2018

INSTRUMENTACIÓN ELECTRÓNICA CON SENSORES DE POSICIÓN PARA ROBOT RECOLECTOR DE PET

HERNÁNDEZ LÓPEZ NEYSER RIQUERMI

TECNOLÓGICO NACIONAL DE MÉXICO,

INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

1-12-2018

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

INFORME TÉCNICO
DE RESIDENCIA PROFESIONAL

INGENIERÍA ELECTRÓNICA

PRESENTA:

Hernández López Neyser Riquermi

NOMBRE DEL PROYECTO.

“Instrumentación electrónica con sensores de Posición para Robot recolector de pet**”**

PERIODO DE REALIZACIÓN:

AGOSTO - DICIEMBRE 2018

AGRADECIMIENTOS

Al finalizar un proyecto tan importante en la vida es grato reconocer el apoyo recibido por parte de mi familia que día a día ha estado a mi lado animándome para seguir adelante y terminar las metas que me he propuesto en esta vida, esta etapa de mi vida que concluyo con una enorme felicidad por haber superado cada obstáculo que se ha presentado a lo largo de este proceso, agradezco a mis padres por su esfuerzo y dedicación para poder darme lo mejor en el tiempo que me ha llevado realizar este proyecto.

Quiero expresar también mi más sincero agradecimiento al ing. ÁLVARO HERNÁNDEZ SOL por su apoyo y participación activa en el desarrollo de este proyecto, gracias por su paciencia y disponibilidad que hizo posible la realización mutua del proyecto elaborado, no cabe duda que su participación ha enriquecido el trabajo realizado y mis conocimientos en la ingeniería aplicada a un caso real.

Gracias a Dios por prestarme vida para terminar un logro más, con una gran satisfacción de haber concluido una etapa muy importante en la mejor institución educativa, con excelentes Docentes.

ÍNDICE

CAPÍTULO 1: GENERALIDADES DEL PROYECTO

1.1 INTRODUCCIÓN

La instrumentación es la base para el buen funcionamiento de un prototipo tecnológico, mediante la instrumentación adecuada y diseño favorable al ambiente donde se desarrollara e implementara el proyecto es posible llegar a obtener resultados óptimos para el beneficio de la sociedad o institución.

Para poder llevar a cabo la instrumentación de un robot recolector es necesario conocer el entorno en el que este va a operar así también como los factores que pueden llegar a ocasionar obstrucciones, problemas o hasta el funcionamiento no requerido del dispositivo, a continuación en el presente documento se hablara las características que se deben de tener en cuenta al aplicar la instrumentación de un robot recolector de pet.

Las características básicas que se deben de tener en cuenta al elegir los instrumentos que servirán al robot para la toma de decisiones y una automatización son: medio de aplicación, tipo de sensor, magnitud de medida, sensibilidad, precisión, resolución, etc.

1.2 DESCRIPCIÓN DEL ÁREA DE TRABAJO:

El TECNM Instituto Tecnológico de Tuxtla Gutiérrez es una institución de educación superior de tecnología, ubicada en la ciudad de Tuxtla Gutiérrez, Chiapas, que forma parte del Sistema Nacional de Institutos Tecnológicos de México, donde se desarrollara el proyecto “Instrumentación electrónica con sensores de posición para robot recolector de pet” en el departamento de Ing. Eléctrica y Electrónica, dentro del área de desarrollo de tecnología realizando la caracterización de sensores aplicados a un robot recolector de pet.

En conjunto con el asesoramiento del Ing. Álvaro Hernández Sol, en el periodo agosto-diciembre con las actividades de búsqueda de información relacionada con encoders, giroscopio, acelerómetros y ultrasónicos, posteriormente realizar la instalación y caracterización de los sensores de acuerdo a su funcionamiento que presente en el recolector de pet, diseño de bases y contenedores, como la realización de pruebas y corrección de errores presentados.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL:

Caracterizar encoders, giroscopios, acelerómetros, ultrasónicos aplicados a un robot recolector de pet.

1.3.2 OBJETIVO ESPECIFICOS:

- Consultar fuentes de información relacionada con sensores aplicados a un robot recolector de pet.
- Caracterizar sensores de posición: encoders, giroscopios, acelerómetro y ultrasónicos.
- Implementar sensores de posición encoders giroscopios, acelerómetro y ultrasónicos.

1.4 JUSTIFICACIÓN

La contaminación por plástico está causando daños irreparables al planeta. Lejos de disminuir, este problema se agudiza cada día.

Anualmente se producen 300 millones de toneladas de plástico, de las cuales un alto porcentaje termina en ríos y mares. Muchos científicos consideran que este es un problema más grave que el cambio climático.

En los océanos hay islas de plástico del tamaño de continentes. Se estima que para el año 2050 habrá más plástico que peces en el mar. Se han encontrado fibras plásticas tanto en el polo norte como en el polo sur.

El plástico tarda cientos de años en degradarse, contiene aditivos y adsorbe metales pesados, antibióticos, pesticidas y otros tóxicos. Estos son transportados por todo el planeta.

Hoy en día una de las problemáticas más grandes y de afectación en general es la contaminación, por ello la importancia de la instrumentación electrónica con sensores de posición para robot recolector de pet para ayudar a reciclar desechos plásticos con la recolección en diversos ambientes he de aquí la utilización de encoders, giroscopios, acelerómetros y ultrasónicos que ayudaran al robot a tomar decisiones y lograr un óptimo resultado.

1.5 PROBLEMA A RESOLVER

El instituto Tecnológico Nacional de México, Instituto tecnológico de Tuxtla es una institución orientada a ingenierías que se preocupa por el medio ambiente es por ello que se pretende ayudar al medio ambiente en el tema de contaminación que hoy en día es el mayor problema a nivel mundial.

En este proyecto se pretende ayudar a la recolección de desechos pet, mediante un robot recolector este será equipado con sensores: encoders, giroscopios, acelerómetros y ultrasónicos. Estos deben ser calibrados y probados para lograr un buen funcionamiento del robot recolector de pet.

1.6 ALCANCES Y LIMITACIONES

1.6.1 ALCANCES

- Que el proyecto realizado sirva como antecedentes a las empresas que requieran construir y ayudar al medio ambiente mediante robots recolectores.
- Que el funcionamiento de los sensores sean empleados de manera adecuada y óptima.
- El diseño del software aporte el rendimiento adecuado al diseño del robot recolector.

1.6.2 LIMITACIONES

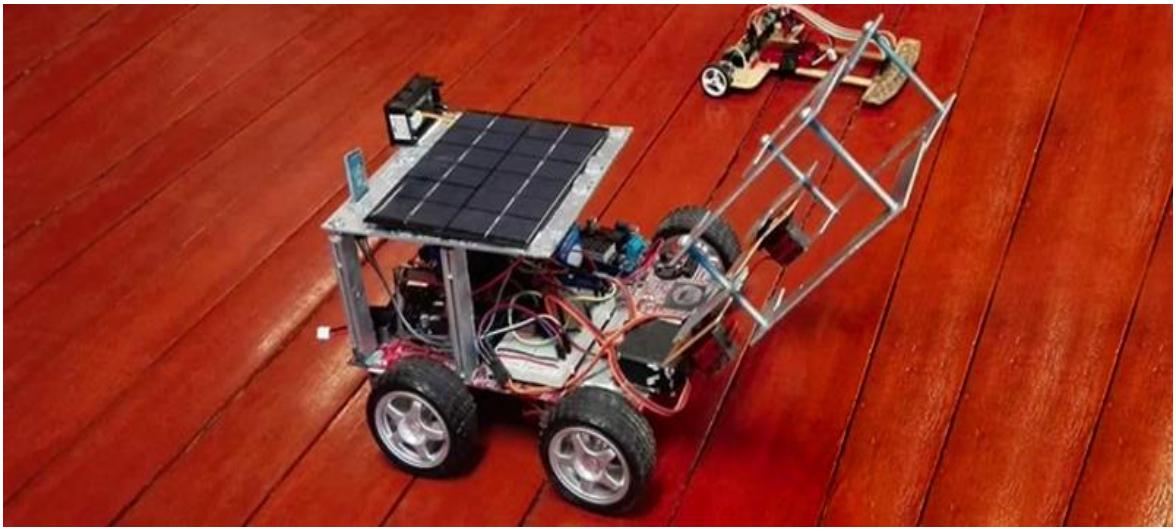
- Por falta de tiempo no se logró realizar todas las pruebas y correcciones necesarias.
- El prototipo utilizado no se pudo manipular los 4 motores al mismo tiempo.
- El prototipo permite hasta un peso límite por la característica de los motores.

CAPÍTULO 2: FUNDAMENTOS TEÓRICOS

2.1 ESTADO DEL ARTE

Crean robot recolector de desechos (Conacyt)

Oaxaca de Juárez, Oaxaca. 11 de mayo de 2016 (Agencia Informativa Conacyt).- Con la intención de crear una herramienta que auxilie el cuidado del medio ambiente, el estudiante de ingeniería en sistemas computacionales del Instituto de Estudios Superiores del Istmo de Tehuantepec (IESIT), Dalí López López, diseñó y fabricó un robot recolector de basura a partir de material reciclado y que puede ser manipulado a través de un teléfono celular o una tableta.



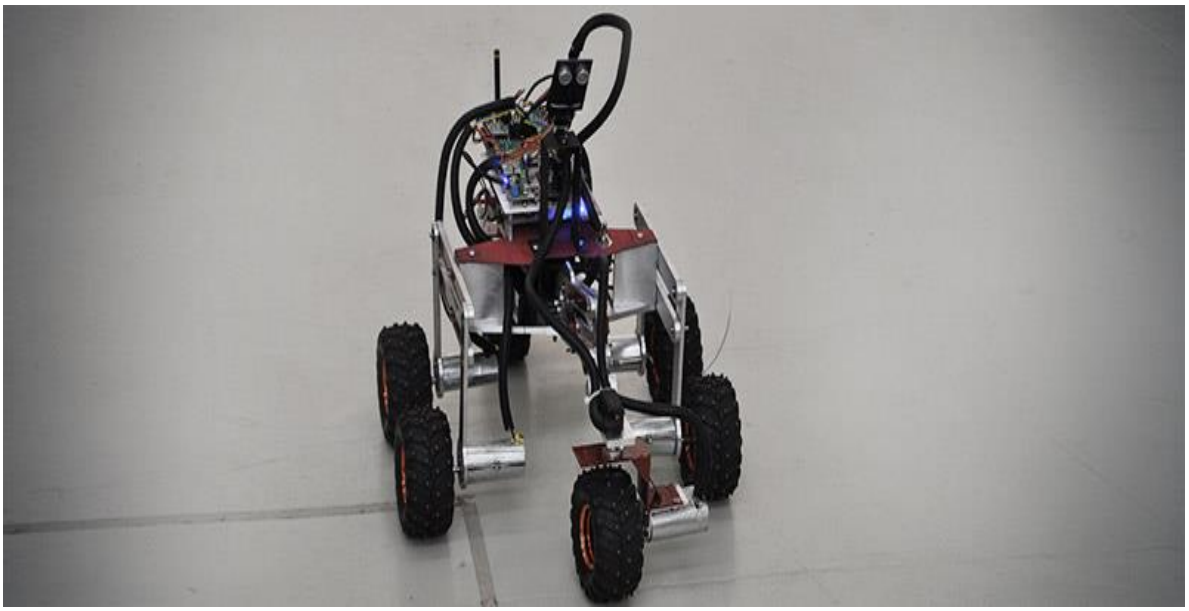
El robot fue nombrado Xtuxhu stinu, que en zapoteco significa "nuestro resplandor", y fue elaborado a partir de materiales electrónicos reciclados, como tarjetas madre de computadoras dañadas, placas fenólicas de circuitos quemados y otros elementos reutilizados como es el caso de servomotores y motor reductores.

En cuanto al funcionamiento, se controla a través de una aplicación móvil con la tecnología Bluetooth. El robot tiene un módulo Bluetooth que se sincroniza con la aplicación en el móvil, una vez sincronizado simplemente se conecta y se puede manipular. La aplicación es compatible con el sistema operativo Android y es muy fácil de operar, tiene los botones de movimiento y de control de las pinzas que auxilian con

la recolección de desechos. Para la fabricación del prototipo contó con la asesoría del ingeniero Heliodoro Jiménez Sánchez, jefe del Departamento de Ingeniería en Sistemas Computacionales del IESIT, con quien también colabora en el Club de Desarrollo Tecnológico “Ba’du Guixhí”, en donde pretenden fomentar el gusto por el área tecnológica auxiliando en la creación de proyectos y prototipos en distintas áreas como robótica, aplicaciones móviles y desarrollo de software.

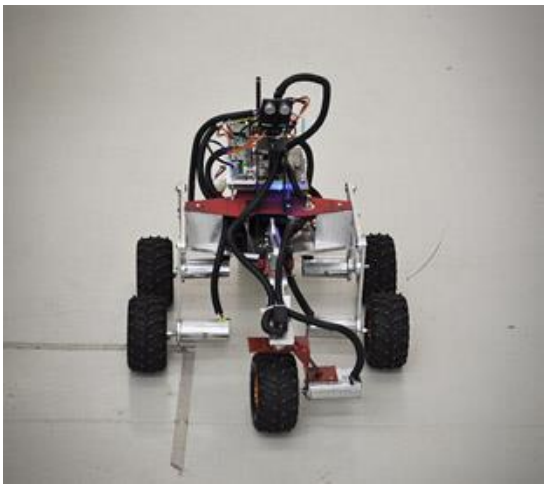
Estudiantes crean robot móvil tele operado

Zacatecas, Zacatecas. 21 de octubre de 2015. (Agencia Informativa Conacyt).- Con apoyo del Consejo Zacatecano de Ciencia y Tecnología (Cozcyt), un equipo de estudiantes de la Unidad Profesional Interdisciplinaria de Ingeniería del Instituto Politécnico Nacional, Campus Zacatecas (IPN UPIIZ) fabricaron un robot móvil tele operado de locomoción híbrida para exploración de terrenos irregulares.



El equipo que realizó la investigación y consecuente realización del robot está conformado por Edgar Eduardo Haro Campos, Omar Moctezuma Barraza, Ulises Muro Barajas y Juan Martín del Río Chacón, bajo asesoramiento del doctor Miguel Ángel Moreno Báez y del maestro Fernando Olivera Domingo. Dicho robot ofrece al usuario, describiéndolo como una herramienta interactiva, práctica, didáctica, así como fácil de operar y maniobrar, ya que se puede controlar a distancia.

Anatomía del robot



El prototipo se divide en varias partes, la primera es el chasís, que es el armazón que sostiene un motor y la carrocería de un vehículo. El chasís del prototipo está diseñado y maquinado sobre material de aluminio, que es el material que cumplió con los lineamientos de masa, ya que a diferencia de materiales más pesados como el acero, el aluminio no se oxida, es más ligero y económico; además de que resulta maleable al momento de ser maquinado.

Las llantas fueron obtenidas con base en un estudio de criterio de selección acerca del mejor prototipo en ámbito mecánico, por lo que se decidió por llantas de uso rudo. Cerebro electrónico

La parte electrónica tiene que ver con todos los componentes electrónicos que permiten a los microcontroladores realizar su tarea. Entonces los realizadores del robot diseñaron, construyeron y llevaron a cabo las pruebas pertinentes dando como resultado las tres placas que componen el robot. En ellas están distribuidos los microcontroladores esclavos y el esclavo maestro (cerebro central). Para diseñarlas utilizaron un software llamado Eagle, que es el que permite diseñar tarjetas de circuito impreso. El cerebro central está compuesto por un Arduino Mega 2560.

La comunicación se lleva a cabo por medio de unas tarjetas llamadas XBee Pro, que tienen un alcance de 600 metros al aire libre. Esto es lo que permite la comunicación tele operada.

El protocolo de comunicación utilizado es el nombrado I^2C, en el cual el usuario realiza indicaciones de velocidad al cerebro central y este a su vez emite la señal a los esclavos. Los esclavos lo señalan al motor y el motor responde trasladándose a la velocidad indicada inicialmente por el usuario, quien lo comprueba visualmente a través de la pantalla.

Energía

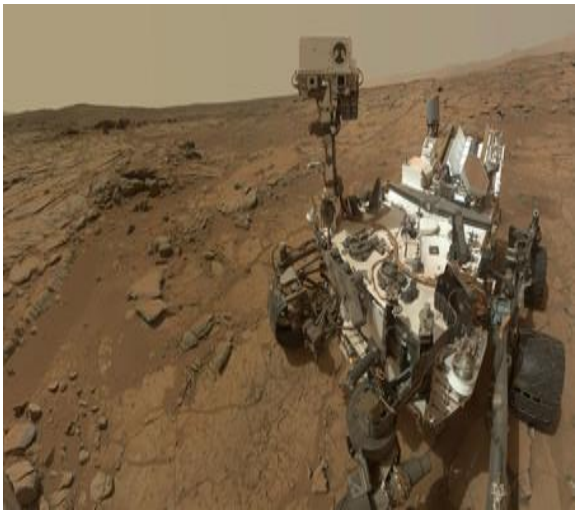
Los motores son alimentados por una batería llamada NiMH de 12 voltios y 10 mil amperios por hora; mientras que los componentes electrónicos se conectan a la batería nombrada Lipo de 11.1 voltios y dos mil 700 amperios por hora.

Robot móvil aeroespacial

La agencia informativa del Consejo Nacional de Ciencia y Tecnología (Conacyt) ha dado a conocer que un grupo de estudiantes del Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), campus Querétaro, representarán a México en la competencia University Rover Challenge que se lleva a cabo cada año por parte de la Mars Society y la NASA.

Son 12 los estudiantes integrantes del equipo Eagle X Robotics que ganó el primer lugar en la competencia de Investigación Espacial del Vive conCiencia 2017. El proyecto que los hizo acreedores de este lugar fue el desarrollo de un robot móvil aeroespacial como el que realiza labores de investigación en la superficie de Marte.

Emiliano Castillo Specia, estudiante de Sistemas Digitales y Robótica, explica que el robot se compone de "un sistema mecánico de 6 llantas con tres ejes de suspensión, lo que le permite navegar por diferentes tipos de terrenos blandos". Además tiene un sistema operativo robótico que administra las tareas de visión para reconocimiento de terreno, inteligencia artificial, potencia de motores, entre otros aspectos necesarios para su operación.



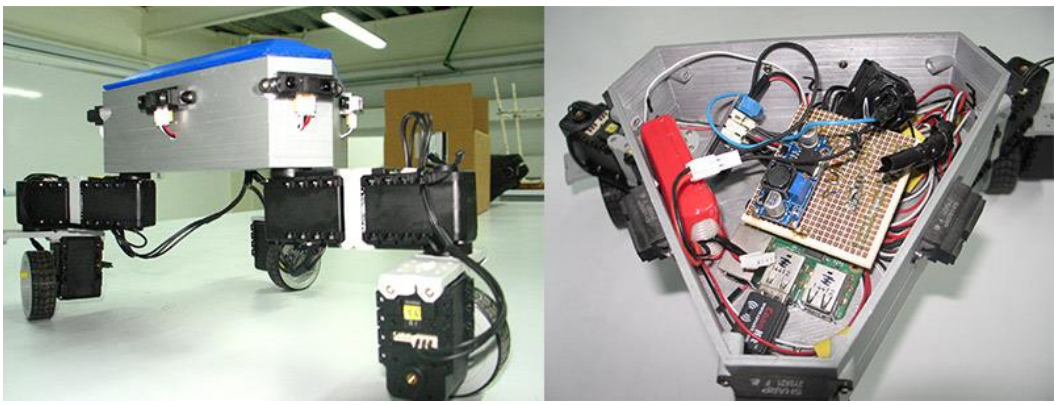
Robot aeroespacial Mars Rover de la NASA

Cuenta con un conjunto de cámaras web para la transmisión de video y un brazo robótico, capaz de manejarse de manera independiente al movimiento, que puede levantar hasta seis kilogramos. Las antenas integradas en el diseño permiten un manejo con alcance de hasta un kilómetro.

Este robot del más puro estilo Mars Rover se ha ganado el derecho a competir en la edición 2018 de la competencia University Rover Challengeen donde deberá cumplir con cuatro tareas entre las que se encuentran pruebas de Ph y humedad, navegación por terrenos irregulares, entre otras. Todo simulando como si se trata de una situación en el planeta rojo.

Inteligencia artificial en robótica

Santiago de Querétaro, Querétaro 28 de noviembre de 2017 (Agencia Informativa Conacyt).- Investigadores del área de Inteligencia Artificial de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro (UAQ), diseñaron el prototipo de un robot móvil omnidireccional, que mediante algoritmos y una interfaz de operación, le permiten el aprendizaje, planeación y control de movimientos.



El diseño y programación de este robot estuvo a cargo del egresado del doctorado en Ingeniería de la UAQ, Daniel García Sillas con la asesoría del profesor investigador de esa facultad, Efrén Gorrostieta Hurtado, quien aseguró que este prototipo de robot busca introducir herramientas de inteligencia artificial para hacerlo más autónomo, capaz de aprender y planear movimientos determinados.

Dr. Efrén Gorrostieta Hurtado."Esta es la quinta versión hablando del diseño del robot, la idea principal era hacer un diseño omnidireccional que tuviera brazos con aplicaciones de algoritmos de inteligencia artificial. Principalmente se trabajaron algoritmos de aprendizaje, que en este caso fue evasiones de obstáculos y planeación, donde el robot tiene que buscar una trayectoria muy específica con ciertos criterios de optimización. Se le enseñaron algunos movimientos y que aprendiera de experiencias al chocar con obstáculos".

Diseñan vehículo robótico para explorar zonas agrestes

Ensenada, Baja California. 14 de noviembre de 2018 (Agencia Informativa Conacyt).- Los robots diseñados para misiones espaciales tienen características que son aplicables en la Tierra, como por ejemplo, su capacidad para transitar por terrenos agrestes.

Especialistas del Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN) trabajan en el diseño y puesta en operación de un vehículo de exploración tipo rover, con el potencial para utilizarse en tareas de búsqueda en terrenos desconocidos y no estructurados.

En entrevista para la Agencia Informativa Conacyt, el doctor Juan Humberto Sossa Azuela, jefe del Laboratorio de Robótica y Mecatrónica del CIC, explicó que el diseño del vehículo robótico conlleva un proceso similar al de los robots que se desarrollan con fines de exploración espacial.

Robots limpiadores de playas para atender problemas ecológicos

Xalapa, Veracruz. 16 de abril de 2018 (Agencia Informativa Conacyt).- La categoría de robots limpiadores de playas, del Torneo Mexicano de Robótica (TMR), intenta contribuir desde la ciencia y la tecnología a la solución de problemas ecológicos que existen actualmente en México, de acuerdo con la Federación Mexicana de Robótica (FMR).

La FMR busca aportar soluciones a los problemas de basura en las playas, así como concientizar a la ciudadanía para tomar acciones al respecto para tener en cuenta cuáles son las causas que producen contaminación con mayor regularidad, derivando daños severos en el ecosistema marino, la salud de las personas y la empresa pesquera: acumulación de basura, descarga de aguas residuales y derrame de petróleo.

Los robots deben buscar los residuos que están dispersos sobre el escenario, llevarlos e introducirlos dentro de un depósito, sin mover o tocar otros objetos, como maniquíes, sombrillas de playa y silla de sol, instalados para elevar la dificultad de la tarea del autómata.

2.2 ROBOTS

Un robot es una entidad virtual o mecánica artificial. En la práctica, esto es por lo general un sistema electromecánico que normalmente es conducido por un programa de una computadora o por un circuito eléctrico. Este sistema electromecánico, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio. La independencia creada en sus movimientos hace que sus acciones sean la razón de un estudio razonable y profundo en el área de la ciencia y tecnología. La limpieza y el mantenimiento del hogar son cada vez más comunes en los hogares. No obstante, existe una cierta incertidumbre sobre el impacto económico de la programación y la amenaza del equipamiento robótico, una ansiedad que se ve reflejada en el retrato a menudo perverso y malvado de robots presentes en obras de la cultura popular. Comparados con sus colegas abe de ficción, los robots reales siguen siendo limitados. La palabra robótica, usada para describir este campo de estudio, fue acuñada por el escritor de ciencia ficción Isaac Asimov. La robótica concentra 3 áreas de estudio: la mecatrónica, la física y las matemáticas como ciencias básicas.

2.2.1 ARQUITECTURA DE LOS ROBOTS

Existen diferentes tipos y clases de robots, entre ellos con forma humana, de animales, de plantas o incluso de elementos arquitectónicos pero todos se diferencian por sus capacidades y se clasifican en 4 formas:

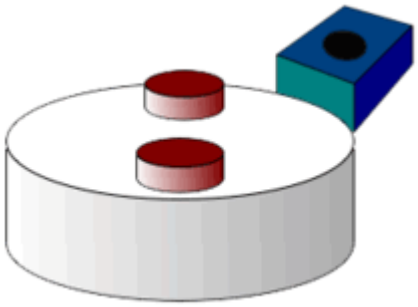
1. Androides: robots con forma humana. Imitan el comportamiento de las personas, su utilidad en la actualidad es de solo experimentación. El principal limitante de este modelo es la implementación del equilibrio en el desplazamiento, pues es bípedo.

2. Móviles: se desplazan mediante una plataforma rodante (ruedas); estos robots aseguran el transporte de piezas de un punto a otro.
3. Zoomórficos: es un sistema de locomoción imitando a los animales. La aplicación de estos robots sirve, sobre todo, para el estudio de volcanes y exploración espacial.
4. Poliarticulados: mueven sus extremidades con pocos grados de libertad. Su principal utilidad es industrial, para desplazar elementos que requieren cuidados.

En esta última se puede clasificar según su morfología en: Robots angulares o antropomórficos, robots cilíndricos, robots esféricos o polares, robots tipo SCARA, robots paralelos, robots cartesianos, entre otros.

En nuestro caso nos enfocaremos en robots móviles “se desplazan mediante una plataforma rodante (ruedas); estos robots aseguran el transporte de piezas de un punto a otro”, esto para llevar a cabo la recolección de desechos pet (plástico) y el transporte de estos en contenedores que el propio robot incluirá.

2.3 SENSOR



Un sensor es todo aquello que tiene una propiedad sensible a una magnitud del medio, y al variar esta magnitud también varía con cierta intensidad la propiedad, es decir, manifiesta la presencia de dicha magnitud, y también su medida.

Un sensor en la industria es un objeto capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica, etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la magnitud que la condiciona o variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo.

Áreas de aplicación de los sensores:Industria automotriz, robótica, industria aeroespacial, medicina, industria de manufactura, etc.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a la toma de valores desde el sensor, una base de datos, etc.

2.4 PROGRAMACIÓN

La programación es un proceso que se utiliza para idear y ordenar las acciones que se realizarán en el marco de un proyecto; al anuncio de las partes que componen un acto o espectáculo; a la preparación de máquinas para que cumplan con una cierta tarea en un momento determinado; a la elaboración de programas para la resolución de problemas mediante ordenadores; y a la preparación de los datos necesarios para obtener una solución de un problema.

En la actualidad, la noción de programación se encuentra muy asociada a la creación de aplicaciones informáticas y videojuegos; es el proceso por el cual una persona desarrolla un programa valiéndose de una herramienta que le permita escribir el código (el cual puede estar en uno o varios lenguajes, tales como C++, Java, Python entre otros) y de otra que sea capaz de “traducirlo” a lo que se conoce como lenguaje de máquina, el cual puede ser entendido por un microprocesador.

2.4.1 PROGRAMAS Y ALGORITMOS

Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que han de seguirse para resolver un problema. Un programa normalmente implementa (traduce a un lenguaje de programación concreto) uno o más algoritmos. Un algoritmo puede expresarse de distintas maneras: en forma gráfica, como un diagrama de flujo, en forma de código como en pseudocódigo o un lenguaje de programación, en forma explicativa.

Los programas suelen subdividirse en partes menores, llamadas módulos, de modo que la complejidad algorítmica de cada una de las partes sea menor que la del programa completo, lo cual ayuda al desarrollo del programa. Esta es una práctica muy utilizada y se conoce como "refino progresivo".

Según Niklaus Wirth, un programa está formado por los algoritmos y la estructura de datos.

La programación puede seguir muchos enfoques, o paradigmas, es decir, diversas maneras de formular la resolución de un problema dado. Algunos de los principales paradigmas de la programación son:

- Programación declarativa
- Programación estructurada
- Programación modular
- Programación orientada a objetos

2.5 ARDUINO

Arduino Uno SMD R3

Información	
Tipo	Single-Board Computer (SBC)
Datos técnicos	
Alimentación	USB/Baterías/Fuentes de poder.
Procesador	Atmel AVR (8bit), ARM Cortex-M0+ (32-bit), ARM Cortex-M3 (32bit), Intel Quark (x86) (32-bit).
Memoria	SRAM
Almacenamiento	Flash, EEPROM.
Soporte	Placas de expansión (shields).
Software	
Sistema operativo	multiplataforma
Otros datos	
Arduino	
https://www.arduino.cc/, http://www.arduino.org/ y http://www.arduino.cc/	

Arduino es una compañía de fuente abierta y hardware abierto así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

Los diseños de las placas Arduino usan diversos microcontroladores y microprocesadores. Generalmente el hardware consiste de un microcontrolador Atmel AVR, conectado bajo la configuración de "sistema mínimo" sobre una placa de circuito impreso a la que se le pueden conectar placas de expansión (shields) a través de la disposición de los puertos de entrada y salida presentes en la placa seleccionada. Las shields complementan la funcionalidad del modelo de placa empleada, agregando circuitería, sensores y módulos de comunicación externos a la placa original. La mayoría de las placas Arduino pueden ser energizadas por un puerto USB o un puerto barrel Jack de 2.5mm. La mayoría de las placas Arduino pueden ser programadas a través del puerto serie que incorporan haciendo uso del Bootloader que traen programado por defecto. El *software* de Arduino consiste de dos elementos: un entorno de desarrollo (IDE) (basado en el entorno de *processing* y en la

estructura del lenguaje de programación Wiring), y en el cargador de arranque (*bootloader,* por su traducción al inglés) que es ejecutado de forma automática dentro del microcontrolador en cuanto este se enciende. Las placas Arduino se programan mediante un computador, usando comunicación serie.

