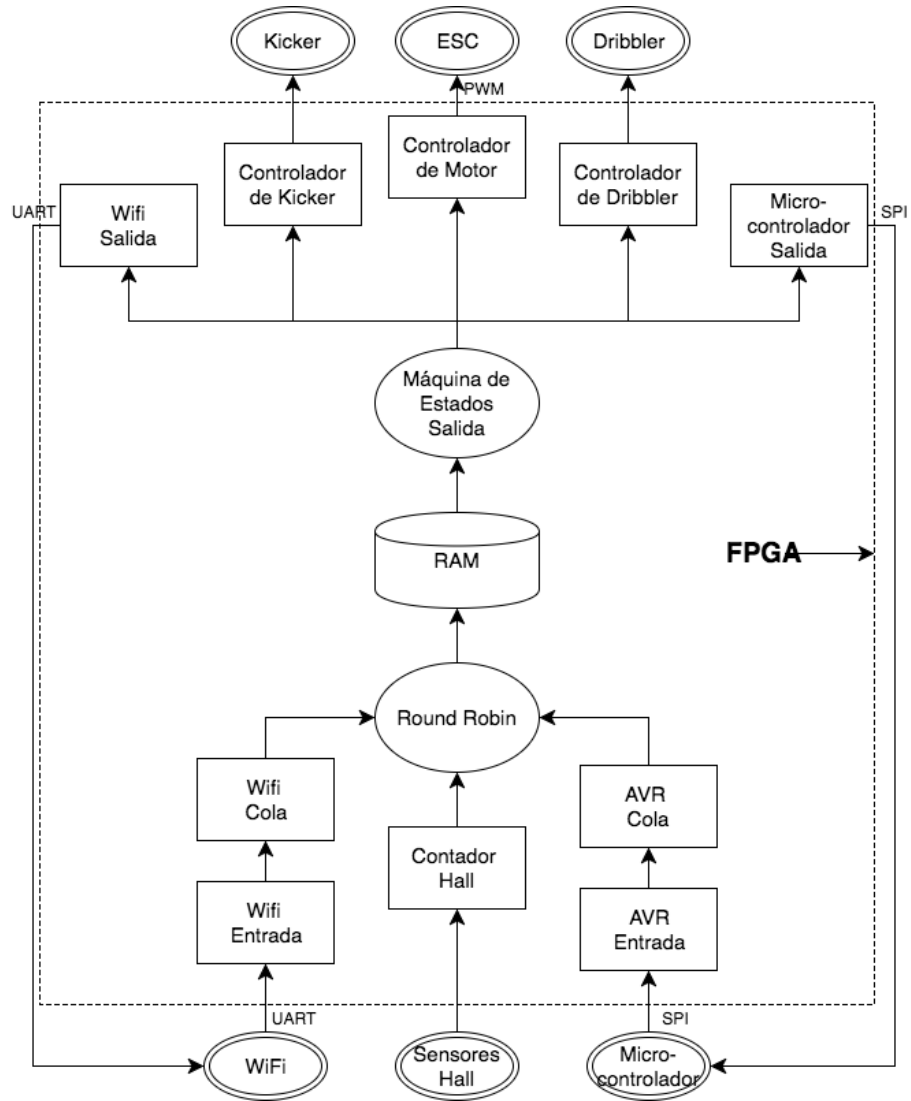


Figura 3.13: Modelo Implementado en la FPGA

II. Interfaz de Entrada/Salida

Se tiene tres posibles fuentes de entrada: la comunicación inalámbrica, sensores hall de los motores y el ATMEL-AVR. Para la comunicación inalámbrica es necesaria una interfaz UART mientras que para el ATMEL-AVR se utiliza una interfaz SPI. Se utiliza una cola para guardar los datos provenientes del AVR y otra para los de la comunicación inalámbrica. Las señales de los sensores Hall se pasan al velocímetro.

Existen cinco salidas a distintos componentes. Para la salida al ATMEL-AVR se utiliza la interfaz SPI y para la comunicación inalámbrica la interfaz UART. Tanto para la salida a los drivers de motores como del Dribbler se utiliza un módulo que convierte el dato a una señal PWM. Por último, la salida al controlador del Kicker es binaria.

III. RAM

La RAM se implementó mediante registros de 8 bits con direccionamiento de 7 bits; la lectura y escritura son independientes. Para la escritura se utiliza un esquema *Round Robin* que toma datos de las colas asociadas al ATMEL-AVR y a la comunicación inalámbrica así como del velocímetro.

Para la lectura se utilizan prioridades en caso de que dos componentes soliciten leer registros en el mismo ciclo de reloj. La prioridad más alta se asignó al ATMEL-AVR, la segunda al controlador de los motores, después a los drivers de Kicker y Dribbler, por último al Wi-Fi. Las prioridades están pre-codificadas en la FPGA, se definieron de acuerdo a la importancia de los procesos que solicitan los datos.

IV. Velocímetro

El cálculo de las velocidades reales de cada motor se realiza a partir de las tres señales de retroalimentación de los motores. La Tabla 3.1 muestra la secuencia seguida cada $\frac{1}{8}$ de revolución. Utilizando un contador por motor, se incrementa con cada cambio de las señales de los sensores Hall. Cada tiempo t constante, el valor del contador se escribe a registros y el contador se establece en 0. Manteniendo la última señal de dos sensores hall por motor, la dirección

Tabla 3.1: Secuencia de los Sensores Hall por Motor

	H^1	H^2	H^3
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	0	0
6	1	0	1

del motor se detecta mediante la función lógica $H_t^1 \oplus H_{t-1}^2$.

V. Ciclo de Control

El microcontrolador calcula y envía a la FPGA la señal de control deseada para cada motor. Cada señal de control es la salida de un lazo de control PI a nivel motor utilizando el error en la velocidad por motor. Las variables del control PI son calculadas manualmente. De la FPGA se obtienen, por motor, los pulsos de los sensores hall p_t contados en una ventana de tiempo F_m (hz). Conociendo los pulsos que genera cada motor por revolución p_r se obtiene la velocidad real de cada motor mediante la ecuación (3.1). Las velocidades de motor deseadas se obtienen a partir de un vector de velocidades de robot deseadas mediante la ecuación (3.2) utilizando los valores mostrados en la Fig. 3.14.

$$v_r = \frac{(p_t)(F_m)}{p_r} \quad (3.1)$$

$$v_m^d = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{bmatrix} +0.8090 & +0.5878 & 1 \\ -0.8090 & +0.5878 & 1 \\ -0.6691 & -0.7431 & 1 \\ +0.6691 & -0.7431 & 1 \end{bmatrix} \begin{pmatrix} v'_x \\ v'_y \\ 85w' \end{pmatrix} \quad (3.2)$$

La velocidad real del robot se puede calcular definiendo D^+ como la pseudo-inversa de D (definida en la ecuación (2.7)) tal que la ecuación (3.3) se cumple.

