

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**DEPARTAMENTO DE INGENIERÍA MECÁNICA**  
**VALPARAÍSO – CHILE**



# **DISEÑO Y FABRICACIÓN DE ROBOT QUE CUMPLA CON LAS REGLAS DE LA CATEGORÍA ROBOTRACER DE LA COMPETENCIA ALL JAPAN MICROMOUSE CONTEST**

**ANDRÉS RODRIGO BERCOVICH MERINO**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO MECÁNICO INDUSTRIAL**

Profesor Guía: Dr. Ing. Danilo Estay B  
Profesor Correferente: Mg. Ing. Felipe Chacana Y

Abril - 2019

## Resumen

Robotracer es una categoría de competición robótica, nacida en Japón el año 1986, cuya popularización ha llegado hasta Latinoamérica. Los robots que participan en esta categoría son en estricto rigor, vehículos autónomos en pequeña escala, que deben recorrer un circuito en el menor tiempo posible, cumpliendo con la reglamentación establecida.

El estudio de estos robots, propone un desafío mecatrónico en los desarrolladores, donde estos se enfrentan a complejos problemas de diseño, que van desde aspectos esencialmente físicos, como el modelamiento de la dinámica del robot, aspectos mecánicos como los sistemas de tracción, aspectos electrónicos y de ciencias computacionales, como la programación de microcontroladores y la comunicación con sensores y transductores.

Para abordar todas las temáticas planteadas anteriormente, esta memoria se separa en cinco capítulos. El primer capítulo, habla sobre el estado del arte de los Robotracer, donde se definirá qué es un Robotracer, y se mostrará la reglamentación que los gobierna. Finalizando se mostrarán y analizarán ligeramente los robots más influyentes de esta categoría hasta el 2017, a partir de la elaboración de un ranking de los últimos cuatro años de la competencia All Japan Micromouse Contest.

En el segundo capítulo se propondrá un diseño personalizado de un Robotracer, considerando aspectos mecánicos, eléctricos y electrónicos, cuyo estudio dinámico se realizará en el capítulo tres. El tercer capítulo describirá analíticamente la dinámica del robotracer propuesto anteriormente, definiendo cómo sería teóricamente el movimiento ideal, para que en el cuarto capítulo de esta memoria se proponga el algoritmo de movimiento del robot, para que se comporte de manera ideal. En el quinto capítulo se comentarán los resultados obtenidos, y se propondrán mejoras al modelo realizado.

## Abstract

Robotracer is a category of robotic competition, born in Japan in 1986, whose popularization has reached Latin America. The robots that participate in this category are in strict rigor, autonomous small-scale vehicles, that must travel a track in the shortest possible time, complying with the established regulations.

The study of these robots, proposed a mechatronic challenge in the developers, where they face complex design problems, ranging from essentially physical aspects, such as the modeling of robot dynamics, mechanical aspects such as traction systems, electronic aspects and computer science, such as microcontroller programming and communication with sensors and transducers.

To cover all the issues raised above, this memory is separated into five chapters. The first chapter, talks about the state of the art of Robotracer, where it will be defined what is a Robotracer, and will show the regulation that governs them. Finishing will show and analyze slightly the most influential robots of this category until 2017, from the development of a ranking of the last four years of the competition "All Japan Robotracer Contest".

In the second chapter a custom design of a Robotracer will be proposed, considering mechanical, electrical and electronic aspects, whose dynamic study will be carried out in chapter three.

The third chapter will describe analytically the dynamics of the robotracer previously proposed, defining how the ideal movement would theoretically be, so that in the fourth chapter of this memory the algorithm of movement of the robot is proposed, so that it behaves in an ideal way. In the fifth chapter, the results obtained will be discussed, and improvements to the model made will be proposed.

## Glosario

**AJMMC:** All Japan Micromouse Contest.

**NTF:** New Technology Foundation.

**VCC:** Voltaje de corriente continua positiva

**RPM:** Revoluciones por minuto

**[m]:** Metros

**[mm]:** Milímetros

**[s]:** Segundo

**[m/s]:** Metros por Segundo

**ARM:** Advanced RISC Machine

**MHz:** Mega Hertz

**CPU:** Central Processesing Unit

**PWM:** Pulse with Modulation

**USB:** Universal Serial Bus

**UART:** Universal Asynchronous Receiver - Transmitter

**SPI:** Serial Peripheral Interface

**I2C:** Inter Integrated Circuits

**CAN Bus:** Controller Area Network

**[V]:** Volt

**$v$ :** Velocidad

**$a$ :** Aceleración

**CW:** Clockwise

**CCW:** Counter-Clockwise

$m$ : Masa

$v_f$ : Velocidad Final

$v_i$ : Velocidad Inicial

**[A]:** Ampere

$x$ : Distancia

$a$ : Aceleración

$F_r$ : Fuerza Fricción

$F_c$ : Fuerza Centrífuga

$a_c$ : Aceleración Centrífuga

## Índice de contenidos

|  |    |
|--|----|
| Resumen  | 1  |
| Glosario   | 3  |
| 1. Introducción                                  | 11 |
| 2. Objetivos                                     | 14 |
| 2.1. Objetivos Generales                         | 14 |
| 2.2. Objetivos Específicos                       | 14 |
| 3. Estado del arte de los Robotracer             | 15 |
| 3.1. Definición de Robotracer                    | 16 |
| 3.2. Reglamentación de categoría Robotracer      | 17 |
| 3.2.1. Reglas para los Robotracer:               | 17 |
| 3.2.2. Reglas para los escenarios:               | 17 |
| 3.2.3. Reglas de la competencia                  | 20 |
| 3.3. Robotracer en Asia                          | 24 |
| 3.4. Robotracer en Latinoamérica                 | 26 |
| 3.5. Ranking de la categoría                     | 27 |
| 3.5.1. Año 2014                                  | 29 |
| 3.5.2. Año 2015                                  | 31 |
| 3.5.3. Año 2016                                  | 33 |
| 3.5.4. Año 2017                                  | 35 |
| 3.5.5. Mejores robots históricos de la categoría | 37 |
| 4. Proposición de diseño personalizado           | 42 |
| 4.1. Hardware Mecánico                           | 43 |
| 4.2. Hardware Electrónico                        | 47 |

|  |    |
|--|----|
| 4.2.1. Subsistema de Potencia:   | 47 |
| 4.2.2. Subsistema de Control:  | 48 |
| 4.2.3. Subsistema de Sensores:   | 50 |
| 4.2.4. Subsistema de Periféricos:                                      | 54 |
| 4.3. Vista general del diseño:   | 55 |
| 4.4. Consideraciones del ensamble completo                             | 56 |
| 5. Descripción Analítica de un Robotracer                              | 59 |
| 5.1. Movimiento Ideal  | 60 |
| 5.2. Movimiento Rectilíneo Uniformemente acelerado:                    | 64 |
| 5.3. Movimiento Curvilíneo Uniforme                                    | 65 |
| 5.4. Mapeo del escenario   | 67 |
| 5.5. Optimización del perfil de velocidades                            | 69 |
| 6. Diseño del algoritmo de control del movimiento                      | 72 |
| 6.1. Consideraciones sobre la programación y el Entorno de Desarrollo. | 73 |
| 6.2. Generación del movimiento   | 73 |
| 6.3. Control de Posición Angular                                       | 76 |
| 6.4. Control de Velocidad Traslacional                                 | 82 |
| 6.5. Combinación de movimientos  | 85 |
| 6.6. Optimización del perfil de velocidades                            | 86 |
| 6.7. Comentarios del diseño y mejoras                                  | 90 |
| 7. Resultados obtenidos  | 92 |
| 8. Conclusiones  | 97 |
| 11. Referencias  | 99 |
| 12. Anexos   | 99 |

## Índice de figuras

|  |           |
|--|-----------|
| <b>Figura 3.1, Intersección de líneas</b>  | <b>17</b> |
| <b>Figura 3.2, Marcas de Inicio y Final</b>  | <b>17</b> |
| <b>Figura 3.3, Área de Meta</b>  | <b>18</b> |
| <b>Figura 3.4, Dimensiones de Checkpoint</b>   | <b>18</b> |
| <b>Figura 3.5, Marcas de cambio de curvatura</b>                                     | <b>18</b> |
| <b>Figura 3.6, Escenario, Taiwan Micromouse &amp; Intelligent Robot Contest 2014</b> | <b>22</b> |
| <b>Figura 3.7, Escenario, AJMMC 2013</b>   | <b>23</b> |
| <b>Figura 3.8, Pista, Etapa preliminar 2014</b>                                      | <b>27</b> |
| <b>Figura 3.10, Pista, Etapa preliminar 2015</b>                                     | <b>29</b> |
| <b>Figura 3.11, Pista, Etapa final 2015</b>  | <b>30</b> |
| <b>Figura 3.12, Pista, Etapa preliminar 2016</b>                                     | <b>31</b> |
| <b>Figura 3.13, Pista, Etapa final 2016</b>  | <b>32</b> |
| <b>Figura 3.14, Pista, Etapa preliminar 2017</b>                                     | <b>33</b> |
| <b>Figura 3.15, Pista, Etapa final 2017</b>  | <b>34</b> |
| <b>Figura 3.16, Cartis04,5 Primer lugar, ranking 2014-2017</b>                       | <b>35</b> |
| <b>Figura 3.17, Ning3, Segundo ranking 2014-2017</b>                                 | <b>36</b> |
| <b>Figura 3.18, Diligent – 4, Tercer lugar, ranking 2014-2017.</b>                   | <b>37</b> |
| <b>Figura 3.19, Slug_III, Cuarto lugar, ranking 2014-2017</b>                        | <b>38</b> |
| <b>Figura 3.20, RS-100, Quinto lugar, ranking 2014-2017</b>                          | <b>39</b> |
| <b>Figura 4.1, Vista isométrica, engranaje-rueda-motor</b>                           | <b>41</b> |
| <b>Figura 4.2, Vista frontal, engranaje-rueda-motor</b>                              | <b>41</b> |
| <b>Figura 4.3, Ensamble Rueda</b>  | <b>42</b> |
| <b>Figura 4.4, Engranaje del sistema de tracción</b>                                 | <b>43</b> |
| <b>Figura 4.5, Detalle, Imán para Encoder</b>  | <b>43</b> |
| <b>Figura 4.6, Ensamble sistema de tracción</b>                                      | <b>42</b> |
| <b>Figura 4.7, Trompa (arreglo de sensores), vista inferior</b>                      | <b>43</b> |
| <b>Figura 4.8, Trompa (arreglo de sensores), vista superior</b>                      | <b>43</b> |



|  |    |
|--|----|
| <b>Figura 4.9, Cuerpo vista superior</b>                         | 44 |
| <b>Figura 4.10, Cuerpo vista inferior</b>                        | 44 |
| <b>Figura 4.11, Sensor lateral izquierdo, Vista superior</b>     | 44 |
| <b>Figura 4.12, Sensor lateral izquierdo, Vista Inferior</b>     | 44 |
| <b>Figura 4.13, Sensor lateral derecho, Vista superior</b>       | 44 |
| <b>Figura 4.14, Sensor lateral derecho, Vista Inferior</b>       | 44 |
| <b>Figura 4.15, Encoders, vista superior e inferior</b>          | 45 |
| <b>Figura 4.16, Dimensiones Motores Portescap 13n88</b>          | 45 |
| <b>Figura 4.17, Circuito de potencia</b>                         | 47 |
| <b>Figura 4.18, Ruteo, circuito subsistema de control</b>        | 48 |
| <b>Figura 4.19, Unidad de procesamiento, Teensy 3.2</b>          | 49 |
| <b>Figura 4.20, Driver para motor, Toshiba TB6612FNG</b>         | 49 |
| <b>Figura 4.21, Conectores para los módulos</b>                  | 50 |
| <b>Figura 4.22, Esquemático, sensor frontal (trompa)</b>         | 51 |
| <b>Figura 4.23, Layout , Sensores frontales (trompa)</b>         | 51 |
| <b>Figura 4.24, Esquemático, Encoders MA700</b>                  | 52 |
| <b>Figura 4.25, Layout Circuito, Encoders MA700</b>              | 52 |
| <b>Figura 4.26, Esquemático, Sensor Lateral Izquierdo</b>        | 53 |
| <b>Figura 4.27, Esquemático, Sensor Lateral Derecho</b>          | 53 |
| <b>Figura 4.28, Layout, Sensores Laterales</b>                   | 53 |
| <b>Figura 4.29, Layout Giroscopio MPU-9250</b>                   | 54 |
| <b>Figura 4.30, Esquemático, subsistema de periféricos</b>       | 54 |
| <b>Figura 4.31, Dimensiones de Ensamble completo</b>             | 55 |
| <b>Figura 4.32, Robot, construido a tamaño real</b>              | 55 |
| <b>Figura 4.33, Unión Cuerpo - Sensor Frontal</b>                | 56 |
| <b>Figura 4.34, Unión Cuerpo - Sensor Lateral</b>                | 56 |
| <b>Figura 4.35, Distancia entre ruedas</b>                       | 57 |
| <b>Figura 4.36, Sistema engranajes: Motor - Rueda - Encoder</b>  | 57 |
| <b>Figura 5.1, Escenario, Robotracer, Preliminar, AJMMC 2016</b> | 60 |
| <b>Figura 5.2, Robotracer en punto de partida</b>                | 61 |
| <b>Figura 5.3, Ejemplo de rapidez durante el primer intento</b>  | 64 |

|   |    |
|---|----|
| <b>Figura 5.4, Partícula en movimiento circular uniforme</b>      | 66 |
| <b>Figura 5.5, Distancia recorrida por cada rueda en un tramo</b> | 67 |
| <b>Figura 6.1, Esquemático, Módulo Driver TB6612FNG</b>           | 73 |
| <b>Figura 6.2, Alineación Sensor Frontal</b>                      | 79 |
| <b>Figura 6.3, Diagrama de control PD, posición angular</b>       | 80 |
| <b>Figura 6.4, Ensamble imán - Encoder</b>                        | 82 |
| <b>Figura 7.1, Gráfico de velocidad vs Tiempo, pista prueba</b>   | 96 |
| <b>Figura 7.2, Atajos en pista de prueba</b>                      | 96 |

## Índice de tablas

|  |           |
|--|-----------|
| <b>Tabla 3.1, Ranking, periodo 2014 -2017</b>                                | <b>26</b> |
| <b>Tabla 3.2, Ranking, Etapa preliminar 2014</b>                             | <b>27</b> |
| <b>Tabla 3.3, Etapa final 2014</b>   | <b>28</b> |
| <b>Tabla 3.4, Ranking, Etapa preliminar 2015</b>                             | <b>29</b> |
| <b>Tabla 3.5, Ranking, Etapa final 2015</b>                                  | <b>30</b> |
| <b>Tabla 3.6, Ranking, Etapa preliminar 2016</b>                             | <b>31</b> |
| <b>Tabla 3.7, Ranking, Etapa final 2016</b>                                  | <b>32</b> |
| <b>Tabla 3.8, Ranking, Etapa preliminar 2017</b>                             | <b>33</b> |
| <b>Tabla 3.9, Ranking, Etapa final 2017</b>                                  | <b>34</b> |
| <b>Tabla 3.10, Especificaciones, Cartis04,5, Documentación NTF 2017</b>      | <b>35</b> |
| <b>Tabla 3.11, Especificaciones, Ning3, Documentación NTF 2017</b>           | <b>35</b> |
| <b>Tabla 3.12, Especificaciones, Diligent - 4, Documentación NTF 2017</b>    | <b>36</b> |
| <b>Tabla 3.13, Especificaciones, Slug_III, Documentación NTF 2017</b>        | <b>37</b> |
| <b>Tabla 3.14, Especificaciones, RS-100, Documentación NTF 2017</b>          | <b>38</b> |
| <b>Tabla 4.1, Especificaciones, Motor Portescap 13n88</b>                    | <b>46</b> |
| <b>Tabla 6.1, Control, Driver TB6612FNG</b>                                  | <b>74</b> |
| <b>Tabla 7.1, Valores primera vuelta, escenario de prueba</b>                | <b>94</b> |
| <b>Tabla 7.2, Optimización de perfil de velocidades, escenario de prueba</b> | <b>95</b> |

## 1. Introducción

La Robótica y Mecatrónica, son quizás de todas las ramas de la ingeniería, las que contemplan la mayor constelación de connotaciones pluridimensionales. Esto se debe a que sus estructuras, se conforman a partir de la relación simbiótica y sinérgica de las ramas basales de la ingeniería, dando espacio para una infinidad de aplicaciones en una infinidad de áreas distintas.

La mecatrónica no es un área nueva de la ingeniería, sino una filosofía de diseño que busca mejorar la calidad de los productos y sistemas. En 1969, Tetsuro Mori, ingeniero de la empresa japonesa Yaskawa Electric Corporation, acuñó el término “Mecatrónica” como un concepto para enfatizar la necesidad de la integración entre diferentes áreas de la ingeniería.

En un principio, esta integración consistía sólo en las áreas de mecánica y electrónica; En 1971 se le otorga el derecho de marca a la empresa, y a partir de 1982, Yaskawa permite el libre uso del término.

La palabra mecatrónica ha sido definida de varias maneras, no obstante, un consenso común es describir a la mecatrónica como una disciplina integradora de las áreas de mecánica, electrónica e informática cuyo objetivo es proporcionar mejores productos, procesos y sistemas.

De acuerdo con lo anterior se hace referencia a la definición de mecatrónica propuesta por Rietdijk en 1989: Mecatrónica es la **combinación sinérgica** de la ingeniería mecánica de

precisión, de la electrónica, del control automático y la ingeniería de sistemas, para el diseño de mejores productos y procesos.<sup>1</sup>

Por su parte, la robótica lo que busca es la sustitución de la mano de obra humana, a partir de los fundamentos mecatrónicos.

Las ventajas del desarrollo e implementación de la robótica y mecatrónica, son evidentes; entre las más importantes, destacan: Mayor precisión y calidad sin cansancio, realización de tareas peligrosas, mayor velocidad, reducción de costos, junto con proteger al humano e ir donde éste no puede. También cabe destacar que el desarrollo de la Robótica y Mecatrónica son unas de las áreas que generan mayor valor agregado a nivel mundial.

La dificultad del desarrollo de la Robótica y Mecatrónica, radica en que requieren de conocimiento hábil de las bases ingenieriles antes mencionadas. Por ejemplo, para poder hacer un artículo mecatrónico, como una “Impresora”, se requiere de conocimiento hábil, tanto de la “Informática” que generará la interfaz virtual “Hombre - Máquina”; de la “Mecánica”, que se deberá encargar del mecanismo de extracción y manipulación de las hojas y la tinta; de la “Electrónica”, que se encargará de automatizar los motores a partir de la información recogida de los Encoder y de la digitalización del documento generado por la interfaz informática; del “Diseño” que se encargará de que todo lo anterior quede encapsulado en un elemento estética y ergonómicamente correcto, y como si fuera poco, de habilidades comerciales, para poder hacer de éste trabajo, un elemento económicamente sustentable.

En este sentido, debido a la dificultad de la enseñanza de la robótica y mecatrónica, países como Japón, han fomentado la especialización en estas áreas, complementando los estudios

---

<sup>1</sup> CONFERENCIA, Mechatronics in Products and Manufacturing (11ª, 1989, Lancaster University, UK). J. A. Rietdijk, Ten propositions on mechatronics).

universitarios, con eventos de competición robótica, donde los desarrolladores pueden entrenarse, para tener un mejor dominio sobre el tema. Estos eventos de competición robótica, se han popularizado y a través del tiempo, se han ido estandarizando, generándose categorías de competición robótica cuyo desafío está reglamentado y homologado. A pesar de que países como Japón y Taiwán, han sido los primeros en homologar estas categorías, en Latinoamérica desde el año 2016, ya se pueden encontrar competencias robóticas homólogas a las asiáticas.

Dentro de las categorías robóticas más populares, la categoría Robotracer es una de las más documentadas, y cuya reglamentación en las competencias robóticas japonesa, posee más de 30 años de antigüedad, convirtiéndose en una de las más tradicionales.

Por esta razón, dado que en los últimos años las competencias robóticas han comenzado a proliferar, y junto con ellas, la cantidad de competidores interesados en la categoría Robotracer, es que esta memoria pretende documentar el proceso de diseño análisis y construcción de un Robotracer.

## **2. Objetivos**

### **2.1. Objetivos Generales**

El objetivo de esta memoria es diseñar, construir y programar un robotracer que cumpla con las especificaciones de la competencia AJMMC. Comenzando por entender la reglamentación que los regula, y los diseños más influyentes en base al ranking de los últimos años de la competencia.

Luego, se propondrá un diseño personalizado, y se hará un breve análisis dinámico de las variables que involucran a los Robotracer, describiendo cuál sería su movimiento ideal, para finalmente proponer un algoritmo de control de movimiento para el robot propuesto.