2.5.1 HARDWARE ARDUINO

Arduino es hardware libre. Los diseños de referencia de hardware se distribuyen bajo licencia Creative Commons Attribution Share-Alike 2.5 y están disponibles en el sitio web de Arduino. Los esquemáticos y archivos de montaje de componentes (PCBs) para algunas versiones de placas también están disponibles.

La mayoría de las placas Arduino constan de un microcontrolador AVR Atmel-8 bits (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560), cada microcontrolador consta de diversas cantidades de memoria flash, pines y funciones. Las placas utilizan pines/cabezales hembra de una o dos hileras que facilitan las conexiones e incorporación en otros circuitos. La mayoría de las placas incluyen un regulador lineal de 5 V y un oscilador de cristal de 16 MHz, o un resonador de cerámica según sea el caso. Algunos diseños, como el LilyPad, funcionan a 8 MHz y prescinden del regulador de voltaje a bordo debido a restricciones de factor/tamaño de forma específicas.

Los modelos de Arduino se categorizan en placas de desarrollo, placas de expansión (*shields*), kits, accesorios e impresoras 3D.

- Placas: Arduino Galileo, Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre (En Desarrollo), Arduino Zero, Arduino Micro, Arduino Esplora, Arduino Mega ADK, Arduino Ethernet, Arduino Mega 2560, Arduino Robot, Arduino Mini, Arduino Nano, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, LilyPad Arduino USB, Arduino Pro Mini, Arduino Fio, Arduino Pro, Arduino MKR1000/Genuino MKR1000, Arduino MICRO/Genuino MICRO, Arduino 101/Genuino 101, Arduino Gemma.

- Placas de expansión (*shields*): Arduino GSM Shield, Arduino Ethernet Shield, Arduino WiFi Shield, Arduino Wireless SD Shield, Arduino USB Host Shield, Arduino Motor Shield, Arduino Wireless Proto Shield, Arduino Proto Shield.
- Kits: The Arduino Starter Kit, Arduino Materia 101.
- Accesorios: Pantalla LCD TFT, Adaptador USB/Serie y MiniUSB/Serie, Arduino ISP.
- Impresoras 3d: Arduino Materia 101.

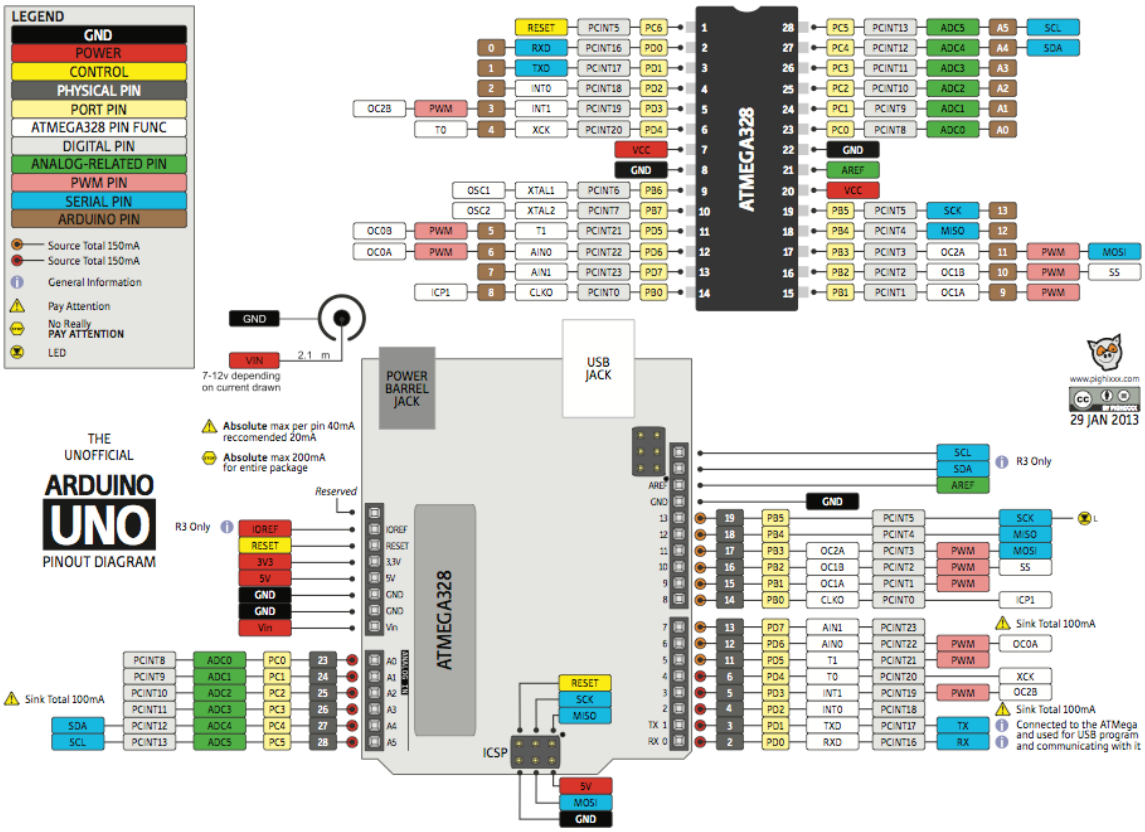
2.5.2 PLACAS ARDUINO Y CARACTERÍSTICAS

	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

La tarjeta de Arduino Uno consiste en una placa electrónica que tiene un microprocesador Atmega328; 14 pines digitales de entrada/salida, de los cuales 6 pueden utilizarse como salidas PWM(modulación de ancho de pulso); 6 entradas analógicas; un resonador cerámico de 16 MHz; una conexión USB; un conector de alimentación; un microcontrolador y un botón de reinicio.

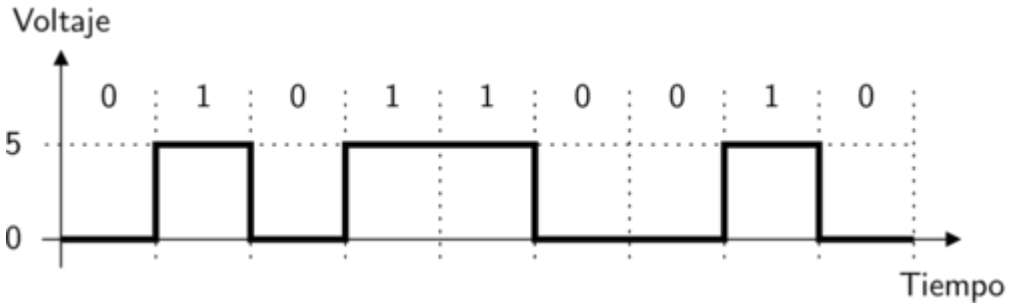
2.5.3 ESTRUCTURA DEL ARDUINO UNO

Prácticamente veremos la tarjeta Arduino UNO pero en general las diferentes tarjetas de Arduino son muy similares, cuentan con pines tanto de salida como de entrada con los cuales podremos leer nuestros dispositivos ya sea una señal de algún sensor u otro parámetro. También enviar señales o datos por los pines de salida los cuales veremos cómo funciona más adelante para usar los Actuadores analógicos y digitales. Aquí se presenta una imagen de la estructura, recordemos que usa un microcontrolador ATMEGA328 para que funcionen todos nuestros dispositivos.

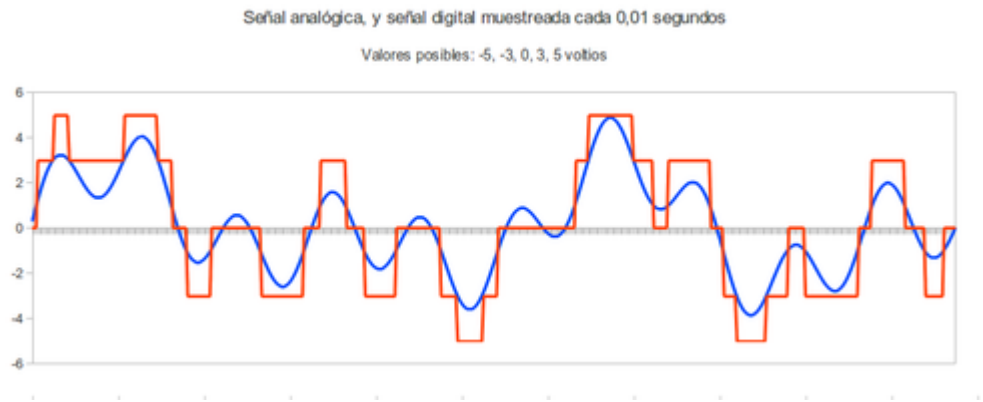


Pines de nuestra tarjeta

Nuestra tarjeta tiene 14 pines digitales del 0 al 13, de los cuales podemos leer y enviar señales digitales que van de 0 a 5 volts, además entre esos pines se cuenta con 6 pines PWM los cuales los veremos más adelante, aquí una imagen de señal digital:



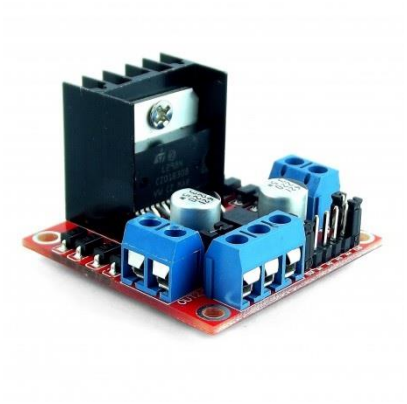
También cuenta con 6 pines analógicos, como lo dice su nombre podremos hacer lecturas analógicas igualmente de 0 a 5 volts, estos pines también se pueden usar como digitales por medio del convertidor analógico digital, aquí una imagen de señal analógica:



Recordemos que cada pin trabaja con voltajes de 0 a 5 volts CC, además que la máxima corriente por pin es de 40mA, si utilizaremos un actuador que pide más corriente que la entregada por un pin es necesario usar un transistor de potencia, pero eso se verá más adelante.

La tarjeta consta de un regulador de voltaje, un 7805 conectado al Jack y al pin vin de la tarjeta con su respectivo diodo de protección, pero esto no evita el tener precaución y no invertir la polaridad en los pines de alimentación haciendo un cortocircuito.

2.6 DRIVER PUENTE H L298N



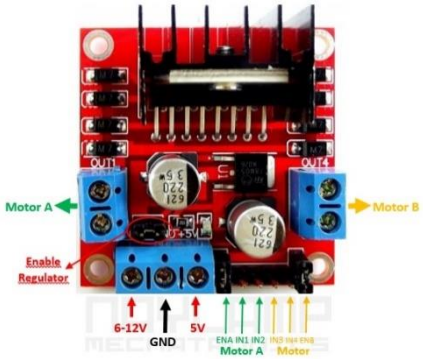
Este módulo posee dos puentes H que permiten controlar 2 motores DC o un motor paso a paso bipolar/unipolar.

El módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi o Launchpads de Texas Instruments.

Tiene integrado un regulador de voltaje de 5V encargado de alimentar la parte lógica del L298N, el uso de este regulador se hace a través de un Jumper y se puede usar para alimentar la etapa de control.

ESPECIFICACIONES TÉCNICAS

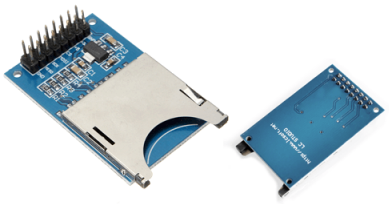
- Chip: L298N
- Canales: 2 (soporta 2 motores DC o 1 motor PAP)
- Voltaje lógico: 5V
- Voltaje de Operación: 5V-35V
- Consumo de corriente (Digital): 0 a 36mA
- Capacidad de corriente: 2A (picos de hasta 3A)
- Potencia máxima: 25W
- Peso: 30g
- Dimensiones: 43 * 43 * 27 mm

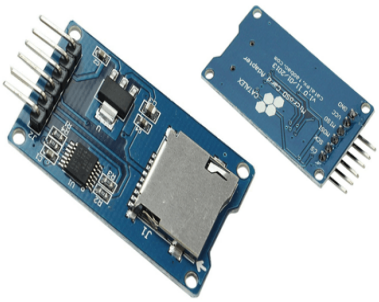


Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805. Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo ya sea un microcontrolador o un Arduino, pero recomendamos que el consumo no sea mayor a 500mA.

Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35V DC. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.

2.7 TARJETA SD O MICRO SD CON ARDUINO





Un lector SD es un dispositivo que permite emplear como almacenamiento una tarjeta SD, que podemos incorporar en nuestros proyectos de electrónica y Arduino.

Las tarjetas SD y micro SD se han convertido en un estándar, desplazando a otros medios de almacenamiento de datos debido a su gran capacidad y pequeño tamaño. Por este motivo han sido integradas en una gran cantidad de dispositivos, siendo en la actualidad componentes frecuentes en ordenadores, tablets y Smartphone, entre otros.

Dentro del mundo de Arduino, es posible encontrar lectores de bajo coste tanto para tarjetas SD como micro SD. Los primeros en aparecer fueron los lectores SD y posteriormente los micro SD. Por tanto, en general, los módulos con micro SD son modelos más modernos que los de SD

En ambos tipos de lectores, la lectura puede realizarse a través de bus SPI. Aunque pueden disponer de otros interfaces, como bus I2C o UART, normalmente es preferible emplear SPI por su alta tasa de transferencia.

Respecto a las tarjetas empleadas, podemos emplear tarjetas SD o SDSC (Standard Capacity) o SDHC (High Capacity), pero no SDXC (Extended Capacity). Deberá estar formateada en sistema de archivos FAT16 o FAT32.

La tensión de alimentación es de 3.3V, pero en la mayoría de los módulos se incorpora la electrónica necesaria para conectarlo de forma sencilla a Arduino, lo que frecuentemente incluye un regulador de voltaje que permite alimentar directamente a 5V.

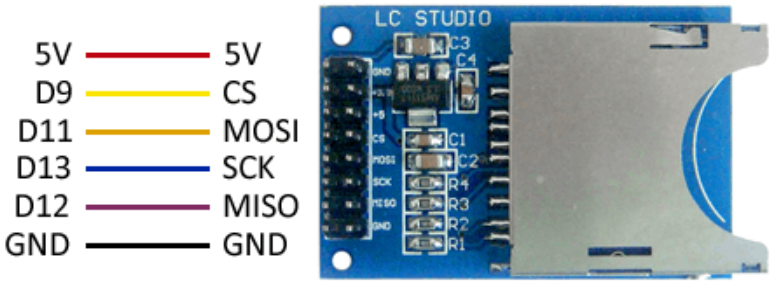
Emplear una tarjeta SD o micro SD en con Arduino tiene la ventaja de proporcionar una memoria casi ilimitada para nuestros proyectos. Además es no volátil (es decir, resiste cuando se elimina la alimentación), y puede ser extraída y conectada a un ordenador con facilidad.

La gran desventaja es que supone una importante carga de trabajo para Arduino. Sólo el programa ocupará el 40% de la memoria Flash, y casi el 50% de la memoria dinámica. El uso del procesador también es exigente.

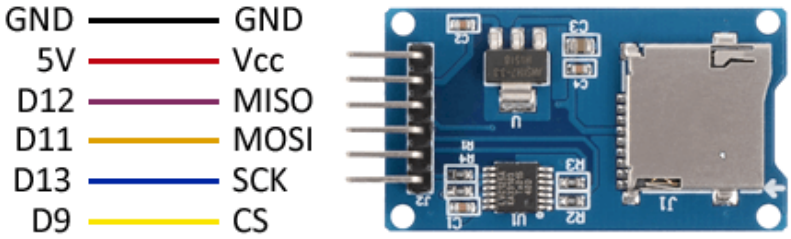
ESQUEMA DE MONTAJE

La conexión es sencilla y similar tanto para lectores SD como Micro SD. Simplemente alimentamos el módulo desde Arduino mediante 5V y Gnd. Por otro lado, conectamos los pines del bus SPI a los correspondientes de Arduino.

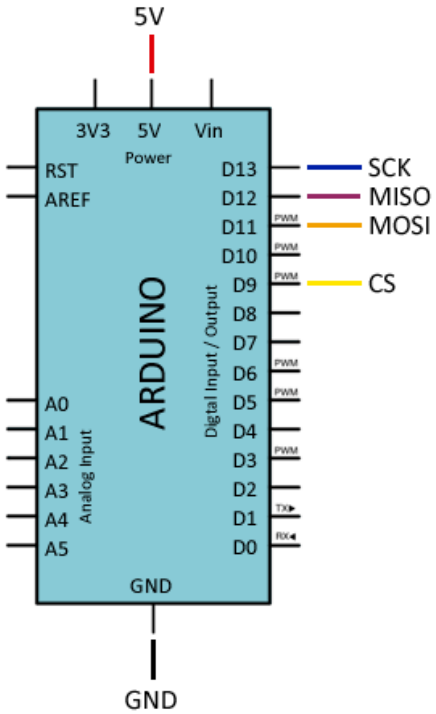
La conexión del lector SD sería la siguiente,



Similar a la de un lector micro SD, que sería la siguiente.



En ambos casos la conexión, vista desde el lado de Arduino, es la misma, y quedaría así.



2.8 SRF05 SENSOR DISTANCIAS ULTRASONIDOS SIMPLE

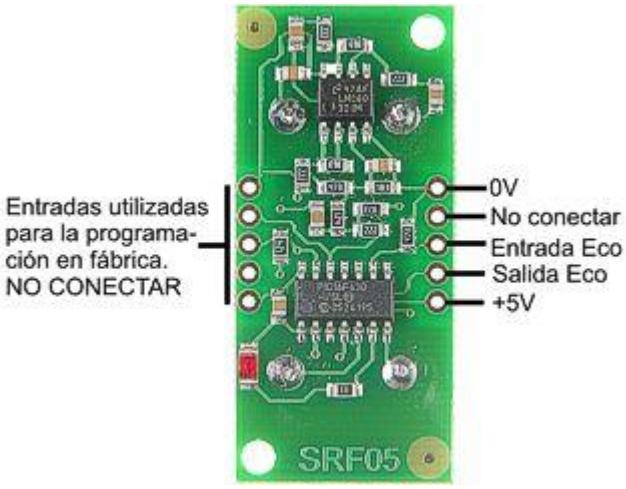
SRF05 es un nuevo sensor de distancias pensado para ser una actualización del clásico SRF04 con el que es compatible, pero además añadiendo nuevas funciones y características. En el modo estándar, el SRF05 se comporta igual que el SRF04 con la diferencia de que el rango de trabajo se ha aumentado de 3 a 4 metros. Esto significa que todo el software que funciona con el SRF04, funciona con el SRF05. Por otro lado, el SRF05 cuenta con un nuevo modo de trabajo que emplea un solo pin para controlar el sensor y hacer la lectura de la medida. Lo que se hace es mandar un impulso para iniciar la lectura y luego poner el pin en modo entrada. Después basta con leer la longitud del pulso devuelto por el sensor, que es proporcional a la distancia medida por el sensor. El SRF05 es mecánicamente igual al SRF04, por lo que puede ser un sustituto de este.

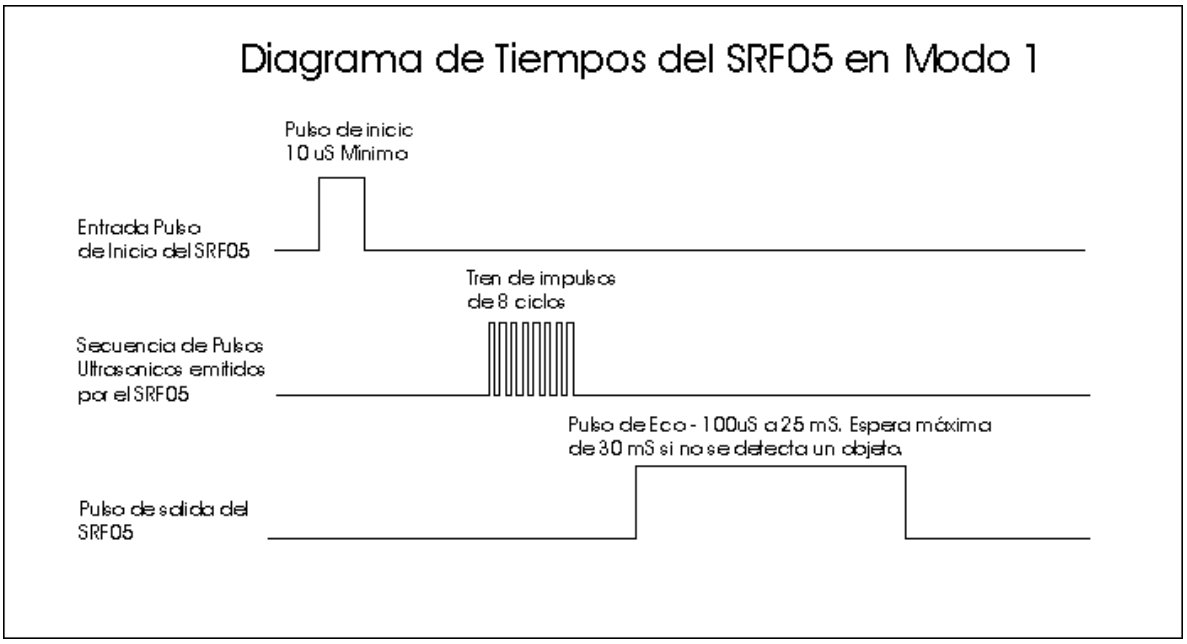
El sensor SRF05 incluye un breve retardo después del pulso de eco para dar a los controladores más lentos como Basic Stamp y Picaxe el tiempo necesario para ejecutar sus pulsos en los comandos. El sensor SRF05 tiene dos modos de funcionamiento, según se realicen las conexiones.

2.8.1 MODOS DE USO

Modo 1 - Compatible con SRF04 - Señal de activación y eco independientes

Este modo utiliza pines independientes para la señal de inicio de la medición y para retorno del eco, siendo el modo más sencillo de utilizar. Todos los ejemplos de códigos para el sensor SRF04 funcionarán para SRF05 en este modo. Para utilizar este modo, simplemente deberá dejar sin conectar el pin de modo - el SRF05 integra una resistencia pull-up en este pin.

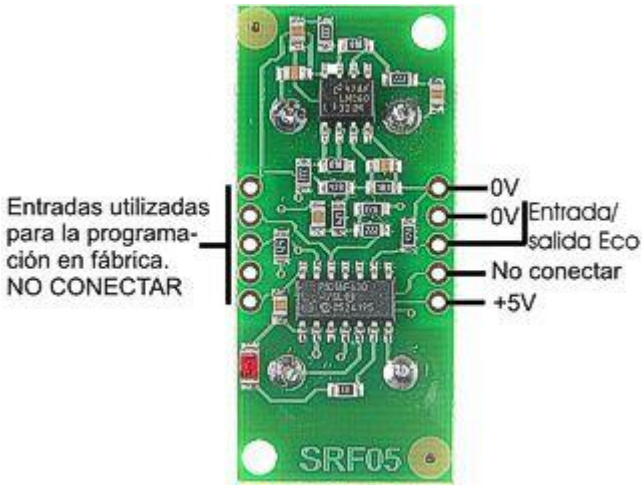


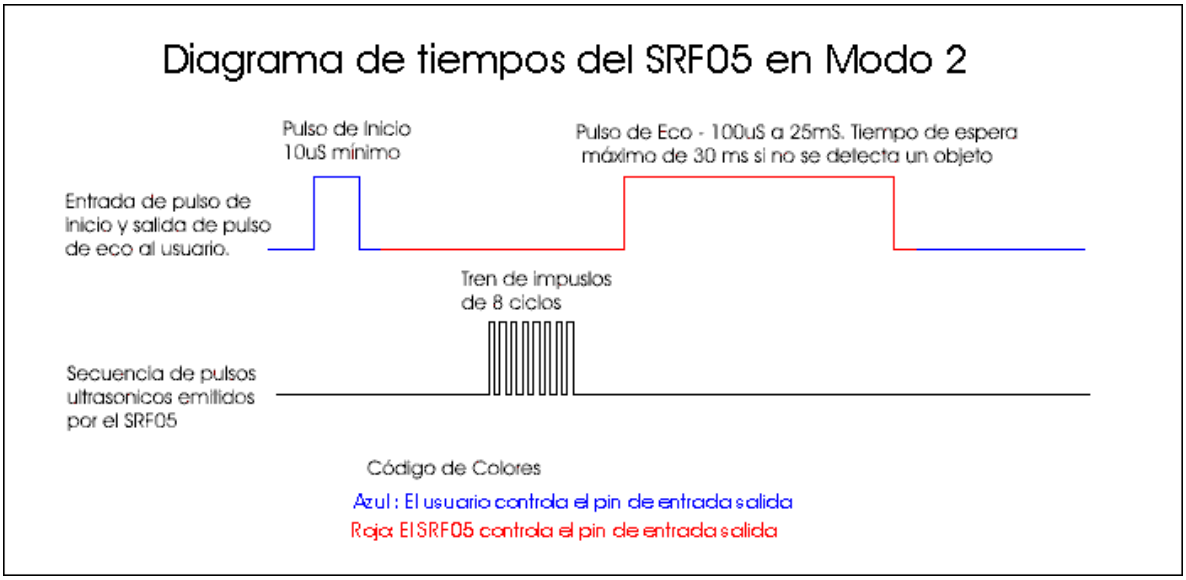


Modo 2 - Pin único para la señal de activación y eco

Este modo utiliza un único pin para las señales de activación y eco, y está diseñado para reducir el número de pines en los microcontroladores. Para utilizar este modo, conecte el pin de modo al pin de tierra de 0v. La señal de eco aparecerá en el mismo pin que la señal de activación. El SRF05 no elevará el nivel lógico de la línea del eco hasta 700uS después del final de la señal de activación. Dispone de ese tiempo para cambiar el pin del disparador y convertirlo en una entrada para preparar el código

de medición de pulsos. El comando PULSIN integrado en la mayor parte de los controladores del mercado lo hace automáticamente.





Cómo calcular la distancia

A continuación, se muestran todos los diagramas de tiempo para el sensor de distancias por ultrasonido SRF05 para cada modo. Deberá suministrar un breve pulso de al menos 10uS para disparar la entrada de comienzo del cálculo de distancia. El SRF05 transmitirá una ráfaga de 8 ciclos de ultrasonidos a 40khz elevando el nivel lógico de la señal del eco (o la línea de activación en el modo 2). Entonces el sensor "escucha" un eco, y en cuanto lo detecta, vuelve a bajar el nivel lógico de la línea de

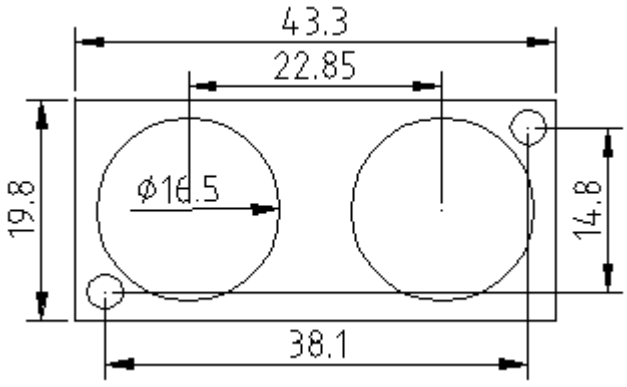
eco. La línea de eco es por lo tanto un pulso, cuyo ancho es proporcional a la distancia respecto al objeto. Registrando la duración del pulso es posible calcular la distancia en pulgadas/centímetros o en cualquier otra unidad de medida. Si no se detectase nada, entonces el SRF05 baja el nivel lógico de su línea de eco después de 30mS.

El SRF05 proporciona un pulso de eco proporcional a la distancia. Si el ancho del pulso se mide en uS, el resultado se debe dividir entre 58 para saber el equivalente en centímetros, y entre 148 para saber el equivalente en pulgadas. uS/58=cm o uS/148=pulgadas.

El SRF05 puede activarse cada 50mS, o 20 veces por segundo. Debería esperar 50ms antes de la siguiente activación, incluso si el SRF05 detecta un objeto cerca y el pulso del eco es más corto. De esta manera se asegura que el "bip" ultrasónico ha desaparecido completamente y no provocará un falso eco en la siguiente medición de distancia.