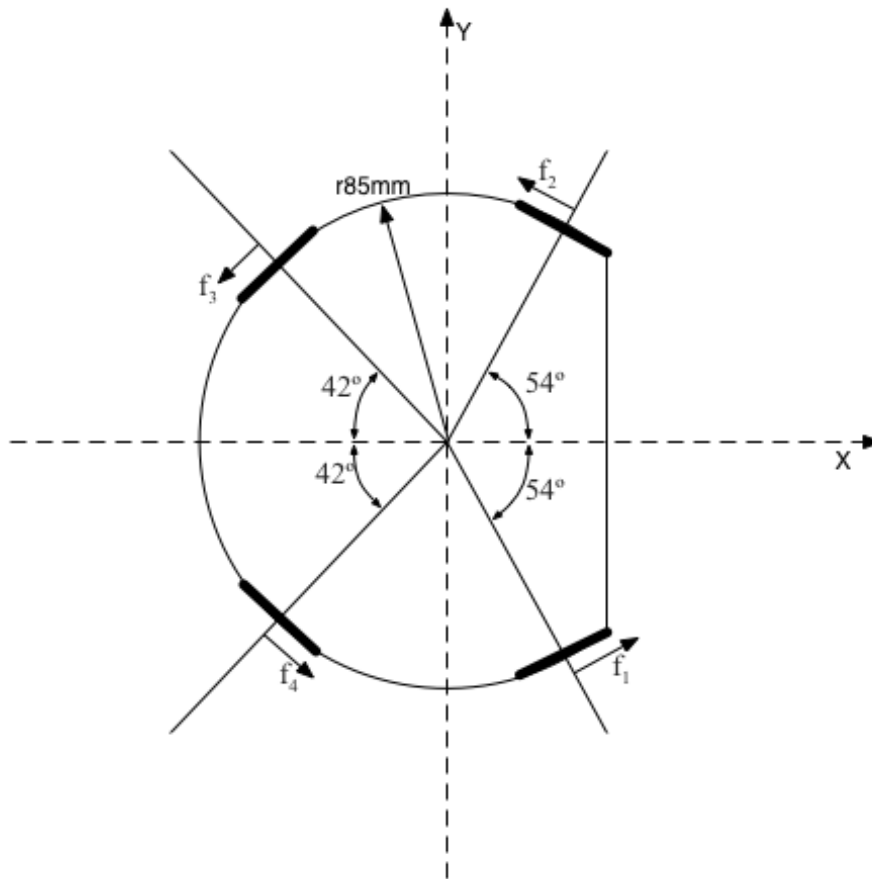


Figura 3.14: Diagrama del modelo implementado

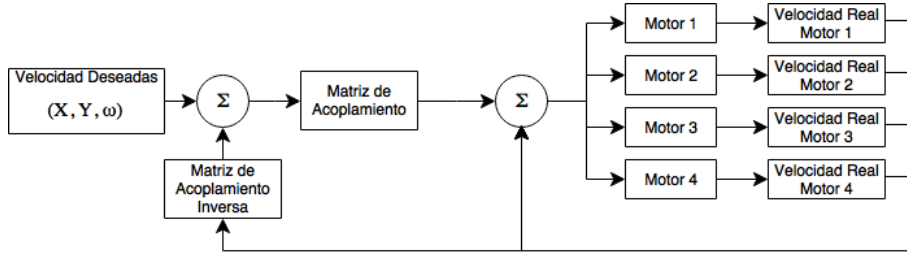


Figura 3.15: Ciclos de Control implementados en el AVR

Definiendo el vector $v_r = (v_1^r \ v_2^r \ v_3^r \ v_4^r)^T$ de velocidades reales de motor se obtiene (3.4) donde V_r es el vector de velocidades reales del robot.

$$D^+ D = I_3 \quad (3.3)$$

$$V_r = D^+ v_r \quad (3.4)$$

Es necesario encontrar la matriz D^+ , para el caso específico $\varphi_1 = \varphi_2$ y $\varphi_3 = \varphi_4$, la pseudoinversa se define en (3.5). A partir de los vectores de velocidades reales y deseadas del robot se obtiene el error utilizado en el segundo lazo de control PI. El esquema completo de control se muestra en la Fig. 3.15.

$$D^+ = \begin{bmatrix} a & -a & -c & c \\ e & e & -g & -g \\ i & i & k & k \end{bmatrix} \quad (3.5)$$

$$\begin{aligned} a &= c \cdot \cos 54 & k &= \frac{1}{\frac{2 \cos 42}{\cos 54} + 2} \\ c &= \frac{1}{\frac{2(\cos 42)(\sin 54)}{\cos 54} + 2 \sin 42} & e &= \frac{1}{2(\cos 54 + \cos 42)} \\ i &= \frac{k \cdot \cos 42}{\cos 54} & g &= e \end{aligned}$$

Antes de calcular la señal de control, se leen de la FPGA tanto las velocidades de motor reales como las de robot deseadas. El ciclo de control se ejecuta

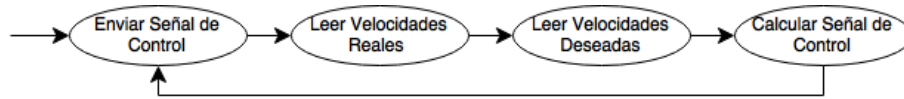


Figura 3.16: Secuencia de Funciones en el microcontrolador

a una frecuencia constante, determinada por una interrupción de reloj en el microcontrolador. Para mantener la sincronía, la señal de control calculada en cada ciclo se envía al inicio del siguiente ciclo. En la Fig. 3.16 se muestra la secuencia de funciones implementadas en el microcontrolador.

Capítulo 4

RESULTADOS

En éste capítulo se presentan experimentos para evaluar el desempeño del robot. Primero, se hace una evaluación de los sistemas implementados. Posteriormente, se realizan pruebas a nivel de robot para observar el comportamiento de los lazos internos de control, el que opera a nivel de velocidades de motor y el que opera a nivel de velocidad de robot. Después se presentan experimentos utilizando un sistema externo para determinar la capacidad del sistema para alcanzar poses a lazo abierto y después a lazo cerrado con trayectorias dinámicas. Por último, se realiza una comparación entre la solución implementada y otros robots participantes en Robocup SSL.

4.1. Resultados por Sistema

A continuación se muestran resultados relacionados al desempeño individual de cada sistema. Una comparación del ensamble completo con el modelo CAD se muestra en la Fig. 4.1.

4.1.1. Chasis

Manufacturar el chasis utilizando plástico HIPS permitió utilizar piezas de grandes dimensiones sin tener deformaciones. El chasis ha soportado satisfactoriamente colisiones entre los robots así como contra objetos estáticos. La base del

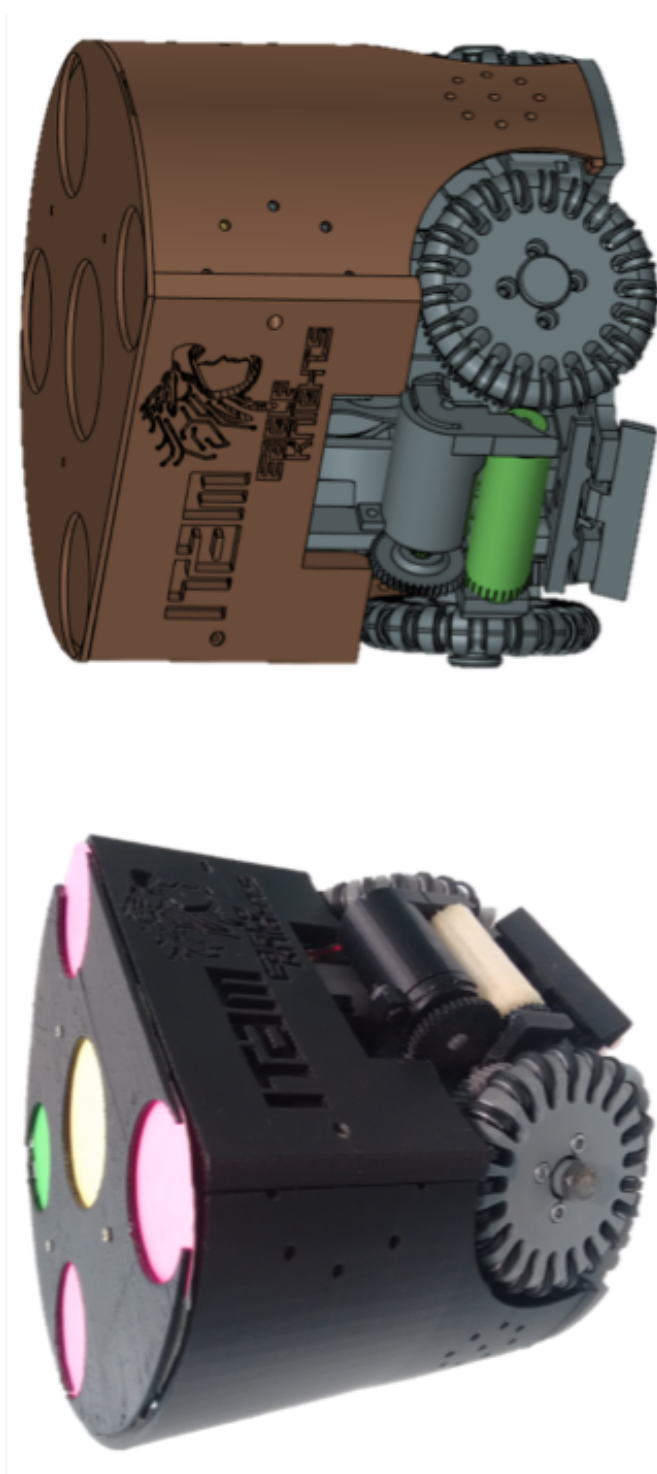


Figura 4.1: Robot Real vs Modelo CAD

CAPÍTULO 4: RESULTADOS

robot soporta a los demás sistemas y no ha presentado problemas que requieran reemplazar ésta pieza. El tener el Standard Pattern como parte del diseño de la tapa ha permitido eliminar problemas de lecturas erróneas de la visión que se presentaban con anteriores diseños de la tapa. Adicionalmente, por la capacidad de retirar la tapa para acceder a los otros sistemas, se facilita el intercambio de pilas e incluso apagar el robot rápidamente.