Los resultados obtenidos se comentarán, y se propondrán mejoras para una futura versión.

### **2.2. Objetivos Específicos**

- Describir el estado del arte de los Robotracer, y su contexto a nivel latinoamericano y asiático.
- Proponer un diseño personalizado de robot.
- Describir analíticamente la dinámica del robot, e identificar las ecuaciones más relevantes que lo gobierna.
- Diseñar el algoritmo de control para el movimiento del robot.
- Testear el robot, posteriormente a su construcción.

### **3. Estado del arte de los Robotracer**

Las competencias Robotracer, se han realizado en el mundo por más de 30 años, logrando recolectar una gran cantidad de información respecto a las técnicas y tecnologías utilizadas para desarrollar los prototipos cada vez más rápidos.

Esta memoria no solo intentará hacer un Robotracer que cumpla con las especificaciones de la reglamentación de la categoría, sino, que intentará proponer una alternativa competitiva a los prototipos que ya existen.

Para esto, se realizará un breve estudio de los competidores históricos de la AJMMC, para enfocar el análisis en los diseños más influyentes de la categoría.

La competencia “AJMMC”, es la competencia robótica vigente, más antigua del mundo, y la categoría Robotracer, ha sido implementada desde el año 1987. Esta competencia posee registros de libre acceso, desde el año 1980, no obstante, los registros detallados sobre los escenarios se encuentran desde el año 2014 en adelante. Por lo tanto, para este estudio, se tomarán los datos desde el año 2014 hasta el 2017, obteniendo como resultado, los competidores más influyentes de la categoría



### 3.1. Definición de Robotracer

Robotracer es una categoría de competición robótica. También se le llaman “Robotracers” a los robots que participan en la categoría “Robotracer”. Un Robotracer (robot), es un vehículo autónomo en pequeña escala, cuyo objetivo es recorrer un escenario establecido bajo reglas estandarizadas, en el menor tiempo posible.

Para cumplir este objetivo, el Robotracer (robot), tiene tres oportunidades para recorrer el escenario, donde se suele utilizar la primera chance para adquirir la mayor información posible del escenario, y de esta forma en las dos oportunidades posteriores, optimizar el recorrido del Robotracer, y disminuir el tiempo en que completa el escenario.

La categoría Robotracer se inicia por primera vez en Japón en el torneo de robótica “All Japan Micromouse Contest<sup>2</sup>”, el año 1986, y fue en esta competencia donde se estableció la estandarización y homologación reglamentaria de esta categoría; por lo tanto, cuando se refiere a las reglas de la categoría, se hace referencia a las reglas establecidas por la New Technology Foundation<sup>3</sup>, entidad organizadora de la competencia “AJMMC”.

Análogamente, se reconoce como “Robotracer”, todas aquellas competencias que cumplan con la reglamentación definida por la New Technology Foundation.

---

<sup>2</sup> All Japan Micromouse Contest, es una competencia que agrupa a los mejores desarrolladores de robótica japoneses. <http://www.ntf.or.jp/mouse/micromouse2018/>

<sup>3</sup> New Technology Foundation (NTF) es la entidad organizadora de la competencia All Japan Micromouse Contest <http://www.ntf.or.jp>

### **3.2. Reglamentación de categoría Robotracer**

Como se mencionó anteriormente, la reglamentación de la categoría Robotracer es establecida por la NTF, cuya descripción se detalla a continuación:

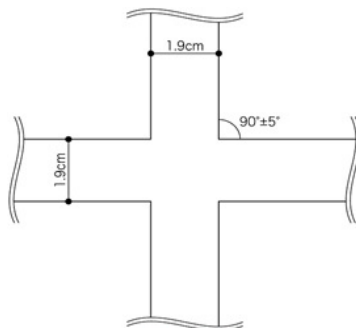
#### **3.2.1. Reglas para los Robotracer:**

- a. Un Robotracer debe ser autónomo. Excepto por la zona de inicio, el concursante no puede mantener ningún tipo de comunicación o control sobre el robot.
- b. Un Robotracer no debe tener ningún hardware o software agregado, eliminado, reemplazado o modificado por el operador durante la competencia. Sin embargo, está permitido hacer reparaciones menores.
- c. El tamaño del Robotracer no debe sobrepasar los 25 cm de largo, 25 cm de ancho y 20 cm de alto.
- d. El Robotracer no debe estar equipado con un mecanismo de succión para aumentar su agarre.

#### **3.2.2. Reglas para los escenarios:**

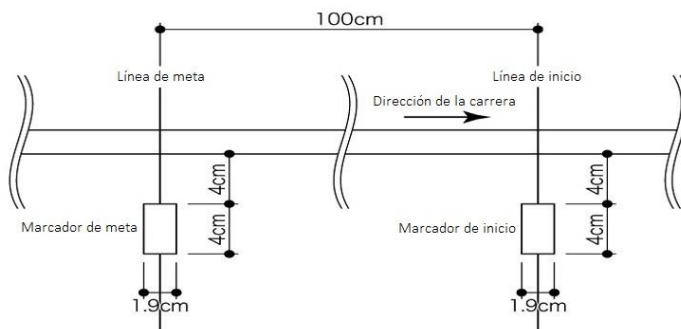
- e. La superficie por la cual se deberá mover el Robotracer, será de color negro mate, y el camino será delineado con una línea blanca de 19 milímetros de ancho. La máxima longitud que puede tener el camino es de 60 metros.
- f. El camino está compuesto por líneas rectas y por arcos. El camino puede tener cruces.
- g. El radio mínimo para cada arco es de 10 centímetros. La distancia entre dos cambios de curvatura consecutivos será de a lo menos 10 centímetros.

- h. Cuando las líneas se crucen, el ángulo de intersección será de  $90 \pm 5^\circ$  (Ver figura 3.1).  
Antes y después del cruce habrá a lo menos 25 cm de línea recta.



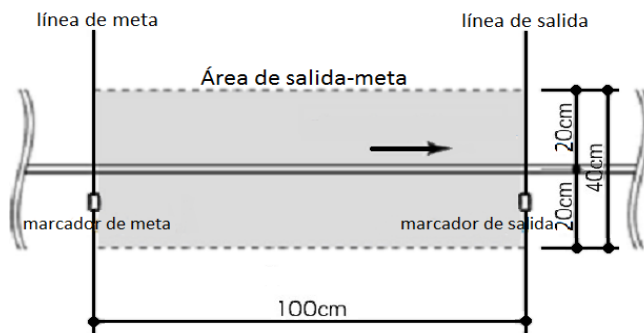
**Figura 3.1, Intersección de líneas**

- i. Las líneas de inicio y final estarán en una sección recta. La línea de finalización estará dispuesta 1 metro antes que la línea de inicio. Las marcas de inicio y fin estarán situadas en el lado derecho del recorrido, (Ver figura 3.2).

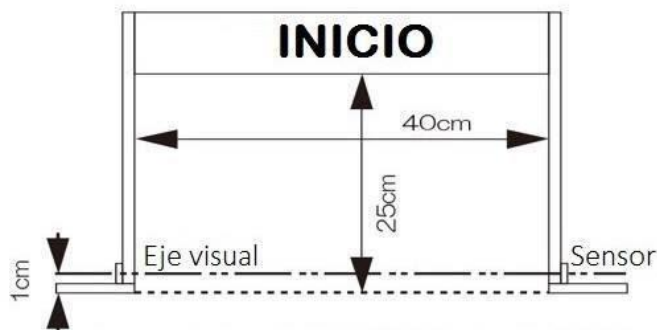


**Figura 3.2, Marcas de Inicio y Final**

- j. Se denominará “Área de Meta” aquella que comprenderá 20 [cm] a la derecha e izquierda de sus líneas de inicio y fin. Junto con lo anterior el ancho del checkpoint será de 40 [cm] y su altura de 25 [cm], (Ver figura 3.3 y 3.4).

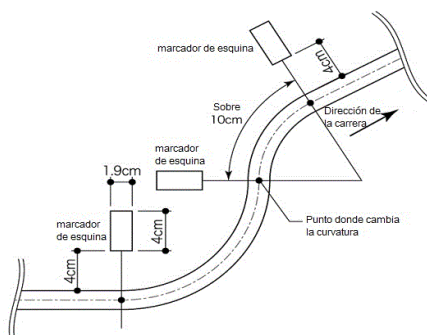


**Figura 3.3, Área de Meta**



**Figura 3.4, Dimensiones de Checkpoint**

- k. Antes y después de las líneas de inicio y fin habrán al menos 25 [cm] de línea recta.
- l. Cuando existan cambios de curvatura, se dispondrán marcadores en las esquinas en el lado izquierdo de la dirección de recorrido. (Ver figura 3.5).



**Figura 3.5, Marcas de cambio de curvatura**

- m. La superficie será plana, existiendo un peralte máximo de 5° aproximadamente.

### **3.2.3. Reglas de la competencia**

1. La proyección del Robotracer con respecto al escenario debe estar siempre sobre la línea a seguir. Si sale por completo se considerará como vuelta inválida.
2. Cada competidor tiene 3 minutos y un máximo de 3 intentos.
3. Por cada intento, el Robotracer debe partir desde el área de meta en la dirección especificada.
4. Después de terminar, el Robotracer debe detenerse por al menos 2 segundos en el área de meta.
5. El recorrido más rápido será considerado como el tiempo oficial.
6. El tiempo de la vuelta empezará a correr cuando el sensor de inicio detecte al Robotracer y terminará cuando el sensor de finalización lo detecte, además, el cuerpo entero del Robotracer debe estar dentro del área de meta, si esto no sucede el tiempo no será medido.
7. Si el Robotracer se apaga o se queda quieto por más de 2 segundos mientras compite, se considerará como vuelta inválida.
8. Luego de que se muestre el escenario a los competidores, estos no podrán cargar ningún tipo de información de este a los Robotracers. Durante la ronda, el operador no tendrá permitido cambiar información sobre el recorrido usando un switch o algo de ese tipo.

9. Los competidores tienen prohibido tocar a los Robotracers a menos que tengan la autorización de un árbitro.
10. El árbitro tiene el derecho de preguntar al competidor sobre su Robotracer, también puede detener una vuelta, declarar descalificación, o pedir instrucciones.
11. Los premios y los criterios de evaluación serán establecidos individualmente en cada competencia.
12. Cada competencia posee una etapa preliminar y una final. Todos los participantes compiten en la preliminar. Se obtienen los tiempos de cada competidor en la etapa preliminar, y pasan a la final los 30 mejores participantes de la fase preliminar.
13. El orden de los lugares es a partir del tiempo que se demoren en completar el trayecto (Tiempo Oficial), llevándose el primer lugar, el prototipo que demore menos en completar la etapa final.

## **Observaciones**

14. El competidor tiene prohibido cargar programas o reemplazar la memoria ROM durante la competencia, tampoco puede enviar ningún tipo de información al Robotracer a través de consolas de desarrollo UNIX u otro programador independiente de la unidad Robotracer.

15. El competidor tiene permitido limpiar el polvo de las ruedas del Robotracer usando cinta adhesiva o similar, sin embargo, tiene prohibido el uso de solventes químicos que pueden ser usados con el objetivo de aumentar la fricción.
16. Después de dar como iniciada una vuelta, si el Robotracer no puede pasar de la línea de inicio, igualmente será considerada como una vuelta.
17. Después de que el Robotracer haya completado la vuelta, este debe detenerse automáticamente, si no lo hace, la vuelta será considerada como inválida.
18. Exceptuando la carrera, los competidores tienen prohibido posicionar sus Robotracer fuera del área de meta con el propósito de ajustar parámetros.
19. En algunos casos, los arcos de curvatura variables aparecen continuamente (ver Figura 3.5).
20. El trayecto está hecho de madera terciada o contrachapado, pintada de negro mate. Las líneas están hechas de cinta blanca 3M de 19 [mm] de ancho. El escenario está construido con una precisión humana, por lo que puede haber errores en las dimensiones en cierto punto. Por lo cual podría haber diferencias en distancia o nivel de aproximadamente 5 [mm] a lo sumo. No se aceptan reclamos sobre el agarre de la superficie.
21. Los lugares donde se ubican los sensores de inicio y final están indicados en las figuras 3.3 y 3.4. Estos son del tipo infrarrojo y su eje óptico se ubica 1 [cm] por encima del piso.





### 3.3. Robotracer en Asia

A partir del año 1986, las competencias de “vehículo o robot autónomos” se fueron popularizando fuera de Japón, no obstante, no todas cumplían la reglamentación de la New Technology Foundation, por lo que estrictamente no se les puede llamar Robotracer.

Por lo general, cada competencia diseñaba su propia reglamentación tanto para los robots, los escenarios y modo de competencia.

No fue hasta mediados del 2014, donde la Lunghwa University of Science and Technology, de Taiwán, a través de su competencia “Taiwan Micromouse & Intelligent Robot Contest” homologó la reglamentación de la competencia con la NTF, generando por primera vez un circuito internacional de esta categoría.

En la figura 3.6 se puede apreciar un escenario de la competencia taiwanesa, en comparación a un escenario japonés, figura 3.7.



**Figura 3.6, Escenario, Taiwan Micromouse & Intelligent Robot Contest 2014**



**Figura 3.7, Escenario, AJMMC 2013**

Actualmente, la AJMMC, cada año convoca en promedio más de 400<sup>4</sup> competidores activos

---

<sup>4</sup> Documento estadístico de la All Japan Micromouse Contest para el año 2017.  
[http://www.ntf.or.jp/mouse/micromouse2017/MM2017recode\\_v101.pdf](http://www.ntf.or.jp/mouse/micromouse2017/MM2017recode_v101.pdf)

### 3.4. Robotracer en Latinoamérica

Como resultado de la propagación de este tipo de competencias alrededor del mundo, llegaron a Latinoamérica las primeras competencias de vehículos seguidores de líneas, pero con reglamentación más ligera. En su gran mayoría, los escenarios eran más sencillos, y el objetivo solo era seguir la línea central.

Desde el año 2016 a la fecha, se ha desarrollado en Chile, la categoría Robotracer, en la “Competencia Robótica<sup>5</sup>”, además desde el año 2016, se han iniciado conversaciones entre organizadores latinoamericanos y japoneses para homologar la reglamentación de esta categoría<sup>6</sup>, convirtiéndola en la categoría robótica con más adherentes en el mundo.

---

<sup>5</sup> Competencia robótica chilena, con más de 15 años de trayectoria. <http://competenciarobotica.cl>

<sup>6</sup> Artículo sobre las negociaciones, diario Las Últimas Noticias, Chile. <http://www.lun.com/Pages/NewsDetail.aspx?dt=2017-12-21&PaginaId=22&bodyid=0>

### 3.5. Ranking de la categoría

Como una manera de enfocar el estudio hacia los modelos más relevantes en los que estará basada la propuesta de diseño, se analizarán la información de la competencia AJMMC, y se elaborará un ranking con los datos obtenidos entre el 2014 y el 2017.

Los datos publicados por la NTF poseen información como las dimensiones de los escenarios y los tiempos que demoraron cada uno de los participantes.

Arbitrariamente se hará una selección de los prototipos más influyentes entre los años 2014 y 2017, bajo los siguientes criterios.

Tratamiento de datos:

- Se hace una lista, de los mejores 10 participantes, de cada etapa preliminar y cada etapa final desde el año 2014 hasta el año 2017.
- Se les asigna puntaje a los desarrolladores, desde 1 a 10 puntos, obteniendo 10 puntos el primer lugar y sucesivamente, el lugar 10 se le asigna 1 punto.
- Se hace un recuento de puntos al final del periodo de medición.
- Finalmente, se hace un ranking con competidores con mayor puntaje acumulado en el periodo de evaluación.

Además, se hace una recopilación con las velocidades promedio para cada año, de los 10 primeros lugares tanto en la fase preliminar como la final.

El ranking mostrado en la tabla en la tabla 3.1, muestra una tendencia clara a obtener los primeros lugares a un conjunto de participantes.

**Tabla 3.1, Ranking, periodo 2014 -2017<sup>7</sup>**

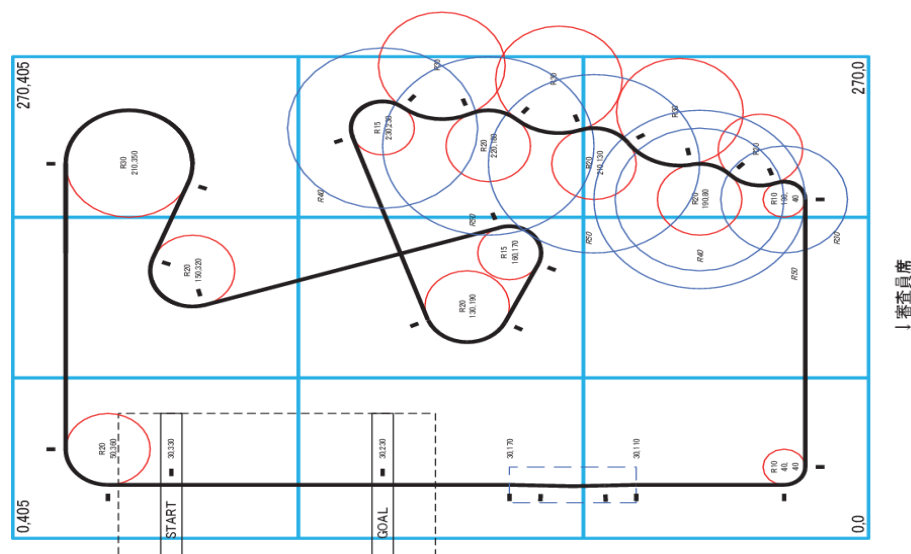
| Ranking | Name           | País     | 2014P | 2014F | 2015P | 2015F | 2016P | 2016F | 2017P | 2017F | Total |
|---------|----------------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1       | Hirai Masataka | Japón    | 10    | 10    | 10    | 10    | 10    | 10    | 9     | 9     | 78    |
| 2       | Ng Beng Kiat   | Singapur | 9     | 5     | 9     | 0     | 0     | 0     | 8     | 8     | 39    |
| 3       | Meng Weilin    | Singapur | 7     | 7     | 5     | 0     | 6     | 6     | 3     | 4     | 38    |
| 4       | Chen, Chao-Wei | Taiwan   | 3     | 0     | 4     | 8     | 4     | 9     | 0     | 0     | 28    |
| 5       | Endono         | Japón    | 0     | 0     | 0     | 0     | 8     | 0     | 10    | 10    | 28    |
| 6       | Ba-Xin Lin     | Taiwan   | 0     | 0     | 6     | 0     | 9     | 8     | 0     | 0     | 23    |
| 7       | Shimura Sōbo   | Japón    | 0     | 0     | 8     | 0     | 5     | 0     | 5     | 0     | 18    |
| 8       | Jun-Guang, Li  | Taiwan   | 0     | 0     | 0     | 0     | 9     | 8     | 0     | 0     | 17    |
| 9       | Ho, Tsung-Chun | Taiwan   | 8     | 8     | 0     | 0     | 0     | 0     | 0     | 0     | 16    |
| 10      | Liao, Huan-Jie | Taiwan   | 8     | 8     | 0     | 0     | 0     | 0     | 0     | 0     | 16    |

En las siguientes tablas se mostrarán los resultados de las competencias desde el año 2014 hasta el 2017, para dimensionar en perspectiva la rapidez promedio de los participantes.

<sup>7</sup> Debido a las limitaciones de idioma, algunos nombres originalmente escritos en japonés, chino, coreano o singapurense, se han traducido fonéticamente al castellano.

### 3.5.1. Año 2014

Etapa preliminar, Longitud total: 14,8 [m] (Ver figura 3.8).



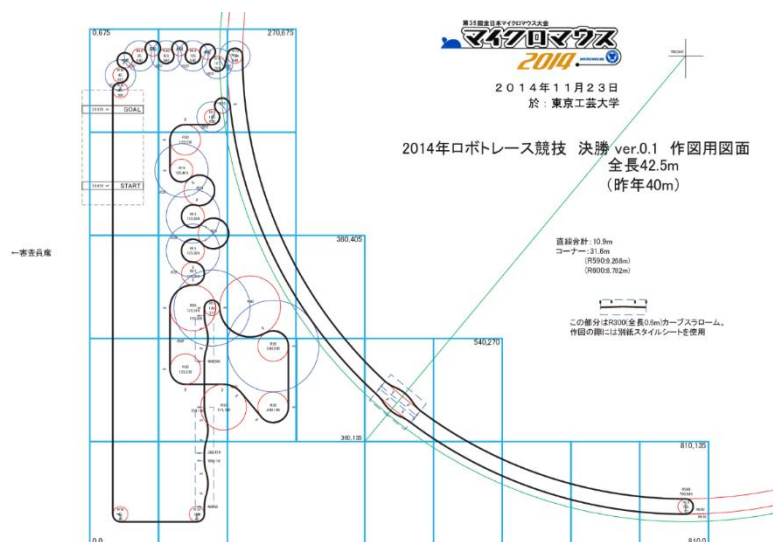
**Figura 3.8, Pista, Etapa preliminar 2014**

Cuyo resultado se muestra en la tabla 3.2.