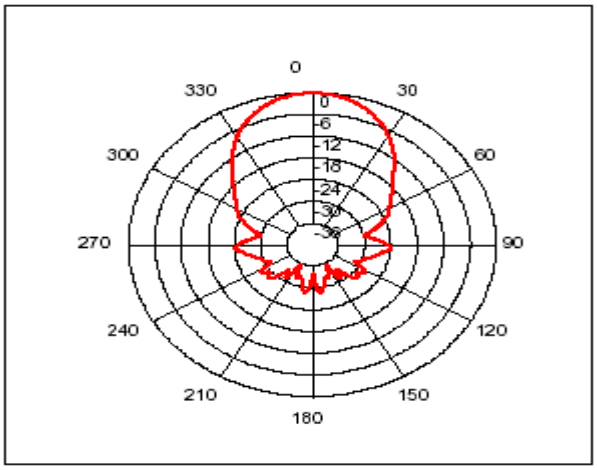
El otro conector de 5 pines

Los 5 pines marcados como "programming pins" (pines de programación) se utilizan sólo una vez durante el proceso de fabricación para programar la memoria Flash en el chip del PIC16F630. Los pines de programación de PIC16F630 se utilizan también para realizar otras funciones en el SRF05, por lo que deberá asegurarse de que nada esté conectado a ellos o se interrumpirá el funcionamiento de los módulos.



Cómo cambiar el patrón y el ancho del haz

No puede hacerlo. Los usuarios de este sensor nos plantean esta consulta muy frecuentemente; sin embargo no existe ninguna manera sencilla de reducir o cambiar el ancho del haz. El patrón del haz del sensor SRF05 es cónico mientras que el ancho del haz es una función del área de la superficie de los transductores y es fijo. El patrón del haz de los transductores utilizados en el SRF05, según la hoja de datos de los fabricantes, es la siguiente:



	HC-SR04	HY-SRF05
Working Voltage	5 VDC	5 VDC
Static current	< 2mA	<2 mA
Output signal:	Electric frequency signal, high level 5V, low level 0V	Electric frequency signal, high level 5V, low level 0V
Sensor angle	< 15 degrees	< 15 degrees
Detection distance (claimed)	2cm-450cm	2cm-450cm
precision	~3 mm	~2 mm
Input trigger signal	10us TTL impulse	10us TTL impulse
Echo signal	output TTL PWL signal	output TTL PWL signal
Pins	1. VCC 2. trig(T) 3. echo(R) 4. GND	1. VCC 2. trig(T) 3. echo(R) 4. OUT 5. GND

2.9 ACELERÓMETRO Y GIROSCOPIO MPU 6050

EL MPU6050 es una unidad de medición inercial o IMU (Inertial Measurment Units) de 6 grados de libertad (DoF) pues combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Este sensor es muy utilizado en navegación, goniometría, estabilización, etc.

Aceleración y acelerómetros

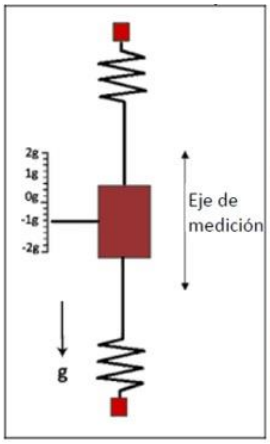
La aceleración es la variación de la velocidad por unidad de tiempo es decir razón de cambio en la velocidad respecto al tiempo:

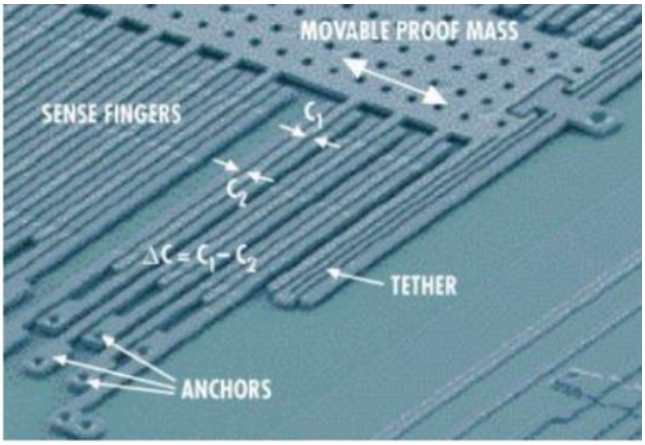
$a=dV/dt$

Así mismo la segunda ley de Newton indica que en un cuerpo con masa constante, la aceleración del cuerpo es proporcional a la fuerza que actúa sobre él mismo:

$a=F/m$

Este segundo concepto es utilizado por los acelerómetros para medir la aceleración. Los acelerómetros internamente tienen un MEMS (Micro Electro Mechanical Systems) que de forma similar a un sistema masa resorte permite medir la aceleración.



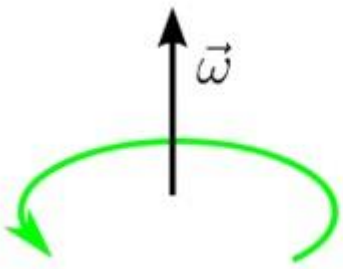


Con un acelerómetro podemos medir esta aceleración, teniendo en cuenta que a pesar que no exista movimiento, siempre el acelerómetro estará censando la aceleración de la gravedad.

Con el acelerómetro podemos hacer mediciones indirectas como por ejemplo si integramos la aceleración en el tiempo tenemos la velocidad y si la integramos nuevamente tenemos el desplazamiento, necesitando en ambos casos la velocidad y la posición inicial respectivamente.

Velocidad Angular y giroscopio

La velocidad angular es la tasa de cambio del desplazamiento angular por unidad de tiempo, es decir que tan rápido gira un cuerpo alrededor de su eje:



$$\omega = \frac{d\theta}{dt}$$

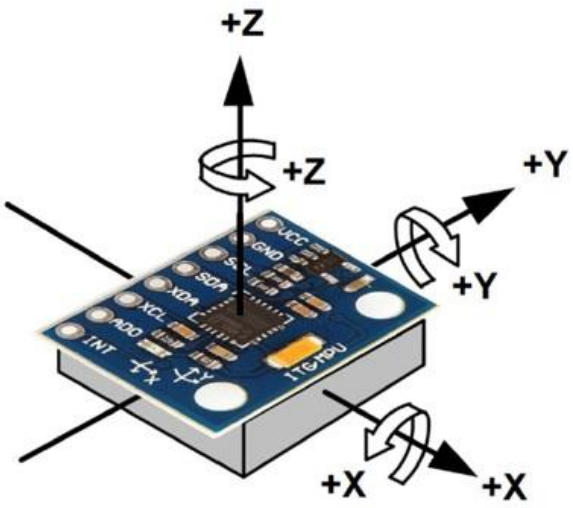
Los giroscopios utilizan un MEMS (MicroElectroMechanical Systems) para medir la velocidad angular usando el efecto Coriolis

Con un giroscopio podemos medir la velocidad angular, y si se integra la velocidad angular con respecto al tiempo se obtiene el desplazamiento angular (posición angular si se sabe dónde se inició el giro)

Módulo Acelerómetro y giroscopio MPU6050

EL módulo Acelerómetro MPU tiene un giroscopio de tres ejes con el que podemos medir velocidad angular y un acelerómetro también de 3 ejes con el que medimos los componentes X, Y y Z de la aceleración.

La dirección de los ejes está indicado en el módulo el cual hay que tener en cuenta para no equivocarnos en el signo de las aceleraciones.



La comunicación del módulo es por I2C, esto le permite trabajar con la mayoría de microcontroladores. Los pines SCL y SDA tienen una resistencia pull-up en placa para una conexión directa al microcontrolador o Arduino.

Tenemos dos direcciones I2C para poder trabajar:

Pin AD0	Dirección I2C
AD0=HIGH (5V)	0x69
AD0=LOW (GND o NC)	0x68

El pin ADDR internamente en el módulo tiene una resistencia a GND, por lo que si no se conecta, la dirección por defecto será 0x68.

El módulo tiene un regulador de voltaje en placa de 3.3V, el cual se puede alimentar con los 5V del Arduino.

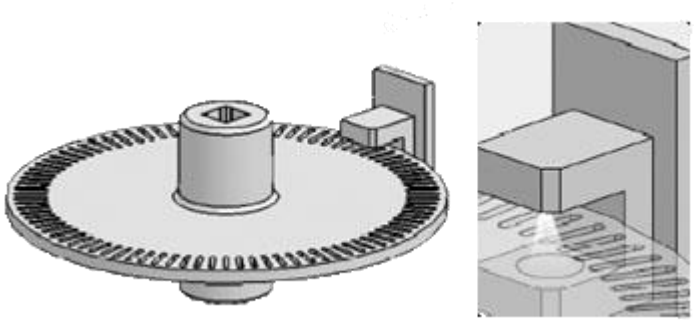
2.10 CODIFICADOR DE EJE ÓPTICO

Con los dos canales de salida del Codificador de cuadratura puede medir tanto la posición como la dirección de rotación de un eje VEX. Esto le permitirá calcular la velocidad del eje, así como la distancia recorrida con el software de programación. Los codificadores se usan típicamente para aplicaciones de "rotación infinita", tales como una rueda motriz. Se necesita un kit de programación para cambiar el programa en el controlador VEX para usar el codificador de cuadratura.

- Medir el viaje angular
- Determinar la dirección de rotación
- Calcular la velocidad del eje
- Calcular la distancia recorrida

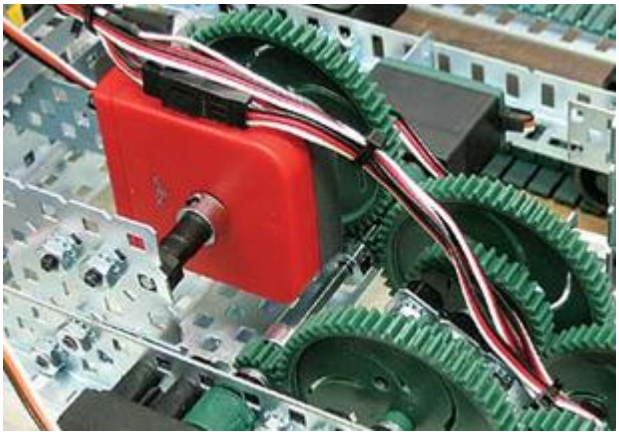
- Aumentar el control de navegación
- Más Funcionalidad Autónoma

El codificador de eje óptico se utiliza para medir tanto la posición relativa como la distancia de rotación recorrida por un eje. Trabaja brillando la luz en el borde de un disco equipado con las ranuras uniformemente espaciadas alrededor de la circunferencia. A medida que el disco gira, la luz pasa a través de las ranuras y es bloqueada por los espacios opacos entre las ranuras. El codificador detecta entonces cuántas hendiduras han tenido luz brillando a través, y en qué dirección el disco está girando.



El codificador de eje óptico se puede utilizar para mejorar un robot de varias maneras. El codificador puede medir la distancia de rotación recorrida y la velocidad, que puede usarse para controlar, por ejemplo, la posición angular de un brazo de agarre de robot o la velocidad de un robot. Conocer estos parámetros puede ayudarle en gran medida a realizar tareas autónomas con su robot.

El codificador de eje óptico puede usarse para rastrear la distancia recorrida, la dirección del movimiento o la posición de cualquier componente rotatorio, tal como un brazo de pinza. El codificador también puede usarse para detectar movimiento, lo que podría facilitar una interacción más rica entre el robot y su entorno (por ejemplo, interacción humano-robot). Si un ser humano mueve un brazo de robot que está conectado a un codificador (por ejemplo, durante un apretón de manos), el robot detecta el movimiento del brazo y la dirección (s) y la distancia recorrida, ayudando al robot a clasificar la interacción como un apretón de manos.



Mientras que el diámetro del disco en el codificador realmente no importa, el diámetro de la rueda o del engranaje cuyo eje pasa a través del codificador hace! La circunferencia de la rueda es igual a su diámetro multiplicado por pi (aproximadamente 3,14). Multiplicar la distancia recorrida que cuando se multiplica por el número de revoluciones da de la distancia recorrida.

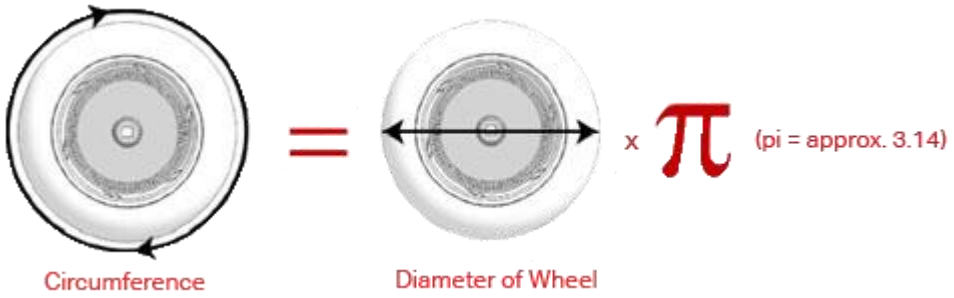


Figura 3. Fórmula de la circunferencia de la rueda

El codificador de eje óptico puede detectar hasta 1.700 pulsos por segundo, lo que corresponde a 18,9 revoluciones por segundo ya 1,133 rpm (revoluciones por minuto). Por tanto, las revoluciones más rápidas no se interpretarán exactamente, lo que podría dar lugar a que se pasen datos de posición erróneos al microcontrolador.

CAPÍTULO 3: DESARROLLO DEL PROYECTO
3.1 MATERIALES UTILIZADOS

Materiales:

- Canal c de vez
- Angulo vez sin ranuras
- Rueda de tracción de 5 pulgadas
- Rueda omnidireccional de 5 pulgadas
- motores VEX EDR 393
- Codificador de eje óptico
- Puente h l298n
- Arduino uno
- Lector de SD
- Batería tipo lipo
- Placa fenólica
- Interruptor on/off
- Mpu6050
- Cables para conexión
- Tornillos
- Tuercas
- Separadores

3.2 DIAGRAMA DE CONEXIÓN (ELÉCTRICO)

Un diagrama electrónico, también conocido como un esquema eléctrico o esquemático es una representación pictórica de un circuito eléctrico. Muestra los diferentes componentes del circuito de manera simple y con pictogramas uniformes de acuerdo a normas, y las conexiones de alimentación y de señal entre los distintos dispositivos. El arreglo de los componentes e interconexiones en el esquema generalmente no corresponde a sus ubicaciones físicas en el dispositivo terminado.

A diferencia de un esquema de diagrama de bloques o disposición, un esquema de circuito muestra la conexión real mediante cables entre los dispositivos. (Aunque el esquema no tiene que corresponder necesariamente a lo que el circuito real aparenta) -- El tipo de dibujo que sí representa al circuito real se llama negativo (o positivo) de la tablilla de circuito impreso.

Para un mejor manejo de conexiones fue necesario realizar un circuito impreso para evitar los falsos contactos y reducir al máximo el cableado que se pudiera requerir. El diseño se basó en la Shelf de arduino uno donde se ubicaron las conexiones para los encoders, puentes h, lector de SD, mpu6050.

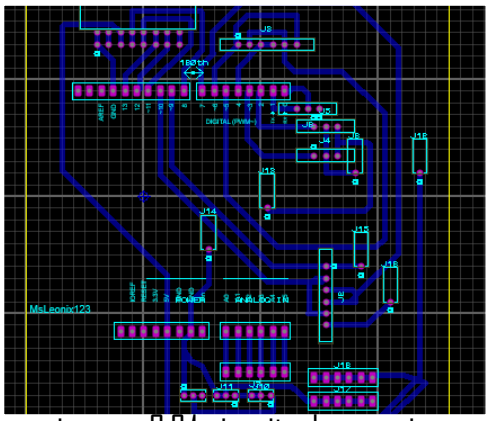


imagen 3.2.1 circuito de conexion realizado en proteus.

3.3 CONSTRUCCIÓN

Como prueba inicial se tenía un prototipo como se puede apreciar en la imagen 3.3.1, este prototipo tenía como fuente de alimentación una fuente de computador es decir no podría mantenerse encendido el sistema sin que la fuente de alimentación se mantuviera conectada directamente a la línea de energía proporcionada por CFE, por esta razón se sustituyó la fuente de alimentación por una pila tipo lipo recargable para poder mantener el sistema encendido sin la necesidad de estar conectada la fuente a la línea de energía.

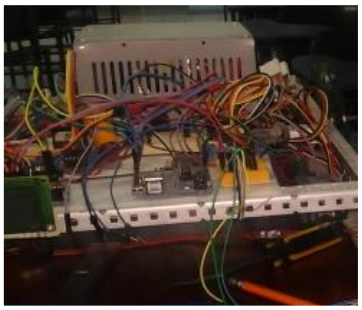


imagen 3.3.1 prototipo proporcionado por la institucion

Otra de las desventajas es que las dimensiones del prototipo no eran dimensionadas con las del proyecto donde se llevaría a cabo la implementación en un ambiente de trabajo.

Entonces se la construcción del prototipo para hacer las pruebas pertinentes y haci caracterizar los sensores de nuestro robot recolector de pet tuvo que ser modificado y haci dimensionarlo 3:1 utilizando canales y ángulos ambos de la marca vex.

- Canal C de VEX

imagen 3.3.2 canal c de vex

Tiene agujeros en incrementos de 0.500 pulgadas. Esta excelente resistencia y a la flexión de este miembro estructural es perfecta para construir robots robustos. Múltiples tamaños disponibles, en dos tipos de materiales diferentes. Hecho del acero laminado o aluminio 5052-H32. Cada canal se segmenta en trozos cortantes de 2.5 pulgadas

- Ángulo VEX sin ranuras

imagen 3.3.3 angulo vex sin ranuras

Típicamente utilizado para el enmarcado estructural. Se puede cortar en incrementos de 0.500 pulgadas. Hecho del acero laminado de 0.046 pulgadas grueso, cinc plateado.

También como prototipo y prueba inicial se utilizó el siguiente tipo de llantas con los motores VEX 393

imagen 3.3.4 rueda de tracción

- Rueda de tracción de 5 pulgadas

Están diseñadas para un agarre óptimo en superficies blandas como las baldosas de espuma utilizadas en la competición de Robótica VEX.

- motores VEX EDR 393

imagen 3.3.5 motor vex 393

Son los motores primarios utilizados para los mecanismos del robot. También ofrecemos engranajes de reemplazo para dar a sus motores un impulso de velocidad.

Como primer sensor fue instalado al momento de ensamblar las llantas y los motores el codificador de eje óptico este nos permitirá obtener y comparar el rendimiento de los motores así también como la distancia que se desplaza cada uno, como poder tener datos numéricos sobre el comportamiento de ambos lados del robot móvil.

- Codificador de eje óptico

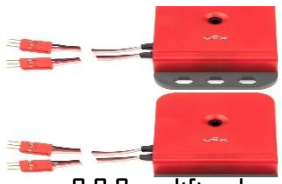


imagen 3.3.6 codificador de eje óptico

Con los dos canales de salida del Codificador de cuadratura puede medir tanto la posición como la dirección de rotación de un eje VEX. Esto le permitirá calcular la velocidad del eje, así como la distancia recorrida con el software de programación.

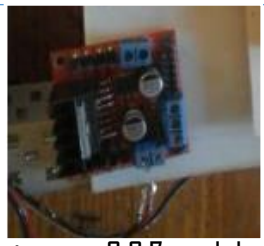


imagen 3.3.7 modulo puente h l298n

Junto con los materiales anteriores fue instalado los módulos puente h para la manipulación de los motores.

Obteniendo como resultado el siguiente prototipo para la caracterización de sensores para robot recolector de pet, en esta base se implementara los sensores y así poder caracterizarlos.

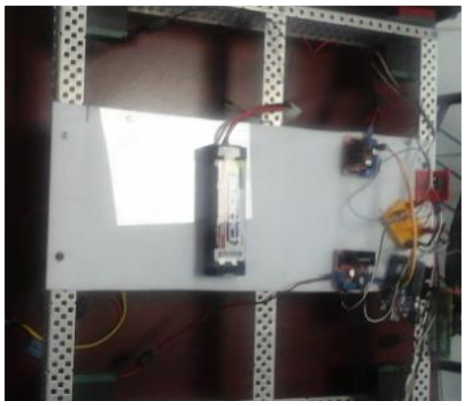


imagen 3.3.8 prototipo robot recolector de pet

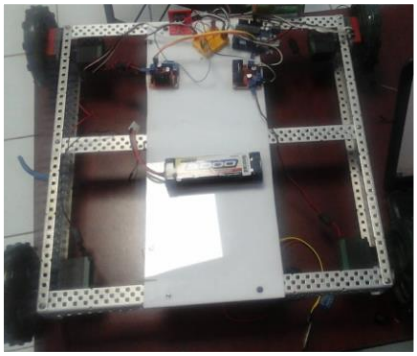


imagen 3.3.8.1 prototipo robot recolector de pet

Para un mejor control y evitar los falsos contactos en las conexiones mediante cables se diseñó una Shelf en proteus tomando en cuenta los sensores que se

necesitara utilizar, las alimentaciones y el elemento de almacenamiento de información que se obtendrá en cada prueba.

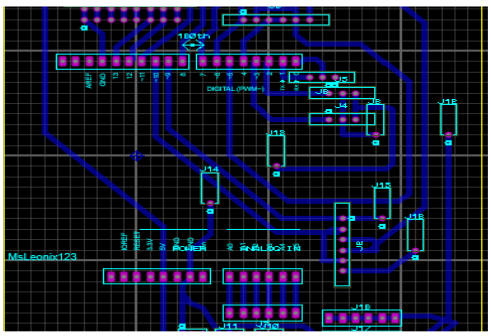


imagen 3.3.9 PCB realizado en proteus

Mediante la técnica del planchado para realizar placas fenólicas se elaboró la Shelf con las conexiones a los sensores y actuadores.

imagen 3.3.10 circuito sobre placa fenolica para realizacion de circuito impreso

El circuito impreso en hoja cauche, se colocó cobre una placa fenólica nueva para luego calentarla usando una plancha hasta que el circuito se transfiera de la hoja a la placa, el tiempo puede variar dependiendo de la temperatura de la plancha y del grosor de la hoja.

Imagen 3.3.11 circuito resultante usando la técnica de planchado

Al terminar el proceso de la elaboración del circuito que utilizaremos, se perforo y soldaron los pines que nos darán la salida para conectar los sensores y actuadores para luego poder montarlo en el prototipo.

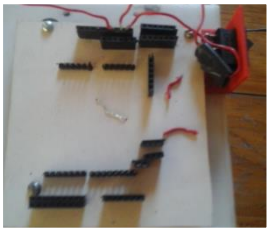


imagen 3.3.12 shelf para arduino

Para tener el control de poder encender y apagar el sistema completo se colocó un interruptor on/off sobre el chasis del prototipo realizado.

Teniendo la Shelf instalada, conectamos el arduino uno en ella. El arduino será nuestro controlador es decir donde programaremos y desde ahí se ejecutaran todas las instrucciones capaz de tomar decisiones de acuerdo a lo que los sensores le manden de información y activar o desactivar los actuadores.

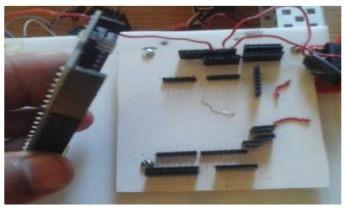
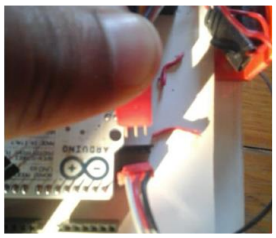


imagen 3.3.13 instalacion de placa arduino uno

Como se diseñó la Shelf se tuvo que hacer la reconexión de los módulos de puente h, usando pwm para el control de la velocidad de los motores y los pines digitales para la activación/desactivación y orientación del giro del motor.

imagen 3.3.14 conexionado de modulos puente h l298n

Por último la conexión de los encoders y el sensor mpu6050 que nos ayudara a ver los grados de giro que obtendremos y nuestro lector de SD para el almacenamiento de los datos obtenidos.



imganen 3.3.15 conexion de encoder

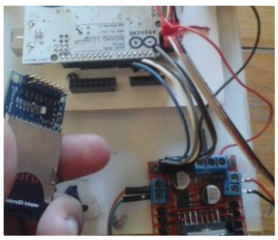


imagen 3.3.16 conexion de lector SD

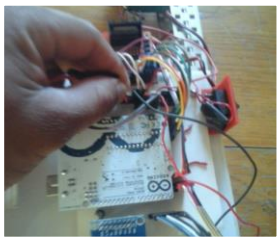


imagen 3.3.17 conexion de mpu6050

Obteniendo el siguiente resultado con todos los materiales ya conectados a nuestra placa y montado en el prototipo de prueba.

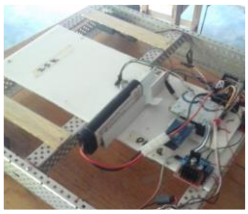


imagen 3.3.18 prototipo terminado

CAPITULO 4: RESULTADOS

4.1 PRUEBAS DE VELOCIDAD

Como primera parte se realizaron pruebas a los motores para ver la velocidad mínima y máxima que alcanza cada uno de estos, debido a que solo se contaba con dos encoders (codificador de eje óptico), las pruebas solo se realizaron en las llantas delanteras etiquetándolos izquierda y derecha en su momento para no tener confusión al momento de obtener los valores y poder graficarlos y compararlos uno con el otro.

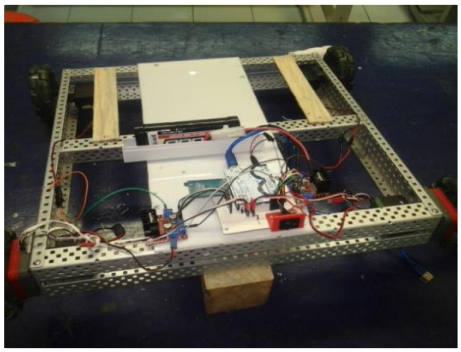


imagen 4.1.1 pruebas de velocidad

Para llevar a cabo estas pruebas las llantas se mantuvieron alzadas para evitar la fricción con el suelo y poder determinar la velocidad que alcanzaría los dos motores sin tener que soportar una carga extra a la de la llanta.

Mediante el puerto serial del arduino se puede visualizar y copiar los datos que los encoders nos arrojaban para luego poder graficarlo estos datos nos arrojaba cuantas interrupciones se generaban en un segundo es decir cuántas veces pasa el haz de luz en la ranura al borde opaco del encoders.

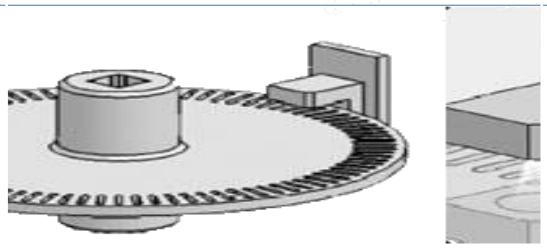


imagen 4.4.2 funcion basica del encoder

Cabe mencionar que el encoders utilizado marca vex cuenta con 90 ranuras por una vuelta girada, es decir que tendríamos 90 interrupciones por cada vuelta que de nuestra llanta.