4.1.2. Energía

Tener tres baterías permite separar eléctricamente los circuitos y facilitar la detección de problemas. Al tener conectores distintos para cada tipo de batería se facilita la conexión de baterías y se minimiza la posibilidad de conexiones incorrectas. La batería que debe ser reemplazada con mayor frecuencia es la del controlador del Kicker mientras que la que menos se reemplaza es la que alimenta al cómputo. El tiempo de duración mínimo de la batería de los motores es de 6.75 minutos, debido a que la utilización de los motores no es homogénea ni continua ni a las especificaciones máximas, su duración es mayor. La duración mínima de la batería del kicker es de 6.75 minutos si estuviera todo el tiempo en modo de carga; dado que en realidad la mayor parte del tiempo está en modo de mantener la carga, la duración de la batería es mayor.

4.1.3. Movimiento

En la Fig. 4.2 se muestra una de las cuatro *instancias* del sistema de movimiento. La pieza principal de la rueda no ha presentado problemas aunque algunos *o-rings* se han roto. Solo se han presentado problemas con el sistema de engranes ocasionados por pelusa que se encuentra en la superficie de pruebas. Por la facilidad de remover cada rueda del resto del robot, el proceso de intercambio es rápido.

4.1.4. Aditamentos de Apoyo

Ambos aditamentos implementados han tenido buen desempeño en diversas pruebas realizadas. El Kicker es capaz de patear la pelota a $1.5 \frac{m}{s}$. El Dribbler es

CAPÍTULO 4: RESULTADOS

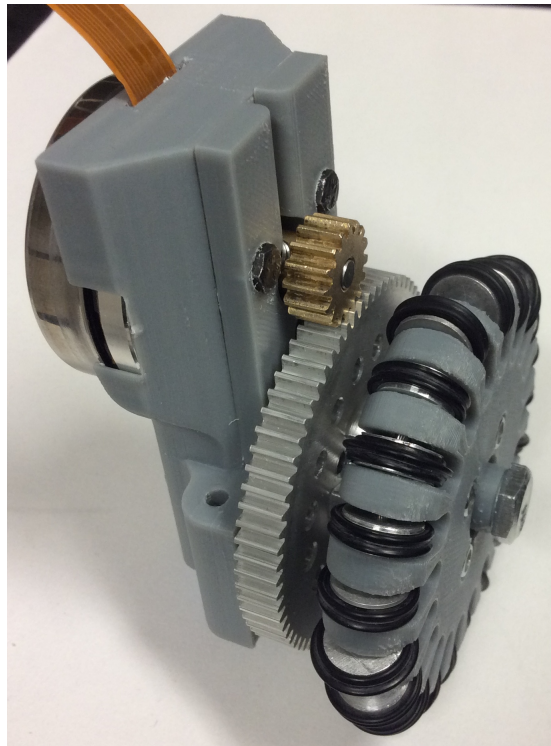


Figura 4.2: Ensamble de la Rueda, Motor y Transmisión

CAPÍTULO 4: RESULTADOS

capaz de acercar la pelota lo suficiente para que el Kicker le pegue. Ambos componentes son independientes del los otros componentes estructurales del robot (solamente dependen de la electrónica), pudiendo ser fácilmente reemplazados.

4.1.5. Cómputo

Las funciones implementadas en el sistema de cómputo son capaces de ejecutarse en menos de 10ms aunque por sincronización se tienen a una frecuencia de 20Hz. La visión determina la frecuencia del sistema al ser el sensor más lento. La utilización de una FPGA con un ATMEL-AVR a demostrado ser efectiva para controlar el robot. La FPGA permite leer numerosas señales y procesarlas sin interferir en el tiempo de cómputo del ATMEL-AVR además de tener mayor resolución en las señales de salida permitiendo mayor resolución en la señal de la velocidad deseada de cada motor. Al utilizar al ATMEL-AVR exclusivamente para calcular las señales de control, se puede minimizar el tiempo total de cómputo por no requerir de interrupciones externas para adquirir señales. El XBee es capaz de recibir los mensajes del sistema externo a la frecuencia que se generan y transmitirlos a la FPGA. Como el reloj de la FPGA es más rápido que el del XBee, no se requieren de protocolos de control de flujo para la información recibida.

4.1.6. Drivers

El driver de cada motor responde rápidamente ante cambios en la señal y no representan una fuente de calor importante. El driver del Kicker es capaz de cargar el capacitor a 200V en menos de 5 segundos. Aunque representa una fuente de calor importante al interior del robot, esto no han generado problemas. Debido a que no se cuenta con la capacidad de hacer circuitos impresos para la electrónica, actualmente esta ocupa mucho espacio y fácilmente se desconectan los cables.

4.2. Resultados de la Integración de los Sistemas

Se presentan los resultados de tres tipos de pruebas realizadas. La primera prueba es a nivel robot para observar el comportamiento de los lazos de control internos del robot que operan a nivel motor y a nivel de la velocidad del robot.

La segunda prueba es para determinar la capacidad del robot de alcanzar poses sin lazos de control externos. Para la tercera prueba se reutiliza el mismo ambiente que con la segunda pero se cierra un tercer lazo de control en el sistema externo. Ésta prueba busca determinar la capacidad de respuesta del robot ante trayectorias dinámicas.

4.2.1. Pruebas a Nivel de Motor

En estas pruebas el robot recibe un vector de velocidad deseada (X, Y, θ) cuyos parámetros se fijan en uno de los siguientes valores: $+V_{cte}$, 0 , $-V_{cte}$. Existen 27 posibles combinaciones presentadas en la tabla 4.1 que se aplican en secuencias espaciadas en intervalos de 10s para medir la respuesta de los motores en dos escenarios como se muestra en la Fig. 4.3. Primero, solamente se utiliza el lazo de control a nivel de motor manteniendo abierto el lazo de control a nivel robot. La velocidad deseada de cada motor permanece constante tras recibir la señal y solo en algunos casos no se alcanzan las velocidades de robot deseadas. Esto sucede cuando la velocidad de motor requerida es superior a la *no-load speed* del motor, $(+V, +V, +V)$ y $(-V, -V, -V)$ para el motor 1. También sucede esto cuando la velocidad de motor requerida es muy baja, presentándose oscilaciones alrededor de esta, $(+V, -V, 0)$ y $(+V, -V, -V)$.

Después se prueba el lazo de control a nivel de robot que determina la velocidad requerida al lazo de control a nivel de motor, por lo que esta última no es constante. En todos los casos se alcanzan y mantienen las velocidades de robot independientemente de que se consigan las velocidades requeridas de motor. En algunos casos, $(+V, +V, +V)$ y $(+V, +V, -V)$, la respuesta es lenta aunque esto se puede mejorar sintonizando k_p y k_i .