**Tabla 3.2, Ranking, Etapa preliminar 2014**

| Lugar | Robot         | Desarrollador   | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|---------------|-----------------|------------|--------------|---------|
| 1     | CartisX04     | Hirai Masataka  | 5,761      | 2,569        | 10      |
| 2     | Ning3         | Ng Beng Kiat    | 6,614      | 2,238        | 9       |
| 3     | Dao-Huei      | Ho, Tsung-Chun  | 6,857      | 2,158        | 8       |
| 4     | Diligent      | meng weilin     | 6,915      | 2,140        | 7       |
| 5     | First Aim     | Sho Onikawa Sho | 6,946      | 2,131        | 6       |
| 6     | Village Stone | Muraishiwa      | 6,951      | 2,129        | 5       |
| 7     | Fooly Cooly   | Antonio Law     | 7,003      | 2,113        | 4       |
| 8     | Ben-Tang      | Chen, Chao-Wei  | 7,049      | 2,100        | 3       |
| 9     | _666_         | Yuk Keun Ho     | 7,640      | 1,937        | 2       |
| 10    | Refulgence01  | Iwasawa Mizuho  | 7,747      | 1,910        | 1       |

Etapa final, Longitud total: 40,5 [m] (Ver figura 3.1).



**Ilustración 3.1, Pista, Etapa final 2014**

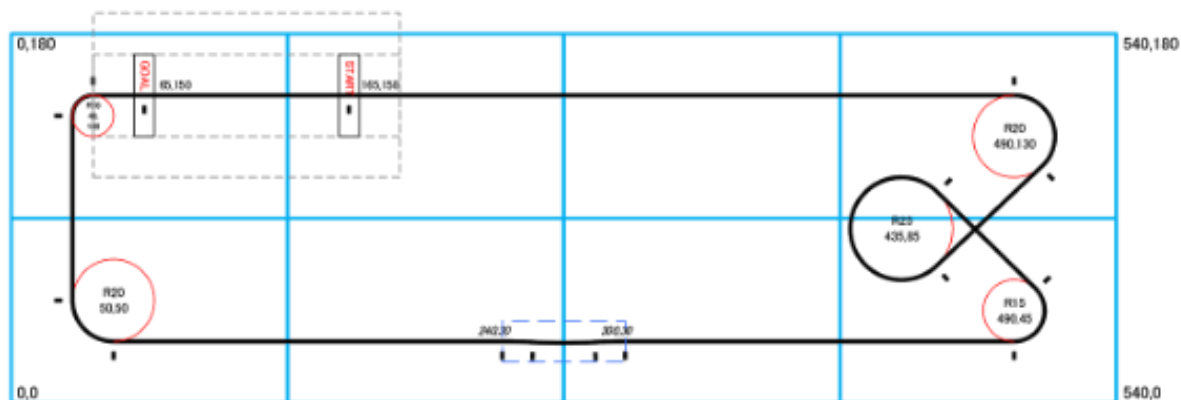
Cuyo resultado se muestra en la tabla 3.3.

**Tabla 3.3, Etapa final 2014**

| Lugar | Robot         | Desarrollador    | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|---------------|------------------|------------|--------------|---------|
| 1     | CartisX04     | Hirai Masataka   | 16,512     | 2,574        | 10      |
| 2     | Snail_II      | Shih, Kai-Hsiang | 18,577     | 2,288        | 9       |
| 3     | Dao-Huei      | Ho, Tsung-Chun   | 19,364     | 2,195        | 8       |
| 4     | Diligent      | meng weilin      | 20,084     | 2,116        | 7       |
| 5     | First Aim     | Sho Onikawa Sho  | 20,433     | 2,080        | 6       |
| 6     | Ning3         | Ng Beng Kiat     | 21,143     | 2,010        | 5       |
| 7     | Fooly Cooly   | Antonio Law      | 21,290     | 1,996        | 4       |
| 8     | _666_         | Yuk Keun Ho      | 21,380     | 1,988        | 3       |
| 9     | Refulgence01  | Iwasawa Mizuho   | 21,576     | 1,970        | 2       |
| 10    | Village Stone | Muraishiwa       | 22,300     | 1,906        | 1       |

### 3.5.2. Año 2015

Etapa Preliminar, Longitud total: 13,7 [m], (Ver figura 3.9).



**Figura 3.9, Pista, Etapa preliminar 2015**

Cuyo resultado se muestra en la tabla 3.4.

**Tabla 3.4, Ranking, Etapa preliminar 2015**

| Lugar | Robot        | Desarrollador                 | Time [s] | V prom [m/s] | Puntaje |
|-------|--------------|-------------------------------|----------|--------------|---------|
| 1     | CartisX04    | Hirai Masataka                | 4,322    | 3,17         | 10      |
| 2     | Ning3        | Ng Beng Kiat                  | 4,532    | 3,02         | 9       |
| 3     | DREAMER      | Shimura Sōbo                  | 4,627    | 2,96         | 8       |
| 4     | Andante      | Motoki Kitajima               | 4,645    | 2,95         | 7       |
| 5     | Amazing-Mach | Hsuan-Sheng Lin & Ba-Xin Lin  | 4,710    | 2,91         | 6       |
| 6     | Diligent2    | Meng Weilin                   | 4,917    | 2,79         | 5       |
| 7     | Slug_II      | CHAO-WEI,CHEN                 | 5,000    | 2,74         | 4       |
| 8     | Starry-recon | Goh Wen Feng                  | 5,007    | 2,74         | 3       |
| 9     | Scout-tracer | Teo Jun Wei                   | 5,047    | 2,71         | 2       |
| 10    | Pico         | Liou,Fong-Chun/Wang,Wen-Liang | 5,072    | 2,70         | 1       |



Etapla final, Longitud total: 50,7 [m] (Ver figura 3.10).

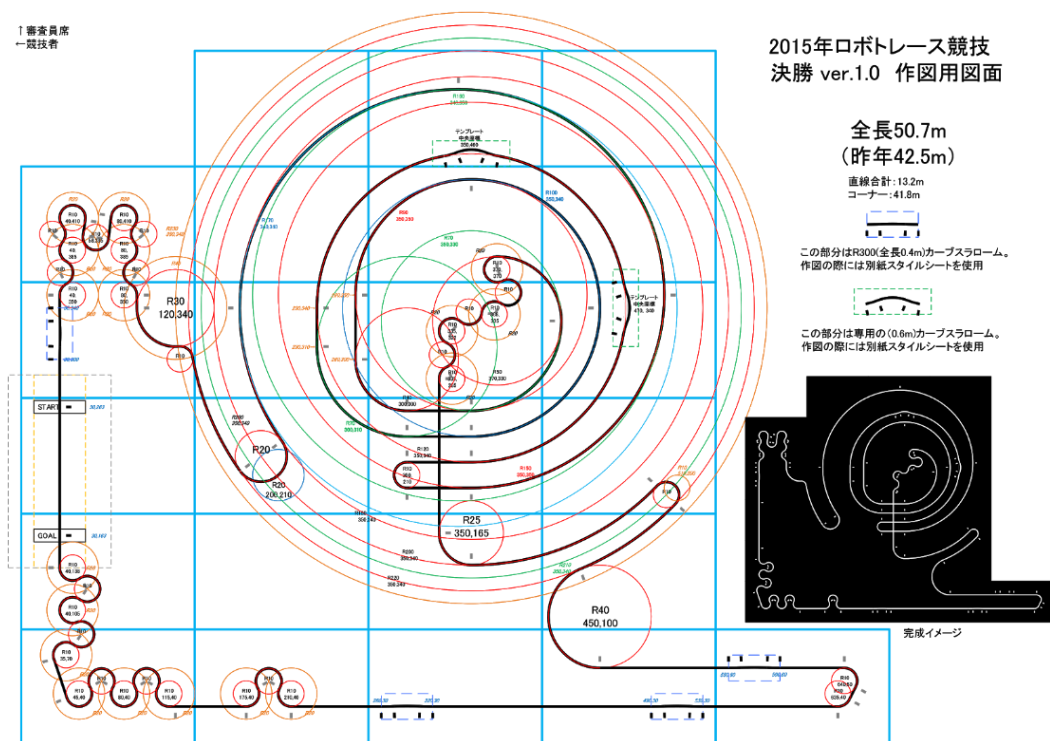


Figura 3.10, Pista, Etapa final 2015

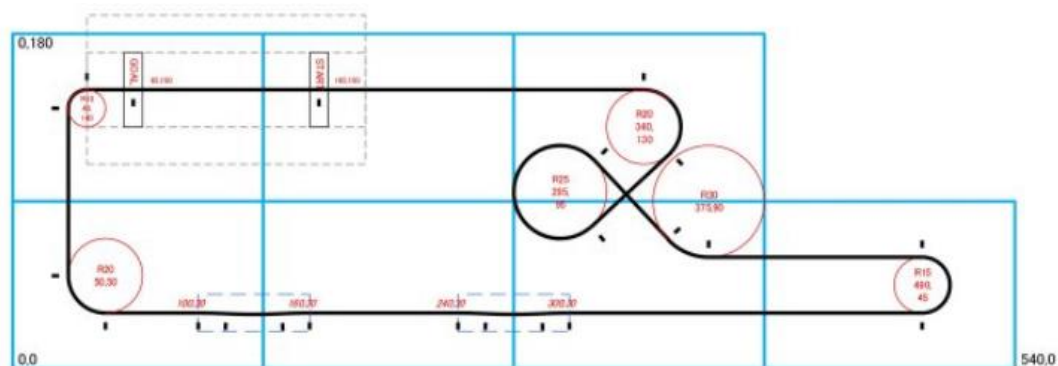
Cuyo resultado se muestra en la tabla 3.5.

Tabla 3.5, Ranking, Etapa final 2015

| Lugar | Robot      | Desarrollador      | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|------------|--------------------|------------|--------------|---------|
| 1     | CartisX04  | Hirai Masataka     | 20,536     | 2,469        | 10      |
| 2     | Pico       | LIU, FONG-CHUN     | 22,118     | 2,292        | 9       |
| 3     | Slug_II    | CHAO-WEI, CHEN     | 22,467     | 2,257        | 8       |
| 4     | Andante    | (Motoki Kitajima)  | 23,709     | 2,138        | 7       |
| 5     | Whitetip X | Seiji Ochiai       | 26,510     | 1,912        | 6       |
| 6     | T,T ~月光~   | Takayanagi         | 27,235     | 1,862        | 5       |
| 7     | 天秤 1 8     | Hatakeyama Kazuaki | 31,148     | 1,628        | 4       |
| 8     | 兜          | Yasunari Oshima    | 31,974     | 1,586        | 3       |
| 9     | Ix         | Kanuma Onuma       | 33,118     | 1,531        | 2       |
| 10    | Gow-Diu    | Cai, Xin-Han       | 33,840     | 1,498        | 1       |

### 3.5.3. Año 2016

Etapla Preliminar, Longitud total: 13,0 [m], (Ver figura 3.11).



**Figura 3.11, Pista, Etapa preliminar 2016**

Cuyo resultado se muestra en la tabla 3.6.

**Tabla 3.6, Ranking, Etapa preliminar 2016**

| Lugar | Robot           | Desarrollador     | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|-----------------|-------------------|------------|--------------|---------|
| 1     | Cartis04,5      | Hirai Masataka    | 4,49       | 2,90         | 10      |
| 2     | Hamutaro, jp    | Ba-Xin, Lin       | 4,934      | 2,63         | 9       |
| 3     | RS-100          | Endomo            | 4,944      | 2,63         | 8       |
| 4     | New-Ns-Build    | Shinnosuke Nomura | 5,351      | 2,43         | 7       |
| 5     | Diligent3       | Meng Weilin       | 5,378      | 2,42         | 6       |
| 6     | Dreamer         | Shimura Sōbo      | 5,401      | 2,41         | 5       |
| 7     | Slug_III        | CHAO-WEI CHEN     | 5,402      | 2,41         | 4       |
| 8     | Flash:RT        | Chan Wei Hao      | 5,419      | 2,40         | 3       |
| 9     | Habitus         | Li, Sheng-Xiang   | 5,548      | 2,34         | 2       |
| 10    | AllegroModerato | Shota Oka         | 5,618      | 2,31         | 1       |

Epata Final, Longitud total: 39,7 [m], (Ver figura 3.12).



**Figura 3.12, Pista, Etapa final 2016**

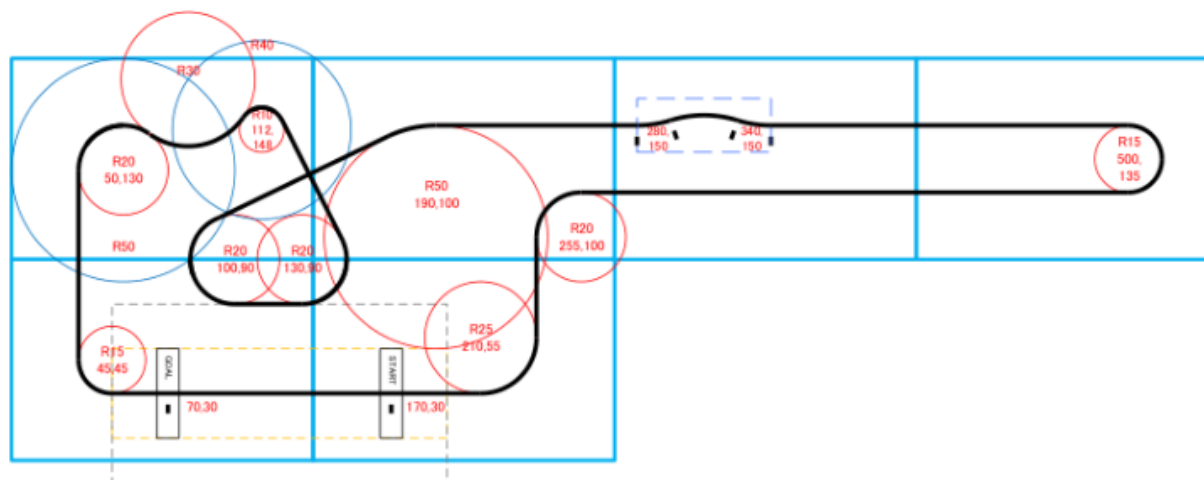
Cuyo resultado se muestra en la tabla 3.7

**Tabla 3.7, Ranking, Etapa final 2016**

| Lugar | Robot           | Desarrollador      | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|-----------------|--------------------|------------|--------------|---------|
| 1     | Cartis04,5      | Hirai Masataka     | 13,803     | 2,88         | 10      |
| 2     | Slug_III        | CHAO-WEI CHEN      | 14,750     | 2,69         | 9       |
| 3     | Hamutaro, jp    | Ba-Xin, Lin        | 14,793     | 2,68         | 8       |
| 4     | Flash: RT       | Chan Wei Hao       | 15,000     | 2,65         | 7       |
| 5     | Diligent3       | Meng Weilin        | 15,857     | 2,50         | 6       |
| 6     | New-Ns-Build    | Shinnosuke Nomura  | 18,350     | 2,16         | 5       |
| 7     | AllegroModerato | Shota Oka          | 18,950     | 2,09         | 4       |
| 8     | シルバーフレーム        | Shintaro Naito     | 19,258     | 2,06         | 3       |
| 9     | Whitechip 7     | Seiji Ochiai       | 20,004     | 1,98         | 2       |
| 10    | 天秤 19           | Hatakeyama Kazuaki | 22,066     | 1,80         | 1       |

### 3.5.4. Año 2017

Etapla preliminar, Longitud total: 13,0 [m], (Ver figura 3.13).



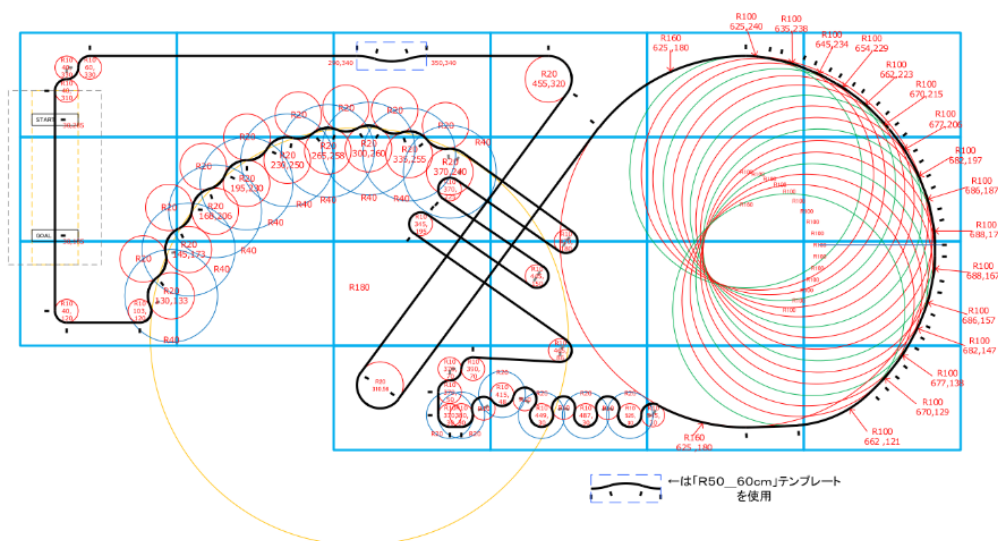
**Figura 3.13, Pista, Etapa preliminar 2017**

Cuyo resultado se muestra en la tabla 3.8.

**Tabla 3.8, Ranking, Etapa preliminar 2017**

| Lugar | Robot           | Desarrollador  | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|-----------------|----------------|------------|--------------|---------|
| 1     | RS-100          | Endomo         | 4,695      | 2,77         | 10      |
| 2     | Cartis04,5      | Hirai Masataka | 4,745      | 2,74         | 9       |
| 3     | Ning3           | Ng Beng Kiat   | 5,013      | 2,59         | 8       |
| 4     | TE-RT           | Takumi Ookami  | 5,289      | 2,46         | 7       |
| 5     | Leo-blue Darryl | Chua Wee Joon  | 5,334      | 2,44         | 6       |
| 6     | DREAMER         | Shimura Sōbo   | 5,542      | 2,35         | 5       |
| 7     | Calibrate       | Muhammad Idris | 5,598      | 2,32         | 4       |
| 8     | Diligent-4      | meng weilin    | 5,682      | 2,29         | 3       |
| 9     | Pico            | Liou,Fong-Chun | 5,689      | 2,29         | 2       |
| 10    | Exelion         | Makoto Yamada  | 5,774      | 2,25         | 1       |

Etapla final, Longitud total: 35,2 [m], (Ver figura 3.14).



**Figura 3.14, Pista, Etapa final 2017**

Cuyo resultado se muestra en la tabla 3.9.

**Tabla 3.9, Ranking, Etapa final 2017**

| Lugar | Robot           | Desarrollador   | Tiempo [s] | V prom [m/s] | Puntaje |
|-------|-----------------|-----------------|------------|--------------|---------|
| 1     | RS-100          | Endomo          | 14,301     | 2,46         | 10      |
| 2     | Cartis04,5      | Hirai Masataka  | 14,377     | 2,45         | 9       |
| 3     | Ning3           | Ng Beng Kiat    | 14,788     | 2,38         | 8       |
| 4     | TE-RT           | Dainichi        | 15,43      | 2,28         | 7       |
| 5     | Leo-blue Darryl | Chua Wee Joon   | 15,574     | 2,26         | 6       |
| 6     | Whitechip 7     | Seiji Ochiai    | 15,653     | 2,25         | 5       |
| 7     | Diligent-4      | meng weilin     | 16,565     | 2,12         | 4       |
| 8     | 寿限無改            | Morita / Shopei | 18,565     | 1,90         | 3       |
| 9     | Super LiZ       | Li Gengwei      | 18,91      | 1,86         | 2       |
| 10    | Du-Lin-Car      | Li,Yu-Ming      | 19,257     | 1,83         | 1       |

### 3.5.5. Mejores robots históricos de la categoría

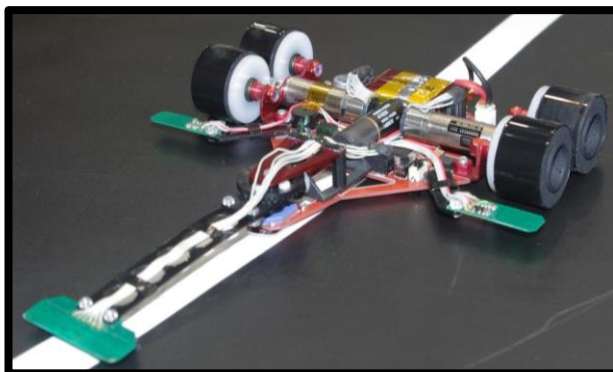
Debido a que la tecnología utilizada en cada robot puede variar de un año a otro, se tomarán las fichas técnicas del último año en que hayan sido registrados

El robot Cartis04,5 puede verse en la figura 3.15, cuyas especificaciones técnicas se encuentran en la tabla 3.10.