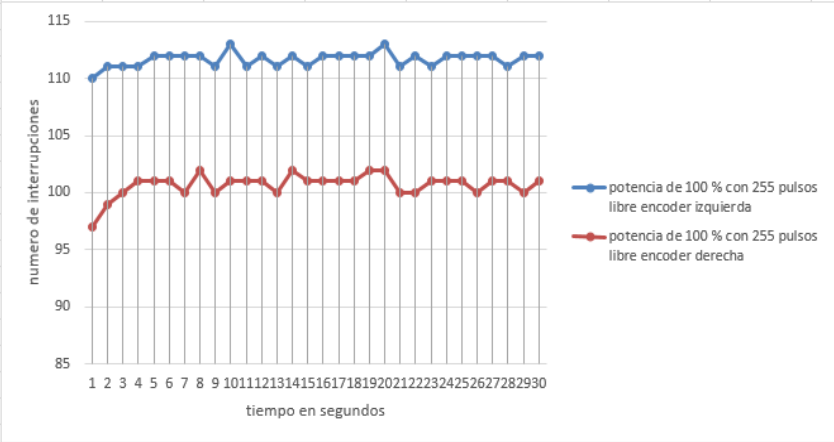
De acuerdo al siguiente programa realizado en arduino se generara los datos.

```
volatile int contador = 0;    //declaramos una funcion volatile para encoder derecha
volatile int contador1 = 0;   //declaramos una funcion volatile para encoder izquierda
 int N1=8;                     //declaramos una variable entera para los pines que
 int N2=9;                    // nos servira para el control de giro de la
 int N3=12;                   //velocidad y sentido del griro
 int N4=13;
 int ENA=10;
 int ENB=11;
int contador2=25;        //declaramos un contador que tendra los pulsos pwm enviados
void setup() {
  Serial.begin(57600);
  attachInterrupt(0,interrupcion0,RISING); //usamos las interrupciones del arduino
   attachInterrupt(1,interrupcion1,RISING); //para contar las ranuras del encoder
   //declaramos como salidas nuestros pines que usaremos
pinMode (N1,OUTPUT);
pinMode (N2,OUTPUT);
pinMode (N3,OUTPUT);
pinMode (N4,OUTPUT);
pinMode (ENA,OUTPUT);
pinMode (ENB,OUTPUT);

}


void loop() {

  delay(999);        //se ejecutara cada segundo
  digitalWrite(N1,HIGH);
  digitalWrite(N2,LOW);
  digitalWrite(N3,HIGH);
  digitalWrite(N4,LOW);
  analogWrite(ENA,contador2);
  analogWrite(ENB,contador2);
  Serial.println(contador);
  Serial.println(contador1);
  Serial.println(contador2);
  contador = 0;      //se colocan en 0 los contadores
   contador1 = 0;


}

void interrupcion0()     //se ejecutan las interrupciones
{
  contador++;

}

void interrupcion1()
{
  contador1++;

}
```

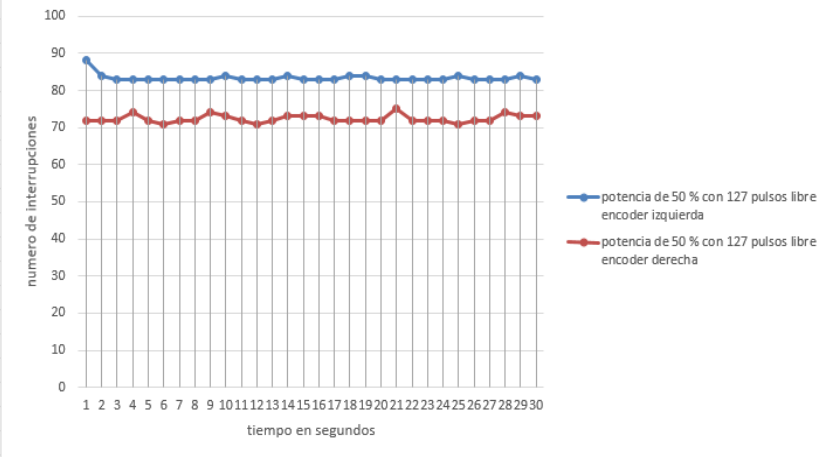
Las interrupciones se ejecutaran cuando pase el estado del encoders de bajo a alto, el tiempo 999 se toma como un segundo ya que con el tiempo de ejecución del programa y los retardos que se crean es aproximadamente 1 segundo el tiempo que se ejecutara cada lectura.

Imagen 4.4.3 código lector de interrupciones en arduino

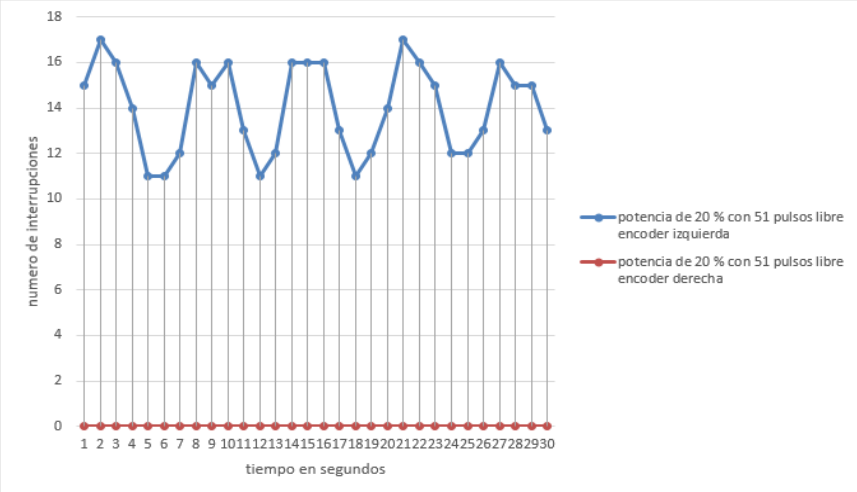
A continuación se muestra el rango del funcionamiento de los motores utilizando control por pulsos pwm.



Usando una potencia de 100% vemos que el motor de la derecha está por debajo del de la izquierda en el número de interrupciones que nos manda los encoders



Cuando llegamos a un 50% de la potencia el resultado en diferencia de velocidad sigue igual es decir la derecha por debajo del motor izquierdo.



El límite de la potencia que se obtuvo fue 20% para el motor de la izquierda mientras que el motor del lado derecho a un 30% ya no mantenía su giro.

Al ver el funcionamiento anterior nos damos cuenta que existe variación en la capacidad de los motores aunque estos son del mismo proveedor y misma marca, ahora mostraremos el resultado que se obtuvo al realizar diferentes corridas para ver el comportamiento del robot móvil así como también ver el funcionamiento de los encoders.

Como el prototipo estaría en movimiento los datos no pueden obtenerse directamente del monitor serial entonces se modificó el programa para empezar a utilizar el lector de tarjeta SD quedando de la siguiente forma.

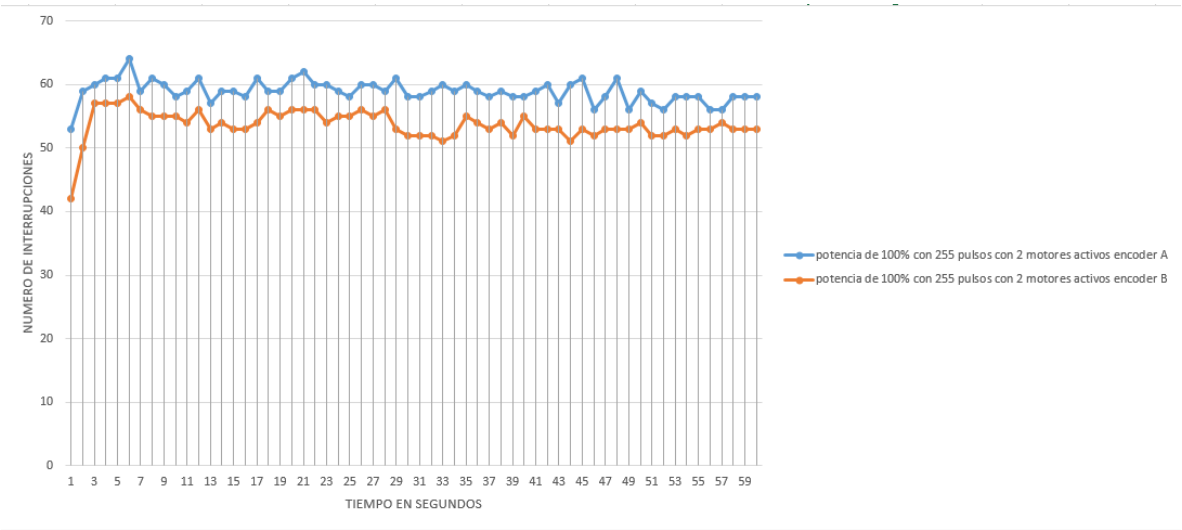
encoder_maestro §

```
#include <SPI.h> //incluimos la librerias
#include <SD.h>
File myFile;
#include <SoftwareSerial.h>
volatile int contador = 0;   //declaramos las variables de las interrupciones
volatile int contador1 = 0;
 int N1=0;
 int N2=4;
 int N3=1;
 int N4=7;
 int ENA=10;
 int ENB=5;
 int ENA1=6;
 int ENB1=9;
int contador2=255; //potencia en pwm
volatile int contiempo=0; //contador de tiempo auxiliar

void setup() {
  Serial.begin(57600);
Serial.print("iniciando sd..."); //inicializamos la tarjeta sd

if (!SD.begin(8)) {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");
  attachInterrupt(0,interrupcion0,RISING); //se aumenta el contador si la interrupcion sucede
   attachInterrupt(1,interrupcion1,RISING);

   attachInterrupt(1,interrupcion1,RISING);
pinMode (N1,OUTPUT); //declaramos nuestros pines de salida
pinMode (N2,OUTPUT);
pinMode (N3,OUTPUT);
pinMode (N4,OUTPUT);
pinMode (ENA,OUTPUT);
pinMode (ENB,OUTPUT);
pinMode (ENA1,OUTPUT);
pinMode (ENB1,OUTPUT);

}
```

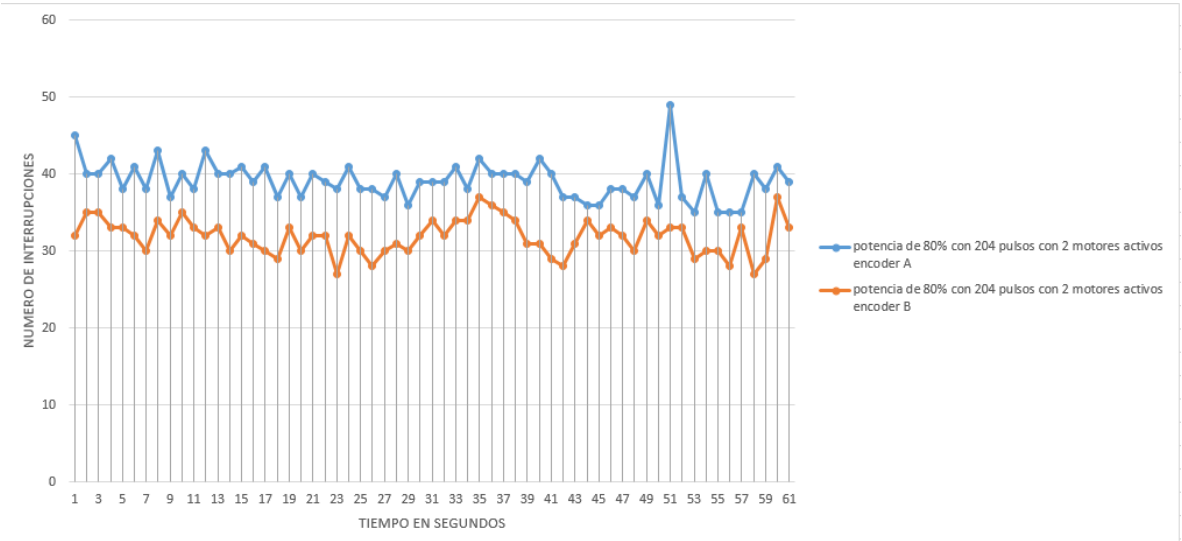
```
void loop() {

  delay(999);              //tiempo de lectura
  digitalWrite(N1,HIGH);
  digitalWrite(N2,HIGH);
  digitalWrite(N3,HIGH);
  digitalWrite(N4,HIGH);
  analogWrite(ENA,contador2);
  analogWrite(ENB,contador2);
  analogWrite(ENA1,contador2);
  analogWrite(ENB1,contador2);
 myFile = SD.open("datalog.txt", FILE_WRITE);//abrimos  el archivo

  if (myFile) {
        Serial.print("Escribiendo SD: "); //escribimos en la SD

        myFile.print("izquierda=");
        myFile.print(contador1);
        myFile.print(", derecha=");
        myFile.print(contador);
        myFile.print(", contador2=");
        myFile.println(contador2);

        myFile.close(); //cerramos el archivo
  } else {
    Serial.println("Error al abrir el archivo");
  }
  delay(10);
  contador = 0;
   contador1 = 0;
```

```
}

void interrupcion0()  //se ejecuta las interrupciones
{
  contador++;

}

void interrupcion1()
{
  contador1++;
}
```

La siguiente grafica es usando únicamente la activación de dos motores al 100% de su potencia.

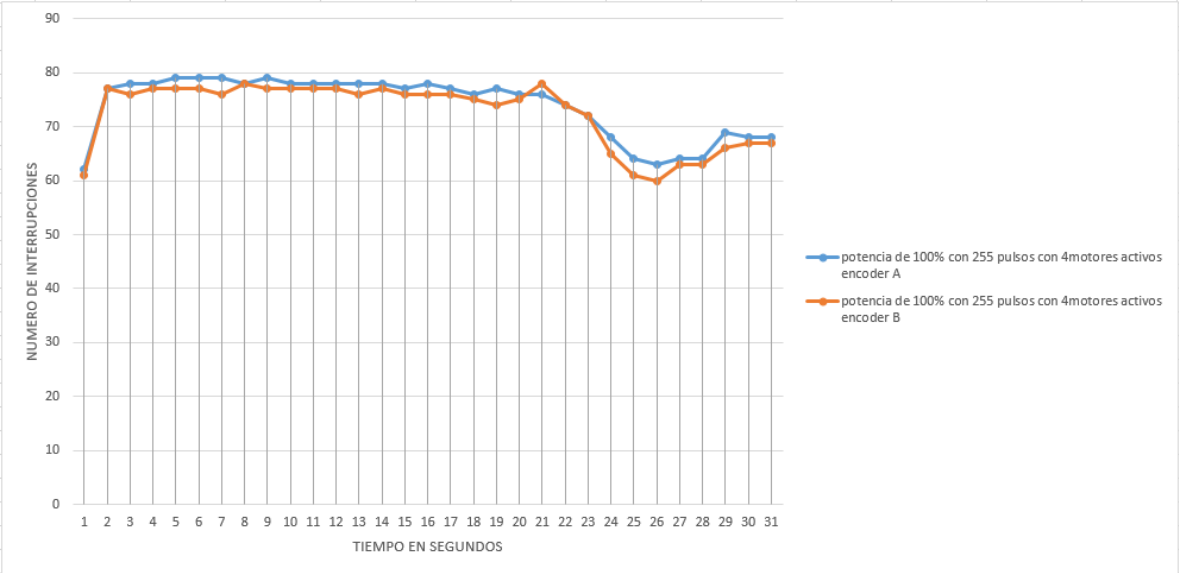


El encoder B pertenece al lado derecho, como se puede observar cuando se genera fricción con el suelo estos baja la velocidad de giro pero aún se mantiene la diferencia entre ambos lados el motor izquierdo por encima del derecho aunque se baje los pulsos pwm a 80% la relación se mantiene como se puede apreciar en la imagen siguiente.

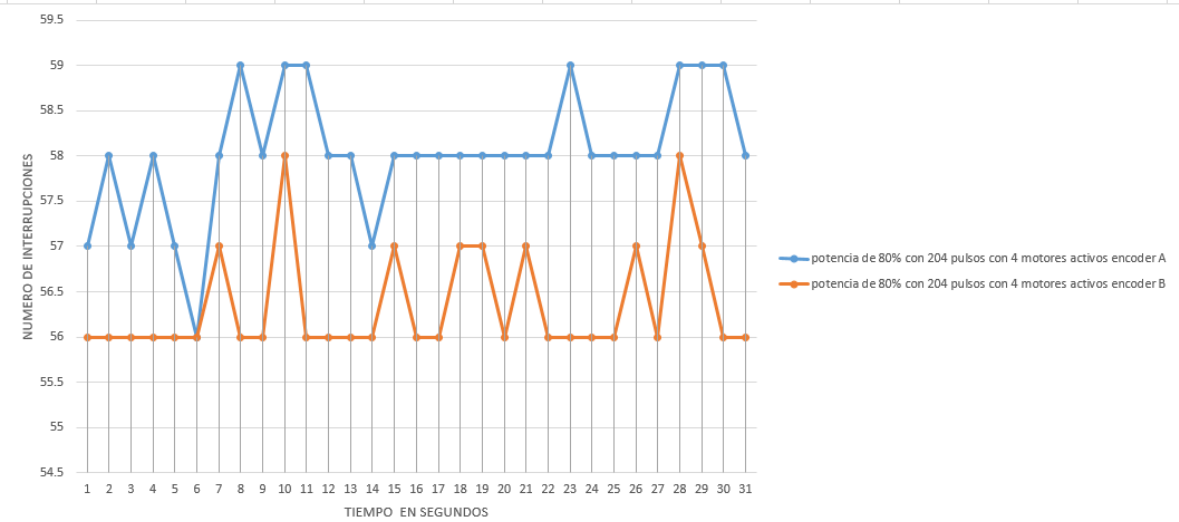


La potencia mínima usando dos motores fue de 60% ya que por debajo de esta el robot se detenía.

Ahora utilizando 4 motores activos vemos que existe una compensación en la velocidad del lado derecho pero aun con esa compensación se mantiene su velocidad por debajo del lado izquierdo esto variación nos causa que nuestro robot móvil tenga una desviación en su dirección y se va acumulando entre más distancia recorra mayor es la desviación.



Usando 80% de la potencia de los motores aún se mantiene la diferencia de velocidad entre un lado del otro.



La potencia mínima de los motores fue de 40% ya que por debajo de este el robot móvil se detenía.

4.2 VELOCIDAD CON LLANTAS DE TRACCIÓN

Como se pudo observar en las pruebas anteriores se generaba una diferencia entre las potencias de los motores es por eso que se corrigió el programa implementado creando una comparativa entre ellos y poder generar el más mínimo error posible.

Programa en arduino para minimizar el error.

```
#include <SPI.h> //se declaran las librerias
#include <SD.h>
File myFile;
#include <SoftwareSerial.h>
volatile int contador = 0;   //se crean las variables volatile
volatile int contador1 = 0;
 int N1=0;  //se crea las variables enteras para control
 int N2=4;
 int N3=1;
 int N4=7;
 int ENA=5;
 int ENB=6;
 int ENA1=10;
 int ENB1=9;
int contador2=255; //se genera 3 contadores uno nos servira
int contador3=255; //como referencia y los otros dos como
int contador4=255; //control de las potencias de los motores
volatile int contiempo=0; //se genera una variable llamada contiempo
//esta variable nos servira para hacer diferentes corridas sin cargar
//a cada instante el programa en el prototipo
void setup() {
  Serial.begin(57600); //se inicializan los puertos seriales
Serial.print("iniciando sd...");  //se inicializa la SD

if (!SD.begin(8)) {
    Serial.println("No se pudo inicializar");
    return;
  }
  Serial.println("inicializacion exitosa");
```

```
  attachInterrupt(0,interrupcion0,RISING); //se declaran las interrupciones
   attachInterrupt(1,interrupcion1,RISING);
pinMode (N1,OUTPUT); //se declaran nuestros pines como salida
pinMode (N2,OUTPUT);
pinMode (N3,OUTPUT);
pinMode (N4,OUTPUT);
pinMode (ENA,OUTPUT);
pinMode (ENB,OUTPUT);
pinMode (ENA1,OUTPUT);
pinMode (ENB1,OUTPUT);

}
void loop() {

  delay(999);  //cada segundo se generara una muestra y nos guardara en la
  //SD se eligio este tiempo por espacio donde se harian las pruebas.
  digitalWrite(N1,HIGH); //se activan los motores
  digitalWrite(N2,HIGH);
  digitalWrite(N3,HIGH);
  digitalWrite(N4,HIGH);
  analogWrite(ENA,contador2); //se le asigna cada salida pwm para el control de un motor
  analogWrite(ENB,contador3);
  analogWrite(ENA1,contador2);
  analogWrite(ENB1,contador3);
 myFile = SD.open("datalog.txt", FILE_WRITE);//abrimos  el archivo

  if (myFile) {
        Serial.print("Escribiendo SD: ");


    //en este apartado se compara los datos de ambos encoders y tambien con el valor
    //de la potencia maxima que tiene es decir que si el encoders derecho es mayor al
    //encoder izquierdo pero tiene un valor de potencia de 255 pulsos pwm, este tendra
    //que bajar para emparejar el numero de interrupciones.
         if((contador>contador1)&&(contador2<=255)){
          contador2--;
        }
        if((contador1>contador)&&(contador3<=255)) {
          contador3--;
        }

        myFile.print("izquierda="); //se escribe en la SD
        myFile.print(contador);
        myFile.print(", derecha=");
        myFile.print(contador1);
        myFile.print(", contador2=");
        myFile.print(contador2);
        myFile.print(", contador3=");
        myFile.print(contador3);
        myFile.print(", contador4=");
        myFile.println(contador4);

        myFile.close(); //cerramos el archivo

        Serial.print("izquierda="); //se manda al puerto serial si esta
        Serial.print(contador);     //conectado.
        Serial.print(", derecha=");
        Serial.print(contador1);
        Serial.print(", contador2=");
        Serial.print(contador2);
```

```
        Serial.print(", contador3=");
        Serial.print(contador3);
        Serial.print(", contador4=");
        Serial.println(contador4);


  } else {
    Serial.println("Error al abrir el archivo");
  }
  delay(10);
  contador = 0; //se iguala a cero los contadores de interrupciones
   contador1 = 0;
contiempo++;

}

void interrupcion0() //se ejecuta el incremento de interrupcion
{
  contador++;

}

void interrupcion1()
{
  contador1++;
}
```

4.2.1 CARGA DE 1.7K

Las llantas de tracción utilizadas fueron de 5 pulgadas de diámetro de vex, el robot móvil en la parte trasera tenía que soportar una carga de 1.7k en este caso se usó una batería de ácido-plomo para realizar las pruebas.

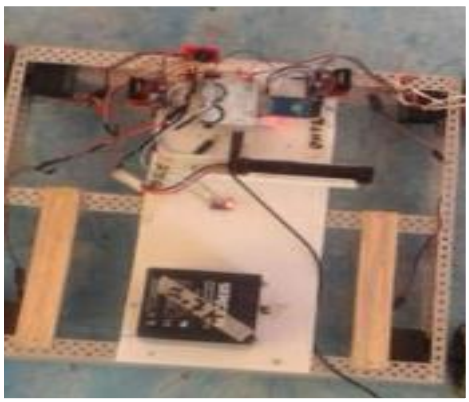
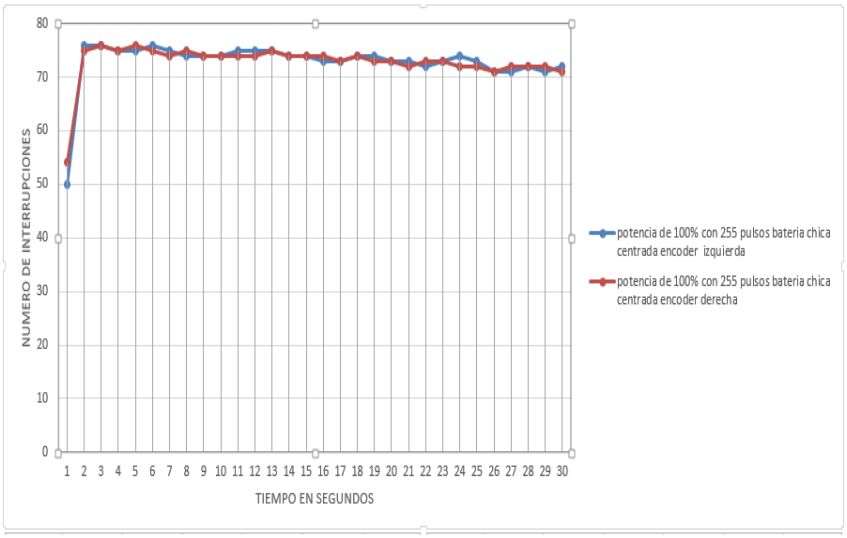


imagen 4.2.1.1 carga de 1.7k

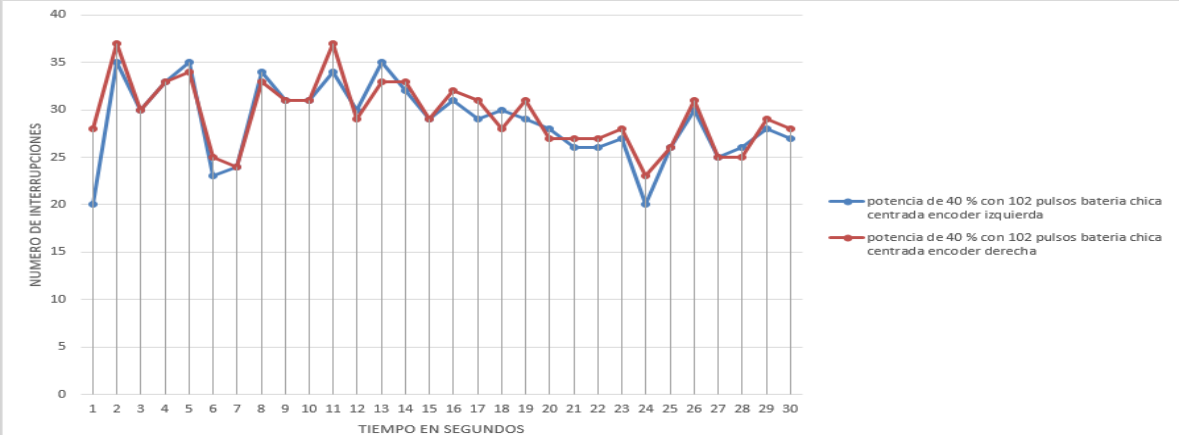
Con este peso se obtuvieron los datos del comportamiento del sistema y la desviación que existía con diferentes posiciones: centrada, izquierda o derecha.

Carga centrada



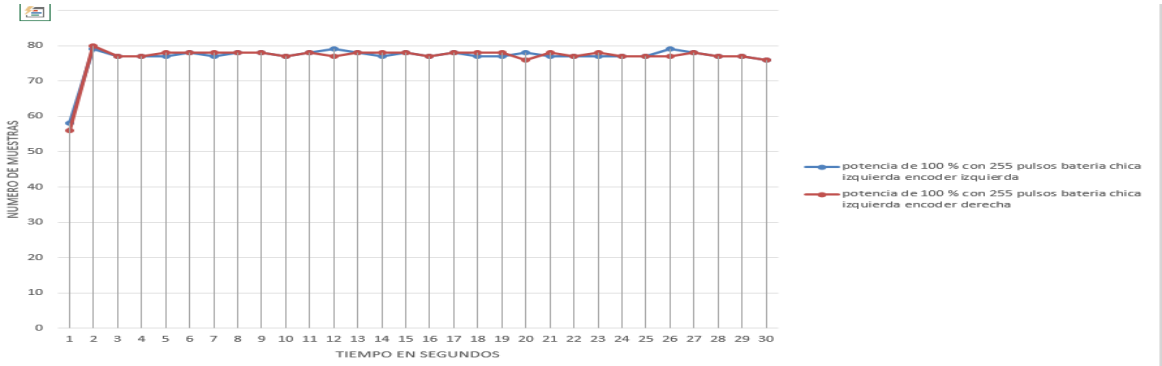
Comportamiento de la velocidad de motores usando el 100% como potencia máxima con 4 motores activos

Como potencia mínima con la carga centrada es de 40% debajo de esta potencia el robot se detiene.

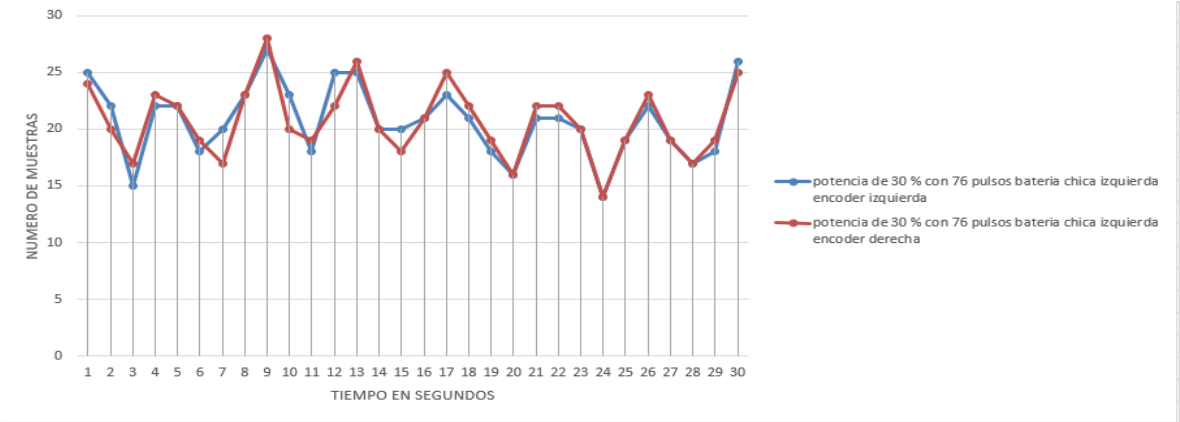


Carga de lado izquierdo

A potencia de 100% con la carga del lado izquierdo obtenemos el siguiente comportamiento.

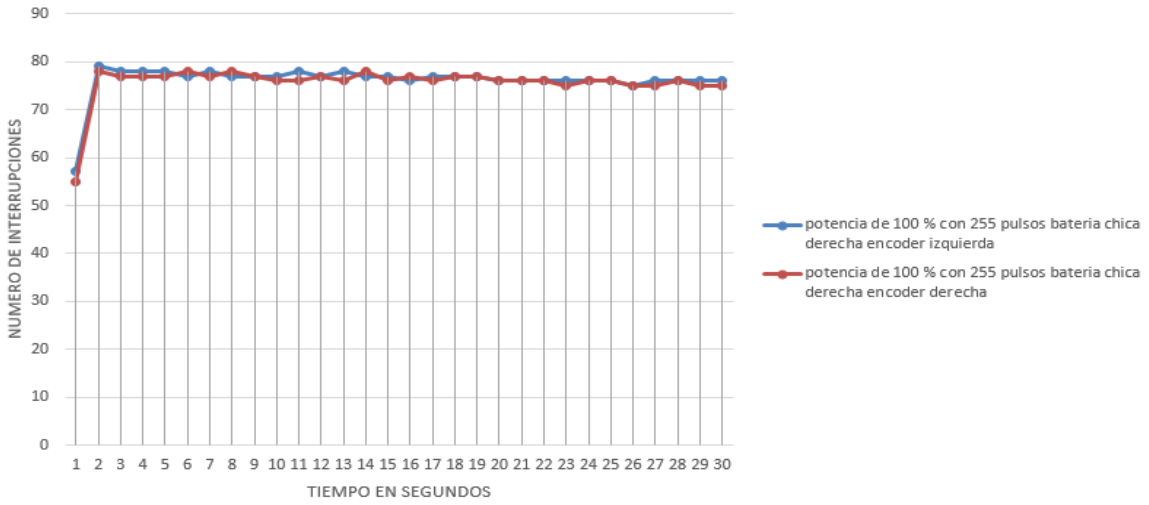


Como potencia mínima para la carga centrada obtenemos que debe de ser 30%.