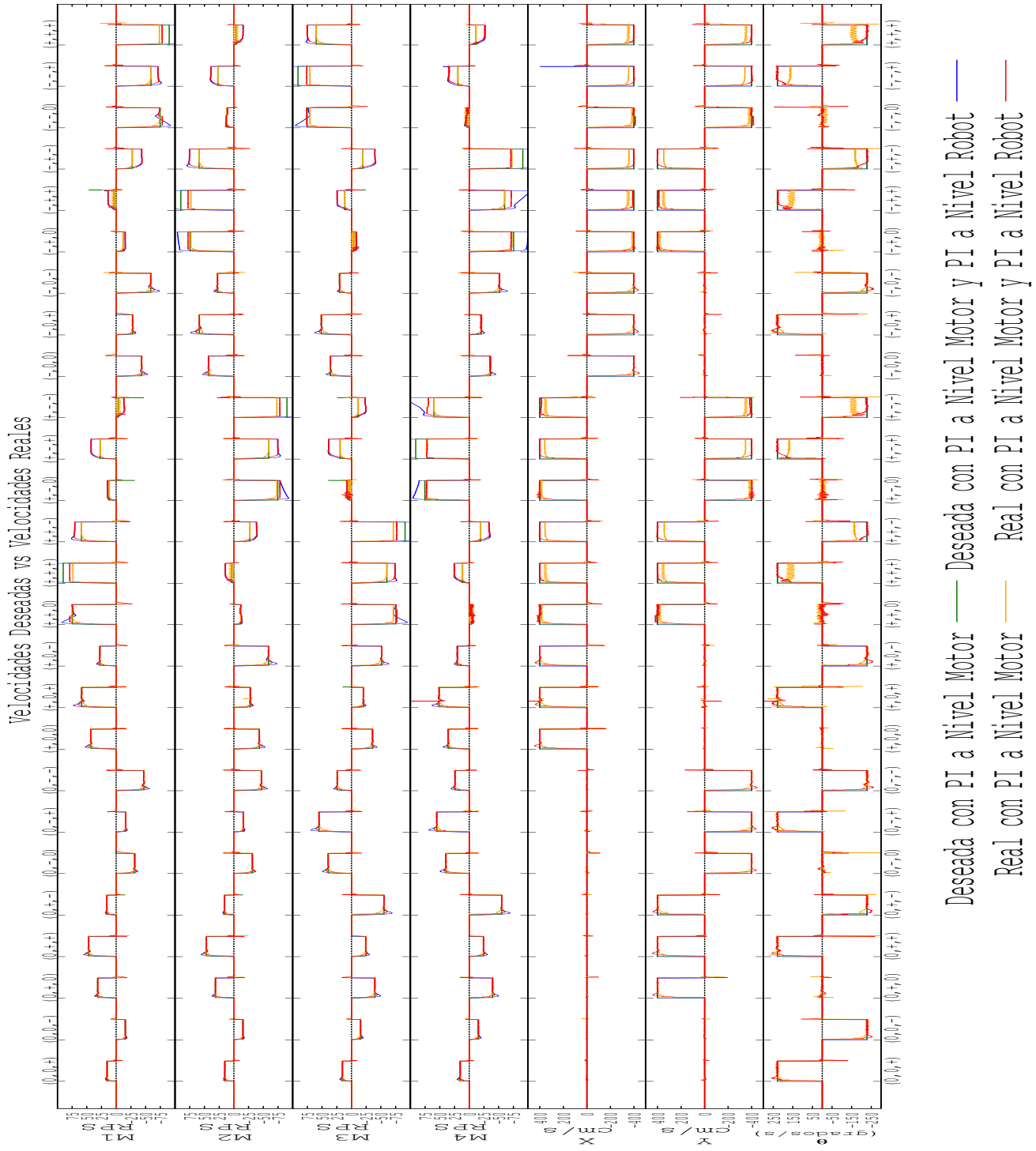


Figura 4.3: Velocidades Deseadas vs Velocidades Reales

Tabla 4.1: Secuencia para las pruebas a Nivel de Motor

#	X	Y	θ	#	X	Y	θ	#	X	Y	θ
0	0	0	0	9	+V	0	0	18	-V	0	0
1	0	0	+V	10	+V	0	+V	19	-V	0	+V
2	0	0	-V	11	+V	0	-V	20	-V	0	-V
3	0	+V	0	12	+V	+V	0	21	-V	+V	0
4	0	+V	+V	13	+V	+V	+V	22	-V	+V	+V
5	0	+V	-V	14	+V	+V	-V	23	-V	+V	-V
6	0	-V	0	15	+V	-V	0	24	-V	-V	0
7	0	-V	+V	16	+V	-V	+V	25	-V	-V	+V
8	0	-V	-V	17	+V	-V	-V	26	-V	-V	-V

4.2.2. Pruebas de Integración con el Sistema Externo

Para las siguientes pruebas, se establecen puntos que el robot debe alcanzar y se utiliza el sistema computacional externo para definir los perfiles de velocidad y captura de datos. Como se muestra en la Fig. 4.4, se define una circunferencia con centro en P_0 y 8 puntos P_A, P_B, \dots, P_H en la circunferencia tales que: $\angle P_n P_0 P_{n+1} = 45^\circ$. La prueba realizada consiste en colocar al robot en P_0 , siempre orientado a P_C . El sistema externo calcula la velocidad (V_x, V_y, ω) para dirigir el robot a cada uno de los 8 puntos.

I. Pruebas Sin Retroalimentación de Visión

En estas pruebas el sistema externo es usado para generar y enviar la velocidad inicial deseada además de capturar las poses del robot en el tiempo. La Fig. 4.4 muestra las trayectorias ideales y 10 repeticiones de trayectorias reales. Cada punto representa el centro del robot, el tamaño del robot en la escala utilizada se muestra en la esquina inferior derecha. Adicionalmente, se muestra una tendencia lineal obtenida a partir de los datos graficados.

Para el movimiento traslacional, se obtuvo el mejor desempeño en dirección al punto P_C y a P_G donde mantiene líneas casi rectas. En dirección a P_B y a P_D las trayectorias son curvas, cruzando al vector de dirección ideal. En dirección a P_F y P_H tenemos los mayores errores aunque el ciclo de control a nivel robot

CAPÍTULO 4: RESULTADOS

corrige el movimiento para llegar al punto objetivo. Las trayectorias hacia P_A y P_E presentan el peor desempeño sin alcanzar el punto objetivo en ninguna ocasión.

En movimiento rotacional, se mantuvo la velocidad en angular en 0 para mantener la orientación del robot. En la Fig. 4.5 se observa la orientación del robot en cada prueba. Se obtuvo el mejor desempeño en dirección a los puntos P_C y P_G . En dirección a P_B, P_D, P_F y P_H , el error es mayor aunque en la mayoría de los casos es menor a 30 grados. Solo en dirección a P_A y P_E , el error llega a sobrepasar los 30 grados, aunque es mayor en las trayectorias al punto P_A . El error en la orientación del robot en las trayectorias a cada punto es consistente con el error en su movimiento traslacional.

En ambos casos, los errores se reflejan en el eje X. Las direcciones del error para los puntos P_1, P_2 y P_8 es la contraria que para los puntos P_4, P_5 y P_6 . Esto indica que al menos para la k_i utilizada en el ciclo de control de velocidades del robot se podría encontrar un valor más adecuado.

4.2.3. Pruebas con Retroalimentación de Visión y Trayectoria Dinámica

El objetivo de éstas pruebas es validar el movimiento del robot ante un ambiente dinámico donde la trayectoria deseada cambia rápidamente. Se utiliza un sistema externo desarrollado sobre Robot Operating System (ROS) con la arquitectura mostrada en la Fig. 4.6. El subsistema de *Visión* utiliza cámaras en la parte superior del área de pruebas para determinar la pose real del robot respecto al mundo. A partir de ésta información, el subsistema *Planeación* determina la ruta deseada para el robot de acuerdo al comportamiento deseado. Para éstas pruebas, el comportamiento consiste en alcanzar cada uno de los puntos P_A, \dots, P_H con la secuencia: $P_A, P_0, P_B, P_0, \dots, P_G, P_0, P_H, P_0$. La ruta generada para las pruebas siempre es una recta entre la posición del robot y el punto objetivo, debido a la ausencia de obstáculos a esquivar. El subsistema de control genera una trayectoria para seguir la ruta establecida. La trayectoria consiste en puntos *atractores* intermedios entre el punto inicial y el objetivo. Los puntos no son necesariamente alcanzados por el robot debido a la alta frecuencia con la que se actualiza el siguiente punto atractor. El control a bajo nivel genera