**Tabla 3.10, Especificaciones, Cartis04,5, Documentación NTF 2017**

|                              |  |
|------------------------------|--|
| <b>Nombre del Robot</b>      | <b>Cartis04,5</b>  |
| <b>Operador</b>              | Masaaki Hirai  |
| <b>Hardware Electrónico</b>  | CPU:STM32F103RE<br>Motor Driver: TB6612FNG<br>Sensor de luz infrarroja: SHARP GP2S700 x10 con Conversor AD   |
| <b>Hardware Mecánico</b>     | Motores Empuje: Maxon DCX10L x2<br>Motor Trompa: Maxon RE10<br>Cuerpo: PCB<br>Encoder: Maxon ENX EASY10<br>Ruedas: 4 ruedas, 25mm diámetro, 12mm de ancho. |
| <b>Baterías</b>              | LiPo 2S 7.4V, 260 mah.   |
| <b>Entorno de desarrollo</b> | CooCox ID  |
| <b>Dimensiones</b>           | Largo x Ancho x Alto: 175 x 150 x 35 [mm]<br>Peso: 141 [g]   |

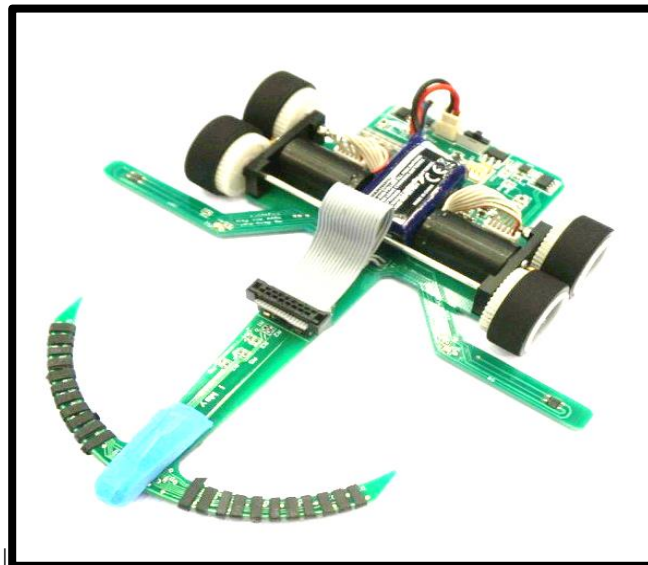
**Figura 3.15, Cartis04,5 Primer lugar, ranking 2014-2017**



El robot Ning3 puede verse en la figura 3.16, cuyas especificaciones técnicas se encuentran en la tabla 3.11.

**Tabla 3.11, Especificaciones, Ning3, Documentación NTF 2017**

|                              |   |
|------------------------------|---|
| <b>Nombre del Robot</b>      | <b>Ning3</b>  |
| <b>Operador</b>              | Ng Beng Kiat  |
| <b>Hardware Electrónico</b>  | CPU: STM32F4<br>Motor Driver: DVR8835<br>Sensor de luz infrarroja: 15 sensores con Conversor AD                           |
| <b>Hardware Mecánico</b>     | Motores Empuje: Faulhaber 1331SR006 x2 + Encoder EI-400<br>Cuerpo: PCB<br>Ruedas: 4 ruedas, 24mm diámetro, 10mm de ancho. |
| <b>Baterías</b>              | LiPo 2S 7.4V, 250mah.   |
| <b>Entorno de desarrollo</b> | CooCox ID   |
| <b>Dimensiones</b>           | Largo x Ancho x Alto: 110 x 130 x 24 [mm]<br>Peso: 120 [g]  |

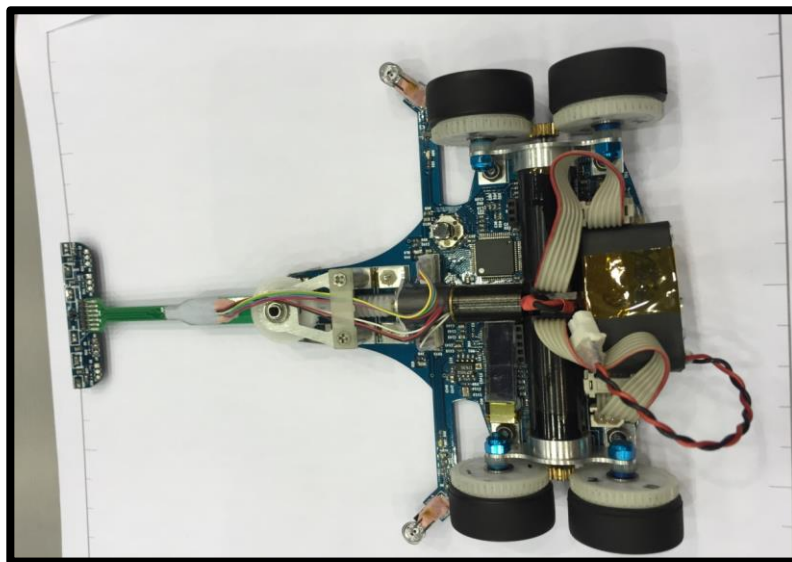


**Figura 3.16, Ning3, Segundo ranking 2014-2017**

El robot Diligent - 4 puede verse en la figura 3.17, cuyas especificaciones técnicas se encuentran en la tabla 3.12.

**Tabla 3.12, Especificaciones, Diligent - 4, Documentación NTF 2017**

|                              |  |
|------------------------------|--|
| <b>Nombre del Robot</b>      | <b>Diligent - 4</b>  |
| <b>Operador</b>              | Meng Weilin  |
| <b>Hardware Electrónico</b>  | CPU: STM32F4<br>Motor Driver: ZXMHC3F381<br>Sensor de luz infrarroja:x8 con Conversor AD   |
| <b>Hardware Mecánico</b>     | Motores Empuje: Faulhaber SR1331006 x 2<br>Motor Trompa: Maxon RE10Cuerpo: PCB<br>Encoder: Maxon ENX EASY10<br>Ruedas: 4 ruedas, 25mm diámetro, 15mm de ancho. |
| <b>Baterías</b>              | LiPo 2S 7.4V, 240mah.  |
| <b>Entorno de desarrollo</b> | Keil   |
| <b>Dimensiones</b>           | Largo x Ancho x Alto: 170 x 130 x 25 [mm]<br>Peso: 125 [g]   |



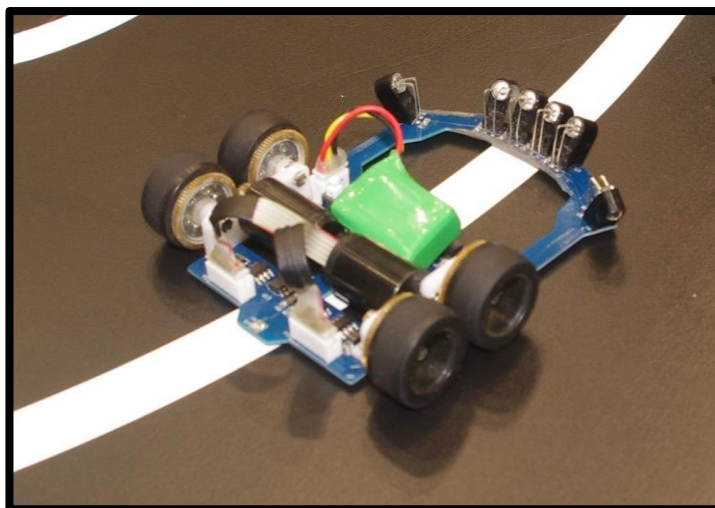
**Figura 3.17, Diligent – 4, Tercer lugar, ranking 2014-2017.**



El robot Slug\_III puede verse en la figura 3.18, cuyas especificaciones técnicas se encuentran en la tabla 3.13.

**Tabla 3.13, Especificaciones, Slug\_III, Documentación NTF 2017**

|                              |  |
|------------------------------|--|
| <b>Nombre del Robot</b>      | <b>Slug_III</b>  |
| <b>Operador</b>              | Chao-Wei Chen  |
| <b>Hardware Electrónico</b>  | CPU: dsPIC33FJ128MC804<br><br>Motor Driver: ZXMHC3A01N8<br><br>Sensor de luz infrarroja:x8 con Conversor AD          |
| <b>Hardware Mecánico</b>     | Motores Empuje: SR1717006 x2<br><br>Encoder: Maxon ENX EASY10<br><br>Ruedas: 4 ruedas, 25mm diámetro, 10mm de ancho. |
| <b>Baterías</b>              | LiPo 2S 7.4V, 240mah.  |
| <b>Entorno de desarrollo</b> | CooCox ID  |
| <b>Dimensiones</b>           | Largo x Ancho x Alto: 120 x 100 x 25 [mm]<br><br>Peso: 120 [g]   |

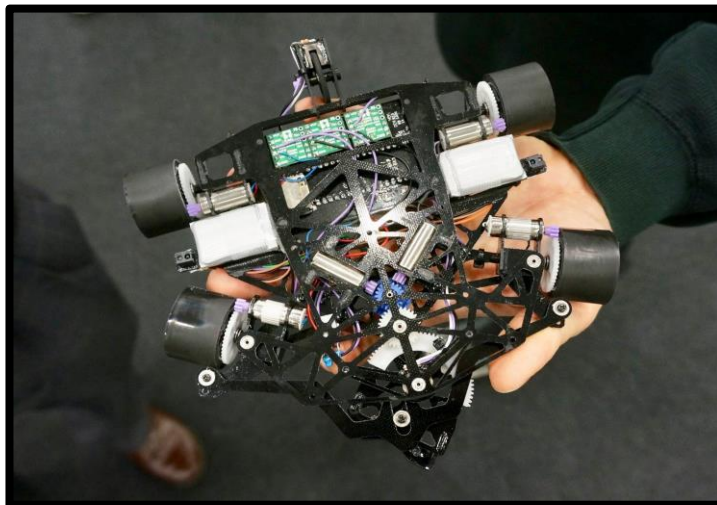


**Figura 3.18, Slug\_III, Cuarto lugar, ranking 2014-2017**

El robot RS-100 puede verse en la figura 3.19, cuyas especificaciones técnicas se encuentran en la tabla 3.14.

**Tabla 3.14, Especificaciones, RS-100, Documentación NTF 2017**

|                              |   |
|------------------------------|---|
| <b>Nombre del Robot</b>      | <b>RS-100</b>   |
| <b>Operador</b>              | Endomo  |
| <b>Hardware Electrónico</b>  | CPU: STM32F303K8T6<br><br>Motor Driver: BD65496<br><br>Sensor de luz infrarroja: Emisor L-31ROPT 1 y receptor RPR-220 x4 con Conversor AD<br><br>Giroscopio: L3GD20 |
| <b>Hardware Mecánico</b>     | Motores Empuje: Sin información x4<br><br>Motores trompa: Sin información x2<br><br>Encoder: GP1A51HRJ00F<br><br>Ruedas: 4 ruedas, 29mm diámetro, 19mm de ancho.    |
| <b>Baterías</b>              | LiPo 2S 7.4V, 240mah.   |
| <b>Entorno de desarrollo</b> | CooCox ID   |
| <b>Dimensiones</b>           | Largo x Ancho x <u>Alto</u> : 178 x 170 x 35 [mm]<br><br>Peso: 145 [g]  |



**Figura 3.19, RS-100, Quinto lugar, ranking 2014-2017**

#### **4. Proposición de diseño personalizado**

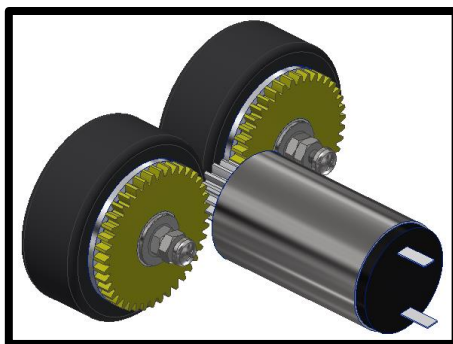
El robot que se propondrá a continuación, cuyo nombre de fantasía será “Lamborghini” es un Robotracer diseñado a partir del conocimiento adquirido por el autor de esta memoria, quien ha participado en la AJMMC desde el año 2015 hasta la fecha de publicación de esta memoria.

Las dimensiones del prototipo están basadas en el robot diseñado por Ng Beng Kiat, que aparece en el segundo lugar en el ranking del capítulo anterior, cuyo hardware considera dos motores, trompa fija (no motorizada), reflectómetros, Encoders externos, giroscopio y una unidad de procesamiento de 32 bits.

Se detallarán las características del prototipo, en los subcapítulos de Hardware Mecánico y Hardware Electrónico, para luego hacer un repaso a la vista general del diseño.

## 4.1. Hardware Mecánico

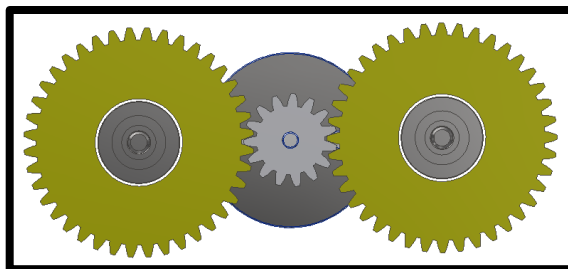
Para propósitos de esta memoria, se ha definido arbitrariamente un modelo simple de construir, tomando como ejemplo el Robotracer de Ng Beng Kiat (Singapur), cuyo robot consta de 2 motores, que generan el movimiento a partir de la velocidad de giro diferenciada para cada motor (ver figura 4.1), donde cada uno de los motores conduce mediante un engranaje, el giro de dos ruedas.



**Figura 4.1, Vista isométrica, engranaje-rueda-motor**

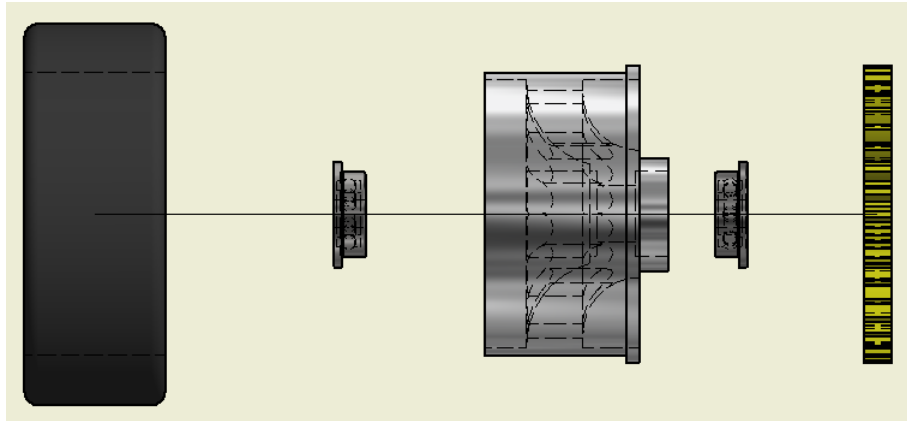
Este diseño se hizo considerando que las ruedas dentadas más comunes para este tamaño, son las de módulo 0.5 [mm].

Uno de los aspectos más importantes para que funcione de manera correcta el ensamble, es que los ejes deben quedar completamente centrados con las ruedas, por lo que se decidió utilizar llantas y neumáticos estandarizados, marca Kyosho, Mini z, y engranajes de 42 dientes y 2 [mm] de grosor (ver figura 4.2).



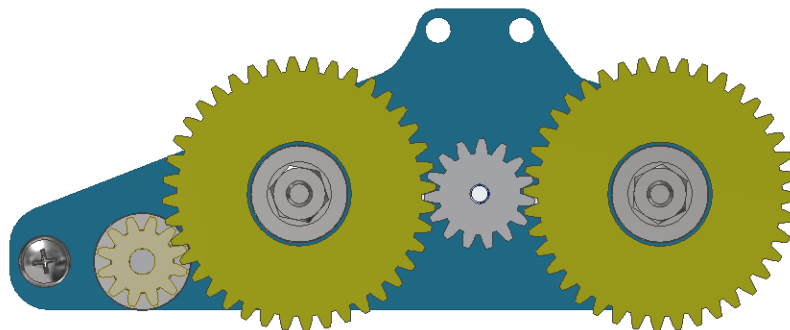
**Figura 4.2, Vista frontal, engranaje-rueda-motor**

De esta manera el engranaje antes mencionado puede caber en el offset de la llanta (ver figura 4.3), y pueden centrarse con la ayuda de un rodamiento recto de bolas de 2x5x3.2 [mm]. Como Eje se utilizará un perno M2, con tuercas del respectivo tamaño.

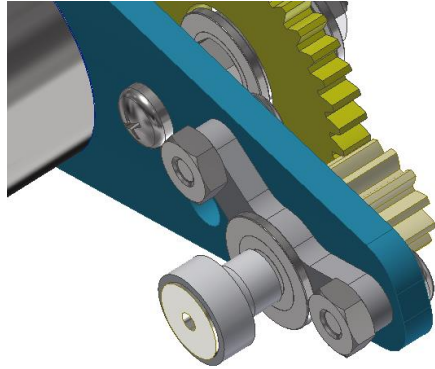


**Figura 4.3, Ensamble Rueda**

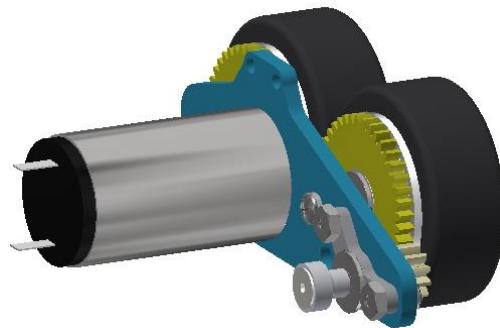
Además de las dos ruedas dentadas para cada rueda, el sistema moverá una cuarta rueda dentada (ver figura 4.4) que estará conectado a un imán que girará cercano al sensor de ángulo (ver figura 4.5), y será sostenido a través de una pieza estructural como se muestra de manera general en la figura 4.6.



**Figura 4.4, Engranaje del sistema de tracción**



**Figura 4.5, Detalle, Imán para Encoder**



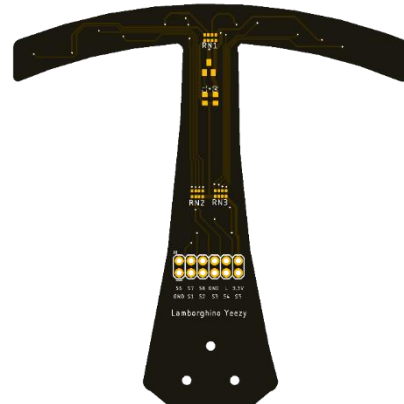
**Figura 4.6, Ensamble sistema de tracción**

El cuerpo del robot está fabricado de PCB (Printed Circuit Board). En cuatro partes.

**Trompa:** Cuyo diseño se puede observar en la figura 4.7 para la vista inferior y figura 4.8 para la vista superior.

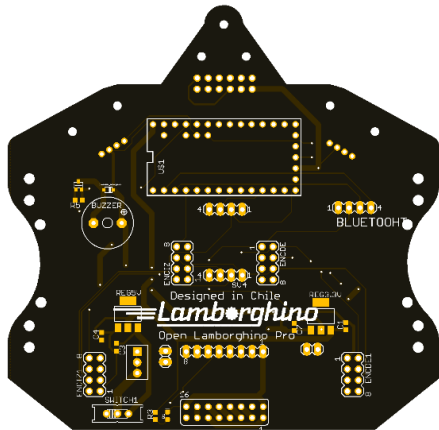


**Figura 4.7, Trompa (arreglo de sensores), vista inferior**

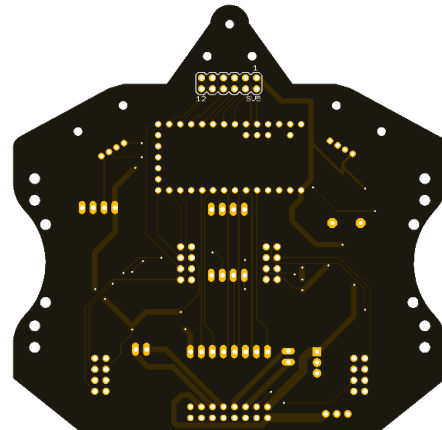


**Figura 4.8, Trompa (arreglo de sensores), vista superior**

**Cuerpo:** Cuyo diseño se puede observar en la figura 4.9 para la vista superior y figura 4.10 para la vista inferior.

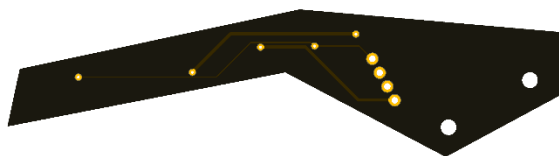


**Figura 4.9, Cuerpo vista superior**

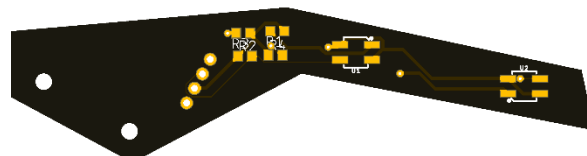


**Figura 4.10, Cuerpo vista inferior**

**Sensor lateral izquierdo:** Cuyo diseño se puede observar en la figura 4.11 para la vista superior y figura 4.12 para la vista inferior.