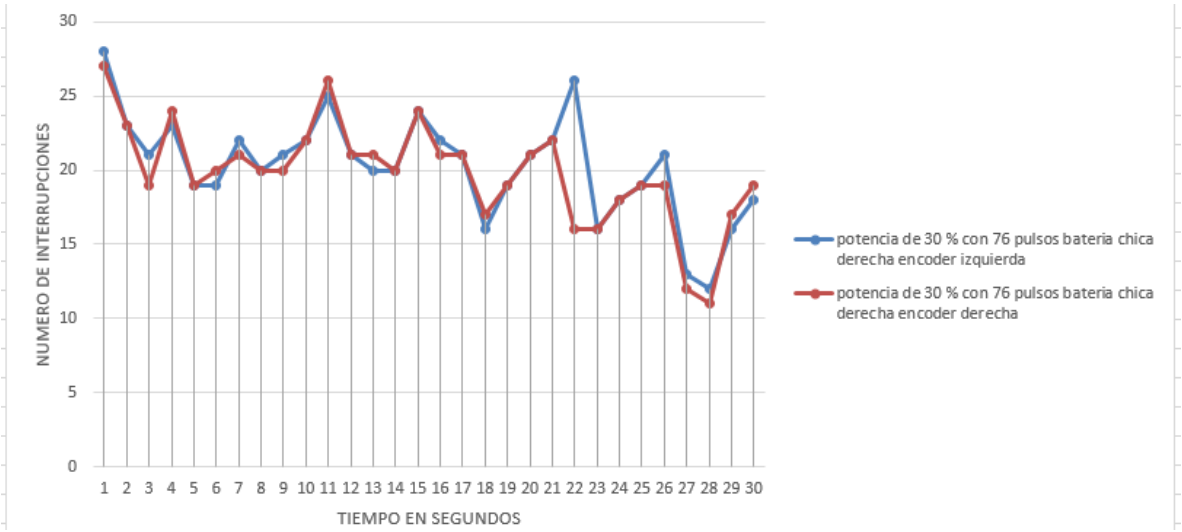


Carga de lado derecho

Con la carga del lado derecho obtenemos el siguiente comportamiento a una potencia del 100%.



La potencia mínima encontrada con carga de 1.7k del lado derecho es 30% si se le manda un pulso por debajo de este el robot se detiene.

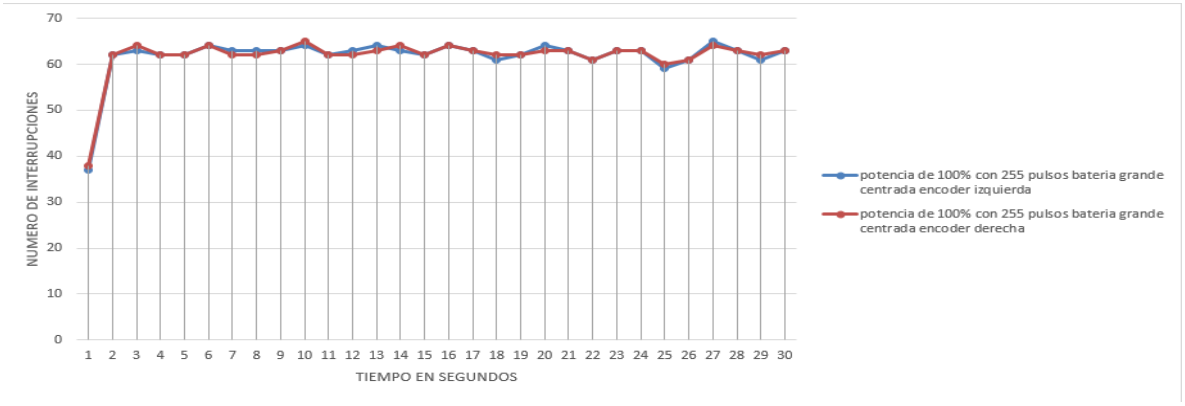


4.2.2 CARGA DE 2.5K

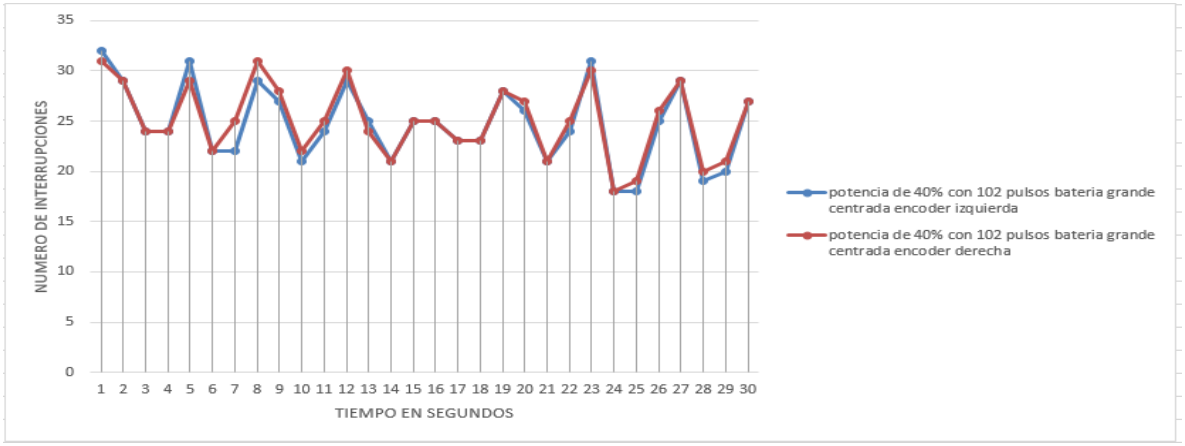
Con una carga de 2.5k para esto se usó una batería de ácido-plomo con ese peso para poder realizar las pruebas del funcionamiento.

Carga centrada

Con una potencia del 100% obtenemos el siguiente comportamiento.

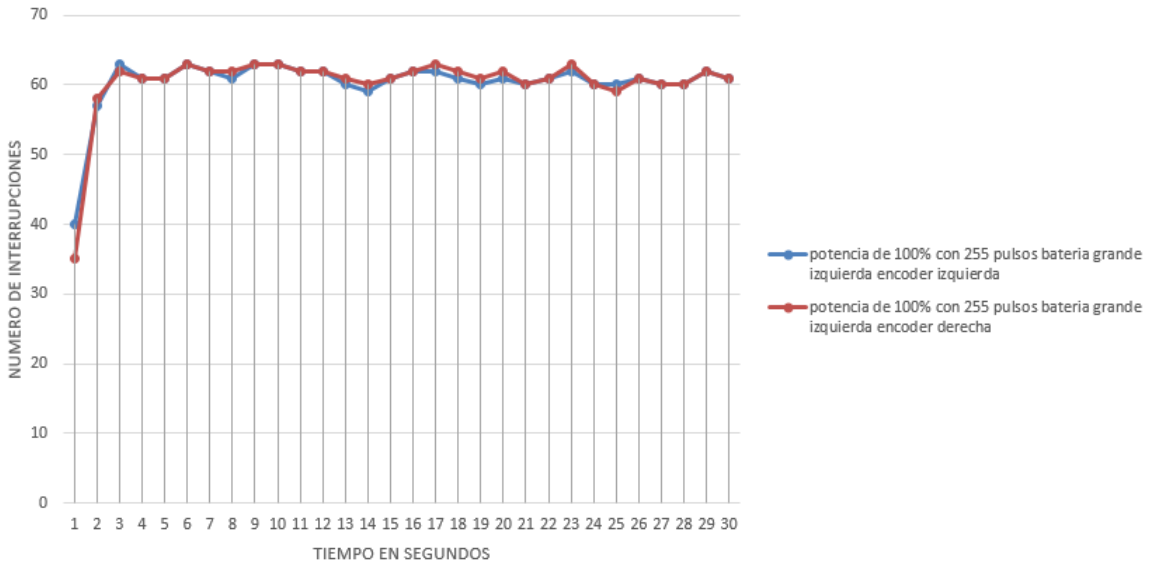


Como potencia mínima se obtuvo como resultado el 40%.

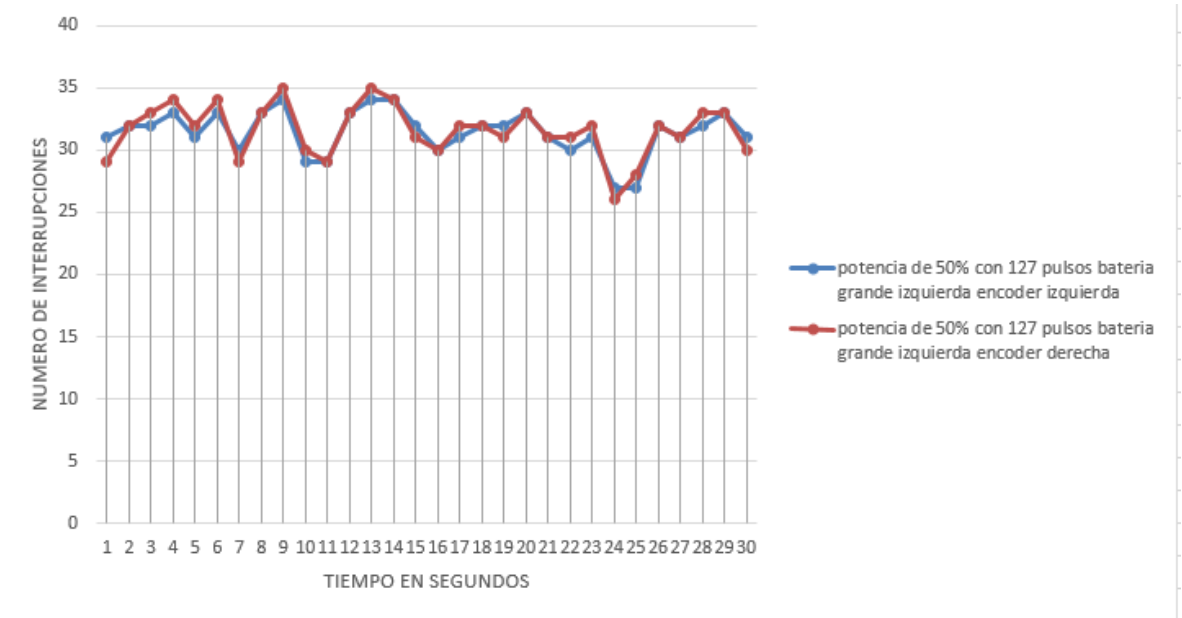


Carga de lado Izquierdo

Si la carga la cambiamos al lado izquierdo el comportamiento del robot móvil nos genera la siguiente gráfica.



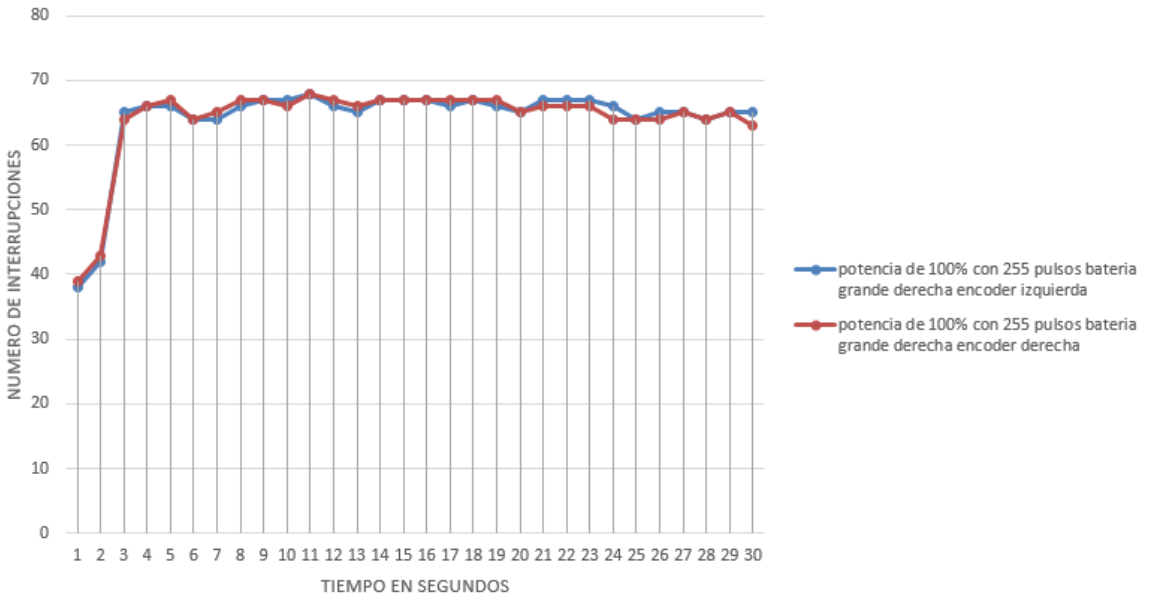
Como resultado a la potencia mínima encontrada con 127 pulsos pwm corresponde al 50% de la potencia total, en esta grafica se puede apreciar que hay cruces en la gráfica.



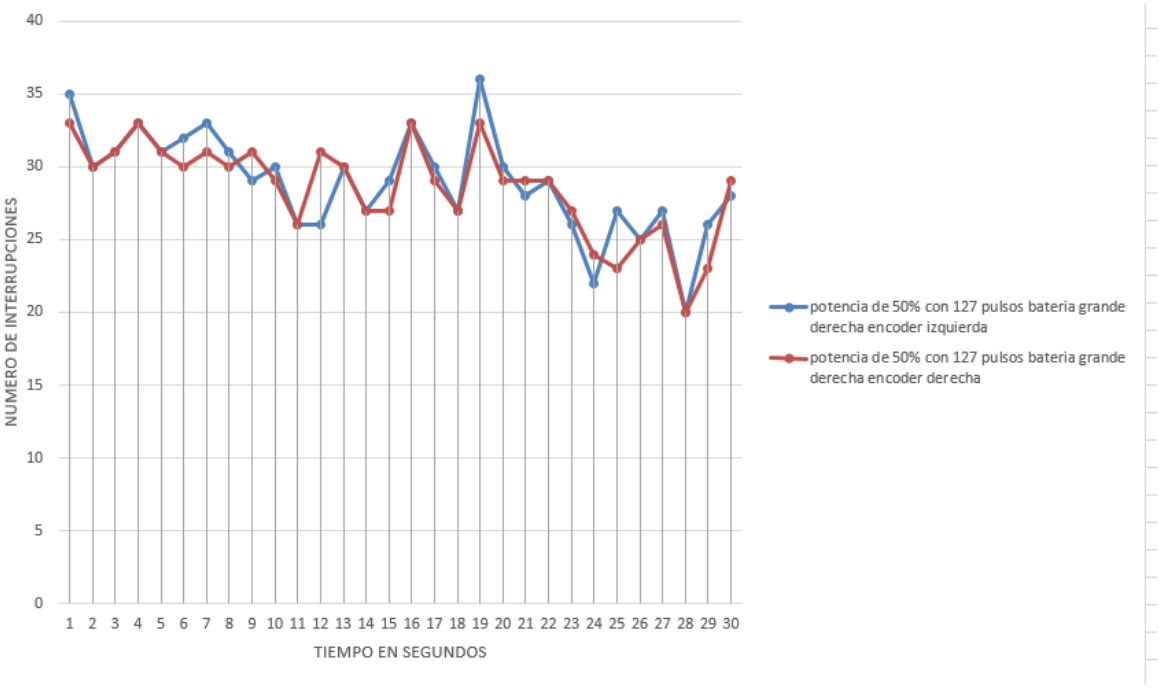
Carga de lado derecho

Como por última prueba con el peso de 2.5k, se cambió la carga al lado derecho y obtuvimos el siguiente resultado donde se ve una mejor estabilidad de la desviación.

.



Para potencia mínima se obtuvo la gráfica siguiente con 127 pulsos pwm.

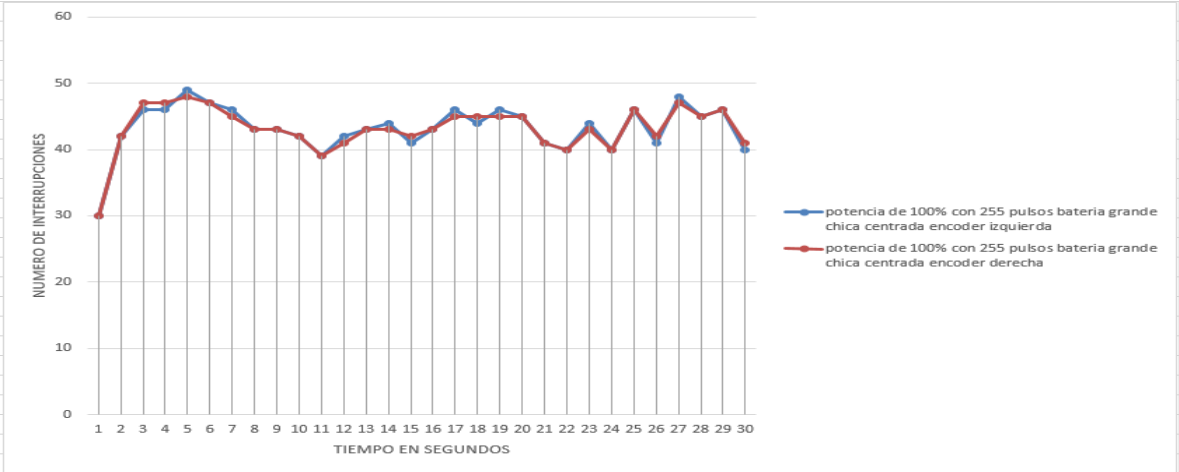


4.2.3 CARGA DE 4.2K

La combinación de la carga de batería chica (1.7k) y la batería grande (2.5k) obtenemos un total de 4.2k las cuales se combinaron para poder ver la carga máxima que nuestro robot soportaba.

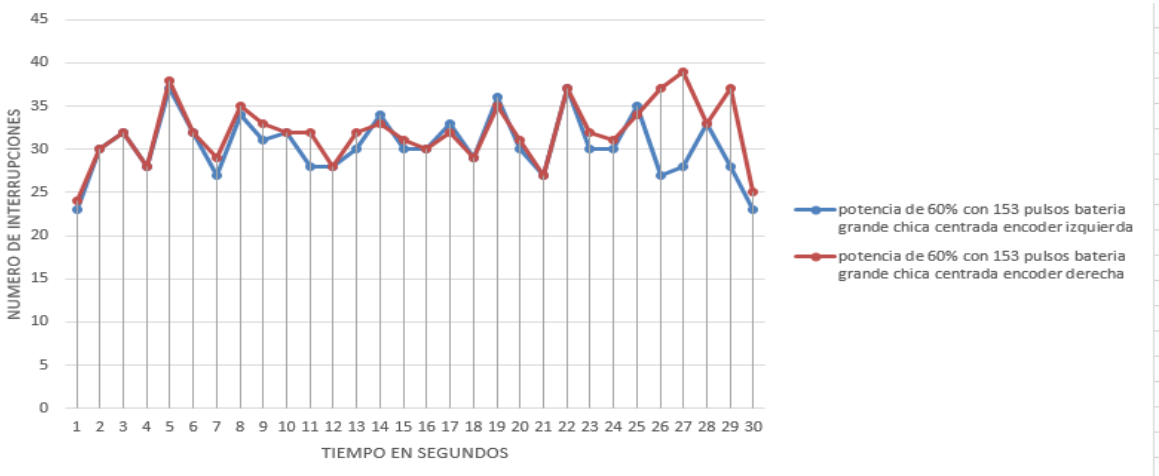
Carga centrada

Para una potencia de 100% el comportamiento de nuestro robot obtenemos.



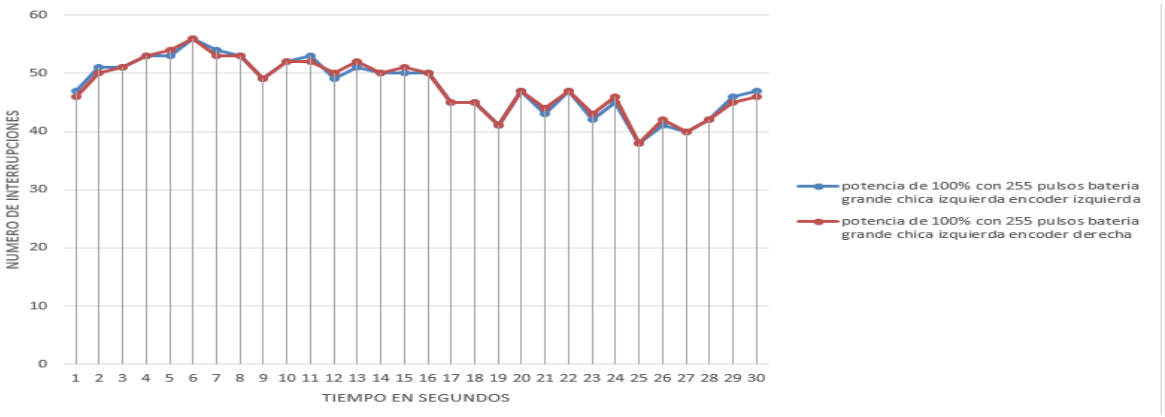
Como se puede observar en la imagen anterior se observa una mejora en la diferencia de potencia de los lados de nuestro robot provocando haci una menor desviación.

Ahora para una potencia mínima se obtuvo el siguiente resultado.

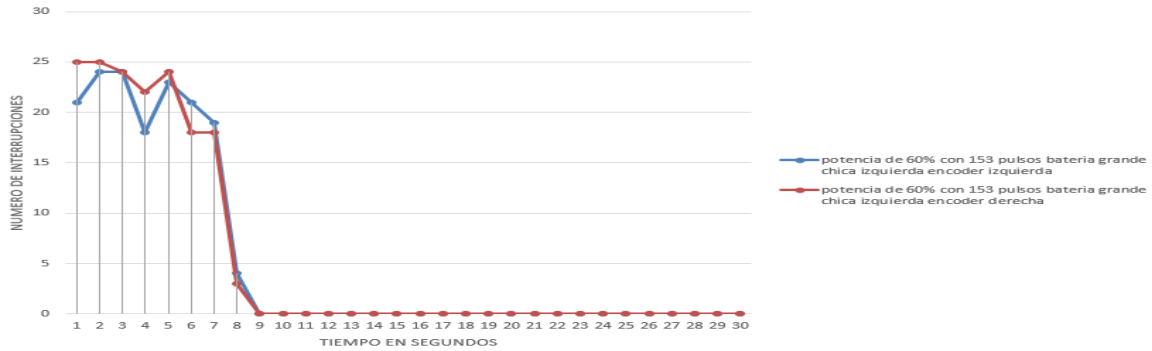


Carga del lado izquierdo

Para una carga de 4.2k colocada del lado izquierdo se obtuvo el siguiente comportamiento.

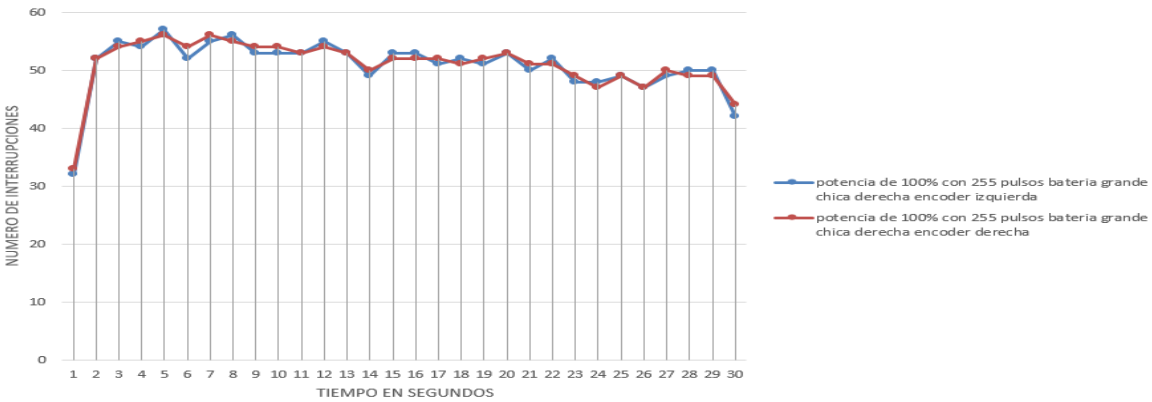


Para la potencia mínima obtenida fue de 60% con 153 pulsos pwm.

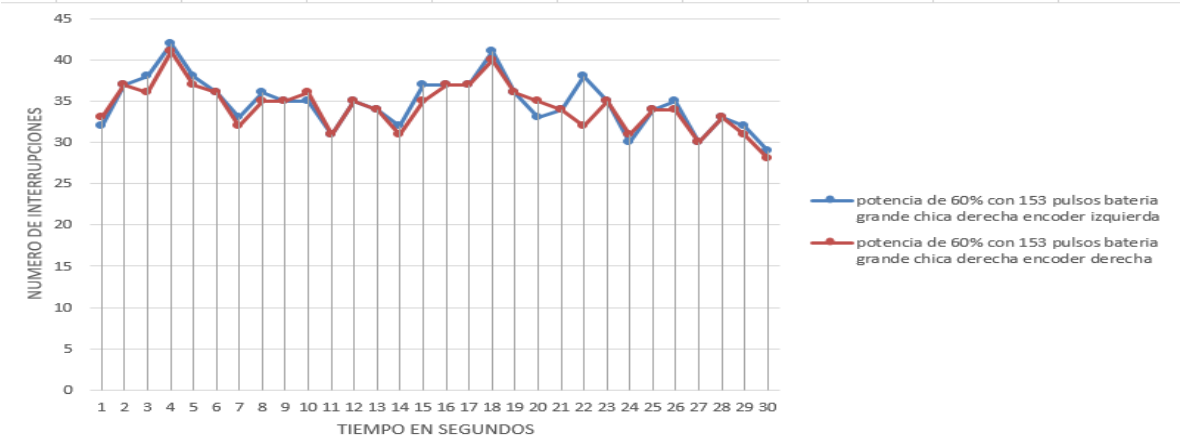


Carga de lado derecho

Se tiene una estabilidad en la desviación con el 100% de la potencia.



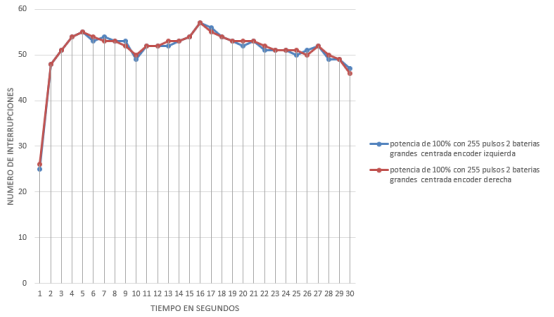
La potencia mínima encontrada fue del 60%.

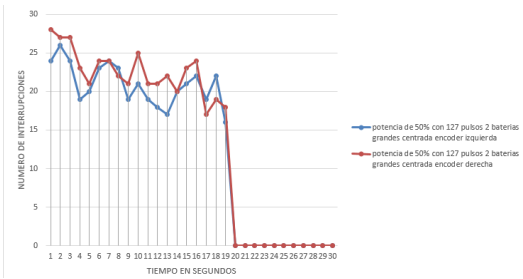


4.2.4 CARGA DE 5K

Carga centrada

Como el peso es mayor la fuerza requerida para que el robot pueda moverse es mayor el rango fue bajando conforme se le aumentaba la carga, en este caso se obtuvo que el robot podría moverse entre el 50% al 100% de la potencia.





Carga de lado izquierdo

El rango de movilidad del robot con una carga de 5k colocada del lado izquierdo va de los 80% al 100% de la potencia. Las graficas de los resultados se encuentran en los anexos

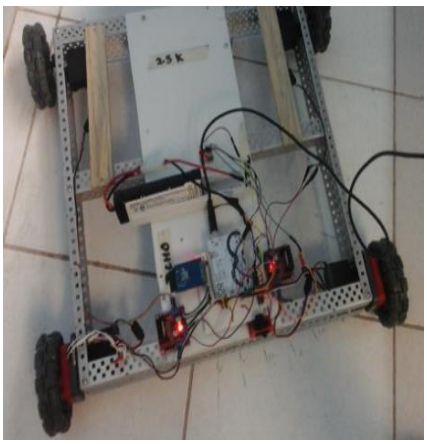
Carga de lado derecho

El rango de movilidad alcanzado con la carga puesta del lado derecho fue mayor ya que nos permitio una movilidad desde el 60% de la potencia hasta el 100% de ella. Los resultados graficados se encuentran en los anexos.