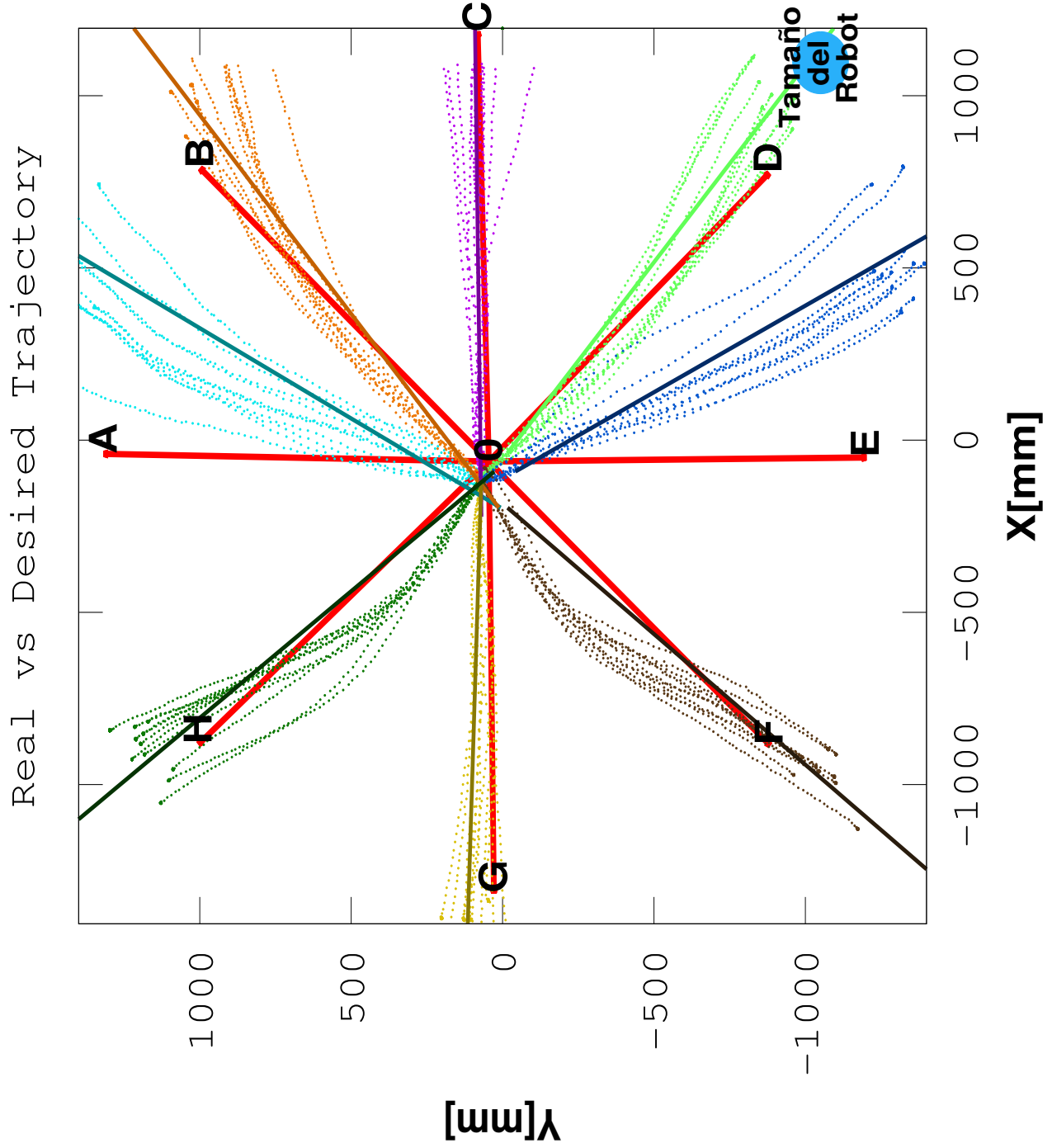


Figura 4.4: Trayectorias sin Retroalimentación de Visión

Error en la Orientación del Robot a Cada Punto

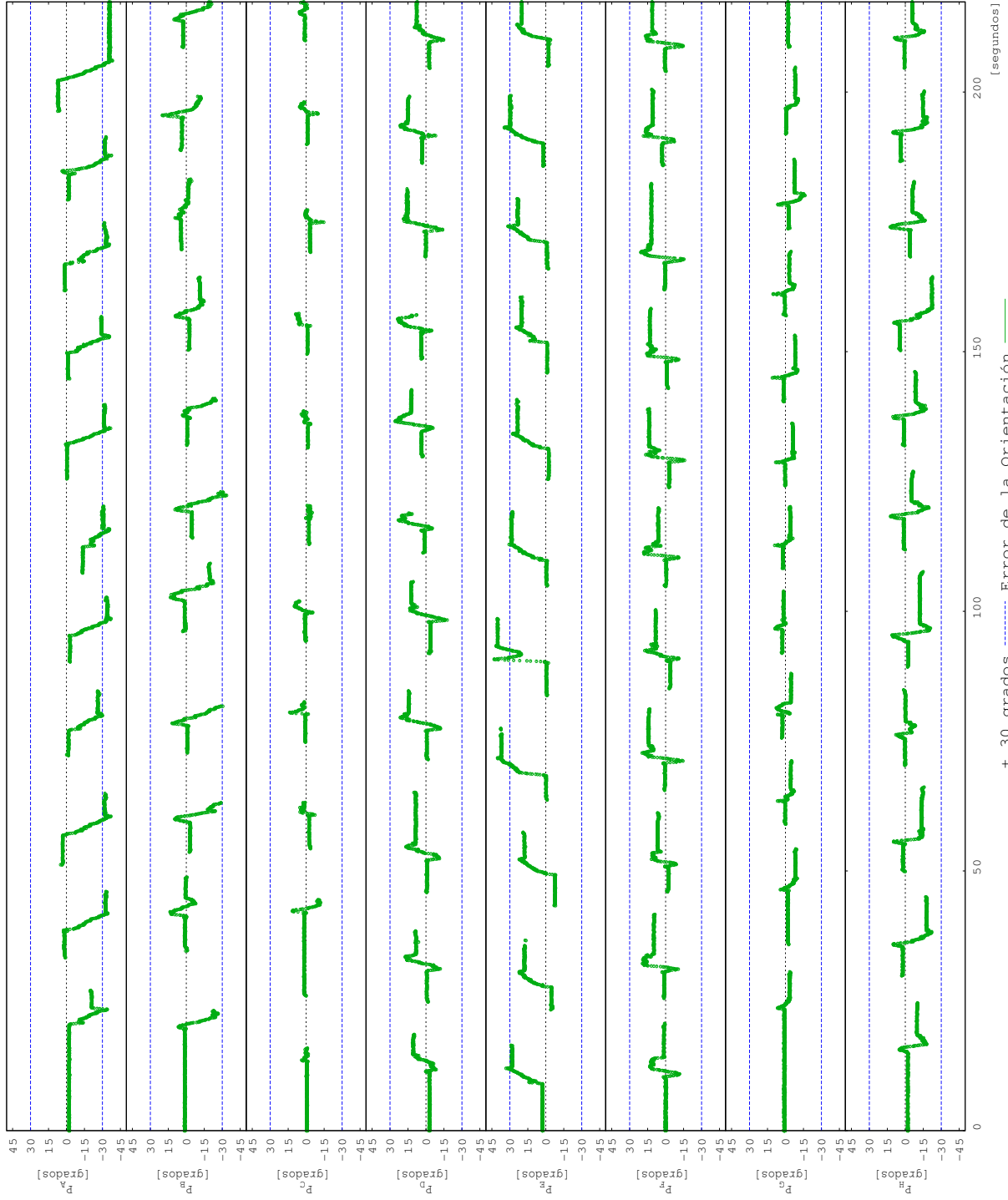


Figura 4.5: Error en la Orientación del Robot en las Pruebas sin Retroalimentación de Visión

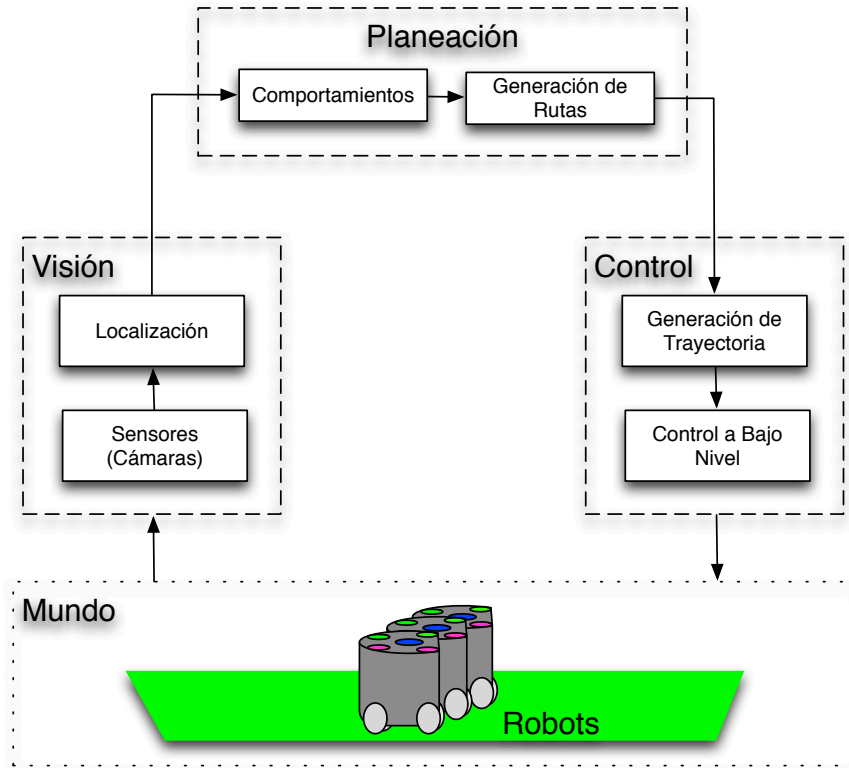


Figura 4.6: Arquitectura del Sistema Utilizado

el perfil de velocidad para dirigir al robot al punto atractor y envía el perfil al robot. El sistema externo está limitado por la frecuencia de 25Hz del subsistema de visión.

En movimiento traslacional, debido a los cambios en el tiempo de los puntos intermedios, las trayectorias no necesariamente son rectas. Sin embargo, el robot siempre alcanza los puntos objetivo deseados. En la Fig. 4.7 se muestra la respuesta del robot en las 10 pruebas realizadas. Para cada prueba se grafica la trayectoria deseada (generada por el subsistema de control desde el sistema externo) así como la trayectoria real seguida por el robot.