**Figura 4.11, Sensor lateral izquierdo, Vista superior**

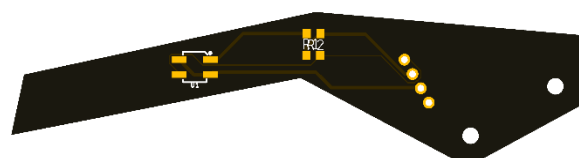


**Figura 4.12, Sensor lateral izquierdo, Vista Inferior**

**Sensor lateral derecho:** Cuyo diseño se puede observar en la figura 4.13 para la vista superior y figura 4.14 para la vista inferior.

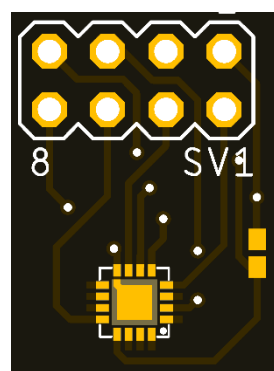
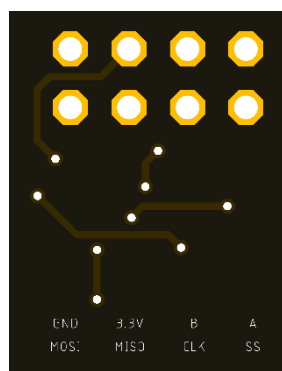


**Figura 4.13, Sensor lateral derecho, Vista superior**



**Figura 4.14, Sensor lateral derecho, Vista Inferior**

**Encoders:** Cuyo diseño se puede observar en la figura 4.15 en vista superior e inferior.

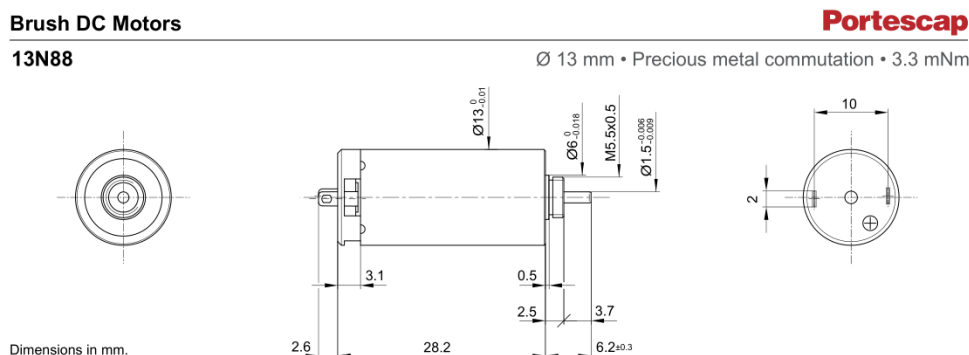


**Figura 4.15, Encoders, vista superior e inferior**



La estructura completa posee una dimensión total de 100x167 [mm], a la cual debe sumarse la distancia generada por el offset del sistema Motor-Engranaje – Rueda.

Los motores seleccionados para este prototipo son los Portescap 13n88, de manufactura india, cuyas dimensiones se muestran en la figura 4.16.



**Figura 4.16, Dimensiones Motores Portescap 13n88**

Además, es importante para saber las características de la electrónica que lo controlará, los datos técnicos que aparecen en la tabla 4.1.

| Electrical Data             | Symbol       | 213E        |
|-----------------------------|--------------|-------------|
| 1 Nominal Voltage           | V            | 6           |
| 2 No-Load Speed             | $n_0$        | 12,290      |
| 3 No-Load Current           | $I_0$        | 25.6        |
| 4 Terminal Resistance       | R            | 4.2         |
| 5 Output Power              | $P_{2max}$   | 2.4         |
| 6 Stall Torque              | mNm          | 6.5 (0.93)  |
| 7 Efficiency                | $\eta_{max}$ | 75          |
| 8 Max Continuous Speed      | $n_{e max}$  | 9,000       |
| 9 Max Continuous Torque     | $M_{e max}$  | 3 (0.47)    |
| 10 Max Continuous Current   | $I_{e max}$  | 0.69        |
| 11 Back-EMF Constant        | $k_E$        | 0.48        |
| 12 Torque Constant          | $k_M$        | 4.58        |
| 13 Motor Regulation         | $R/k^2$      | 200.0       |
| 14 Friction Torque          | $T_F$        | 0.12 (0.02) |
| 15 Rotor Inductance         | L            | 0.07        |
| 16 Mechanical Time Constant | $\tau_m$     | 5.6         |
| 17 Rotor Inertia            | J            | 0.28        |

**Tabla 4.1, Especificaciones, Motor Portescap 13n88**

## 4.2. Hardware Electrónico

El Hardware electrónico está separado en cuatro subsistemas: Potencia, Control, Sensores y Periféricos

### 4.2.1. Subsistema de Potencia:

El subsistema de potencia, tiene por objetivo alimentar correctamente todos los componentes del robot, y para esto se ha creado un circuito que comienza con una entrada de batería de 7.4 volts (Litio – Polímero), controlada por un switch al cual se conecta un regulador de voltaje de 5 Volts, y a su vez este se conecta a un regulador de voltaje de 3.3 Volts. Todos los componentes asociados a sus respectivos capacitores de desacople.

De esta manera se tiene un circuito de potencia desde el cual se pueden alimentar componentes de 3.3 volts, de 5.0 volts y directamente de la fuente principal como se muestra en la figura 4.17.

### POWER

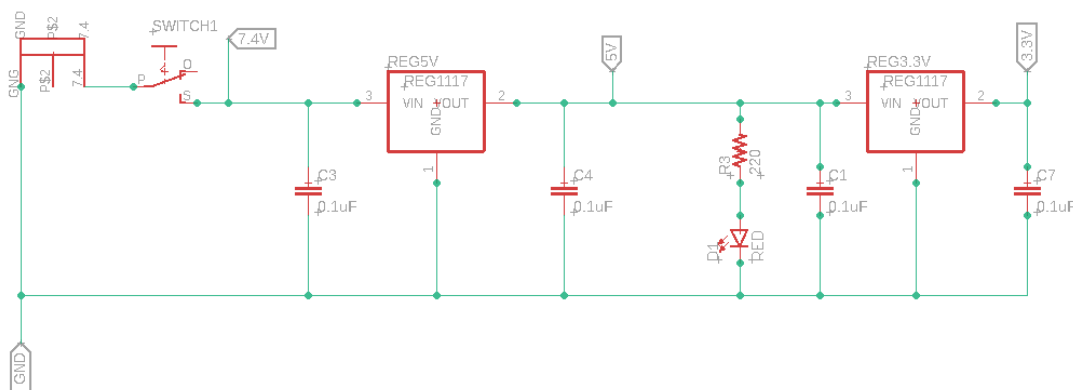
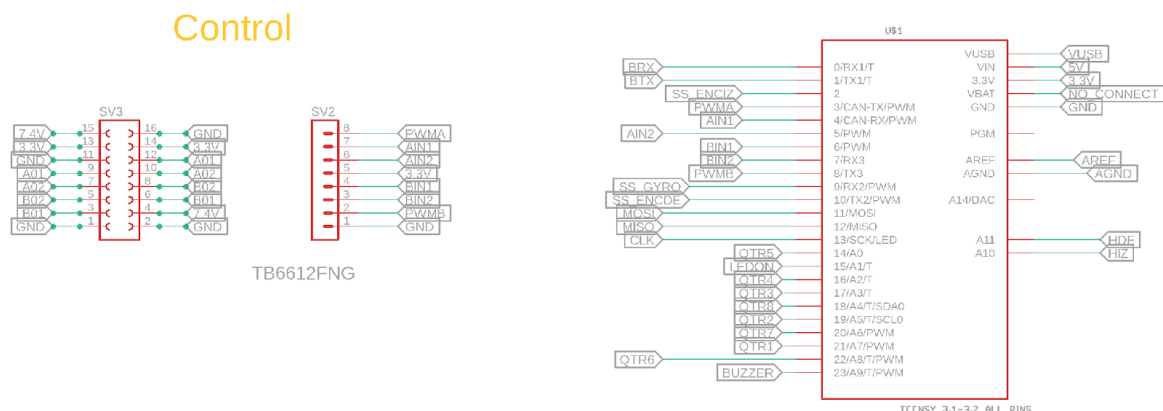


Figura 4.17, Circuito de potencia

Este subsistema se encuentra embebido en la placa principal (cuerpo).

## 4.2.2. Subsistema de Control:

El subsistema de control es el encargado de procesar y comunicar todos los componentes del circuito, el cual se separa en dos partes. El Microcontrolador y el Driver de Motor cuyos pines de salida se muestran en la figura 4.18.



**Figura 4.18, Ruteo, circuito subsistema de control**

Como Microcontrolador se utilizará el MK20DX256, en forma de placa embebida. (ver figura 4.19)



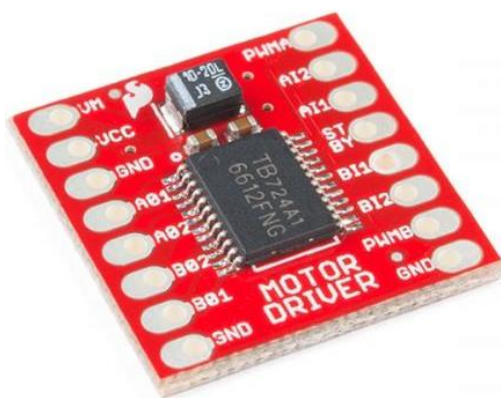
**Figura 4.19, Unidad de procesamiento, Teensy 3.2**

Esta placa se puede encontrar en el mercado local con el nombre Teensy 3.2, cuyas características son las siguientes:

- 32-bit ARM Cortex-M4 72 MHz CPU
- Memoria Flash 256K, RAM 64K, EEPROM 2K
- 21 entradas analógicas de alta resolución (13 bits utilizables, 16 bit hardware)
- 34 entradas/salidas digitales (tolerancia de 5V) • 12 salidas PWM
- 7 temporizadores de intervalo/retardo separados del PWM
- USB con DMA dedicado
- 3 puertos UARTs
- SPI, I2C, I2S, CAN Bus, modulator

El Driver de motor se utilizará como puente H, para alimentar a los motores con la fuente principal, sin dejar de controlarlos con el microcontrolador. En este caso se utilizó un driver Toshiba TB6612FNG (ver figura 4.20), cuyas características son:

- Voltaje de alimentación:  $V_M = 15\text{ V máx.}$ ,  $V_{CC} = 2,7 - 5,5\text{ V}$
- Corriente de salida:  $I_{out} = 1,2\text{A (promedio)} / 3,2\text{A (peak)}$
- Modos de control de motor: CW, CCW, freno corto y parada.
- Circuito integrado de apagado térmico y circuito de detección por bajo voltaje
- Capacitores en ambas líneas de alimentación



***Figura 4.20, Driver para motor, Toshiba TB6612FNG***

### 4.2.3. Subsistema de Sensores:

Para poder facilitar el diseño del robot, se trabajó en una estructura modular donde la placa principal (cuerpo), posee conectores donde van a su vez los distintos módulos, como se muestra en la figura 4.21.

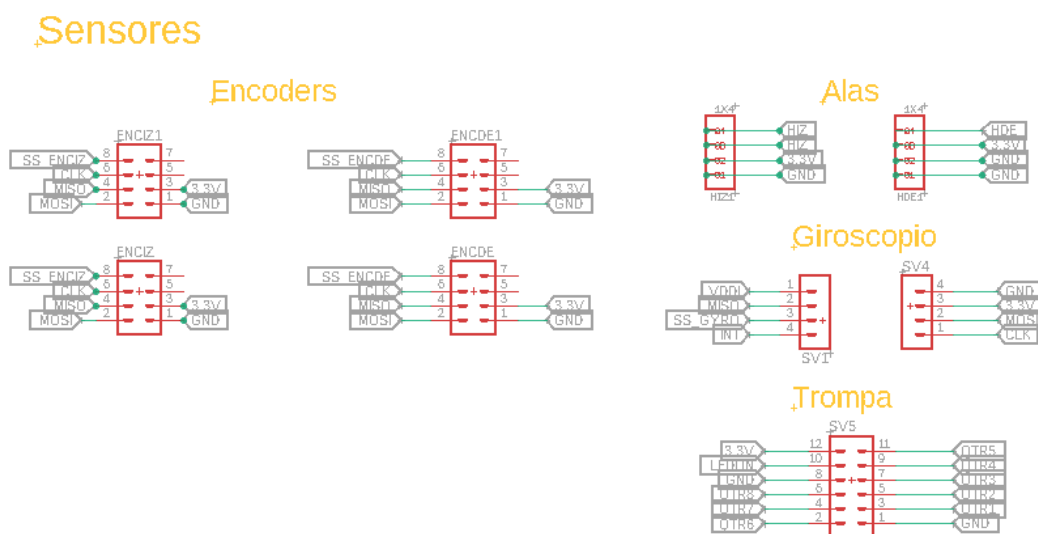
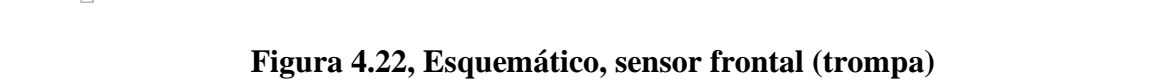


Figura 4.21, Conectores para los módulos

#### Trompa:

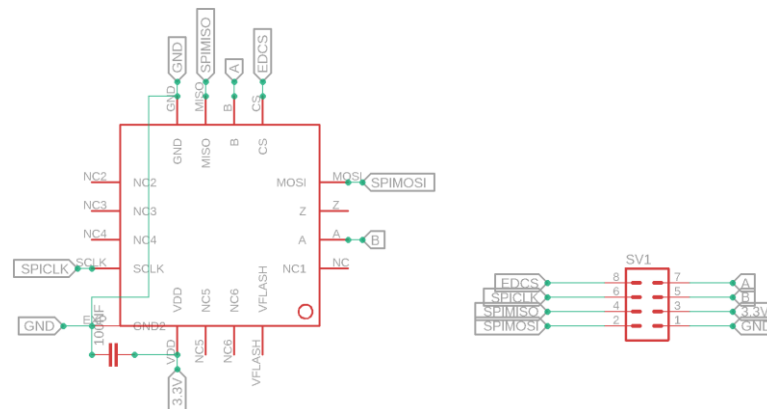
Este componente tiene por objetivo leer la reflectancia del lugar por donde está pasando, convirtiendo en una señal analógica el nivel de reflectancia. Donde el voltaje estará entre 0 volts y VCC (3.3V), cuando el sensor pase entre el negro y el blanco respectivamente (en una gama de colores grises).

La trompa está compuesta por un circuito que contiene 8 emisores y sensores de luz infrarroja (reflectómetros), QRE1113GR (ver figura 4.22), que son energizados con 3.3V, a través de un MOSFET, para controlar a través de pulsos PWM la potencia de los emisores de luz infrarroja. Cada una de las salidas de los sensores, van conectados a una entrada analógica del microcontrolador. El conector que une la trompa con el cuerpo, es un Berg de 2x6 pines, con un paso de 2.54 [mm] como se muestra en el layout de la trompa en la figura 4.23.



## Encoders:

Los Encoders tienen por objetivo, leer el ángulo de rotación del imán que se encuentra en el juego de engranajes, a través del efecto hall. Esta dimensión es procesada por el chip MA700 (ver figura 4.24) embebido en el módulo mostrado anteriormente en la figura 4.15 para finalmente comunicarse con el microcontrolador a través del protocolo SPI.

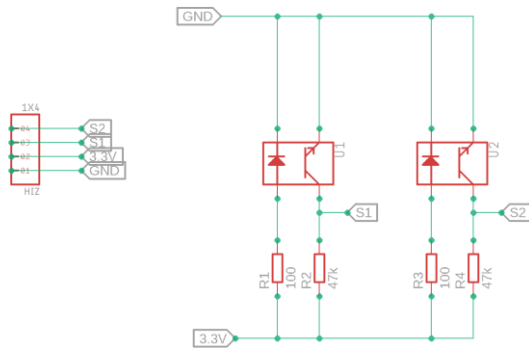


**Figura 4.24, Esquemático, Encoders MA700**

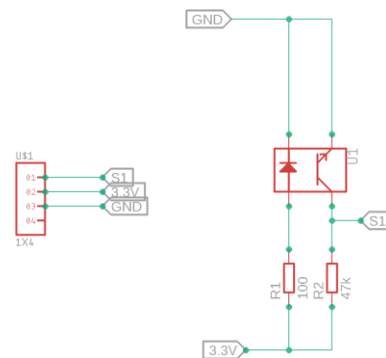
De esta manera, por cada giro del motor, el microcontrolador podrá obtener un feedback del ángulo del imán, con una resolución de 12 bits, es decir, 11.37 cuentas por cada grado (4095 cuentas /360 grados).

## Sensores laterales:

Los sensores laterales tienen por objetivo captar el instante cuando el robot pase por sobre una marca lateral. Este sensor, al igual que la trompa, está compuesto por el sensor QRE1113GR (ver figura 4.26 para el sensor lateral izquierdo y 4.27 para el sensor lateral derecho). Cabe destacar que el sensor izquierdo posee 2 de estos sensores y el sensor derecho posee solo uno. Esto es debido a que el sensor de la izquierda debe captar las marcas laterales en curvas cerradas tanto en sentido horario como en sentido antihorario, de esta manera el sensor amplía el rango de distancia en el cual medir la marca.



**Figura 4.26, Esquemático, Sensor Lateral Izquierdo**



**Figura 4.27, Esquemático, Sensor Lateral Derecho**



## Giroscopio:

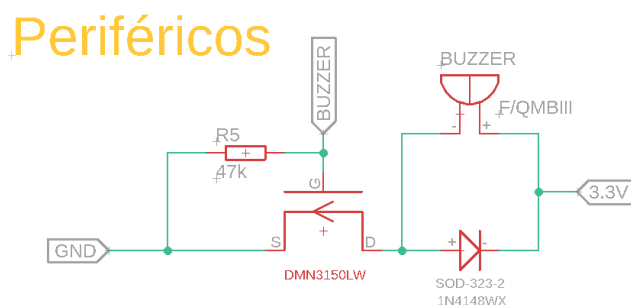
El subsistema de Giroscopio (ver figura 4.29) tiene por objetivo obtener la velocidad angular del robot. Para esto se utilizará el módulo MPU-9250, vendido por la empresa Sparkfun, cuya comunicación con el microcontrolador será a través del protocolo SPI.



**Figura 4.29, Layout Giroscopio MPU-9250**

### 4.2.4. Subsistema de Periféricos:

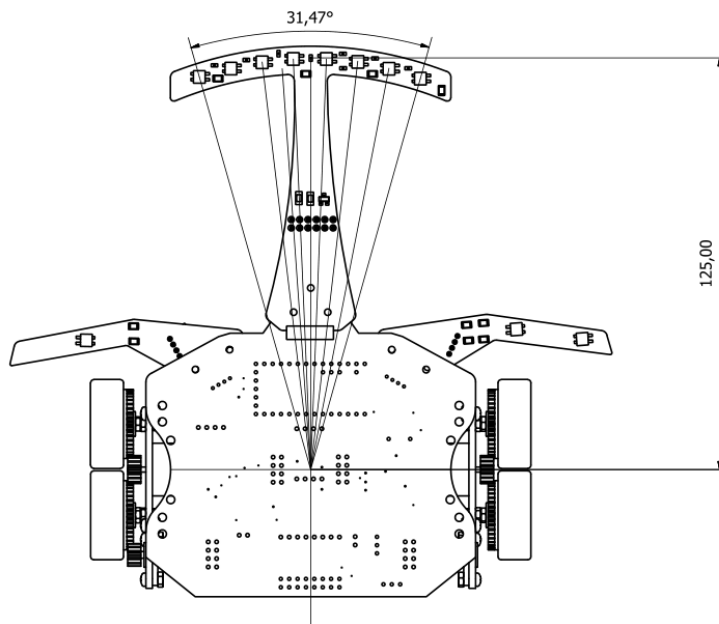
El subsistema de periféricos tiene por objetivo dar feedback desde el robot hacia el operador, para entregar una referencia de la ejecución del sistema operativo del robot y está compuesto por un buzzer, cuyo esquemático se puede ver en la figura 4.30. Este sistema de feedback es en solo una dirección, (desde el robot hasta el operador), por lo tanto, cumple con la reglamentación sobre la comunicación con robot.