4.3 VELOCIDAD CON LLANTAS OMNIDIRECCIONAL

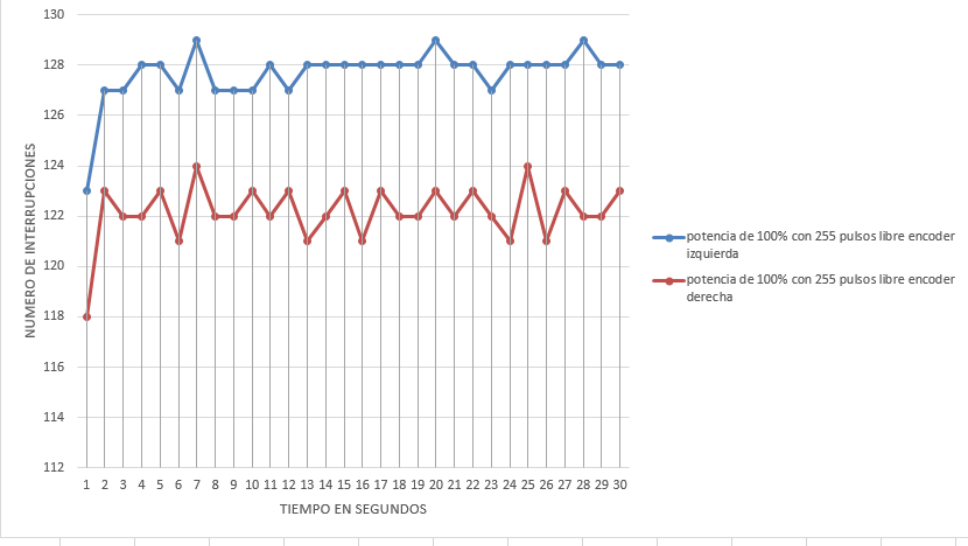
Para mejorar la fricción se decidió cambiar las llantas de tracción por llantas omnidireccional de la misma marca vex, la razón por la que estas cuentan con rodillos de un plástico menos rígido al de la llanta de tracción.



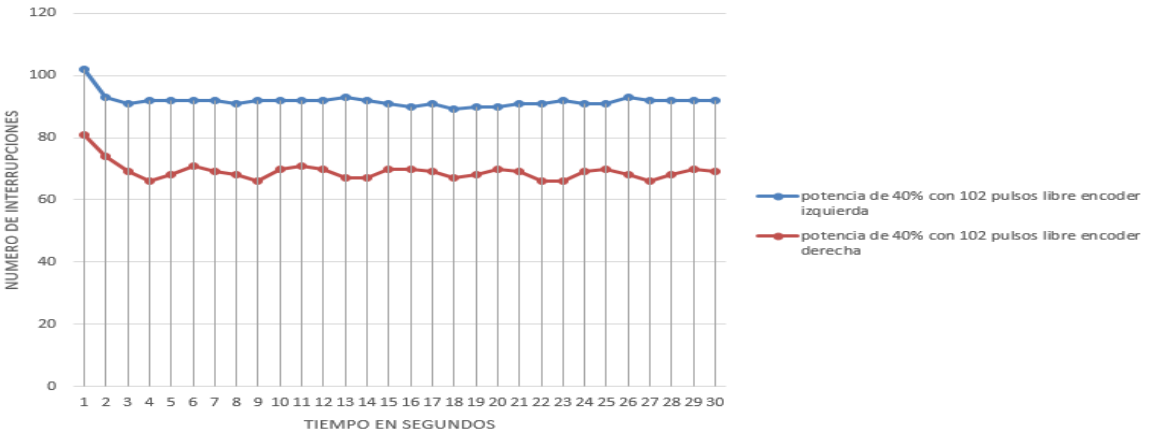


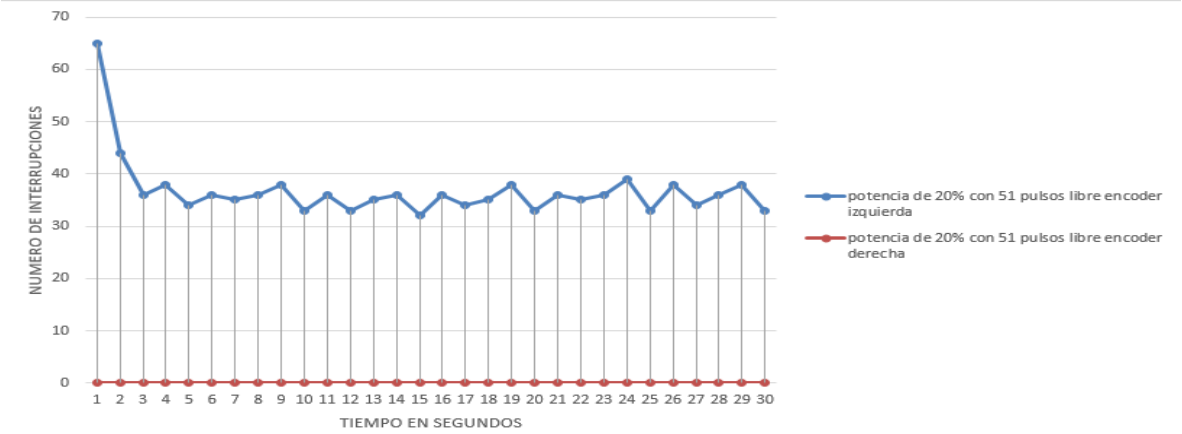
Como prueba inicial al igual que las pruebas con las llantas de traccion se realizo pruebas con las llantas omnidireccional con dos motores delanteros sin tener contacto con el suelo para poder ver el rango de velocidad.

El resultado de la potencia al 100% nos arojo como resultado la siguiente grafica.

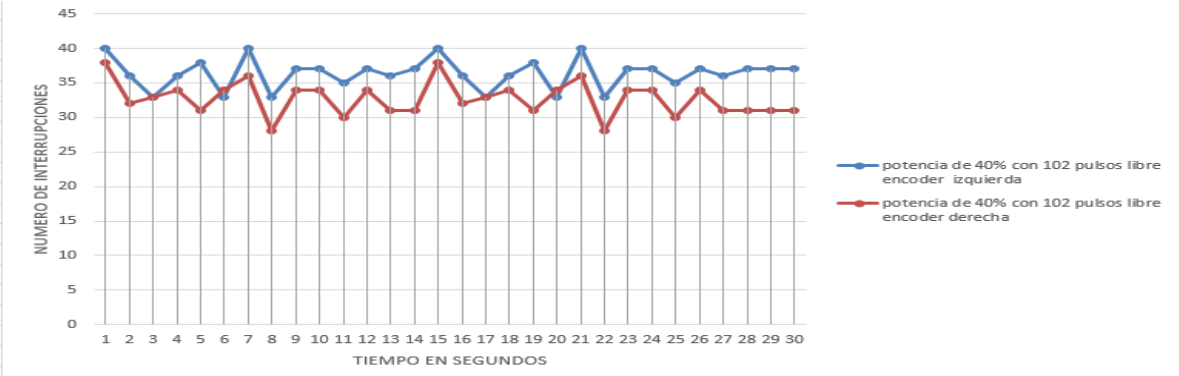


Como se puede observar en la gráfica siguiente la potencia mínima de funcionamiento de los motores es 20 % para el lado izquierdo y para el lado derecho es de 40% por debajo de esos niveles los motores se detienen.

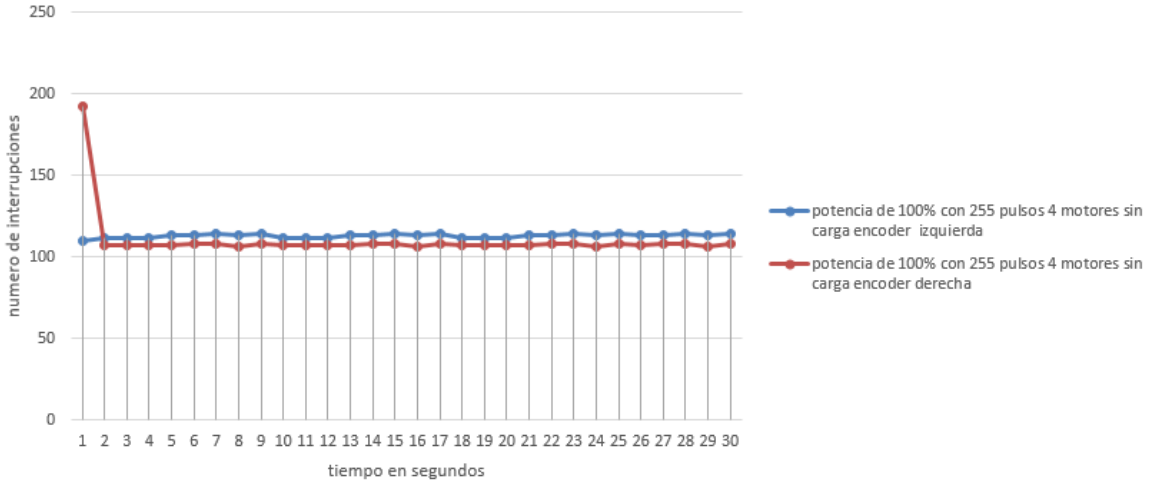




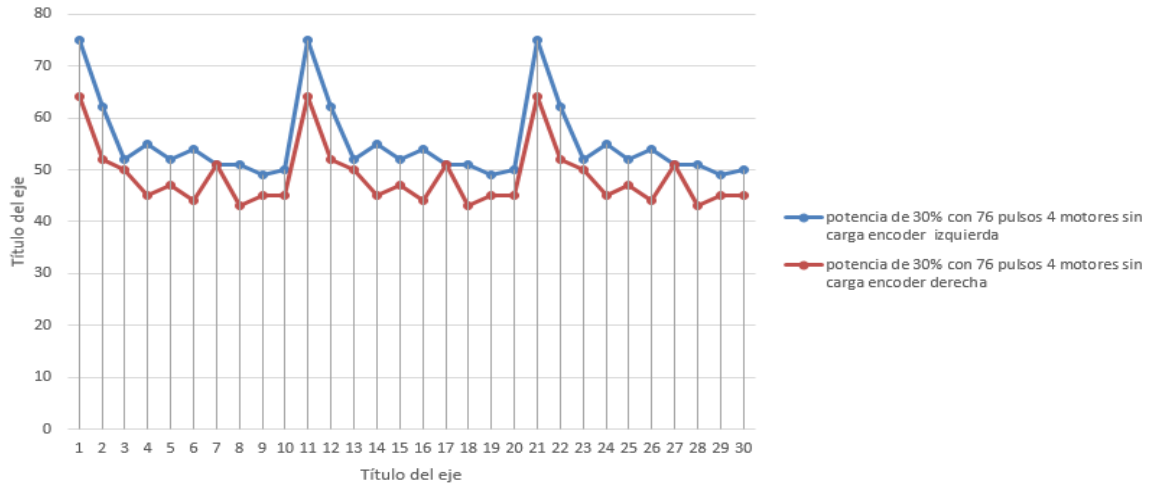
Como resultado usando los dos motes pero haciendo fricción con el suelo soportando únicamente el peso del prototipo se obtiene los siguientes resultados para potencia mínima lado izquierdo y lado derecho fue de 40%.



Incluyendo la potencia de los cuatro motores en contacto con el suelo con una potencia de 100% se genera el siguiente gráfico.



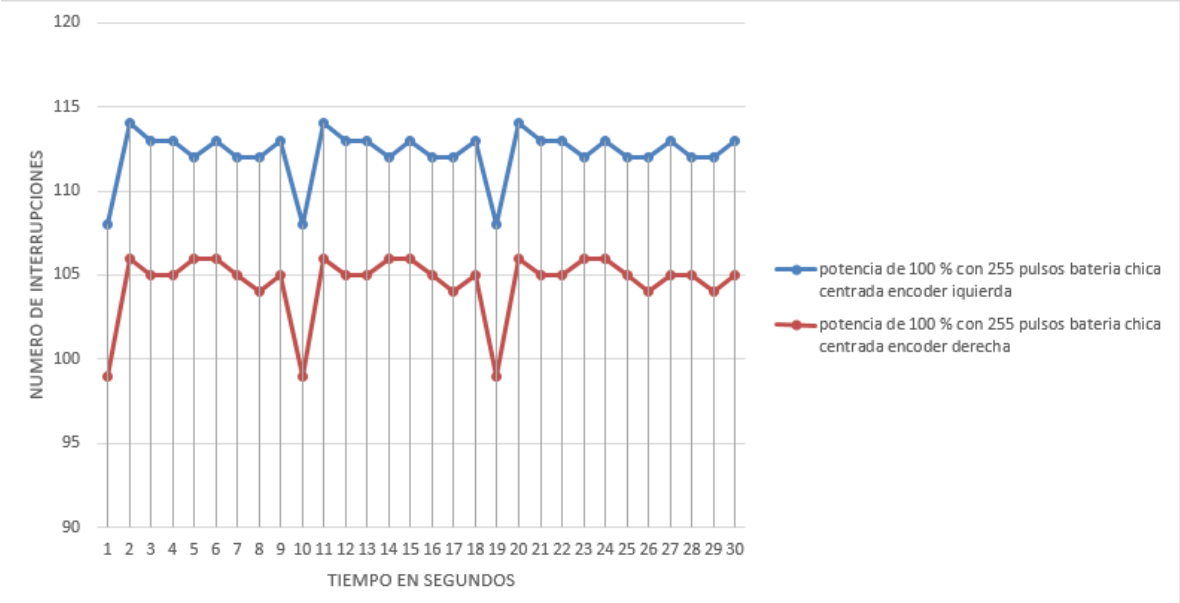
Como resultado la potencia mínima alcanzada es de 30% de la potencia total por debajo de esta los motores se detienen.



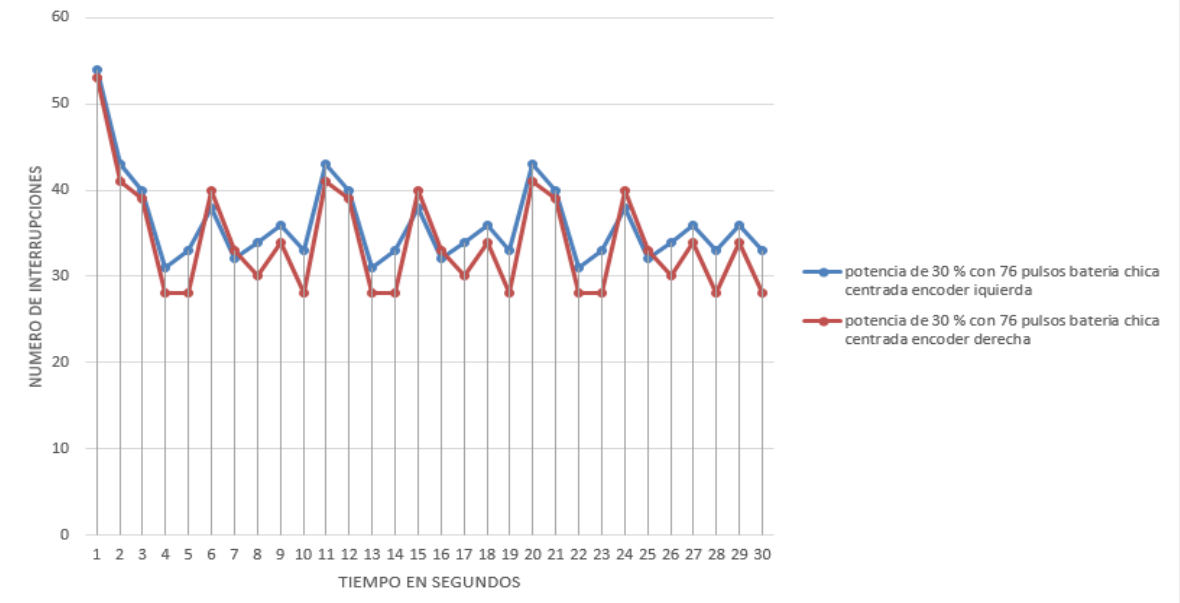
4.3.1 CARGA DE 1.7K

Centrada

A la máxima potencia con una carga de 1.7k extra al del prototipo la gráfica resultante nos genera.

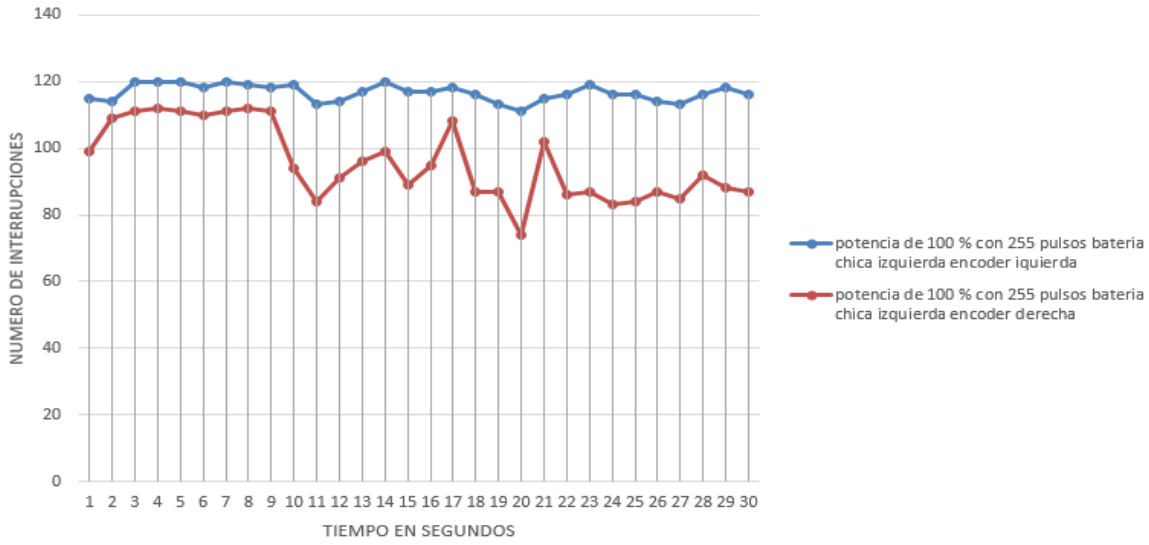


La potencia mínima soportada para que el robot no se detenga con una carga de 1.7k es 30%

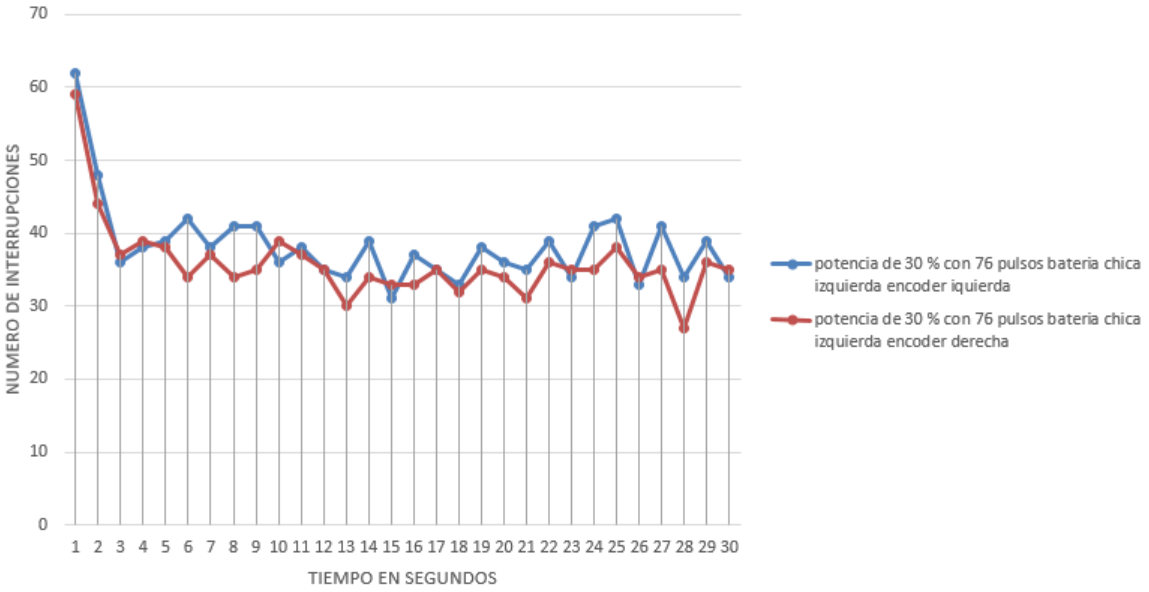


Carga del lado Izquierdo

Con una potencia de 100% el resultado generado es la siguiente.

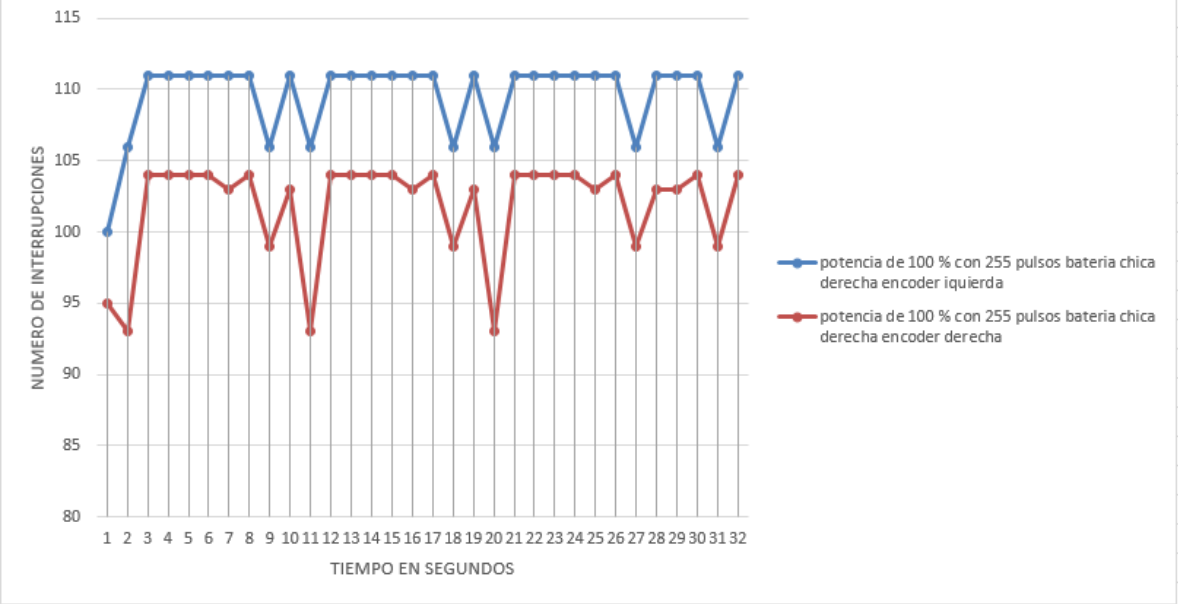


Como potencia mínima obtuvimos la siguiente grafica con 30% de la potencia total equivalente a 76 pulsos pwm por debajo de esta potencia el robot se detiene.

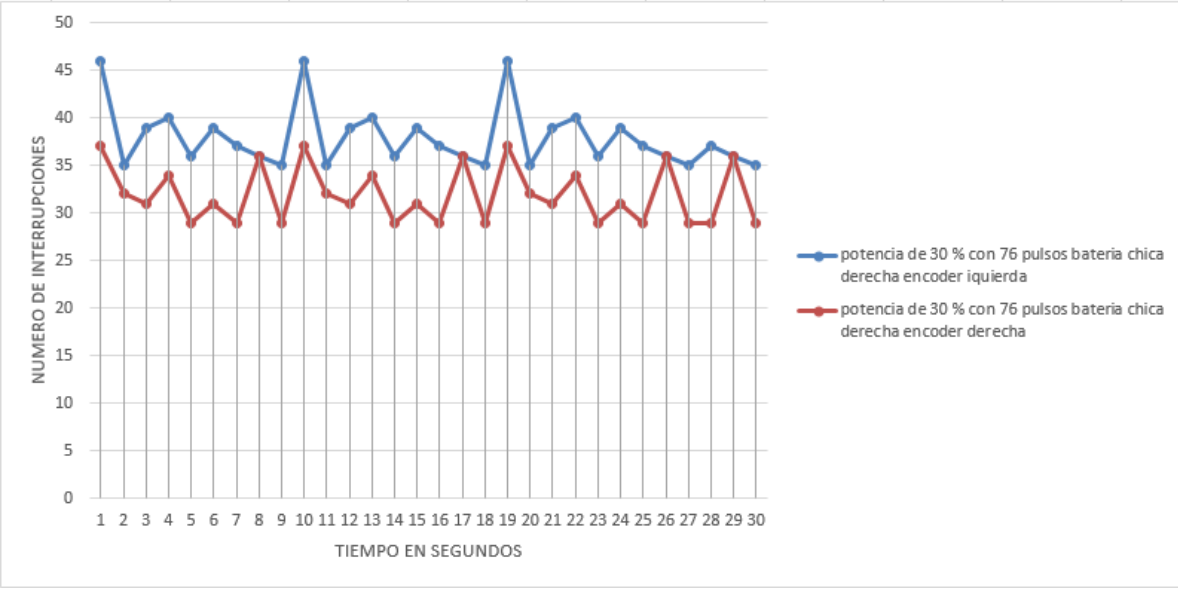


Carga del lado Derecho

A una potencia de 100% como resultado nos genera.



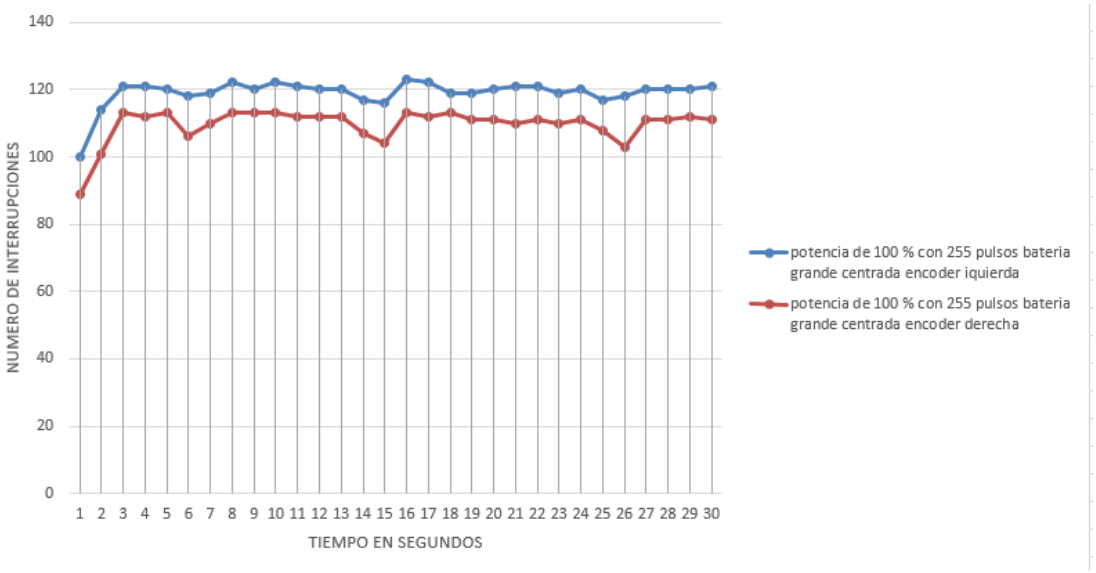
Como potencia mínima 30% resultante nos da el siguiente comportamiento.



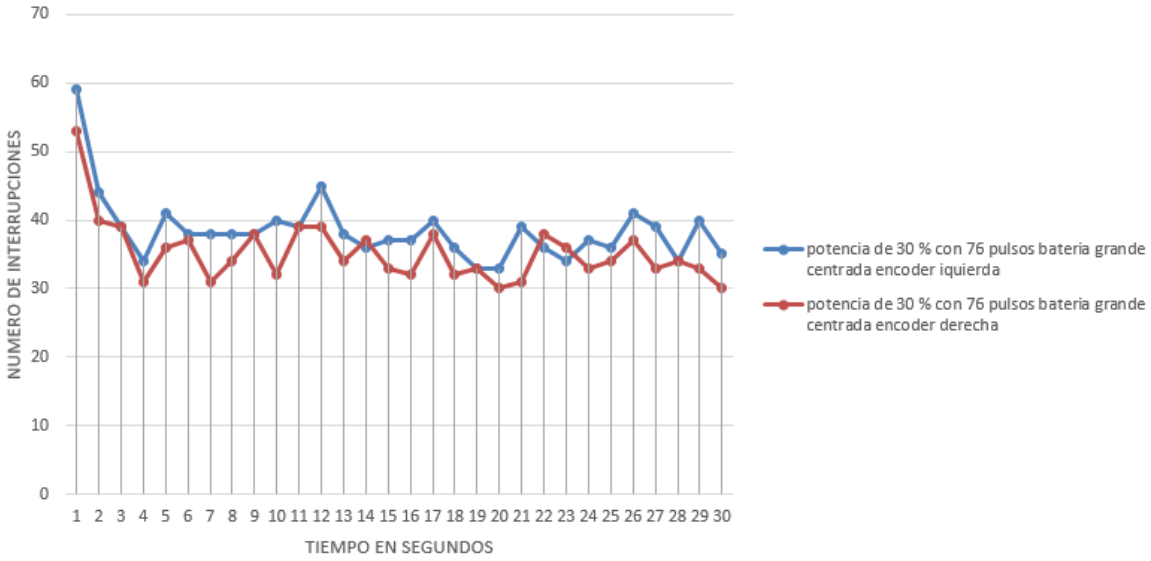
4.3.2 CARGA DE 2.5K

Carga centrada

Con una carga mayor con una potencia de 100% el comportamiento generado es el siguiente.