El error en la orientación del robot en el tiempo se muestra en la Fig. 4.8. El error es mínimo solo llegando en una ocasión a ser de 60 grados. El subsistema

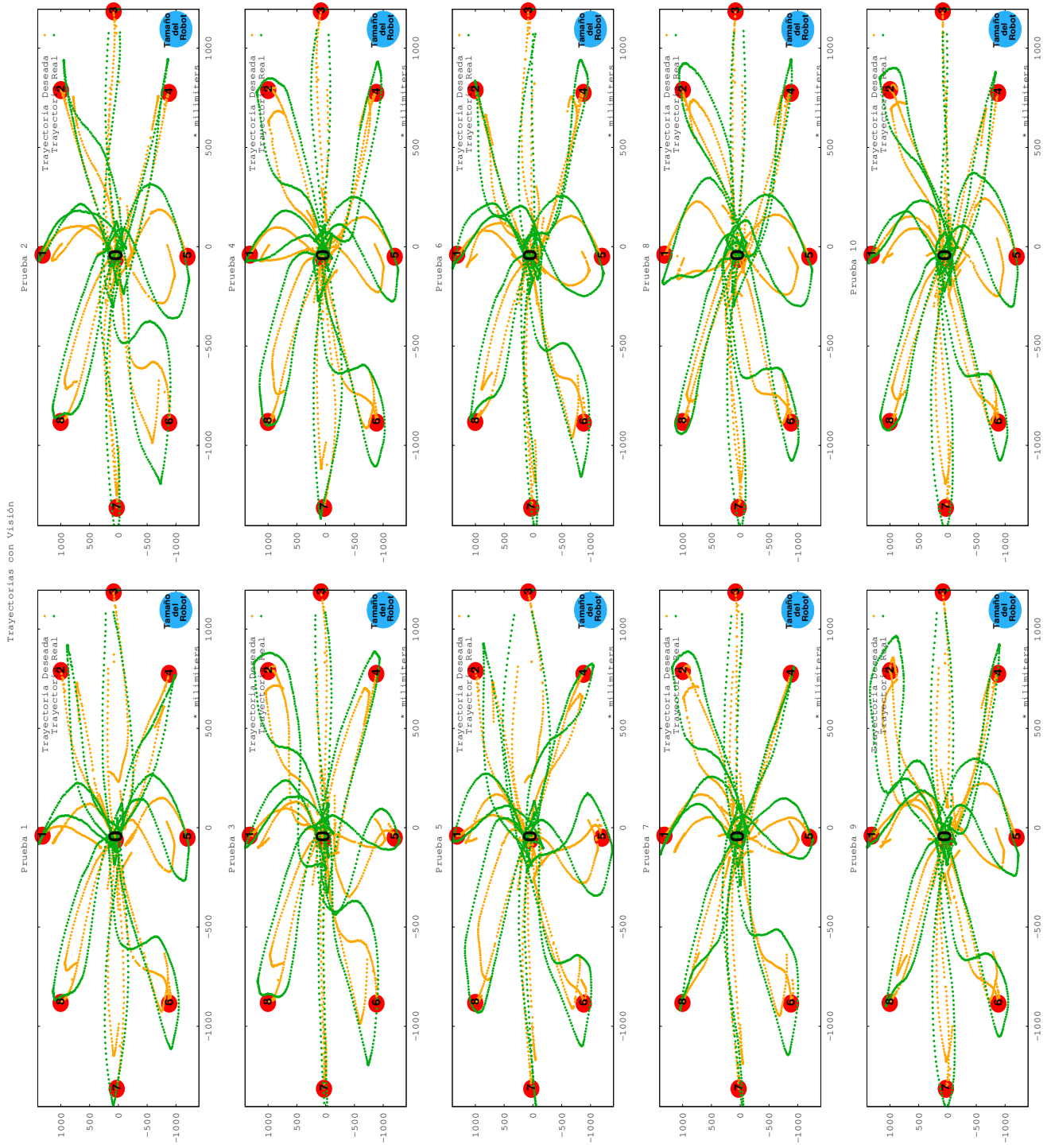


Figura 4.7: Movimiento del Robot con Retroalimentación de Visión y Trayectorias Dinámicas

CAPÍTULO 4: RESULTADOS

Punto	Sin Visión			Con Visión		
	e_X	e_Y	e_θ	e_X	e_Y	e_θ
A	27.95	70.94	2.2231	5.04	21.42	1.3923
B	26.06	26.15	0.5271	7.22	8.26	0.7735
C	22.91	0.82	0.1261	6.57	1.54	0.2521
D	10.66	11.16	0.2521	3.62	3.33	0.2578
E	5.72	14.28	0.3724	1.54	4.06	0.3266
F	5.89	8.75	0.1891	2.34	2.78	0.2807
G	8.88	0.65	0.0688	2.77	0.66	0.1318
H	4.97	6.75	0.1375	1.66	2.16	0.1089

Tabla 4.2: Norma cuadrada de los errores e_X [mm], e_Y [mm] y e_θ [grados]

de control en el sistema externo considera una tolerancia para la orientación de ± 6 grados por lo que el sistema no genera velocidad rotacional para corregir si el error es menor a la tolerancia.

La tabla 4.2 muestra una comparación de la norma cuadrada de los errores en pose (e_X, e_Y, e_θ) que muestra el mejor desempeño esperado cuando se usa retroalimentación con visión. En la mayoría de los casos el error a lazo abierto es de 3 a 5 veces el obtenido a lazo cerrado. En orientación no se obtienen tales mejoras, pero hay que notar que desde antes el error era mínimo pues la velocidad angular deseada era muy baja y en el caso de lazo cerrado no necesariamente es el caso. Lo mismo aplica para la velocidad en Y en dirección a P_C y a P_G .

4.3. Comparación con Otros Robots SSL

A continuación se realiza una comparación con los cinco mejores equipos de Robocup SSL 2016. Se toman los datos reportados en los *Team Description Paper* de cada equipo. Los equipos comparados son: *MRL* [(Poudeh y cols., 2016), (Adhami-Mirhosseini, Bakhshande Babersad, Jamaati, Asadi, y Ganjali, 2012)], *CMDragons* [(Biswas, Mendoza, Zhu, Klee, y Veloso, 2013), (Zickler,

CAPÍTULO 4: RESULTADOS

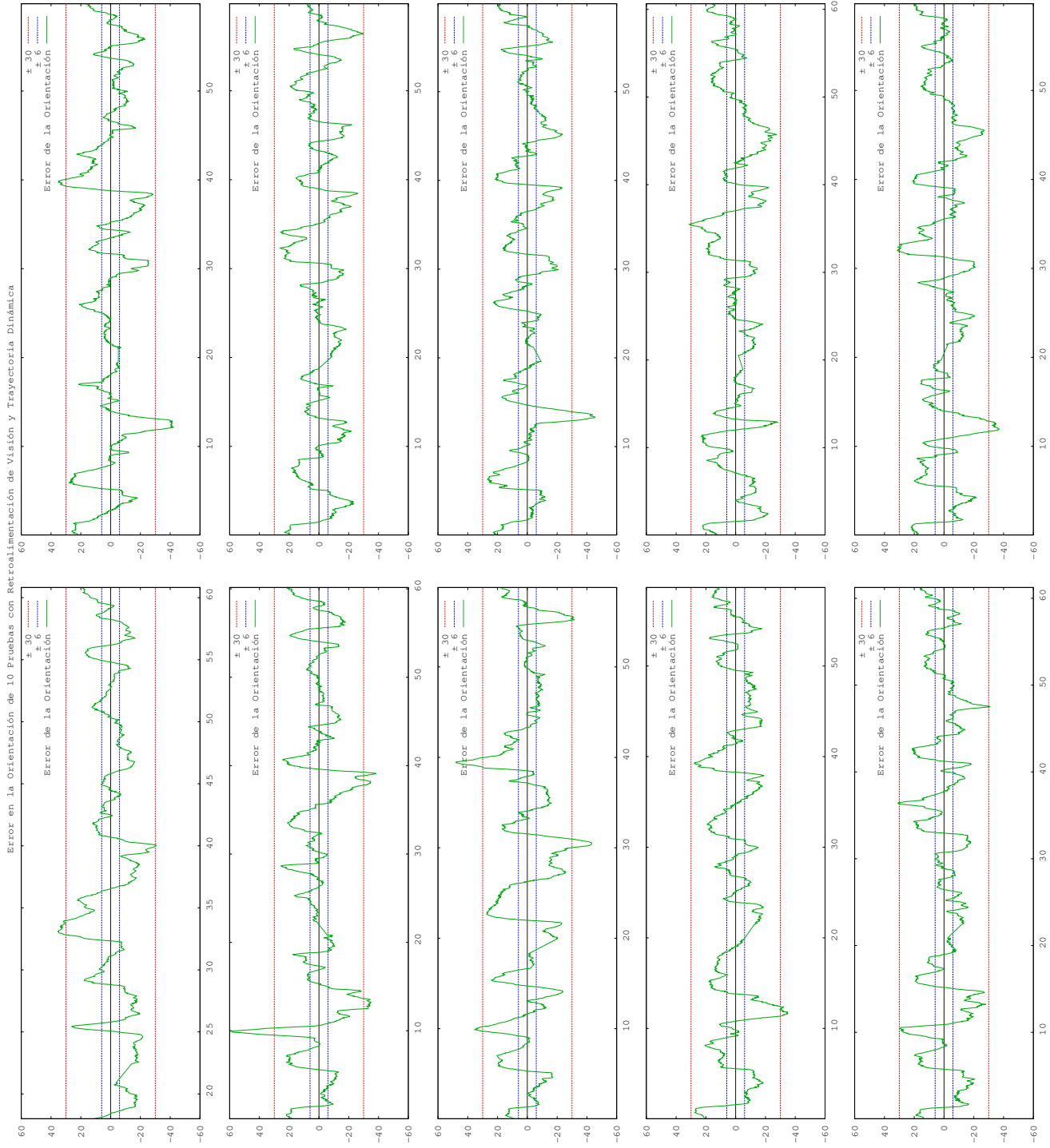


Figura 4.8: Velocidades Deseadas vs Velocidades Reales

CAPÍTULO 4: RESULTADOS

Biswas, Luo, y Veloso, 2010)], *ZJUNlict* [(Zhao y cols., 2013)], *Robodragons* [(Adachi y cols., 2016)] y *ER-Force* [(Lobmeier y cols., 2016), (“ER-Force Team DEscription Paper for Robocup 2014”, s.f.)]. Todos los equipos han estado compitiendo por al menos tres años de manera continua.