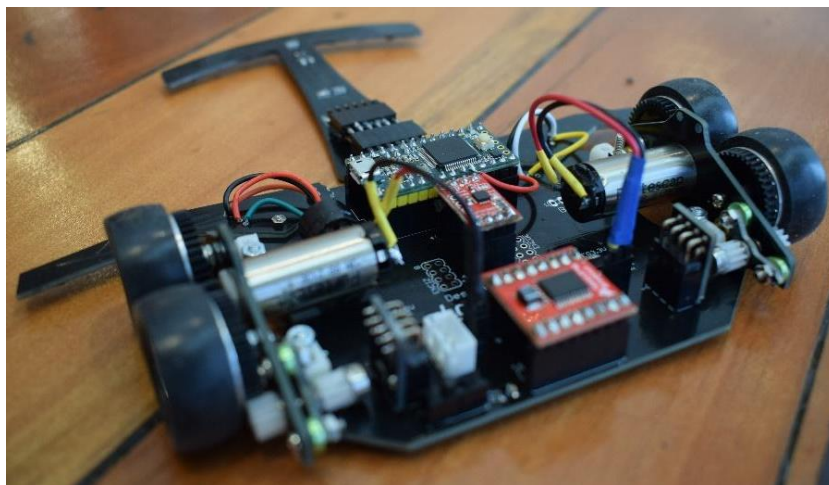
**Figura 4.30, Esquemático, subsistema de periféricos**

### 4.3. Vista general del diseño:

En la figura 4.31 se muestra el ensamble completo y en la figura 4.32 se puede ver el robot ya construido.



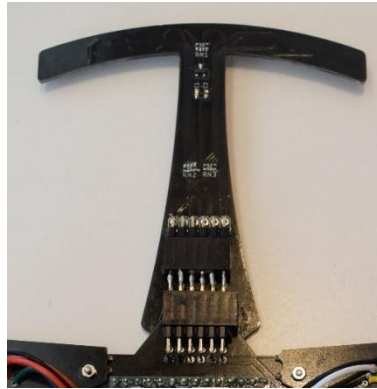
**Figura 4.31, Dimensiones de Ensamble completo**



**Figura 4.32, Robot, construido a tamaño real**

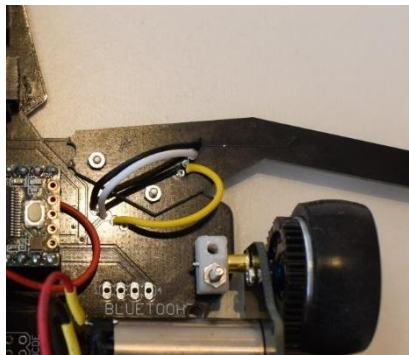
#### 4.4. Consideraciones del ensamble completo

La trompa se ha ensamblado mecánicamente con adhesivo y las conexiones se han hecho a través de dos conectores Header Hembra soldados, haciendo una unión Macho Hembra – Hembra Macho como se muestra en la figura 4.33.



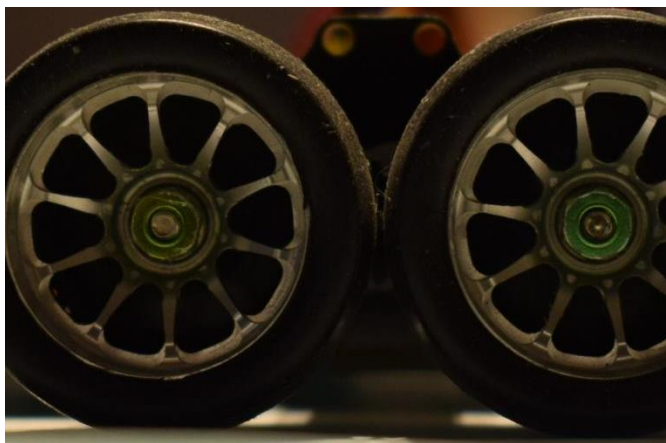
**Figura 4.33, Unión Cuerpo - Sensor Frontal**

La unión mecánica entre las alas y el cuerpo se ha hecho con tornillos de M1.7 y las conexiones se han hecho con cable como se muestra en la figura 4.34.



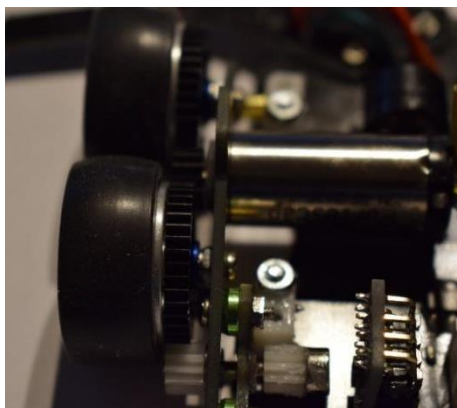
**Figura 4.34, Unión Cuerpo - Sensor Lateral**

La distancia entre ruedas logró quedar robusta y sin que se toquen entre sí (ver figura 4.35).



**Figura 4.35, Distancia entre ruedas**

El ensamble del sistema de engranajes considera un piñón de motor de 15 dientes, que conducirán a su vez dos ruedas dentadas rectas de 42 dientes. Además, se le agregó una cuarta rueda dentada de 12 dientes, que se encargarán de mover un imán que entregará un feedback sobre la posición angular de las ruedas, como se muestra en la figura 4.36.



**Figura 4.36, Sistema engranajes: Motor - Rueda - Encoder**

Los motores Portescap 13n78 a 6.0 Volts tienen una velocidad de 12.290 RPM, que, conducido a los engranajes de las ruedas, y sin considerar pérdida de eficiencia, sería de 4.389 RPM. Las ruedas tienen un diámetro exterior de 27[mm], lo que permitirá una

velocidad de 6.2 [m/s], sin considerar pérdidas de eficiencia cuando el motor se alimente con 6 volts.

Además, el sistema de engranajes conduce al cuarto engranaje de 12 dientes, lo que haría girar al imán 3.5 vueltas, por cada revolución de la rueda. Para una vuelta de la rueda (que recorre  $27 * \pi \approx 84.78 \text{ mm}$ ) el imán dará 3.5 vueltas, y ya que el encoder es de 12 bits de resolución, por cada vuelta el encoder contará  $2^{12} * 3.5$  cuentas =  $14.336 \left[ \frac{\text{cuentas}}{\text{revolución de rueda}} \right]$ , dando a sistema completo una resolución de medición de 169 [cuentas/mm].

Como referencia, el robot del desarrollador Ng Beng Kiat, posee unos motores Faulhaber 1331T006SR con encoder IE-400, con 400 pulsos por revolución del eje del motor, dándole 1600 cuentas por revolución del eje del motor, y considerando que utiliza una reducción de 1:3, y ruedas de 27mm de diámetro, el robot de Beng Kiat posee una resolución aproximada de 57 [cuentas/mm].

## 5. Descripción Analítica de un Robotracer

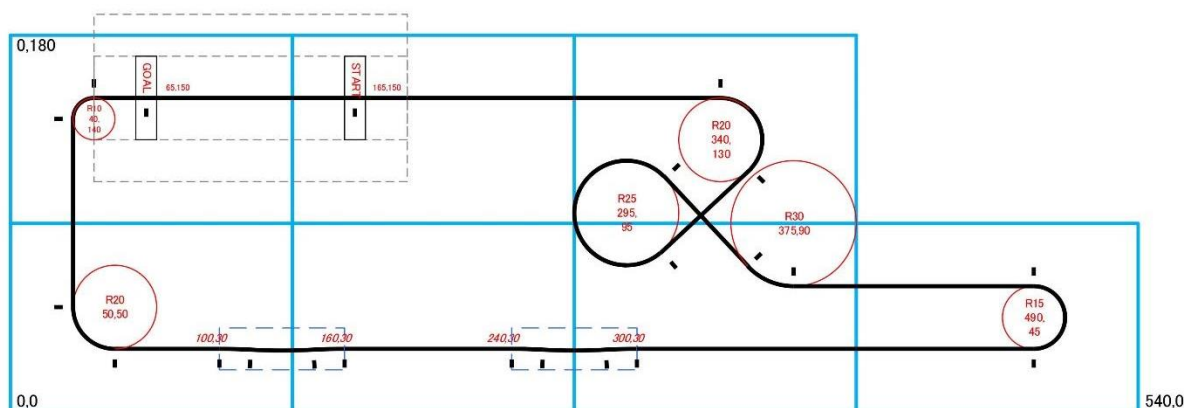
Considerando la reglamentación acerca de los escenarios, se puede observar que la pista está construida a partir de rectas y curvas, donde aparecerá una marca a 40 milímetros a la izquierda de la línea central, cada vez que exista un cambio en la curvatura del camino.

El comportamiento del Robotracer se observará como una partícula, que debe seguir la trayectoria determinada. Esta partícula se verá enfrentada tanto a movimiento rectilíneo uniformemente acelerado, como movimiento circular uniforme, donde deberá controlar su velocidad a partir de las limitaciones físicas que posea el robot, para no generar una fuerza que lo pueda sacar de la pista.

En este capítulo, se describirá la dinámica del movimiento ideal de un Robotracer modelado como una partícula, para luego adaptar el movimiento ideal al Robotracer propuesto en el capítulo 4.

## 5.1. Movimiento Ideal

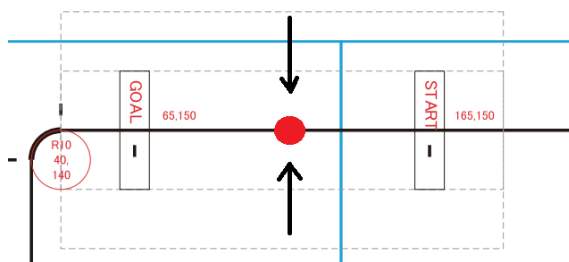
Para describir cómo sería el movimiento ideal de un Robotracer en la pista, se tomará como ejemplo el escenario utilizado por la competencia AJMMC 2016, para la categoría Robotracer, en la fase preliminar, cuyas dimensiones se ven en la figura 5.1.



**Figura 5.1, Escenario, Robotracer, Preliminar, AJMMC 2016**

Según las reglas de la competencia mencionadas en el subcapítulo 3.2.3, el competidor tiene hasta tres oportunidades para recorrer la pista, quedando registrado como tiempo de competición, el menor tiempo de las tres oportunidades. Como el objetivo es recorrer el escenario en el menor tiempo posible, cumpliendo con la reglamentación, se debe programar el movimiento del Robotracer considerado a lo menos una vuelta de reconocimiento, a partir de la cual se obtengan las dimensiones del escenario, y así en los intentos posteriores, poder optimizar el movimiento del robot.

Para comenzar con el primer intento, el Robotracer, representado por una partícula elemental, debe ser colocado en el área de meta, alineado con la pista y en un punto equidistante entre las dos marcas laterales derechas, como muestra la figura 5.2.



**Figura 5.2, Robotracer en punto de partida**

Una vez puesto el robot en la zona de partida, se debe esperar la instrucción del juez, para activar la partida del robot. Una vez activada la partida, no se puede volver a manipular el robot, ni mantener ningún tipo de comunicación hacia este. No obstante, antes de cada intento se pueden hacer limpiezas de las ruedas, o navegar en el sistema operativo del robot, para la ejecución de una rutina específica.

El robot debe comenzar su movimiento y recorrer la pista a una rapidez constante, y lo suficientemente baja como para poder recorrer la pista sin sufrir desviaciones debido al derrape, que puedan alterar la correcta medición de las dimensiones del escenario.

Durante la ejecución del primer intento, el robot deberá alcanzar la rapidez de navegación con una aceleración constante, de tal manera que no provoque ni derrape, ni que genere inclinaciones que puedan descontrolar el robot. Durante la ejecución del intento, el robot se verá enfrentado a curvas y a rectas, que podrán ser interpretados como movimiento rectilíneo uniformemente acelerado y movimiento circular uniforme.

El robot medirá con ayuda de los Encoder y de los sensores laterales, la distancia de cada tramo y además su radio de curvatura, y con esta información deberá generar el perfil de velocidades óptima para que, en el segundo y tercer intento, disminuya el tiempo en el que completa el circuito.



El comportamiento direccional del robotracer durante su desplazamiento por la pista define su respuesta a las acciones ejercidas por el controlador, así como aquellas ejercidas por el medio, que pueden afectar a la dirección del movimiento: la fuerza centrífuga, aerodinámica, irregularidades de la pista, etc. Este estudio se enfocará en la fuerza centrífuga y en la aceleración direccional, desestimando las fuerzas ejercidas en el robot debido a la aerodinámica o posibles irregularidades de la pista.

Además, el estudio se enfocará asumiendo que la partícula tiende seguir la línea de forma natural, es decir, que la partícula tenderá siempre a estar alineado con el camino, y que la única manera de salirse de la línea es que su rapidez traslacional la haga derrapar en las curvas, sacándola de esta.

El controlador puede modificar los parámetros de velocidad de giro de cada motor, en función de las condiciones en las que circula en la pista. A partir de la velocidad de giro de cada motor, se puede controlar tanto la velocidad traslacional como la dirección de giro del Robotracer.

Como en el primer intento la partícula recorrerá la pista con rapidez constante, esta partícula solo se verá enfrentada a aceleraciones para llegar a la rapidez antes mencionada y cuando se vea enfrentado a un trayecto curvo, es decir a movimiento rectilíneo uniformemente acelerado y movimiento circular uniforme respectivamente.

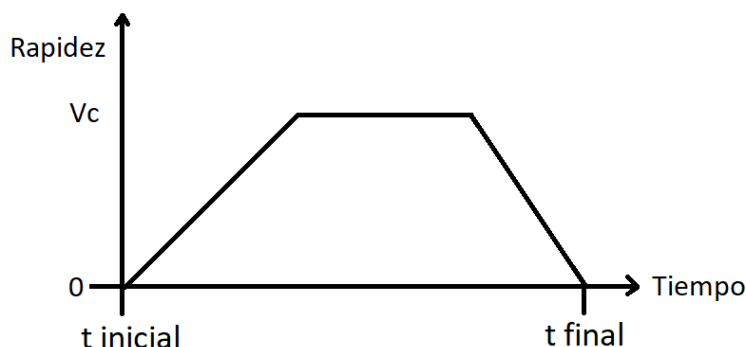
El fenómeno del derrape puede ocurrir tanto en el movimiento rectilíneo uniformemente acelerado como en el movimiento circular uniforme. En el caso del movimiento rectilíneo uniforme, puede provocar derrape o patinaje, si existe deslizamiento del neumático debido a que la aceleración direccional del robot provoca una fuerza mayor a la que efectivamente pueden soportar los neumáticos sin deslizarse; Mientras que, en el movimiento circular

uniforme, el derrape se generará cuando la fuerza centrífuga aplicada, sea superior a la fuerza de roce estático de los puntos de contacto entre los neumáticos y la pista.

La fuerza de empuje traslacional máxima antes de provocar el patinaje de las ruedas del robot será obtenida experimentalmente, observando para distintas aceleraciones y desaceleraciones, el comportamiento del prototipo. Una vez encontradas las aceleraciones límites, cualquier aceleración en magnitud inferior al límite, será aceptable para el robot.

## 5.2. Movimiento Rectilíneo Uniformemente acelerado:

El robot debe partir con rapidez cero, hasta la rapidez crucero, que será la rapidez definida para dar la primera vuelta de manera segura (ver figura 5.3).



**Figura 5.3, Ejemplo de rapidez durante el primer intento**

La aceleración con la que se genera este movimiento, como se comentó anteriormente, debe ser de magnitud, más baja que la máxima soportada empíricamente por el robot, y análogamente para la magnitud de desaceleración máxima, de esta manera, la aceleración generará una fuerza en dirección de la trayectoria ideal, que podrá controlar el robot, sin generar derrapes.

Es necesario recalcar, que el robot, debido a sus limitaciones de diseño, tendrá una aceleración y una desaceleración traslacional máxima antes de derrapar, y son estas las que generarán las limitaciones a la hora de generar el perfil de velocidades ideal para el recorrido del robot.

Como la velocidad del robot dependerá de la aceleración, y de la distancia del trayecto, entonces, la manera más idónea de observar gráficamente la velocidad del robot, es a partir de una gráfica de rapidez versus distancia, a partir de la ecuación (1), según referencia [1] que relaciona la velocidad inicial, velocidad final, aceleración y distancia.

$$v_f^2 - v_i^2 = 2 * a * x \quad (1)$$

Siendo  $a$  la aceleración constante resultante de la suma vectorial de la aceleración centrípeta y la aceleración tangencial ( $a = \sqrt{a_c^2 + a_t^2}$ ),  $v_f$  la velocidad final de la partícula, constante en el primer intento,  $v_i$  la velocidad inicial de la partícula, constante e igual a cero, y  $x$  la distancia recorrida.

### 5.3. Movimiento Curvilíneo Uniforme

Cuando la partícula con masa  $m$  y rapidez constante  $v$ , se ve enfrentada a una trayectoria curvilínea, la partícula tenderá a seguir una trayectoria rectilínea en dirección a la tangente del círculo, en su punto actual, por lo que para cambiar su dirección hacia la de la trayectoria, se verá afectada por una fuerza centrípeta. La fuerza centrípeta es la sumatoria de las fuerzas de roce de cada uno de los puntos de apoyo del robot, cuya sumatoria se le denominará como ( $F_r$ ).

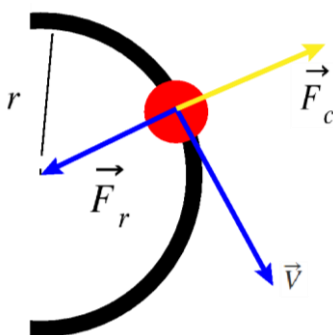
Por otro lado, el robot se verá afectado por una fuerza que intentará que el robot siga una trayectoria rectilínea, llamada fuerza centrífuga ( $F_c$ ), que tendrá una magnitud proporcional a la masa de la partícula y al cuadrado de la velocidad, e inversamente proporcional al radio de la curva.

La aceleración centrípeta ( $a_c$ ), que experimenta el robot, está representada por la fórmula (2),

$$F_c = m * a_c = m * \frac{v^2}{r} \quad (2)$$

Del movimiento curvilíneo uniformemente (MCU), donde  $v$  es la rapidez del robot en la curva, o velocidad angular, y  $r$  es el radio de la curva. La aceleración radial máxima soportada por el robot se puede calcular experimentalmente.

En el caso del movimiento curvilíneo uniforme, el fenómeno del derrape ocurrirá cuando la fuerza centrífuga que se ejerce sobre el robot, supera la fuerza de roce estático entre las ruedas y la pista (ver figura 5.4).



**Figura 5.4, Partícula en movimiento circular uniforme**

Por otro lado, la aceleración transicional ( $a_t$ ), está limitada por las características físicas del robot, cuyo valor límite, se puede calcular experimentalmente.

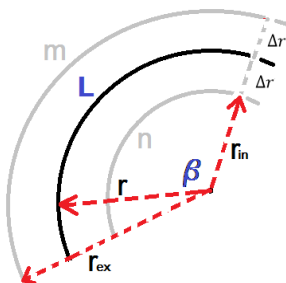
Para que el robot no derrape en la pista, todos los movimientos que este realice, no deberán superar las aceleraciones límites, cuyos valores serán calculados de manera experimental. Por lo tanto, el movimiento ideal del robot dentro de la pista, se hará estimando que deberá ir a la máxima velocidad posible, respetando las aceleraciones máximas.

## 5.4. Mapeo del escenario

El mapeo o reconocimiento del escenario es fundamental para lograr una correcta aplicación del perfil de velocidades que harán que el robot recorra en el menor tiempo posible el escenario. Dentro de las magnitudes que se requieren para producir una modelación correcta, son la cantidad total de tramos, junto a la distancia total del tramo y la curvatura de este, entendiendo como tramo a la sección de pista que hay entre dos marcas laterales.

Para poder obtener estos valores, se utilizarán los Encoders ubicados en las ruedas del robot, y los sensores laterales. Cada vez que el sensor lateral detecte una marca lateral, los Encoders de ambas ruedas medirán las distancias recorridas por cada rueda, para ese tramo.

Siendo  $m$  la distancia recorrida por la rueda izquierda y  $n$  la distancia recorrida por la rueda derecha, como se observa en la figura 5.5.