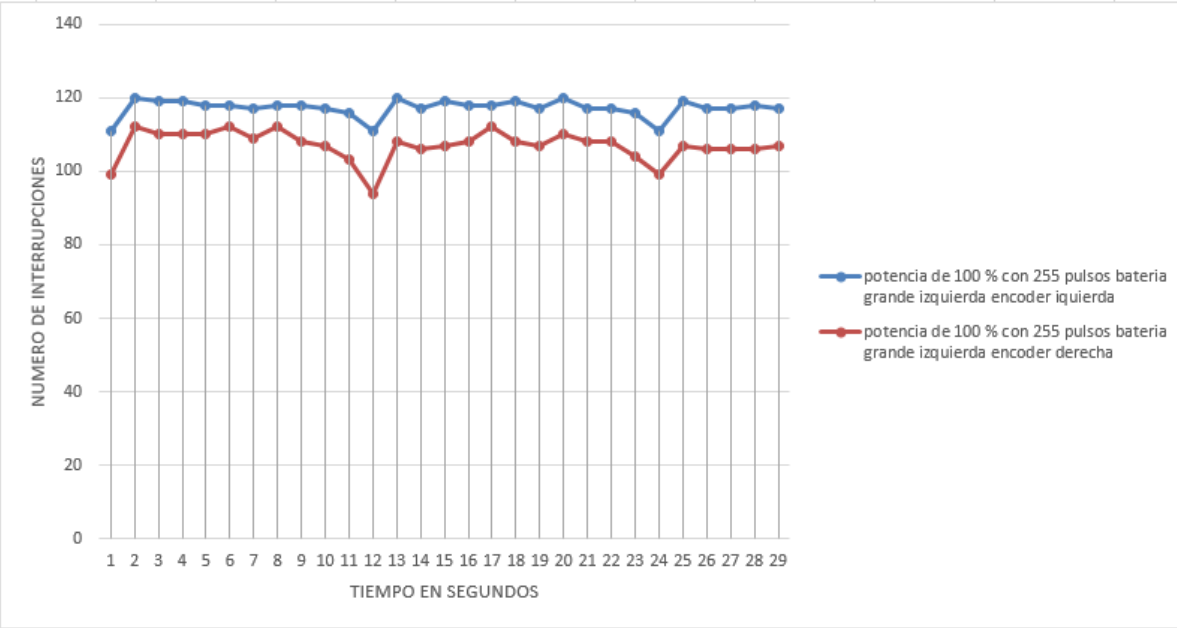


Al reducir la velocidad el robot se mantenía en movimiento hasta bajar a la potencia de 30% con 76 pulsos pwm.

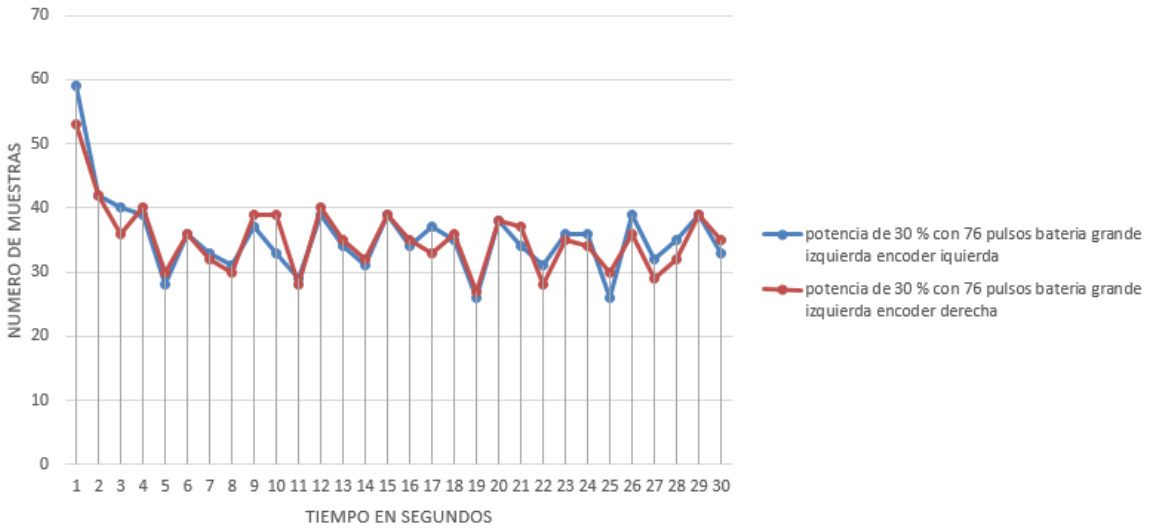


Carga del lado Izquierdo

Al mover la carga al lado izquierdo con una potencia de 100% los motores reaccionan de la forma siguiente.

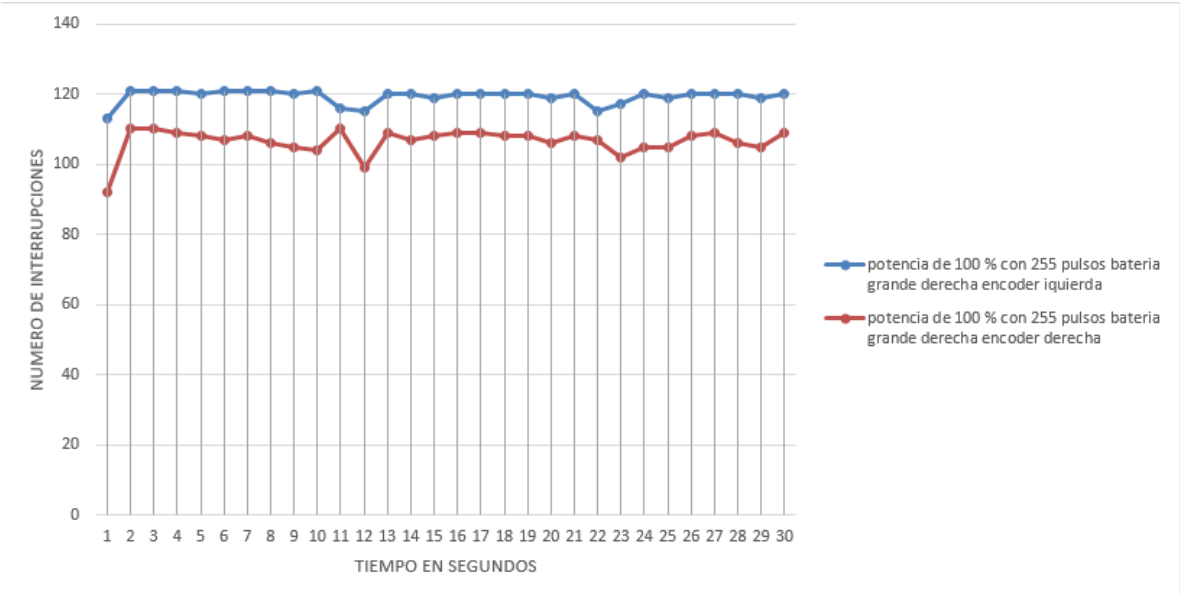


Como potencia mínima encontramos que es del 30%.

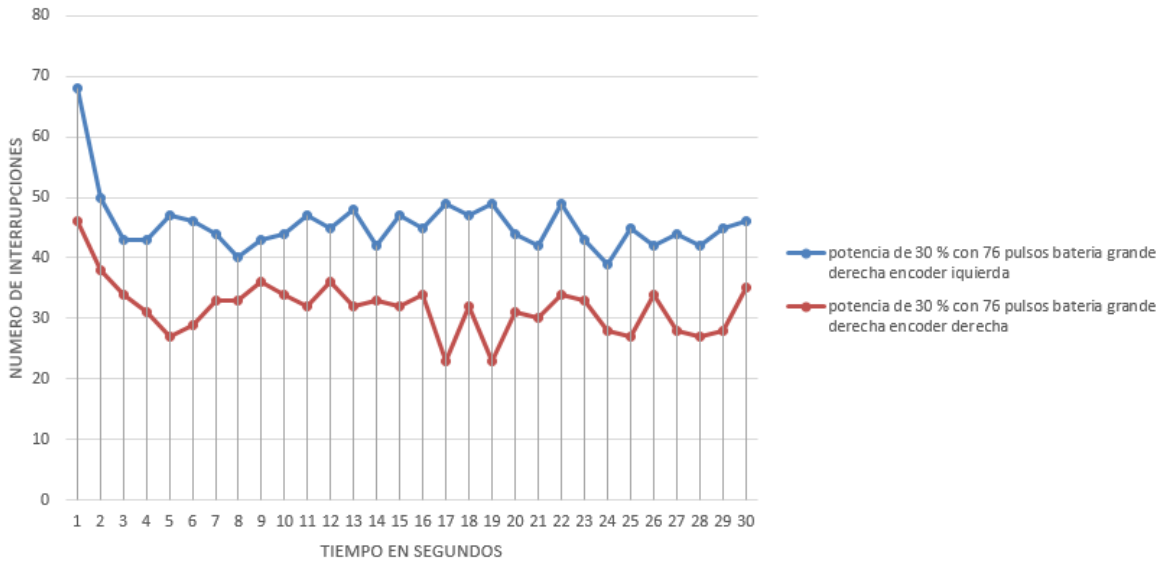


Carga del lado Derecho

Con el 100% de la potencia se logró obtener los datos que se muestran a continuación graficados.



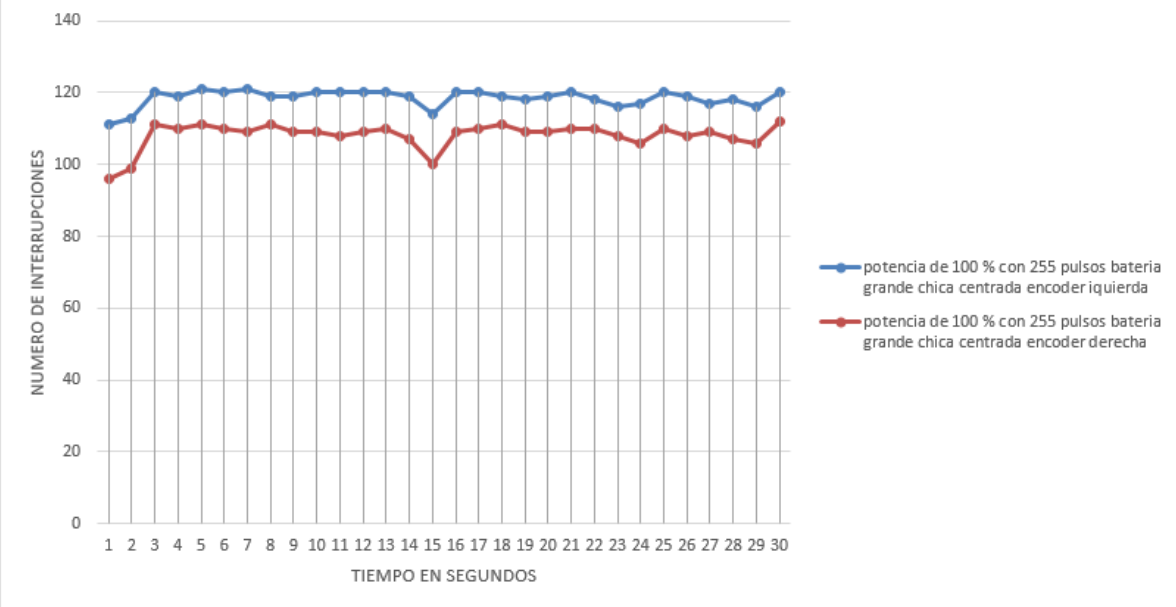
La potencia mínima para que nuestro robot se mantuviera en movimiento fue de teniendo como resultado lo siguiente.



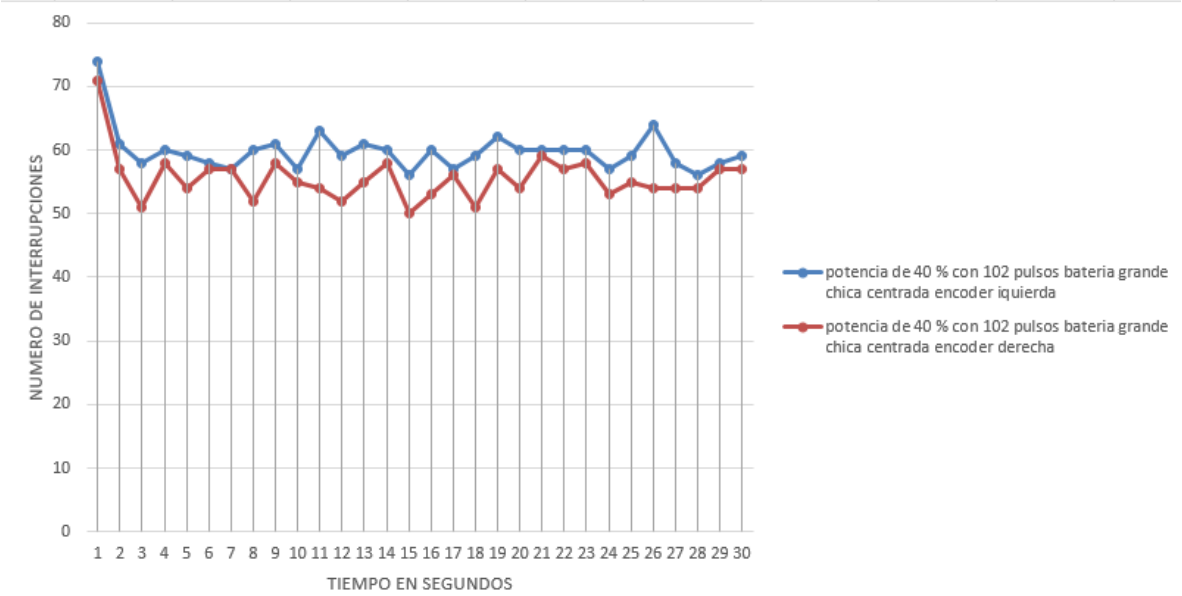
4.3.3 CARGA DE 4.2K

Carga centrada

Para una potencia del 100% para los motores los datos obtenidos nos generan el siguiente gráfico.

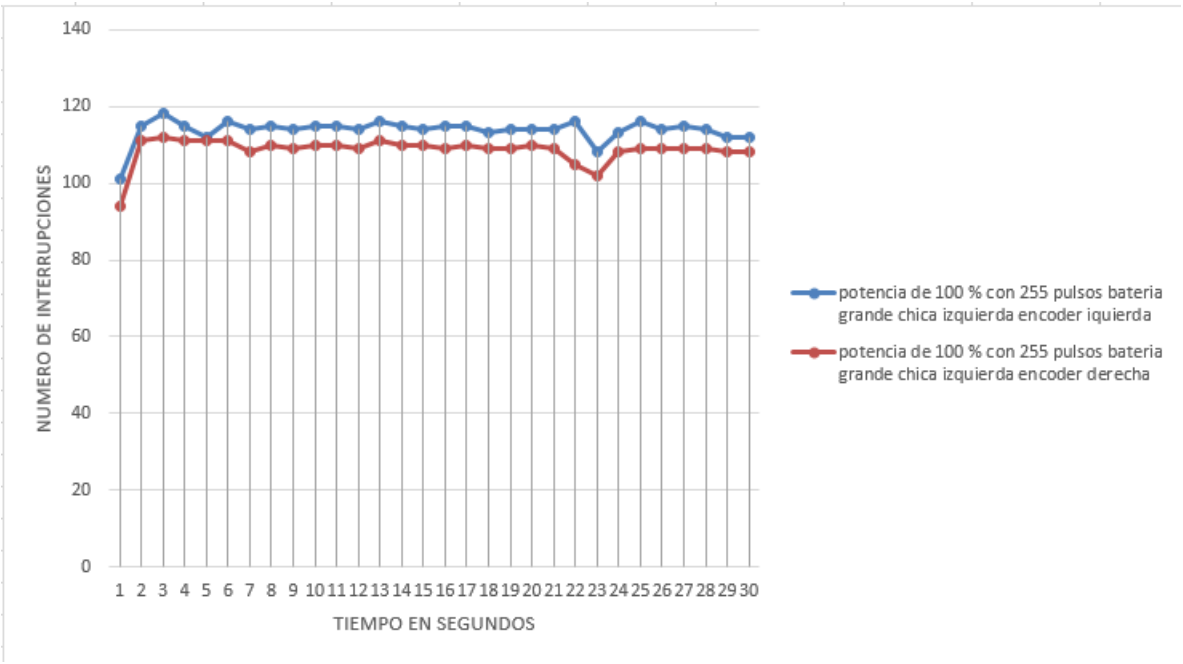


Con una potencia mínima del 40% como el peso aumento la potencias mínimas es mayor a las anteriores.

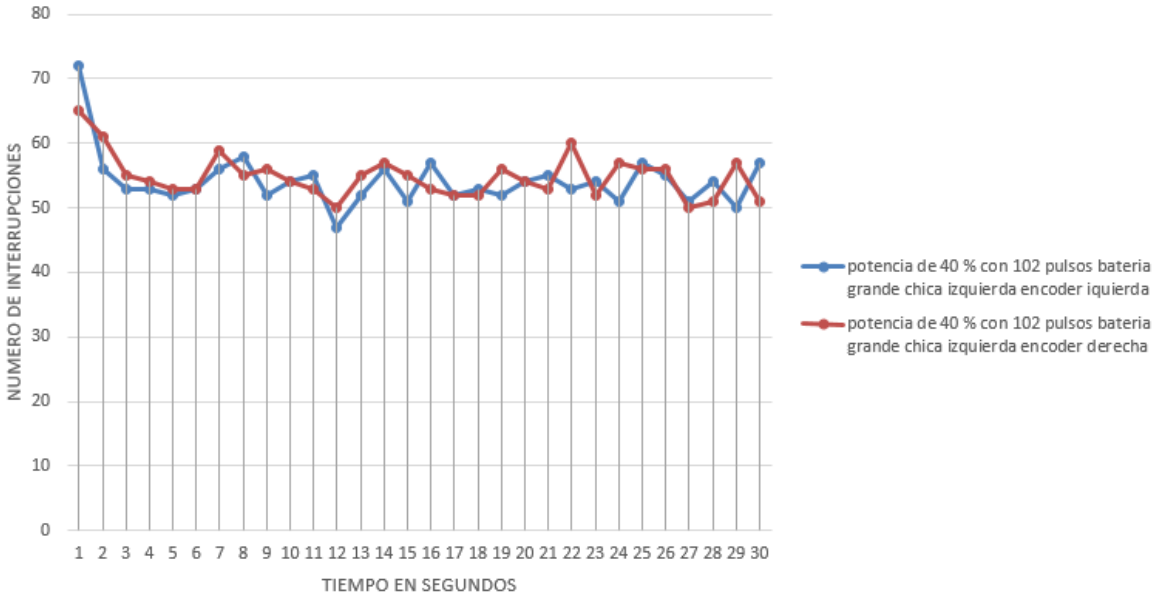


Carga del lado Izquierdo

Usando la potencia máxima de los motores con la carga del lado izquierdo el comportamiento de la velocidad del giro se muestra en la siguiente gráfica.

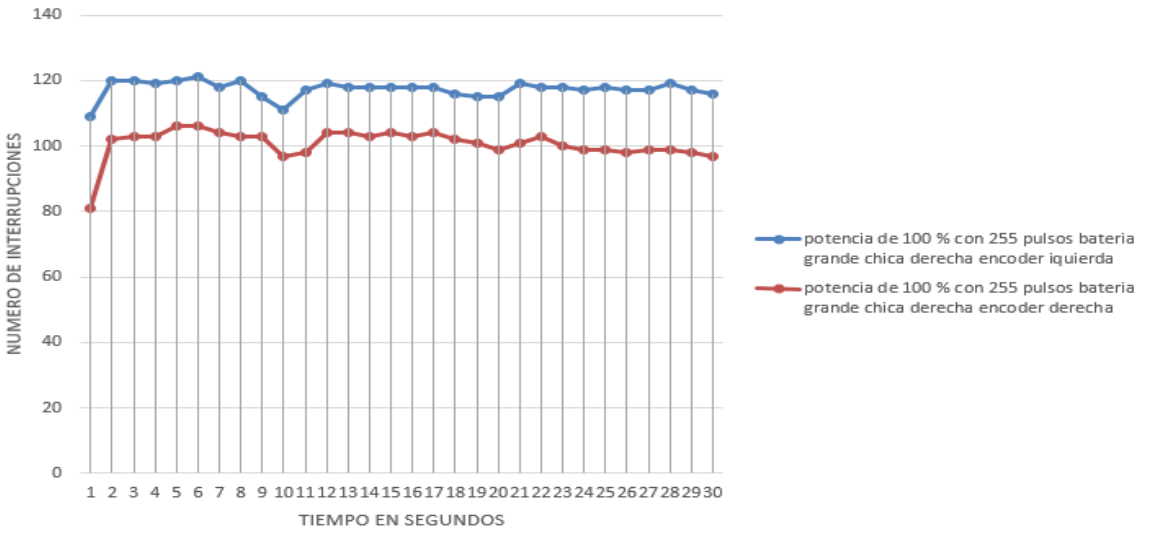


La potencia mínima para que se mantuviera en movimiento es de 40%.

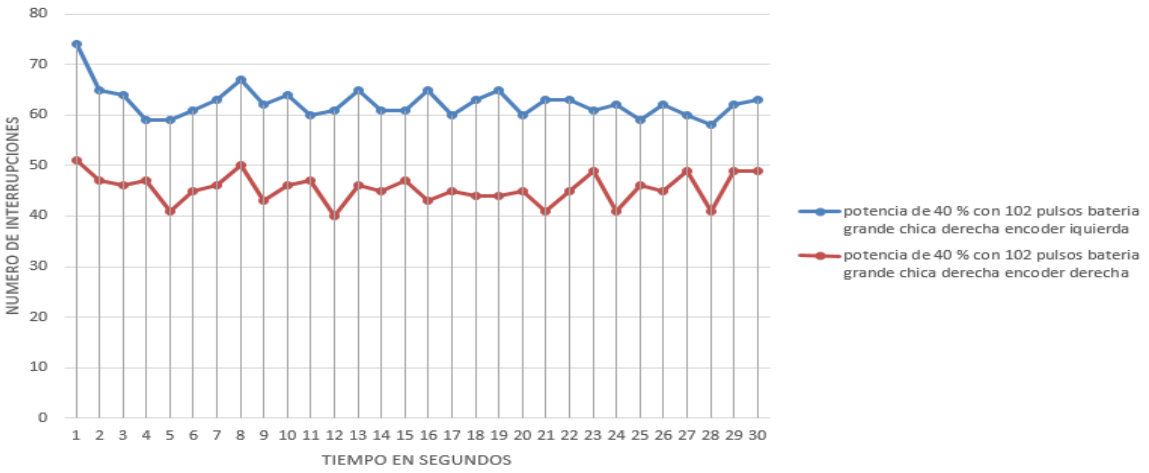


Carga del lado Derecho

Con una potencia de 100% el comportamiento de nuestro sistema resulto.



Y como potencia mínima obtuvimos que es el 40%



4.3.4 CARGA DE 5K

El rango de la potencia con una carga de 5k ronda entre 40% al 100% como se muestra en las siguientes graficas menor a esta potencia el robot se detendrá. Ya sea que la carga está centrada, en el lado izquierdo o del lado derecho

4.4 PRUEBAS DE GIRO CON MPU6050

Para poder manejar el mpu6050 es necesario calibrarlo para ello hacemos uso del siguiente programa.

```
// Librerias I2C para controlar el mpu6050
// la libreria MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implicito
MPU6050 sensor;

// Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
int ax, ay, az;
int gx, gy, gz;

//Variables usadas por el filtro pasa bajos
long f_ax,f_ay, f_az;
int p_ax, p_ay, p_az;
long f_gx,f_gy, f_gz;
int p_gx, p_gy, p_gz;
int counter=0;

//Valor de los offsets
int ax_o,ay_o,az_o;
int gx_o,gy_o,gz_o;

void setup() {
  Serial.begin(57600);    //Iniciando puerto serial
  Wire.begin();            //Iniciando I2C
  sensor.initialize();     //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");
```

```
  // Leer los offset los offsets anteriores
  ax_o=sensor.getXAccelOffset();
  ay_o=sensor.getYAccelOffset();
  az_o=sensor.getZAccelOffset();
  gx_o=sensor.getXGyroOffset();
  gy_o=sensor.getYGyroOffset();
  gz_o=sensor.getZGyroOffset();

  Serial.println("Offsets:");
  Serial.print(ax_o); Serial.print("\t");
  Serial.print(ay_o); Serial.print("\t");
  Serial.print(az_o); Serial.print("\t");
  Serial.print(gx_o); Serial.print("\t");
  Serial.print(gy_o); Serial.print("\t");
  Serial.print(gz_o); Serial.print("\t");
  Serial.println("nnEnvie cualquier caracter para empezar la calibracionnn");
  // Espera un caracter para empezar a calibrar
  while (true){if (Serial.available()) break;}
  Serial.println("Calibrando, no mover IMU");

}

void loop() {
  // Leer las aceleraciones y velocidades angulares
  sensor.getAcceleration(&ax, &ay, &az);
  sensor.getRotation(&gx, &gy, &gz);

  // Filtrar las lecturas
  f_ax = f_ax-(f_ax>>5)+ax;
  p_ax = f_ax>>5;
```

```
  f_ay = f_ay-(f_ay>>5)+ay;
  p_ay = f_ay>>5;

  f_az = f_az-(f_az>>5)+az;
  p_az = f_az>>5;

  f_gx = f_gx-(f_gx>>3)+gx;
  p_gx = f_gx>>3;

  f_gy = f_gy-(f_gy>>3)+gy;
  p_gy = f_gy>>3;

  f_gz = f_gz-(f_gz>>3)+gz;
  p_gz = f_gz>>3;

  //Cada 100 lecturas corregir el offset
  if (counter==1){
    //Mostrar las lecturas separadas por un [tab]
    Serial.print("promedio:"); Serial.print("t");
    Serial.print(p_ax); Serial.print("\t");
    Serial.print(p_ay); Serial.print("\t");
    Serial.print(p_az); Serial.print("\t");
    Serial.print(p_gx); Serial.print("\t");
    Serial.print(p_gy); Serial.print("\t");
    Serial.println(p_gz);

    //Calibrar el acelerometro a 1g en el eje z (ajustar el offset)
    if (p_ax>0) ax_o--;
    else {ax_o++;}
    if (p_ay>0) ay_o--;


     else {ay_o++;}
     if (p_az-16384>0) az_o--;
     else {az_o++;}

     sensor.setXAccelOffset(ax_o);
     sensor.setYAccelOffset(ay_o);
     sensor.setZAccelOffset(az_o);

     //Calibrar el giroscopio a 0°/s en todos los ejes (ajustar el offset)
     if (p_gx>0) gx_o--;
     else {gx_o++;}
     if (p_gy>0) gy_o--;
     else {gy_o++;}
     if (p_gz>0) gz_o--;
     else {gz_o++;}

     sensor.setXGyroOffset(gx_o);
     sensor.setYGyroOffset(gy_o);
     sensor.setZGyroOffset(gz_o);

     counter=0;
   }
   counter++;


  if((p_az==16384)){//&&(p_ax==0)&&(p_ay==0)&&(p_gx==0)&&(p_gy==0)&&(p_gz==0)){
  Serial.print("bueno:"); Serial.print("t");
    Serial.print(p_ax); Serial.print("\t");
    Serial.print(p_ay); Serial.print("\t");
    Serial.print(p_az); Serial.print("\t");
    Serial.print(p_gx); Serial.print("\t");
    Serial.print(p_gy); Serial.print("\t");
    Serial.println(p_gz);
    delay(5000);
  }
}
```

Cuando el valor de az es cercano a 16384, se cierra el puerto serial o simplemente se desconecta el arduino de la computadora para que este valor quede guardado en el arduino y posterior mente lo reconozca.