La mayor parte de las piezas de los robots de los equipos están fabricadas con aluminio con algunas partes hechas con otros materiales. De manera similar al diseño presentado, todos los equipos utilizan cuatro ruedas omnidireccionales con un tren de engranes y motores Maxon. Cada equipo utiliza ruedas propias, con diferente número de rodillos y diámetro de rueda, aunque la presentada en éste trabajo es la mayor. La proporción de reducción del tren de engranes varía en cada equipo aunque está entre 3 y 4. La carcasa de cada equipo es diferente, salvo ER-Force que solamente cuenta con tapa, los demás cuentan con protección lateral. Todos los equipos han implementado algún tipo de solución para acceder rápidamente a los componentes internos del robot aunque ninguno presenta una solución para acoplar/desacoplar la tapa con imanes. Destaca que los equipos que cuentan con protección lateral utilizan una sola pieza para la carcasa, mientras que la solución propuesta utiliza en total 4 piezas.

La electrónica de todos los equipos es compacta al estar integrada mediante Printed Circuit Board (PCB). Una de las principales diferencias entre los equipos radica en los sistemas de cómputo. De manera similar a la solución implementada, MRL, CMDragons y Robodragons distribuyen el cómputo en una FPGA y un microcontrolador. Todos utilizan la FPGA para el manejo de señales aunque MRL implementó las funciones de control directamente en la FPGA, utilizando el microcontrolador para otros procesos. ZJUNlict implementa todo mediante una FPGA donde implementa un procesador para el manejo de funciones específicas. Solamente ER-Force utiliza un microcontrolador para implementar todas las funciones requeridas. Aunque todos los equipos utilizan la retroalimentación de motores, solamente ZJUNlict detalla el algoritmo utilizado, siendo similar al implementado en este trabajo.

Todos los equipos implementan funciones de control a nivel motor ya sean PI o PID. Ninguno de los equipos considerados implementa un segundo lazo de control a nivel robot como en la solución presentada. Solamente ER-Force especifica que utilizan el método *Ziegler-Nichols* para la sintonización manual del PID. Destaca el diseño completamente modular de ER-Force en la electróni-

CAPÍTULO 4: RESULTADOS

ca, manteniendo PCBs independientes para cada componente. Ningún equipo detalla en las baterías utilizadas.

Todos los equipos presentan soluciones únicas respecto a los aditamentos para el juego. Para el Dribbler, a diferencia de la solución presentada, todos utilizan motores sin escobilla. Para el Kicker, utilizan tanto un circuito de carga a 200V aunque los tiempos de carga varían. Es en estos componentes donde es común que se hagan cambios cada año tanto de forma como de material.

Para la comunicación inalámbrica, solamente Robodragons utiliza radios WiFi aunque todos utilizan comunicación en la banda de 2.4Ghz. La principal razón que presentan para no utilizar WiFi es poder definir un protocolo específico de comunicación entre el sistema y el robot, pudiendo tener comunicaciones más rápidas y con menos interferencia. Una de las ventajas de la solución implementada es que no depende de una tecnología de comunicación inalámbrica específica, pudiendo cambiar el módulo fácilmente. A pesar de esto, WiFi utilizando UDP a probado ser efectivo en las pruebas realizadas.

No se realizan comparaciones de desempeño debido a que no se cuenta con los datos necesarios de los otros equipos.

Capítulo 5

CONCLUSIONES Y TRABAJO FUTURO

El diseño del robot omnidireccional presentado en éste trabajo es una solución modular enfocada a la Robocup SSL a bajo costo. La arquitectura modular de la solución facilita la actualización de los sistemas que la conforman y permite reemplazar sistemas completos si alguna aplicación futura así lo requiere. El diseño se realizó para utilizar Modelado por Deposición Fundida para la manufactura de las piezas.

El diseño y fabricación de una rueda propia permite un importante ahorro de espacio, además de reducir el costo total. El diseño de la rueda lo conforman pocas piezas únicas, facilitando su ensamblado. Adicionalmente, en las diversas pruebas realizadas no se han reportado fracturas en las ruedas. Si bien el sistema *Movimiento* es el más caro, también es un sistema crítico que requiere de componentes de alta calidad para funcionar adecuadamente.

La carcasa representó un reto importante debido al tamaño de las piezas. La solución encontrada (utilizar plástico HIPS en lugar de ABS) favorece la resistencia a impactos de las piezas. La carcasa ha recibido múltiples impactos durante las pruebas que se han realizado y no se han reportado daños.

Los aditamentos para la pelota son efectivos. Para el caso del Kicker, aunque el componente realizado con ABS es capaz de patear la pelota a alta velocidad,

CAPÍTULO 5: CONCLUSIONES Y TRABAJO FUTURO

se debería explorar con otros materiales que transfieran mejor la energía (como algún metal). Similarmente, en el caso del Dribbler, se pueden explorar otros materiales para mejorar su capacidad de mantener la posesión de la pelota.

El sistema de electrónica se encuentra modularizado por lo que ocupa mucho espacio. Si se quisiera liberar espacio (por ejemplo: para tener pilas que duren más), se debe hacer con un *circuito impreso*. Adicionalmente, esta solución facilitaría el armado de los robots y detección de fallas.

Para el sistema de cómputo, la utilización de una FPGA y un ATMEGA-AVR resulta muy adecuada principalmente por el número de entradas y salidas que se manejan en el sistema. Utilizando la FPGA principalmente para manejar entradas y salidas, le permite al ATMEGA-AVR realizar rápidamente el cómputo para determinar la siguiente señal de control. Actualmente, por la forma en que se mide la velocidad real del motor, la frecuencia del sistema es relativamente baja (20 Hz). Si se modifica el método para calcular la velocidad real de los motores, la frecuencia del sistema se podría acelerar considerablemente, reflejándose en un mejor desempeño del robot.

Una de las mayores limitantes que se tienen actualmente en el sistema es la determinación de las constantes k_p y k_i para ambos algoritmos de control implementados. Debido a que actualmente se determinan a *prueba y error*, no se puede garantizar obtener valores óptimos. La implementación de un método que determine éstas constantes de manera automática disminuiría el error que presenta el robot en su movimiento. Adicionalmente, se puede considerar implementar algoritmos de aprendizaje de máquina en el sistema que controla el robot para que desde el sistema se compense el error del robot.

GLOSARIO

ABS Acrilonitrilo Butadieno Estireno.

ATMEL-AVR Microcontrolador programable desarrollado por ATmel.

CAD Computer Aided Design.

Electronic Speed Controller Circuito especializado en el control de motores sin escobillas.

FPGA Field-Programmable Gate Array.

HIPS High Impact Polystyrene.

ITAM Instituto Tecnológico Autónomo de México.

MISO Master In Slave Out.

Modelado por Deposición Fundida Proceso de manufactura principalmente utilizado para prototipaje basado en depositar en capas el material por medio de un extrusor..

MOSI Master Out Slave In.

PWM Pulse Width Modulation.

ROS Robot Operating System.

SCK Serial Clock.

GLOSARIO

SPI Serial Peripheral Interface.

SS Slave Select.

SSL Small Size League.

Standard Pattern Patrón utilizado en la tapa superior de todos los robots para ser localizados por la visión global; consistente de 5 círculos distribuidos con medidas específicas y colores de acuerdo al equipo y número de robot.

UART Universal Asynchronous Receiver/Transmitter.

UDP User Datagram Protocol.

Wi-Fi Forlamente llamado IEEE 802.11, es un estandard de comunicación para redes locales inalámbricas.

WLAN Wireless Local Area Network.