**Figura 5.5, Distancia recorrida por cada rueda en un tramo**

Se puede determinar fácilmente las magnitudes requeridas, sabiendo que:

$$\beta = \frac{L}{r} = \frac{m}{r_{ex}} = \frac{n}{r_{in}} \quad (3)$$

Donde

$$r_{in} = r - \frac{\Delta r}{2} \quad (4)$$

$$r_{ex} = r + \frac{\Delta r}{2} \quad (5)$$

Siendo  $\Delta r$  la distancia entre las ruedas de la izquierda y las ruedas de la derecha (conocida),

Se puede reescribir la ecuación como:

$$\frac{m}{r - \frac{\Delta r}{2}} = \frac{n}{r + \frac{\Delta r}{2}}$$

Despejando  $r$

$$r = \Delta r \left( \frac{m+n}{m-n} \right), \text{ con } (m-n) \neq 0 \quad (6)$$

Por otro lado, la distancia del tramo será el promedio aritmético de lo recorrido por ambas ruedas.

$$d = \frac{m+n}{2} \quad (7)$$

### **5.5. Optimización del perfil de velocidades**

Para el segundo y tercer intento, el robot ya tendrá registrado las dimensiones del escenario en que se debe desenvolver, y el objetivo es utilizar esta información para recorrer el escenario en el menor tiempo posible.

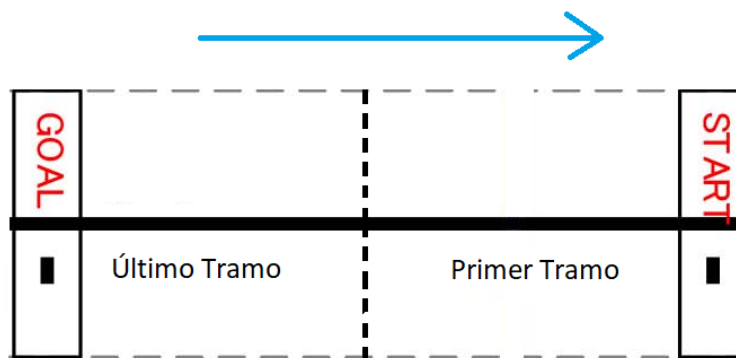
Un menor tiempo en recorrer el escenario será posible si se aumenta la velocidad en cada tramo, utilizando la información recolectada, además según las limitaciones explicadas anteriormente, es posible optimizar el perfil total de velocidades, limitando su velocidad para que no supere las aceleraciones límites.

En este sentido la velocidad máxima que alcanzará el robot en cada tramo, dependerá de la aceleración y desaceleración máxima, de la distancia de cada tramo, del radio de curvatura de cada tramo, y de los tramos posteriores.

Para poder hacer un modelo simple de perfil de velocidades se iniciará el cálculo, considerando que cada tramo tiene una velocidad de entrada y una velocidad de salida, donde la velocidad de salida de un tramo, es igual a la velocidad de entrada del tramo siguiente.

Además, se sabe que la velocidad de entrada del primer tramo y la velocidad de salida del último tramo es cero. Además, para términos de cálculo, se considerará como primer tramo, los últimos 500 milímetros del área de meta, y como último tramo los primeros 500 milímetros del área de meta (ver figura 5.1).





**Ilustración 5.1, Primer y último tramo**

Considérese entonces que existe una aceleración traslacional máxima  $a_t$ , que para términos prácticos también será la desaceleración traslacional máxima, y también habrá una aceleración centrífuga máxima  $a_c$ , que podrá soportar el robot en un trayecto curvilíneo.

Además, se hablará directamente de aceleración en vez de fuerza, ya que la masa del robot es constante, y ayudará a simplificar los cálculos.

La velocidad de salida para el primer tramo, debe ser calculada como la menor velocidad entre el límite de velocidad impuesto por el movimiento rectilíneo uniformemente acelerado, cuya fórmula es  $v_f = \sqrt{2 * a * x + v_i^2}$ , obtenido de la ecuación (1), sabiendo que tanto la aceleración, la velocidad inicial y la distancia, son conocidos, y el límite de velocidad impuesto por el movimiento curvilíneo uniforme, cuya fórmula es  $v = \sqrt{a * r}$ , obtenido de la ecuación (2), sabiendo que tanto la aceleración centrípeta y el radio de la curva son conocidos

Como la velocidad de salida del primer tramo es igual a la velocidad de entrada del segundo tramo, y a su vez la velocidad de entrada del segundo tramo, puede estar limitada aún más

que la velocidad de salida del tramo anterior, entonces, la velocidad de salida del tramo anterior debe también limitada por la velocidad de entrada del tramo siguiente.

Es decir, la velocidad de salida de cada tramo, debe ser la menor velocidad entre el límite impuesto por el movimiento rectilíneo uniformemente acelerado, por el límite impuesto por el movimiento circular uniforme y por el límite impuesto por la velocidad de entrada del tramo siguiente.

Esta iteración se debe realizar, desde el primer hasta el último tramo, y una vez realizado, debe recalcular si efectivamente las aceleraciones para cada tramo son las soportadas por el robot. En el caso de que haya alguna aceleración superior a la máxima soportada, se debe iterar las velocidades de entrada y de salida de los tramos que entran en conflicto, hasta llegar a las velocidades correctas.

Una vez obtenida las velocidades de entrada y salida de cada tramo, se definirá la rapidez instantánea del robot para cada parte del trayecto, (notar que se define la velocidad en función del trayecto, no del tiempo), de manera que acelere y desacelere de manera controlada para entrar y salir de cada trayecto de la manera más óptima posible, respetando las limitaciones.

## 6. Diseño del algoritmo de control del movimiento

Al principio del capítulo anterior, se habló sobre el movimiento ideal que debería tener un Robotracer, mostrando que existen dos clases movimientos complementarios que debe ejercer el robot para recorrer la pista.

La primera clase de movimiento es el que se encarga de mantener el robot alineado con la línea blanca central, cualquiera sea la velocidad traslacional que este lleve, que, para efectos de esta memoria, llamaremos “**Control de Posición Angular**”, mientras que la segunda clase de movimiento es el que le da al robot la velocidad de traslación en la pista, que para efectos de esta memoria llamaremos “**Control de Velocidad Traslacional**”.

Además, se propone una optimización del perfil de velocidades a partir de las limitantes físicas que posea el robot, para que pueda recorrer cualquier escenario reglamentado, en todo momento a la máxima velocidad posible, respetando las máximas aceleraciones que pueda tener, antes de derrapar.

De esta manera, este capítulo se centrará en el control del robot, separando modularmente el software en tres tipos de control. El control posición angular, control de velocidad traslacional, y la optimización del perfil de velocidades.

## 6.1. Consideraciones sobre la programación y el Entorno de Desarrollo.

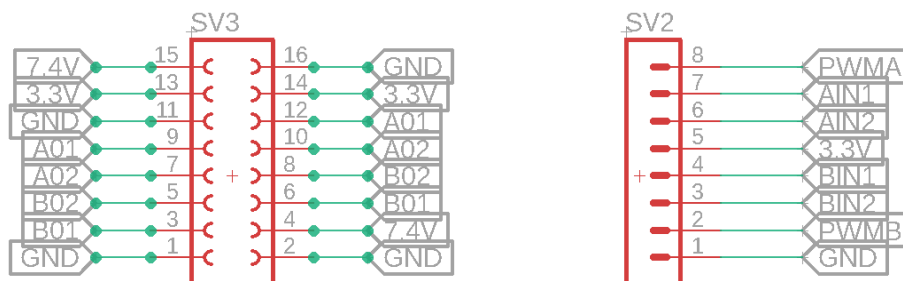
Para efectos de esta memoria, y debido al microcontrolador que se está utilizando, se ha determinado utilizar como entorno de desarrollo la plataforma Arduino IDE, cuya documentación se puede obtener en la website [Arduino.cc](http://Arduino.cc).

Además de utilizar el IDE de Arduino, se utilizarán otras plataformas de edición de texto, para la construcción de librerías para facilitar la construcción del código del software del robotracer.

## 6.2. Generación del movimiento

Para generar su movimiento el robot utiliza dos motores de corriente continua, que son energizados a por una batería LiPo de 7.4 Volts, a través del driver TOSHIBA TB6612FNG, que funciona como puente H. A su vez, el movimiento de los motores es controlado a través de pulsos PWM, generados por el microcontrolador.

El módulo TB6612FNG (ver figura 6.1) se conecta a la placa central, a través de pin Header hembra.



**Figura 6.1, Esquemático, Módulo Driver TB6612FNG**

Como se muestra en la tabla 6.1, obtenida del datasheet del Toshiba TB6612FNG, el control del motor se genera a través del siguiente esquema.

**Tabla 6.1, Control, Driver TB6612FNG**

| Input |     |     |      | Output                  |      |             |
|-------|-----|-----|------|-------------------------|------|-------------|
| IN1   | IN2 | PWM | STBY | OUT1                    | OUT2 | Mode        |
| H     | H   | H/L | H    | L                       | L    | Short brake |
| L     | H   | H   | H    | L                       | H    | CCW         |
|       |     | L   | H    | L                       | L    | Short brake |
| H     | L   | H   | H    | H                       | L    | CW          |
|       |     | L   | H    | L                       | L    | Short brake |
| L     | L   | H   | H    | OFF<br>(High impedance) |      | Stop        |
| H/L   | H/L | H/L | L    | OFF<br>(High impedance) |      | Standby     |

Cada canal posee tres entradas digitales y dos salidas de potencia. Dos de las entradas digitales general el sentido de giro, y la tercera entrada digital determina la potencia de salida a través de pulsos PWM. Por otro lado, las salidas de potencia se conectan directamente a los bornes del motor de corriente continua.

Para mejorar la programación del robot, se ha diseñado una librería que facilita la programación del movimiento a través de la siguiente función:

OpenTB6612FNG.Motores(int a, int b)

Donde “a y b” son variables de tipo entera entre -255 y 255, cuyo signo indica el sentido de giro del accionamiento (a: motor izquierdo, b: motor derecho), y el valor absoluto de la

variable indica la potencia inyectada a los motores, entre 0 y 255, siendo 0 lo mínimo y 255 lo máximo.

A modo de ejemplo:

OpenTB6612FNG.Motores(0, 0), ambos motores en potencia cero.

OpenTB6612FNG.Motores(125, 125), ambos motores al 50% de la potencia máxima, sentido horario.

OpenTB6612FNG.Motores(255, 0), Motor izquierdo al 100%, sentido horario, y motor derecho a 0%.

OpenTB6612FNG.Motores(-255, 255), Motor izquierdo al 100%, sentido antihorario, y motor derecho a 100%, sentido horario.

Cabe destacar que cuando el motor izquierdo gira en sentido antihorario, las ruedas izquierdas avanzan hacia adelante, y cuando el motor derecho gira en sentido horario, las ruedas de la derecha se mueven hacia adelante. La librería se muestra en los anexos.

### **6.3. Control de Posición Angular**

El control de línea tiene por objetivo que el robot se encuentre en todo momento centrado en la línea. Para lograr esta estabilización, se requiere tener una medición constante de la posición angular del robot respecto a la línea. Esto se logra con la ayuda de los sensores y receptores de luz infrarroja ubicados en la trompa.

Como se observó en la figura 4.31, la trompa posee 8 sensores QRE1113, capaces de medir la reflectancia de la superficie sobre la que se encuentra. Cada uno de los sensores están equidistantes en el arco que forma su estructura.

Para obtener una lectura de la alineación del robot respecto a la línea, se hará una lectura analógica de cada uno de los sensores QRE1113, a una resolución de 10 bits, Esto quiere decir, que se cuantificará el nivel de reflectancia de la superficie con un valor entre 0 y 1023, donde 0 equivaldrá a una lectura de blanco absoluto, y 1023 a una lectura de negro absoluto.

La obtención de la posición angular del robot respecto a la línea se hará con el siguiente algoritmo.

## 1. Lectura

Para cada lectura, se considera un valor residual de ruido debido a la luz infrarroja ambiental. Cabe recordar que la lectura que se hace es de la cantidad de luz infrarroja que rebota en la superficie en la que se está midiendo, por lo que la luz infrarroja ambiental, también puede rebotar en esta superficie. Para poder reducir el error en la medición de reflectancia, se hará una medición de esta, apagando los emisores de luz infrarroja, y luego se hará la medición con el emisor de luz infrarroja prendido. Finalmente, la reflectancia de la superficie, filtrando la luz residual será la diferencia entre la cantidad medida con el emisor prendido menos la cantidad medida con el emisor apagado.

Este proceso se hará para cada uno de los sensores de la trompa, almacenando el valor de la reflectancia en una matriz de con valores entre 0 y 1023 (10 bits). El valor analógico obtenido a través del conversor análogo digital será una cuantificación de la reflectancia de la superficie, donde mientras menos reflectante sea la superficie, mayor será el valor obtenido por el Conversor análogo digital (ADC); Esto quiere decir que, para superficies blancas, el ADC obtendrá valores cercanos a cero, y para superficies negras, el ADC obtendrá valores cercanos a 1023.

Cabe también mencionar que este algoritmo es funcional solo para el caso en que la luz infrarroja residual sea considerablemente menor a la luz infrarroja emitida por el diodo del sensor de reflectancia. Es decir, si el escenario está afectado por luz infrarroja residual de luminosidad cercana a los emisores que posee la trompa del robot, las lecturas no serán confiables y por lo tanto este algoritmo no será funcional.



## 2. Transformación

Los valores almacenados en la matriz deben ser transformados para cuantificar de la alineación del robot respecto al centro. Para eso, se utilizará el promedio ponderado de la reflectancia de cada uno de los sensores.

Luego de medir todos los puntos y almacenar los valores la, se hará una transformación aritmética, para que los valores obtenidos en las superficies oscuras sean cercanos a cero y los obtenidos en superficies blancas sean cercanos a 1023. Este valor se calculará de la siguiente manera.

$$\text{Valor corregio} = 1023 - \text{Valor medido}$$

Para determinar la posición angular relativa de robot, se utilizará el método propuesto en la publicación “An intelligent line-following robot project for introductory robot courses”, referencia [2], donde se propone calcular el promedio ponderado, para determinar la posición relativa del robot respecto a la línea.

Entonces:

$$\text{Posición} = \frac{1000 * \sum_{n=0}^7 VC_{n+1} * n}{\sum_{n=0}^7 VC_{n+1}} \quad (8)$$

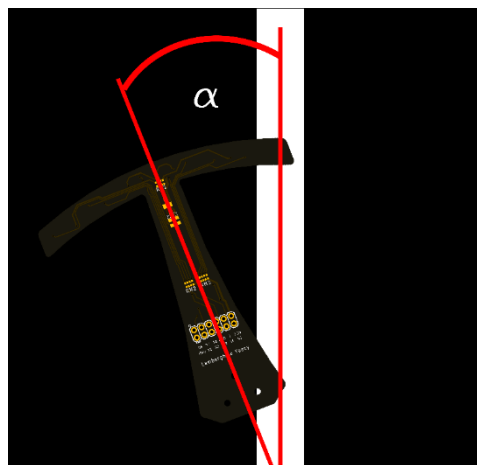
Con  $VC_n$  el valor de reflectancia corregido, almacenado en la matriz.

Finalmente, la posición dará como resultado un número entre 0 y 7000, siempre y cuando  $\sum_{n=0}^7 VC_{n+1} \neq 0$ .

Para simplificar el código, tanto la calibración como la obtención de la posición relativa de los sensores sobre la línea, se obtendrá a partir de la librería QTRSensors.H, desarrollada por la empresa estadounidense Pololu, cuya documentación se encuentra en los anexos.

De esta manera, para obtener el valor de la posición relativa del robot se hará llamando a la función ReadLine() cuyo retorno es un número entero entre 0 y 7000, que representa la posición relativa de la línea sobre los sensores, donde 3500 es el punto central de la trompa.

La trompa encontrará alineada con la pista, a medida que  $\alpha$  se acerque a cero, como muestra la figura 6.2.



**Figura 6.2, Alineación Sensor Frontal**

La función para obtener

$$\alpha = \frac{(Posición - 3.500) * (15,74)}{3.500} \quad (9)$$

$\alpha$  a partir del valor obtenido de la función ReadLine(), es

Una vez obtenido  $\alpha$ , se realizará un control de lazo cerrado para que el movimiento de los motores tienda a que  $\alpha$  se haga cero.

El controlador de lazo cerrado elegido para este procedimiento es un controlado PD. (Proporcional y Derivativo), como se muestra en la figura 6.3.

Este controlador trabaja recolectando los datos de la posición angular relativa entre el centro de la trompa y la línea blanca, para controlar la potencia de los motores a partir de pulsos PWM.

El proceso de estabilización comienza obteniendo la posición actual del robot, respecto a la línea, y esta posición es comparada con el setpoint, o posición ideal, que para este caso es 0 ( $\alpha = 0^\circ$ ). Esta comparación se hace encontrando el error de la posición actual respecto a la idea, siendo el error siendo  $pos_e(t) = pos\_Setpoint - Posición\ actual$

Cada medición se hará en periodos regulares de tiempo, y se obtendrá de este error una magnitud proporcional a este y una magnitud proporcional a la derivada de este error, de modo que:

$$pos\_Proporcional = K_p * e(t) \quad (10)$$

$$pos\_Derivativo = K_d * \frac{d(e(t))}{dt} \quad (11)$$

Finalmente, estas magnitudes se suman, obteniendo una magnitud final de salida, para el control angular. Los motores del robot serán potenciados con el valor de la magnitud final, considerando un sentido de giro opuesto para cada motor.

A modo de ejemplo, si

$$P_{giro} = pos\_Proporcional + pos\_Derivativo$$

Entonces, para mantener el control angular, se deben accionar los motores con la siguiente sentencia:

$$\text{OpenTB6612FNG.Motores}(-P_{giro}, P_{giro})$$

Este algoritmo controlará los motores para que la trompa tienda a centrarse en la posición  $\alpha = 0$ .

Consideraciones:

Las constantes  $K_p$  y  $K_d$  serán encontradas de manera manual.

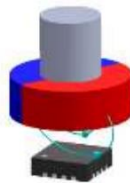
## 6.4. Control de Velocidad Traslacional

Como ya se habló anteriormente, una vez que el robot sea capaz de mantenerse centrado en la línea, es necesario generar otro tipo de control, en este caso es el control de velocidad traslacional, que, para efectos de esta memoria, se le llamará control de pista.

Para poder realizar el control de pista es necesario obtener un feedback fidedigno de la velocidad de giro, y para esto, se ha implementado en el diseño, un sensor de posición MA700.

Cuando el motor gira, su piñón mueve las ruedas dentadas de las ruedas, y estas a su vez mueven una pequeña rueda dentada ubicada en la parte trasera del sistema de engranajes, que mueve a su vez un eje, en cuya punta se ubica un imán polarizado diametralmente, como se observó en la figura 4.35.

Como bien se especifica en las consideraciones del ensamble completo, el sensor detecta la posición angular absoluta del imán permanente, magnetizado diametralmente, ubicado al final del eje trasero (ver figura 6.4). Este sensor posee la capacidad de adquisición y procesamiento rápido de datos, pudiendo medir velocidades de hasta 60.000 RPM.



**Figura 6.4, Ensamble imán - Encoder**

Estos datos son obtenidos en cada ciclo de trabajo, a través del protocolo SPI.

Para obtener un feedback de la velocidad del robot a partir de los datos obtenidos de los sensores de posición angular, se creó la librería Encoders.h, cuyo algoritmo se detalla en los anexos. Llamando a la función `iz_Get_speed_mm()`, se obtendrá la velocidad promedio de ambas ruedas en milímetros por segundo.

Midiendo la velocidad traslacional promedio del robot, se podrá hacer un control de lazo cerrado de esta, con un procedimiento similar al del control de lazo abierto de la posición angular, antes vista.

El control de pista comienza obteniendo la rapidez del robot, y esta posición es comparada con el setpoint que puede ser variable, dependiendo del perfil de velocidades que se quiera dar al robot. Esta comparación se hace encontrando el error de la velocidad actual respecto a la ideal, siendo el error  $e_{vel}(t) = Setpoint_{vel} - Velocidad_{actual}$

Cada medición se hará en periodos regulares de tiempo, y se obtendrá de este error una magnitud proporcional a este y una magnitud proporcional a la derivada de este error, de modo que:

$$vel\_Proporcional = vel\_K_p * e_{vel}(t)$$
$$vel\_Derivativo = vel\_K_d * \frac{d(e_{vel}(t))}{dt}$$

Finalmente, estas magnitudes se suman, obteniendo una magnitud final de salida, para el control de velocidad. Los motores del robot serán potenciados con el valor de la magnitud final.

A modo de ejemplo, si

$$P_{vel} = vel\_Proporcional + vel\_Derivativo$$

Entonces, para mantener el de velocidad, se deben accionar los motores con la siguiente sentencia:

$$\text{OpenTB6612FNG.Motores}(P_{vel}, P_{vel})$$

Este algoritmo controlará los motores para que la velocidad del robot tienda a estabilizarse en la velocidad previamente definida como `vel_setpoint`, en [mm/s].

Consideraciones:

Las constantes  $vel\_K_p$  y  $vel\_K_d$  serán encontradas de manera manual.

## 6.5. Combinación de movimientos

Como se definió anteriormente, el control de posición angular como el control de velocidad traslacional, se ejecutan mediante las funciones

$$\text{OpenTB6612FNG.Motores}(-P_{giro}, P_{giro})$$

Para el control de posición angular, y

$$\text{OpenTB6612FNG.Motores}(P_{vel}, P_{vel})$$

Para el control de velocidad traslacional.

Para combinar ambos movimientos, solo se necesitan sumar las potencias suministradas para cada motor por separado de la siguiente manera:

$$\text{OpenTB6612FNG.Motores}(P_{vel} - P_{giro}, P_{vel} + P_{giro})$$

Consideraciones:

Se debe tener en consideración las limitaciones de la potencia suministrable por software al sistema, sabiendo que tanto  $P_{vel} - P_{giro}$  como  $P_{vel} + P_{giro}$  deben estar en el rango de -255 a 255.



## 6.6. Optimización del perfil de velocidades

Como se mostró en el análisis del movimiento ideal, el robot fallará cuando exista una fuerza que lo haga derrapar; y el derrape se generará cuando la fuerza ejercida sobre el vehículo para sacarlo de la línea, sea superior a la fuerza de roce que lo mantiene en la línea. Por lo tanto, la velocidad ideal del robot dentro de la pista, se estimará considerando en todo momento, la máxima velocidad posible, considerando un movimiento sin derrape.

Para poder decidir cuál es la mejor velocidad para el vehículo, es necesario conocer las características de la curva. Para eso, se debe realizar un análisis de los datos recopilados por el vehículo, para lograr obtener las magnitudes claves de cada tramo.

El objetivo es obtener para cada tramo del escenario, entendiéndose como tramo la distancia entre dos marcas laterales izquierdas continuas, tanto la distancia total del tramo como el radio. Estos datos se pueden adquirir con la ayuda de los Encoders, con el siguiente procedimiento:

Todos los tramos serán tratados como curvas, incluso aunque sean visiblemente rectas, de esta manera, las rectas serán curvas de radio infinito, teóricamente.

Consideración: Antes de aplicar la ecuación 3, en la sección de mapeo de escenario, se asegurará que  $(m - n) \neq 0$ , para evitar un overflow en el microcontrolador. En el caso que  $(m - n) = 0$ , se establecerá el radio como 10.000 [mm], y se saltará el procedimiento de cálculo de radio.

Con la recolección de estos datos, se comenzará a calcular la optimización del perfil de velocidades, con el siguiente método:

- Se considera que tanto la aceleración centrípeta máxima  $a_c$  como la aceleración traslacional máxima  $a_t$  han sido calculadas empíricamente para este ejercicio.
- Se generará una tabla de n filas y 8 columnas.
- Cada fila corresponderá a cada tramo, enumerados desde el cero en adelante.
- Las columnas tendrán los siguientes datos:
  - Columna 1: Número del tramo (Tramo):

Corresponde al tramo, numerado desde el cero en adelante, donde el tramo cero es la distancia entre el centro del área de meta, hasta la primera marca lateral derecha.

- Columna 2: Distancia del tramo (Dist):

Corresponde a la distancia en milímetros entre dos marcas laterales, exceptuando el tramo cero y el tramo final, que corresponden cada uno a 500 milímetros. Se calcula con la ecuación (7)

- Columna 3: Radio del tramo (Radio)

Es el radio del tramo, en milímetros, calculado con la ecuación (6), teniendo en consideración que si  $(m - n) = 0$ , entonces el radio para el tramo se establece arbitrariamente en 10.000 milímetros.

- Columna 4: Velocidad límite (VL)

La velocidad límite es la velocidad en milímetros por segundo, calculada a despejando  $v$  de la fórmula  $a_{cm} = \frac{v^2}{r}$ . Esta velocidad se calcula para todos los tramos.

- Columna 5: Velocidad límite corregida (VLC)

Es la velocidad límite corregida, obtenida despejando  $v_f^2$  de la fórmula a partir de la ecuación del movimiento rectilíneo uniformemente acelerado  $v_f^2 - v_i^2 = 2 * a_{tm} * x$ , donde  $x$  es la distancia del tramo.

- Columna 6: Velocidad de entrada (Vin)

Es la velocidad de entrada para cada tramo, en milímetros por segundo

- Columna 7: Velocidad de salida (Vout)
- Es la velocidad de salida para cada tramo, en milímetros por segundo
- Columna 8: Aceleración necesaria. (Acc)

Recálculo de la aceleración, utilizando la ecuación de movimiento rectilíneo uniformemente acelerado  $v_f^2 - v_i^2 = 2 * a * x$ , donde  $v_i$  es la velocidad de entrada y  $v_f$  la velocidad de salida de del tramo

- Primero se comienza relleno las tres primeras columnas de la tabla. (el número de tramo, la distancia de cada tramo y el radio de cada tramo)
- A partir de esta información, se rellena la columna de velocidad límite, utilizando el radio antes calculado, y la aceleración centrípeta máxima, para cada uno de los tramos.
- Se establece la velocidad de entrada del tramo 0 como cero m/s.

A partir de este punto se deben repetir las siguientes instrucciones para cada tramo, uno por uno, siguiendo el orden correlativo desde el tramo cero hasta el final.

- Se calcula la velocidad límite corregida solo para el tramo.

- Se calcula la velocidad de salida solo para el tramo, que será igual al mínimo valor entre la velocidad límite, la velocidad límite corregida y la velocidad límite del siguiente tramo.
- Se establece la velocidad de entrada del tramo siguiente igual a la velocidad de salida del tramo actual.
- Se repiten los últimos cuatro pasos hasta el último tramo.
- Se comienza nuevamente desde el tramo cero, correlativamente hasta el último de los tramos, a calcular la aceleración necesaria, considerando la velocidad de entrada y de salida para cada tramo.
- Finalmente se comienza con un proceso de iteración, desde el último tramo, hasta el tramo cero, donde cada uno de los tramos cuya aceleración sea mayor a la permitida, comience a disminuir su velocidad de entrada, hasta que la aceleración máxima del tramo, esté bajo la magnitud permitida.

Consideraciones: Se tomará como punto de partida el centro del área de meta, y para eso, se generará una marca virtual equidistante entre las dos marcas laterales derechas que limitan el área de partida, de esta manera el tramo cero será el que está entre la marca virtual del inicio y la primera marca a la derecha.

El mismo punto de partida, será considerado como punto final, donde el robot deberá detenerse, por lo tanto, el último tramo será entre la última marca a la derecha y la marca virtual ubicada en el centro del área de meta.

## 6.7. Comentarios del diseño y mejoras

Uno de los problemas que se pudo observar dentro de la ejecución de las rutinas, es que si bien una trompa larga (sensor frontal), ayuda a que el robot reaccione anticipadamente a los giros bruscos, también provoca problemas en la alineación del robot, pudiendo generar que los sensores laterales no siempre pasen por sobre las marcas laterales.

En este sentido una de las mejoras que puede tener el robot, es hacer un estudio de la distancia ideal que deben tener los sensores laterales, para generar el perfecto complemento entre reacción, y lectura.

Una de las irregularidades que tomó más tiempo adaptar fue la medición correcta de la velocidad de giro del motor, ya que los gráficos mostraban oscilaciones periódicas de la velocidad crucero. En un principio se pensó que podía ser debido a la vibración interna del motor, pero luego de mediciones del subsistema de motores y enconder externas al robot, se comprobó que las oscilaciones provenían de una excentricidad entre el imán y el sensor de ángulo MA700. Para corregir este error, el fabricante del chip MA700 agrega una herramienta interna dentro del chip, para contrarrestar la excentricidad matemáticamente. Después de este ajuste, la velocidad crucero bajó considerablemente su oscilación.

Otro punto a considerar es que, debido a la rugosidad de las ruedas, éstas debían ser limpiadas constantemente por la acumulación de polvo. En las competencias tanto de Japón como de Taiwán, los escenarios están contruidos de madera MDF, previamente lijada y finalmente pintada con pintura negro mate; mientras que, en las competencias nacionales, se suelen utilizar melaminas negras cuya rugosidad es mucho menor. Por esta razón, en las competencias nacionales los robots deben disminuir sus aceleraciones constantes para realizar una correcta optimización del perfil de velocidades.

Un punto que no se tocó en esta memoria, fue la posibilidad de generar un algoritmo que considere atajos en los trayectos, por ejemplo, cuando se ve enfrentado a curvas pequeñas y seguidas, que no provocan desviación del trayecto del robot, dando la oportunidad al robot de omitir esa parte del escenario, tomándola como una recta.

Respecto a la ejecución del sistema operativo, se comprobó que el robot falla mucho menos en la ejecución de su rutina, cuando se consideran aceleraciones conservadoras para el segundo y tercer intento, por lo que se recomienda a futuros constructores, que la variable más importante a cambiar en el segundo y tercer intento, es la aceleración predeterminada, tanto para el movimiento curvilíneo uniforme como el movimiento rectilíneo uniformemente acelerado.

## 7. Resultados obtenidos

Para poner a prueba los resultados del diseño del control de movimiento se construyó una pista de las mismas dimensiones de la etapa preliminar del torneo All Japan Robotracer Contest 2015 cuyas dimensiones se pueden apreciar en la figura 3.9 mostrada en el tercer capítulo, con un total de 13,7 [m] de largo.

En un principio se realizaron pruebas de derrape y control para obtener empíricamente la máxima aceleración centrípeta que puede soportar el robot en las curvas; determinándose que la máxima aceleración centrípeta que puede soportar es de  $10.000 \left[ \frac{mm}{s^2} \right]$ . Además, se realizó el mismo ejercicio con la aceleración línea, y se pudo observar que la trompa del robot se levantaba cuando el robot sufría aceleraciones por sobre los  $12.000 \left[ \frac{mm}{s^2} \right]$ . Por esta razón, se decidió dejar como la aceleración que gobernará el sistema en  $8.000 \left[ \frac{mm}{s^2} \right]$ ,

Como la máxima aceleración del sistema se configuró en  $8.000 \left[ \frac{mm}{s^2} \right]$ , la velocidad de la primera vuelta se estableció 800 [mm/s], de esta manera si hubiese una curvatura de 100 [mm] de radio, el robot sufriría una aceleración de  $6.400 \left[ \frac{mm}{s^2} \right]$ , que podría ser soportada sin generar descontrol en este.

Además, para evitar el descontrol de la posición angular del robot, se ha limitado la velocidad máxima a la que éste puede avanzar, a 3.500 [mm/s], una vez que esté ejecutando la rutina con el perfil de velocidades optimizado.

Luego de la primera vuelta el robot obtuvo los valores para la distancia recorrida en cada tramo junto con su radio de curvatura, cuyos valores se pueden observar en la tabla 7.1. siendo Rec la distancia recorrida entre cada marca lateral en milímetros, Rec acum la distancia recorrida acumulada en milímetros y el tiempo de ejecución, el tiempo que demoró el robot en cada tramo.

| Tramo                 | Rec [mm] | Rec acum [mm] | Radio [mm]   | tiempo ejecución [s] |
|-----------------------|----------|---------------|--------------|----------------------|
| 0                     | 515      | 515           | 9.922        |                      |
| 1                     | 3.248    | 3763          | 79.693       | 4,060                |
| 2                     | 563      | 4326          | 189          | 0,704                |
| 3                     | 725      | 5051          | 7.458        | 0,906                |
| 4                     | 1.158    | 6209          | 251          | 1,448                |
| 5                     | 672      | 6881          | 715          | 0,840                |
| 6                     | 393      | 7274          | 151          | 0,491                |
| 7                     | 1.899    | 9173          | 21.826       | 2,374                |
| 8                     | 148      | 9321          | 2.780        | 0,185                |
| 9                     | 313      | 9634          | 5.781        | 0,391                |
| 10                    | 147      | 9781          | 10.092       | 0,184                |
| 11                    | 1.917    | 11698         | 5.294        | 2,396                |
| 12                    | 297      | 11995         | 177          | 0,371                |
| 13                    | 922      | 12917         | 2.555        | 1,153                |
| 14                    | 156      | 13073         | 98           | 0,195                |
| 15                    | 251      | 13324         | 2.212        | 0,314                |
| 16                    | 502      | 13826         | 1.411        |                      |
| Distancia competición |          | 12.809        | Tiempo total | 16,011               |

**Tabla 7.1, Valores primera vuelta, escenario de prueba.**



En los datos obtenidos se puede apreciar que la distancia total del escenario medida por el robot es de 13,826 [m]; valor muy cercano a los 13,7 [m] de longitud del valor teórico del recorrido total (error experimental menor al 1%). La diferencia entre ambas se puede explicar por errores en la construcción de la pista, por el derrape de las ruedas o un error atribuible a otra fuente no estudiada en esta memoria.

Luego de la primera vuelta, se ejecutó el algoritmo de optimización de velocidades, cuyos resultados se pueden observar en la tabla 7.2.

| Tramo                         | limite at [mm/s] | limite ac [mm/s] | vin [mm/s] | vout [mm/s] | Punto de corte [mm] | tiempo ejecución [s] |
|-------------------------------|------------------|------------------|------------|-------------|---------------------|----------------------|
| 0                             | 8.909            | 2.870            | 0          | 2870        | 515                 |                      |
| 1                             | 25.249           | 7.208            | 2870       | 1229        | 1414                | 1,028                |
| 2                             | 1.229            | 3.001            | 1229       | 1229        | 282                 | 0,458                |
| 3                             | 7.724            | 3.405            | 1229       | 1417        | 378                 | 0,355                |
| 4                             | 1.417            | 4.304            | 1417       | 1417        | 579                 | 0,817                |
| 5                             | 2.391            | 3.279            | 1417       | 1099        | 311                 | 0,350                |
| 6                             | 1.099            | 2.507            | 1099       | 1099        | 197                 | 0,357                |
| 7                             | 13.213           | 5.512            | 1099       | 1538        | 986                 | 0,713                |
| 8                             | 4.715            | 1.538            | 1538       | 1538        | 74                  | 0,086                |
| 9                             | 6.800            | 2.237            | 1538       | 1533        | 156                 | 0,167                |
| 10                            | 8.944            | 1.533            | 1533       | 1533        | 74                  | 0,086                |
| 11                            | 6.507            | 5.538            | 1533       | 1189        | 929                 | 0,712                |
| 12                            | 1.189            | 2.179            | 1189       | 1189        | 149                 | 0,249                |
| 13                            | 4.521            | 3.840            | 1189       | 885         | 441                 | 0,468                |
| 14                            | 885              | 1.579            | 885        | 885         | 78                  | 0,175                |
| 15                            | 4.206            | 2.003            | 885        | 2003        | 226                 | 0,163                |
| 16                            | 3.359            | 2.834            | 2003       | 0           | 126                 |                      |
| Tiempo Total de ejecución [s] |                  |                  |            |             |                     | 6,184                |

**Tabla 7.2, Optimización de perfil de velocidades, escenario de prueba.**

Siendo “límite at” la velocidad límite considerando la aceleración del sistema y la longitud del tramo, “límite ac” la velocidad límite considerando la aceleración centrípeta máxima del sistema, “vin” la velocidad de entrada al tramo, “vout” la velocidad de salida del tramo, “punto de corte”, la distancia desde el punto de inicio del tramo a la que el robot debe comenzar a desacelerar.

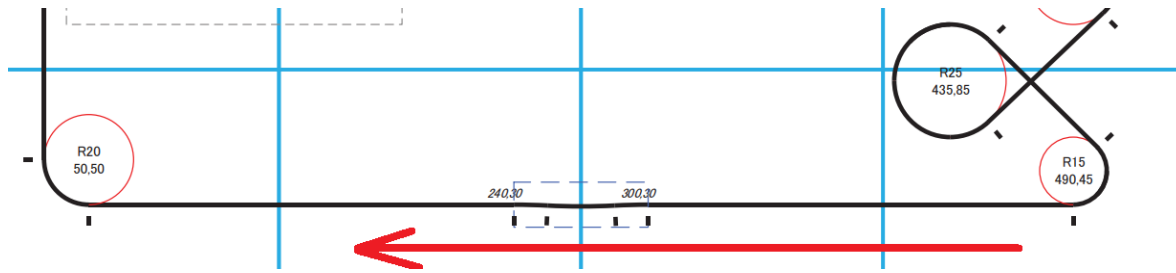
Como se puede observar, la optimización del perfil de velocidades redujo el tiempo de ejecución en 9,827 [s], respecto al tiempo de ejecución obtenido a la velocidad crucero de la primera vuelta.

El robot utiliza la optimización del perfil de velocidades para recorrer la pista a la máxima velocidad posible, evitando el descontrol del vehículo, como se puede observar gráficamente en la figura 7.1



**Figura 7.1, Gráfico de velocidad vs Tiempo, pista prueba**

Una de las optimizaciones que no se abordó en esta memoria fue la posibilidad de tomar atajos cuando el robot se enfrenta a marcas sin cambio aparente de curvatura como se observa en la figura 7.2.



**Figura 7.2, Atajos en pista de prueba**

Esta optimización podría haber sido posible gracias a la implementación del giroscopio del robot, haciendo que el robot deshabilite los sensores frontales y se guíe a partir de la desviación angular obtenida por el giroscopio aplicando un filtro de Kalman o similares.

## 8. Conclusiones

Se ha logrado describir el estado del arte de los Robotracers, su contexto latinoamericano y asiático y además la reglamentación que los gobierna. Además, se propuso un ranking de los mejores robots históricos de la categoría para basar un diseño a partir de uno de ellos.

Con esta información en el cuarto capítulo se propuso un diseño personalizado de robotracer considerando los componentes necesarios para poder implementar el algoritmo de optimización del perfil de velocidades. Algunos de los componentes propuestos fueron el microcontrolador, los motores, sensores de luz infrarroja, motores, encoders, drivers, y giroscopio.

Además, se propuso el diseño de un sistema de tracción que utiliza dos motores y cuatro ruedas, utilizando un sistema de engranajes.

Con el hardware del robot ya diseñado, se comenzó a describir analíticamente el robotracer, estableciendo el movimiento ideal que debe tener tanto del movimiento rectilíneo uniformemente acelerado como del movimiento curvilíneo uniforme; seguido del algoritmo para el mapeo del escenario, a partir de la información recolectada por los sensores y los encoders.

En el sexto capítulo de esta memoria se diseñó el algoritmo de control de movimiento, tanto para el control de la posición angular como para el control de la velocidad traslacional, basado en las ecuaciones que gobiernan el movimiento del robot en la pista.

Finalmente, el robot se probó en una pista cuyas dimensiones eran idénticas a un escenario real de la competencia All Japan Micromouse Contest, mostrando que la optimización logró reducir considerablemente el tiempo de ejecución de la performance del robot en la pista.

En esta memoria se ha documentado la experiencia del memorista, diseñando, construyendo y programando un Robotracer. Estas competencias debido a su progresiva normalización, serán cada año más populares alrededor de mundo, y se hace necesario generar documentación al respecto, para mejorar el nivel de los desarrolladores.

La experiencia de los desarrolladores japoneses, ha sido clave para crear esta documentación, ya que la mayor parte de este trabajo está basado en las experiencias adquiridas en competencias robóticas japonesas como la All Japan Micromouse and Robotracer Contest.

La proposición del diseño personalizado, requirió de mucha experiencia en el desarrollo de este tipo de robots, además de habilidades de diseño de circuitos electrónicos. El diseño y programación de este robot demoró más de dos años, no obstante, lo aprendido puede ser aplicado fácilmente en la industria para mejorar sus productos o procesos.

Este tipo de robots, que son técnicamente vehículos eléctricos autónomos en pequeña escala, pueden fácilmente relacionarse dinámicamente con el auto común, no obstante, la diferencia en la generación del movimiento, desde tracción trasera, y dirección delantera, hasta la generación de tracción a partir de la diferenciación de las velocidades de ambas ruedas, hizo un desafío no menor.

Utilizando el análisis Top - Down, es posible convertir un gran problema en muchos problemas pequeños. Esto fue lo que se hizo para generar el algoritmo de movimiento, partir de la determinación del movimiento ideal. Cuando se ve enfrentado a problemas mecatrónicos, utilizar este sistema es lo más recomendable.

Finalmente se pudo optimizar el perfil de velocidades, se pudo comprobar la efectividad del algoritmo en la competencia All Japan Robotracer Contest 2018, donde el robot obtuvo la posición 21, de más de 100 participantes.

## **9. Referencias**

- [1] Sears Zemansky, Física Universitaria, Decimosegunda Edición Addison – Wesley, 2009
- [2] (Juing-Huei Su, 2010)

## **10. Anexos**