REFERENCIAS

- 1/2" bore 32 pitch aluminum hub gears.* (2017). Descargado de https://www.servocity.com/32-pitch-50-bore-aluminum-hub-gears#.VdT0VrJ_0ko
- 4mm bore 32 pitch, 16t shaft mount pinion gear.* (2017). Descargado de <https://www.servocity.com/4mm-bore-32p-16t-shaft-mount-pinion-gear>
- Adachi, Y., Kusakabe, H., Yamanaka, Y., Ito, M., Murakami, K., y Naruse, T. (2016). *Robodragons 2015 extended team description.*
- Adăscăliței, F., y Doroftei, I. (2011). Practical applications for mobile robots based on mecanum wheels—a systematic survey. *Gh. Asachi Technical University of Iasi, Mechanical Engineering Faculty, Theory of Mechanisms and Robotics Department, B-dul D. Mangeron*, 61–63.
- Adhami-Mirhosseini, A., Bakhshande Babersad, O., Jamaati, H., Asadi, S., y Ganjali, A. (2012). Mrl extended team description 2012. En *Proceedings of the 15th international robocup symposium, mexico city, mexico.*
- Afro esc (Manual de software informático n.º 30A). (s.f.). Descargado de <https://hobbyking.com/media/file/745353836X1335174X28.pdf>
- Ahn, S.-H., Montero, M., Odell, D., Roundy, S., y Wright, P. K. (2002). Anisotropic material properties of fused deposition modeling abs. *Rapid prototyping journal*, 8(4), 248–257.
- Ashmore, M., y Barnes, N. (2002). Omni-drive robot motion on curved paths: The fastest path between two points is not a straight-line. En *Australian joint conference on artificial intelligence* (pp. 225–236).
- Bischoff, R. (2011). Kuka youbot—a milestone for education and research in mobile manipulation. En *Ieee icra workshop—a new generation of educational*

REFERENCIAS

- robots. kuka laboratories gmbh, shanghai international convention center, augsburg, china* (Vol. 76).
- Biswas, J., Mendoza, J. P., Zhu, D., Klee, S., y Veloso, M. (2013). Cmdragons 2014 team description.
- Borgolte, U., Hoyer, H., Bühler, C., Heck, H., y Hoelper, R. (1998). Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22(3-4), 233–253.
- Chaiso, K., y Sukvichai, K. (s.f.). *Skuba 2011 extended team description*.
- Committee, S. S. L. T. (2016, April). *Laws of the robocup small size league 2016*. http://wiki.robocup.org/images/1/18/Small_Size_League_-_Rules_2016.pdf.
- Corke, P. (2011). *Robotics, vision and control: Fundamental algorithms in matlab*. Springer Berlin Heidelberg. Descargado de <https://books.google.com.mx/books?id=6-kLBwAAQBAJ>
- Cox, J. (2006). *Electric motors*. Special Interest Model Books.
- Davis, P. G., Huynh, Q. B., y Peck Jr, J. C. (2013, noviembre 19). *Serial peripheral interface*. Google Patents. (US Patent 8,589,717)
- dual ball bearing hub*. (2017). Descargado de https://www.servocity.com/dual-ball-bearing-hub#.VdTN37J_0ko
- D’Andrea, R., Kalmar-Nagy, T., Ganguly, P., y Babish, M. (2000). The cornell robocup team. En *Robot soccer world cup* (pp. 41–51).
- Ec 45 flat 42.9 mm, brushless, 30 watt (Manual de software informático n.º 200142). (2012, 5). Descargado de <http://edge.rit.edu/edge/P13021/public/DDR/Maxon%20Generator%2030W.pdf>
- Egorova, A., Glove, A., Liers, A., Rojas, R., Schreiber, M., Simon, M., ... Wiesel, F. (2003). Fu-fighters 2003 (global vision). En *7th international workshop on robocup*.
- Er-force team description paper for robocup 2014. (s.f.).
- Ferrein, A., y Steinbauer, G. (2016). 20 years of robocup. *KI-Künstliche Intelligenz*, 30(3-4), 225–232.
- Gear, H., Edition, S., V3.3, S., V3.3, S., V3.2, S., (1A), S., ... Driver, S. e. a. (2017). *Hobby motor - gear - rob-11696 - sparkfun electronics*. Descargado de <https://www.sparkfun.com/products/11696>
- GmbH, N. (2016, 5). Omni-drive-module (Vol. 2.4.2) [Manual de software

REFERENCIAS

- informático].
- Hanselman, D. (2003). *Brushless permanent magnet motor design*. Writers' Collective.
- Houshangi, N., y Lippitt, T. (1999). Omnibot mobile base for hazardous environment. En *Electrical and computer engineering, 1999 IEEE Canadian conference on* (Vol. 3, pp. 1357–1361).
- Instruments, T. (2005). L293 quadruple half-h, drivers data sheet. *Dallas, Texas*.
- International, D. (2015). Xbee wi-fi [Manual de software informático].
- Lobmeier, C., Blank, P., Buehlmeier, J., Burk, D., Eischer, M., Hauck, A., ... Eskofier, B. M. (2016). Er-force 2016 extended team description.
- Mano, M., y Kime, C. (2007). *Logic and computer design fundamentals*. Prentice Hall.
- Maxon. (2014, 11). Dc ec key information [Manual de software informático].
- Mojo fpga development board (Manual de software informático n.º V3). (s.f.). Descargado de <https://embeddedmicro.com/products/mojo-v3.html>
- Nise, N. (2014). *Control systems engineering*. Wiley. Descargado de <https://books.google.com.mx/books?id=X6DalwEACAAJ>
- Poudeh, A. G., Esmaeelpourfard, S., Hosseinikia, A., Tafti, H., y Adhami-Mirhosseini, A. (2016). Mrl extended team description 2016. En *Proceedings of the 18th international robocup symposium*.
- Purwin, O., y D'Andrea, R. (2003). Cornell big red 2003. *Robocup*.
- Robocup-SSL. (2015). *Small size league*. <http://wiki.robocup.org/wiki/Small-Size-League.html>.
- Rodríguez, J. (2009, June). *Robotics architectures for the RoboCup small-size league* (Inf. Téc.). Mexico City, Mexico: Computer Engineering Dept, ITAM.
- Rojas, R. (2005). A short history of omnidirectional wheels. *See also URL* <http://robocup.mi.fu-berlin.de/buch/shortomni.pdf>.
- Rojas, R., y Förster, A. G. (2006). Holonomic control of a robot with an omnidirectional drive. *KI-Künstliche Intelligenz*, 20(2), 12–17.
- Ruiz, M. D. S. (2007). *Diseño experimental para el mejoramiento del subsistema de control de la pelota para los robots de la liga small size* (Inf. Téc.). Mexico City, Mexico: ITAM.

REFERENCIAS

- Ryll, A., Ommer, N., Geiger, M., Jauer, M., y Theis, J. (2014). Tigers mannheim.
- Salt, J. (2012). *Understanding rc lipo batteries*.
- Square-profile oil-resistant buna-n o-rings*. (2017). Descargado de <https://www.mcmaster.com/#catalog/123/3550/=17rqkey> (Wd.: 3/32 Fractional (0.103. Actual): 109)
- Tanenbaum, A. (2003). *Computer networks* (n.º p. 3). Prentice Hall PTR. Descargado de <https://books.google.com.mx/books?id=Pd-z64SJRBA>
- Torres, E. (2009, Abril). *Coordination strategies for multi-robot systems in RoboCup small-size league* (Inf. Téc.). Mexico City, Mexico: Computer Engineering Dept, ITAM.
- Trimberger, S. (1994). *Field-programmable gate array technology*. Springer US.
- Ulrich, K., y Eppinger, S. (2009). *Diseño y desarrollo de productos (4a. ed.)*. McGraw-Hill Interamericana. Descargado de <https://books.google.com.mx/books?id=g8xtAQAACAAJ>
- Valvano, J. (2012). *Embedded systems: Introduction to the arm® cortex(tm)-m microcontrollers*. Jonathan W. Valvano. Descargado de <https://books.google.com.mx/books?id=3MeBMAEACAAJ>
- Wright, P. (2001). *21st century manufacturing*. Prentice Hall.
- Xbee wi-fi (Manual de software informático n.º S6B). (2015). Descargado de https://www.digi.com/pdf/ds_xbeewifis6b.pdf
- Z-abs material data sheet [Manual de software informático]. (2014, 1 de 1). (Update 13.12.2016)
- Zhao, Y., Xiong, R., Tong, H., Li, C., y Fang, L. (2013). Zjunliet: Robocup 2013 small size league champion. En *Robot soccer world cup* (pp. 92–103).
- Z-hips material data sheet [Manual de software informático]. (2014, 12 de 12). (Update 20.09.2016)
- Zickler, S., Biswas, J., Luo, K., y Veloso, M. (2010). *Cmdragons 2010 team description*. Carnegie Mellon University.