Para la lectura del mpu6050 y guardar los datos se creó un nuevo programa en este nos arroja el ángulo medido con un error de ±1 grado, este error se le agrego como una tolerancia para evitar el cabeceo al momento de girar ya que para que del ángulo exacto estaría tratando de evitar los errores aunque sean mínimos.

```
#include <SPI.h> //se declaran las librerias para el mpu6050
#include <SD.h> //y para el lector de SD
File myFile;
#include <SoftwareSerial.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
volatile int contador = 0;//se declaran dos variables tipo
volatile int contador1 = 0; //volatile
 int N1=4; //se declaran los pines que usaremos
 int N2=5;
 int N3=7;
 int N4=10;
 int ENAdcha=6;
int ENBizqu=9;
int contador2=70;
int contador3=65;
int contadoraux=0;
int angulo=90;//se declara el angulo de giro de referencia
MPU6050 sensor;
int gx, gy, gz;
long tiempo_prev, dt; //se declaran variables para el giroscopio
float girosc_ang_x, girosc_ang_y, girosc_ang_z;
float girosc_ang_x_prev, girosc_ang_y_prev, girosc_ang_z_prev;
void setup() {

  Serial.begin(57600); //se inicializa los puertos seriales y sensores
   Wire.begin();
 sensor.initialize();
 if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");
  else Serial.println("Error al iniciar el sensor");

  tiempo_prev=millis();

  Serial.println("inicializacion exitosa");
  attachInterrupt(0,interrupcion0,RISING);
   attachInterrupt(1,interrupcion1,RISING);
 // LOW, CHANGE, RISING, FALLING
pinMode (N1,OUTPUT);//se declaran nuestros pines como salida
pinMode (N2,OUTPUT);
pinMode (N3,OUTPUT);
pinMode (N4,OUTPUT);
pinMode (ENAdcha,OUTPUT);
pinMode (ENBizqu,OUTPUT);
}

 void loop() {
  if(contadoraux==0){
    delay(5000);
  }
  sensor.getRotation(&gx, &gy, &gz); //se empieza la lectura del mpu6050
  dt = millis()-tiempo_prev;
  tiempo_prev=millis();
  girosc_ang_z = (gz/131)*dt/1000.0 + girosc_ang_z_prev;
  girosc_ang_z_prev=girosc_ang_z;
  if(angulo>0){
    if(girosc_ang_z<(angulo-1)){//si el angulo leido es menor al angulo de referencia se mantiene el giro
    digitalWrite(N1,HIGH);
  digitalWrite(N2,LOW);
  //para giro a la derecha n3=high y n4=low para la izquierda n3=low y n4=high
  digitalWrite(N3,LOW);
  digitalWrite(N4,HIGH);
```

```
  analogWrite(ENAdcha,contador3);
  analogWrite(ENBizqu,contador2);
 contador2++;
 contador3++;
  }
  if((angulo+1)<girosc_ang_z){ //se compara si el angulo girado es menor al angulo deceado se mantiene girando
  digitalWrite(N1,LOW);
  digitalWrite(N2,HIGH);
  //para giro a la derecha n3=high y n4=low para la izquierda n3=low y n4=high
  digitalWrite(N3, HIGH);
  digitalWrite(N4,LOW);
  analogWrite(ENAdcha,contador3);
  analogWrite(ENBizqu,contador2);
 contador2++;
 contador3++;
//  }
  }
  if((girosc_ang_z>(angulo-1))&&(girosc_ang_z<(angulo+1)))//si el angulo leido esta dentro del rango de error
  {                                                       //los motores se detienen.
  digitalWrite(N1,LOW);
  digitalWrite(N2,LOW);
  //para giro a la derecha n3=high y n4=low para la izquierda n3=low y n4=high
  digitalWrite(N3,LOW);
  digitalWrite(N4,LOW);
  analogWrite(ENAdcha,contador3);
  analogWrite(ENBizqu,contador2);
 contador2=0;
 contador3=0;
 }


   sensor.getRotation(&gx, &gy, &gz);
   dt = millis()-tiempo_prev;
   tiempo_prev=millis();
   girosc_ang_z = (gz/131)*dt/1000.0 + girosc_ang_z_prev;
   girosc_ang_z_prev=girosc_ang_z;
   if(contador1<contador){
     contador3--;
   }
    if(contador1>contador){
     contador2--;
   }
          Serial.print("izquierda=");
          Serial.print(contador1);
          Serial.print(", derecha=");
          Serial.print(contador);
          Serial.print(", contador2=");
          Serial.print(contador2);
          Serial.print(", contador3=");
          Serial.print(contador3);
          Serial.print("   tRotacion en z: ");
          Serial.println(girosc_ang_z);

  contador = 0;
   contador1 = 0;
 }
 if(contador2>254){      //para el contador 2 y 3 se dieron estos valores limites por
  contador2=contador2-1;//la diferencia de potencia que existe entre ellos
 }
 if(contador3>249){
 contador3=contador3-1;


 }

   contadoraux++;
   }

void interrupcion0() //se llevan acabo las interrupciones
{
   contador++;

}

void interrupcion1()
{
   contador1++;
}
```

4.4.1 GIRO SIN CARGA

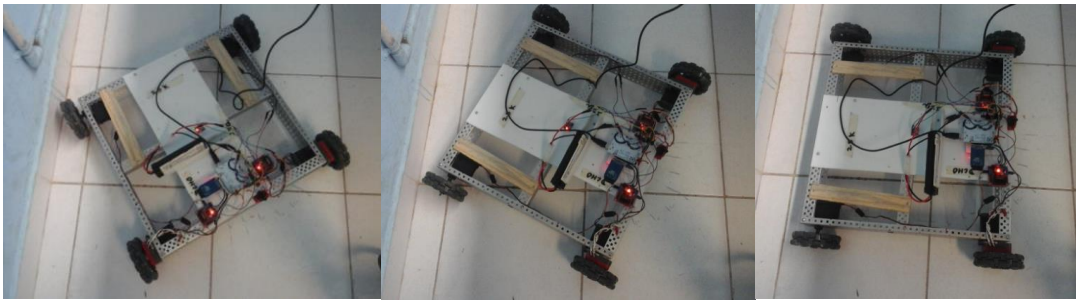
Como prueba inicial se llevó acabo la prueba de giro sin tener ningún peso que soporte los datos obtenidos como referencia se utilizara los ángulos medidos más lejanos en cada prueba, para tener una tabla comparativa entre el ángulo ideal, el ángulo medido y el ángulo real.

Ángulos		
ideal	Medido	real
10	5.49	5.3
20	19.09	18.90
30	29.19	30.3
40	39.30	39
50	49.15	48.5
60	59.09	57.08
70	69.02	69
80	79.11	78.3
90	89.35	85

Como se puede observar en la tabla, el ángulo real, medido e ideal son distintos.

La diferencia que se tiene entre el ángulo ideal y medido es el error en el programa creado para evitar el cabeceo al tratar de corregir el error obtenido.

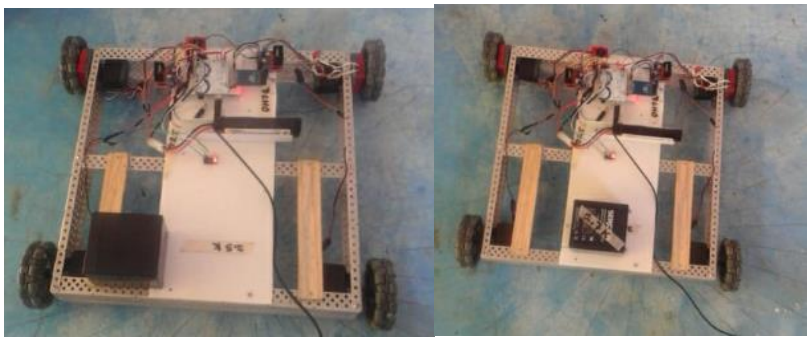
La diferencia que hay entre el ángulo medido y el ángulo real se debe al desplazamiento tanto en “Y”, como en “X” ya que al desviarse el punto central del robot móvil, el eje de referencia se mueve mientras el sensor mpu6050 da una lectura correcta físicamente no se aprecia por la desviación.



4.4.2 GIRO CON CARGA

En el giro con carga los valores cambian drásticamente esto se debe a la fuerza ejercida por el peso en la parte trasera del robot móvil. Al igual que la tabla anterior, a continuación se presenta una tabla donde se muestra la comparativa de los ángulos con pesos diferentes y en posiciones (centrada, izquierda y derecha).

Ángulos con carga de 1.7k		
Centrada		
ideal	Medido	real
30	29.05	35
60	59.16	71
90	90.03	100
Izquierda		
ideal	Medido	real
30	29.10	35
60	59.47	75
90	90.43	108
Derecha		
ideal	Medido	Real
30	29.15	40
60	59.23	80
90	89.96	110

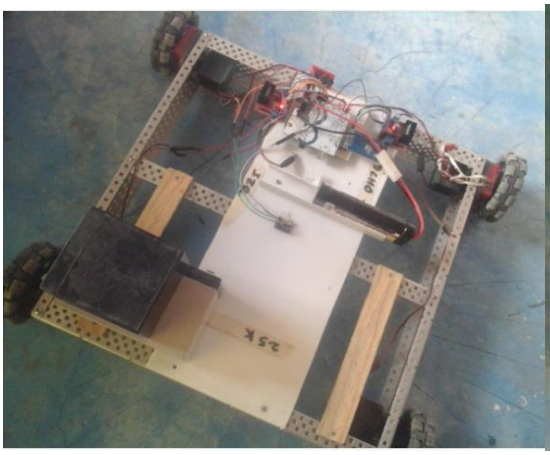


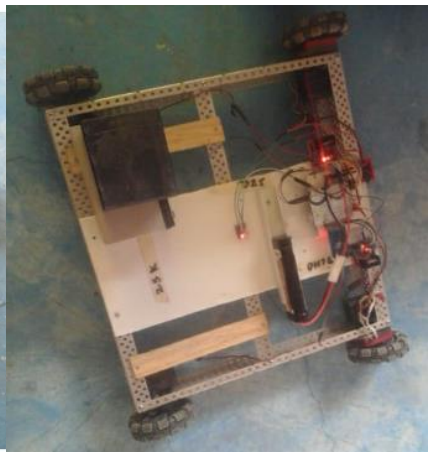
Ángulos con carga de 2.5k		
Centrada		
ideal	medido	Real
30	29.06	38
60	59.03	75
90	89.12	112
Izquierda		
ideal	medido	Real
30	29.17	42
60	59.52	80
90	89.72	110
Derecha		
ideal	medido	real
30	29.16	50
60	59	85
90	89.25	115

Entre mayor es la carga que moverá el robot la desviación del punto de referencia se incrementa por esta razón si vemos los datos de la primer tabla con un peso de 1.7k y comparamos con la de 2.5k vemos que el valor real se encuentra más lejos que el valor medido por el mpu6050.

Ángulos con carga de 4.2k		
Centrada		
ideal	medido	real
30	29.17	45
60	59.16	80
90	89.04	112
izquierda		
ideal	medido	real
30	29.02	46
60	59.12	83
90	89.45	114
derecha		
ideal	medido	real
30	30.29	55
60	59.15	92
90	90	115

Al seguir aumentando el peso el error de medición aumenta es decir no tenemos una lectura que nos permita tener certeza en los datos que obtenemos con el valor real girado, dado en ángulos.





CONCLUSIÓN

El robot recolector de pet, es de mucha ayuda en la aplicación de este dentro de la sociedad ya que con este proyecto se mejoraría la limpieza de calles, playas, escuelas, etc., una de las ventajas es que funciona mediante baterías recargables lo cual puede llegar a ser sustentable con un sistema fotovoltaico.

La instrumentación e implementación del robot recolector de pet, requiere de un sistema con una exactitud muy buena para lograr el buen funcionamiento de este es por ello que se hicieron muchas pruebas con el prototipo, la función de los sensores como son los encoders nos brinda la oportunidad de ver el funcionamiento tanto el de los motores como la distancia recorrida del recolector de pet, gracias a un sistema muy sencillo pero practico.

Al implementar un nuevo programa diseñado en arduino con el fin de estabilizar y emparejar las potencias de los motores resulto buena con una velocidad entre el 40% y 70% ya que en ese rango la desviación del prototipo es menor que utilizando el 100% de la potencia en cada uno de los motores, es recomendable usar ese rango de velocidad para fines prácticos o implementar un nuevo sistema donde la llanta delantera y la trasera del mismo lado estén sincronizadas esto puede llegar a realizarse mediante engranes o haciendo uso de llantas tipo oruga.

El uso del MPU6050 para manipulación de giros mediante comparaciones de ángulos es necesario tomar en cuenta un rango de tolerancia o error, con esto evitar el cabeceo al momento de que el robot trate de llegar al ángulo deseado, entre mayor sea la tolerancia menor es el cabeceo. También se recomienda hacer un programa nuevo ya que al aplicarle diferentes pesos la desviación del robot respecto al punto de referencia se vuelve mayor provocándonos un error fatal entre el ángulo medido al ángulo real girado. La desviación se daba por la fuerza ejercida en las llantas traseras, mientras las llantas traseras giraban muy lento las llantas delanteras giraban más rápido por esa razón es conveniente que las llantas delanteras estén sincronizadas con las traseras.

FUENTES DE INFORMACIÓN

arduino. (2018). *arduino .* Obtenido de https://www.arduino.cc/

blogspot. (4 de septiembre de 2012). *robot recolector de objetos.* Obtenido de http://robotrecolectordeobjetoscecyt3.blogspot.com/

datasheet. (s.f.). *datasheet.* Obtenido de https://www.st.com/resource/en/datasheet/l298.pdf

fundacion wikimedia, inc. (28 de diciembre de 2018). *wikipedia.* Obtenido de https://es.wikipedia.org/wiki/Robot

fundacion wikimedia, inc. (6 de diciembre de 2018). *wikipedia.* Obtenido de https://es.wikipedia.org/wiki/Programaci%C3%B3n#Programas_y_algoritmos

fundacion wikimedia, inc. (01 de 03 de 2019). *wikipedia.* Obtenido de https://es.wikipedia.org/wiki/Sensor

fundacion wikimedia, inc. (1 de enero de 2019). *wikipedia.* Obtenido de https://es.wikipedia.org/wiki/Arduino

Inc., I. (9 de agosto de 2013). *InvenSense.* Obtenido de https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

INTPLUS. (3 de enero de 2019). *Super Robotica.* Obtenido de http://www.superrobotica.com/S320111.htm

llamas, l. (16 de octubre de 2016). *ingenieria, informatica y diseño.* Obtenido de https://www.luisllamas.es/tarjeta-micro-sd-arduino/

mendez, e. (agosto de 2018). *mecatronica latam.* Obtenido de https://www.mecatronicalatam.com/es/tarjeta/arduino/que-es

naylamp. (2016). *naylamp mechatronics.* Obtenido de https://naylampmechatronics.com/blog/45_Tutorial-MPU6050-Acelerómetro-y-Giroscopio.html

naylamp mechatronics AC. (2016). */naylampmechatronics.* Obtenido de https://naylampmechatronics.com/blog/11_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html

Naylamp Mechatronics SAC. (2018). *Naylamp Mechatronics.* Obtenido de

https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html

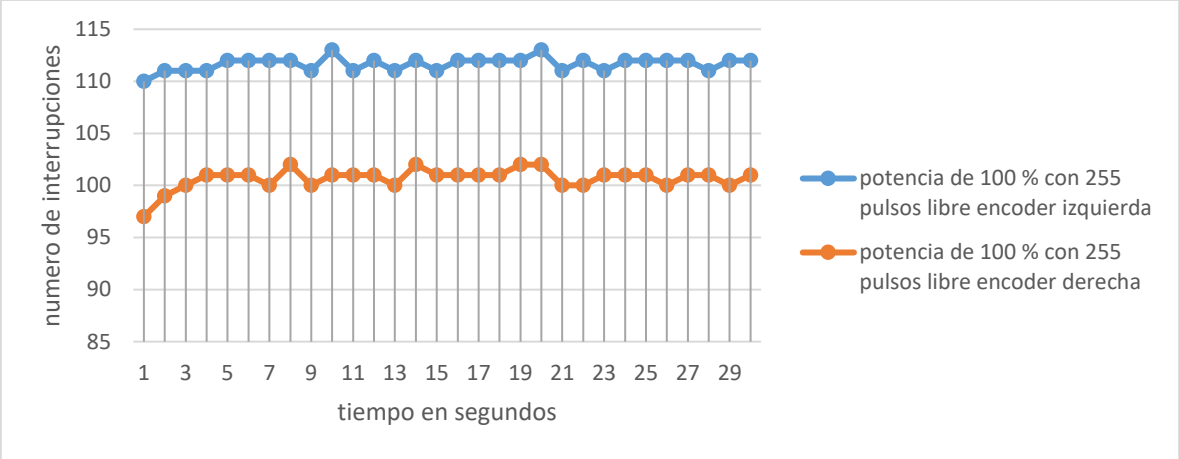
Valis, D. (11 de mayo de 2016). *conacyt.* Obtenido de http://www.conacytprensa.mx/index.php/tecnologia/robotica/7333-crean-robot-recolector-de-desechos

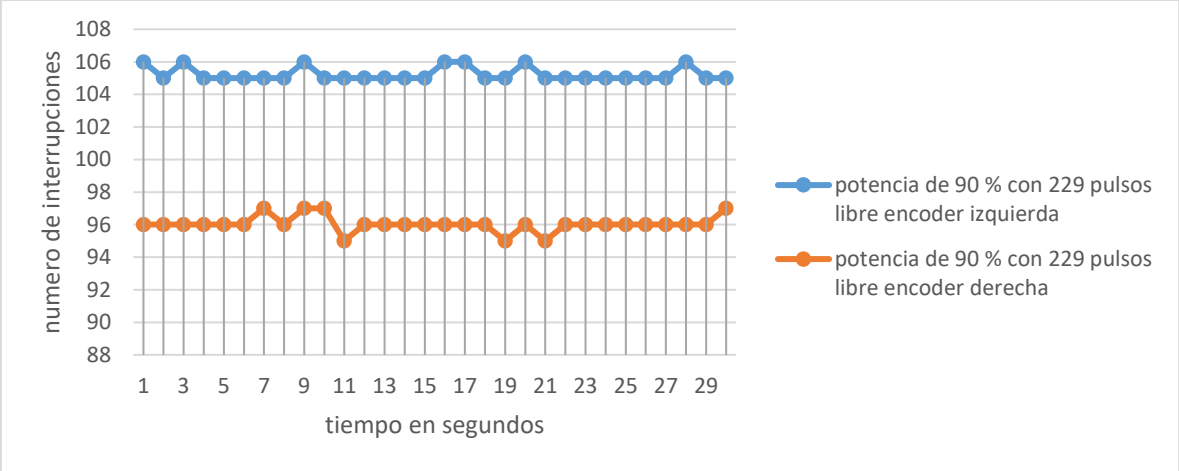
vexrobotics. (s.f.). *VEX EDR.* Obtenido de http://www.vexrobotics.com.mx/vexedr/descripcion.php?id=276-2156

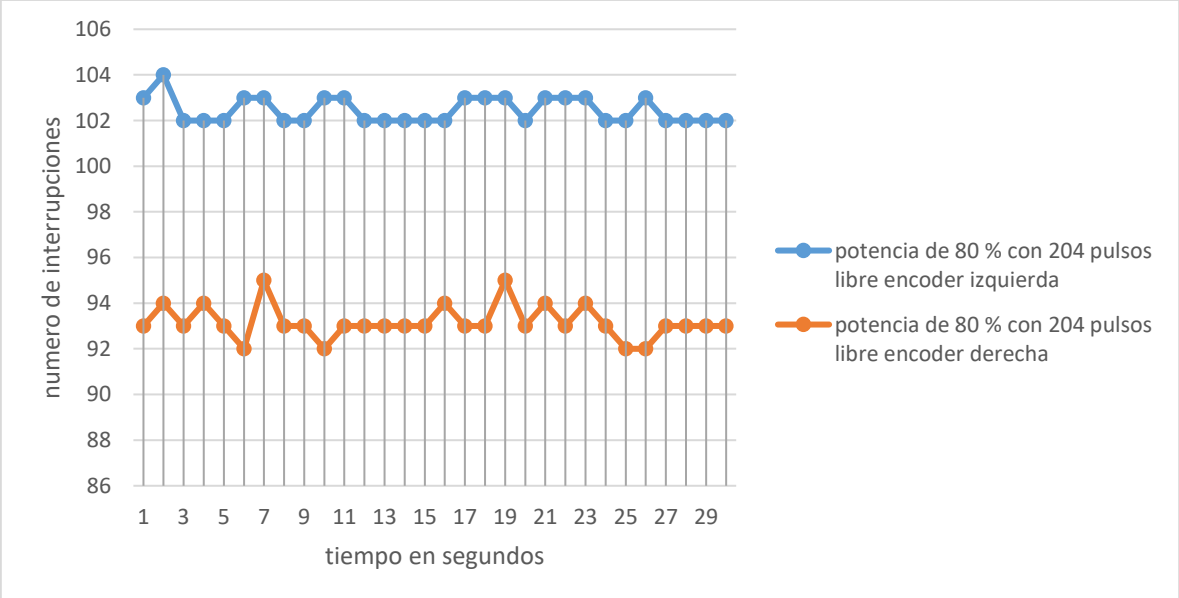
Welle, D. (16 de septiembre de 2010). *dw.* Obtenido de https://www.dw.com/es/el-robot-que-recoge-la-basura/a-6003437

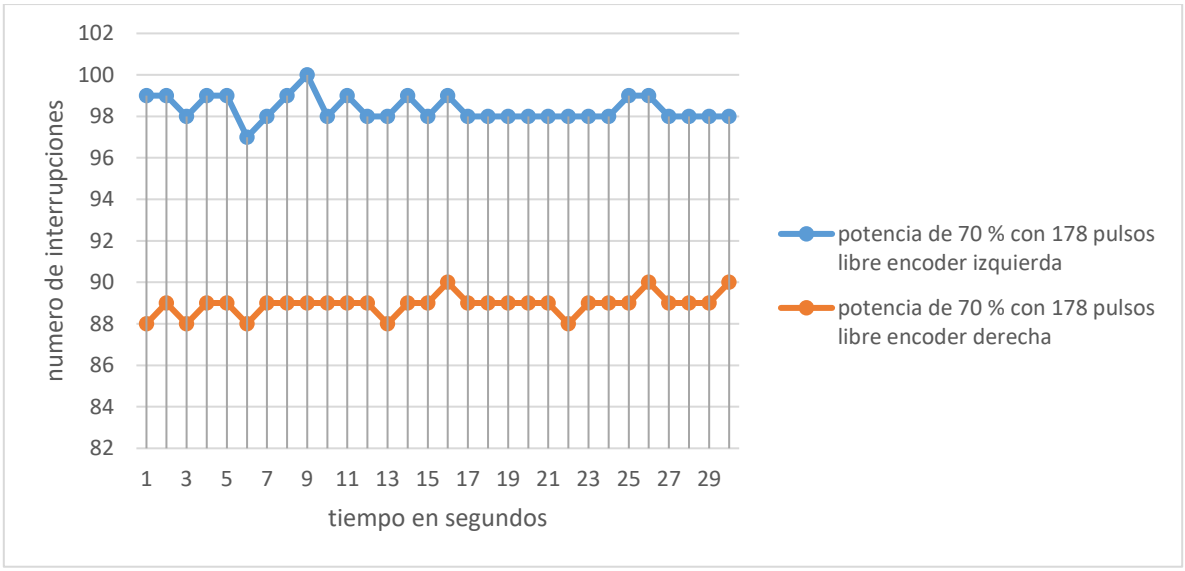
Anexos

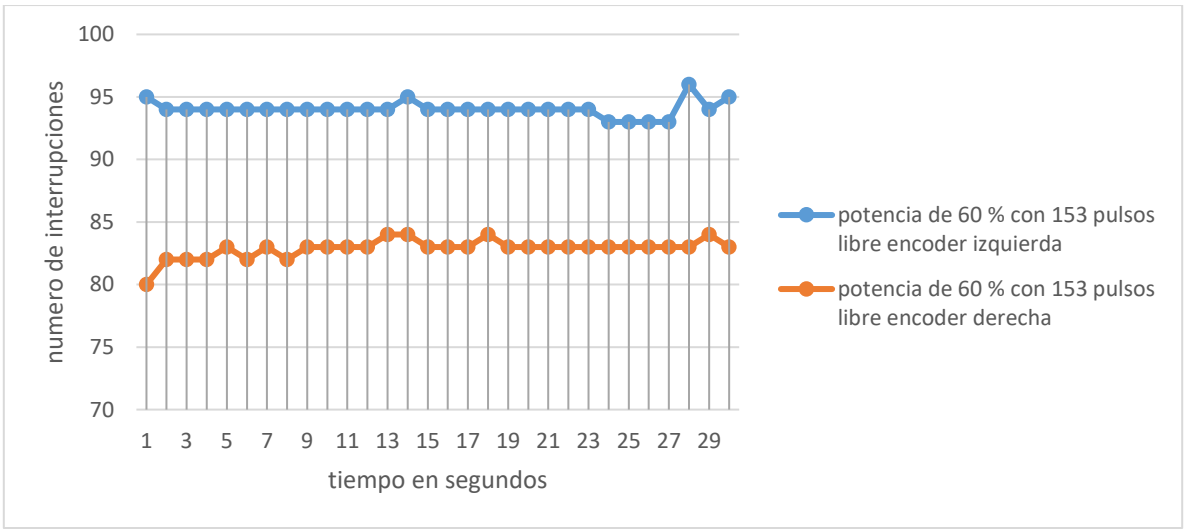
Pruebas de velocidad usando Llantas de tracción, sin fricción

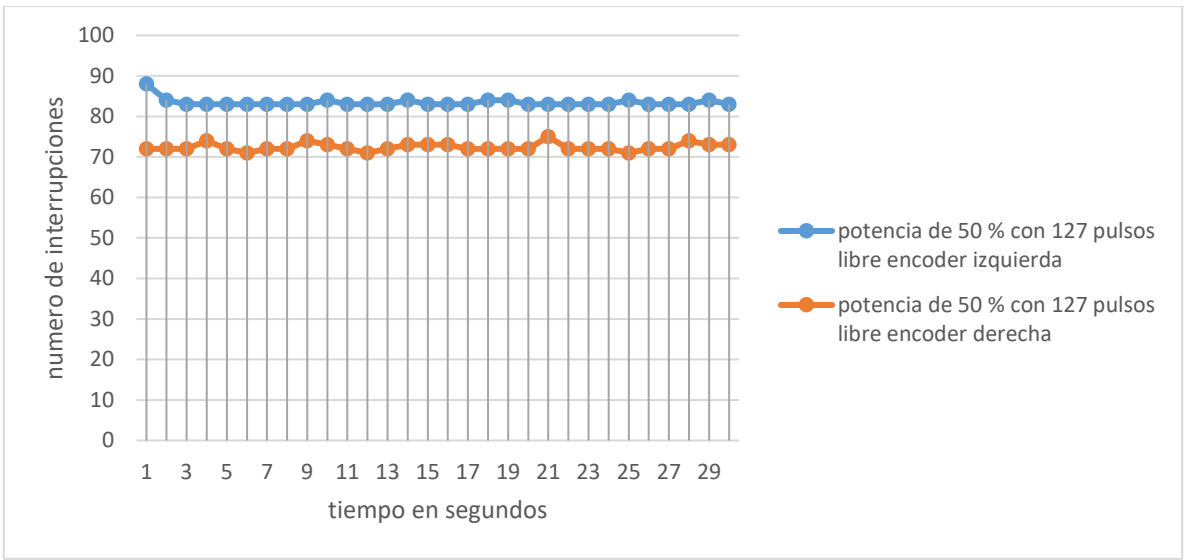


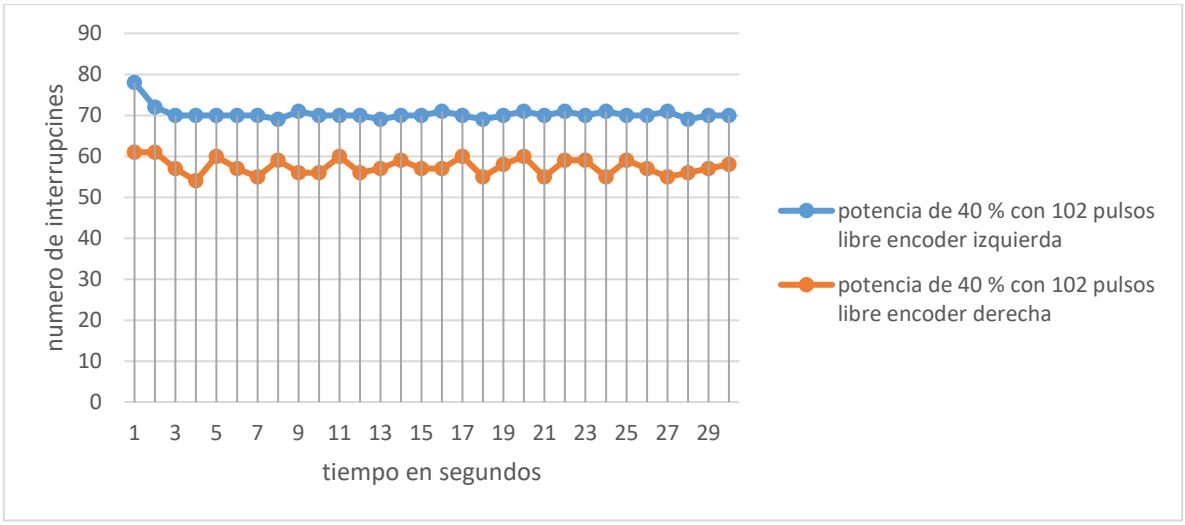


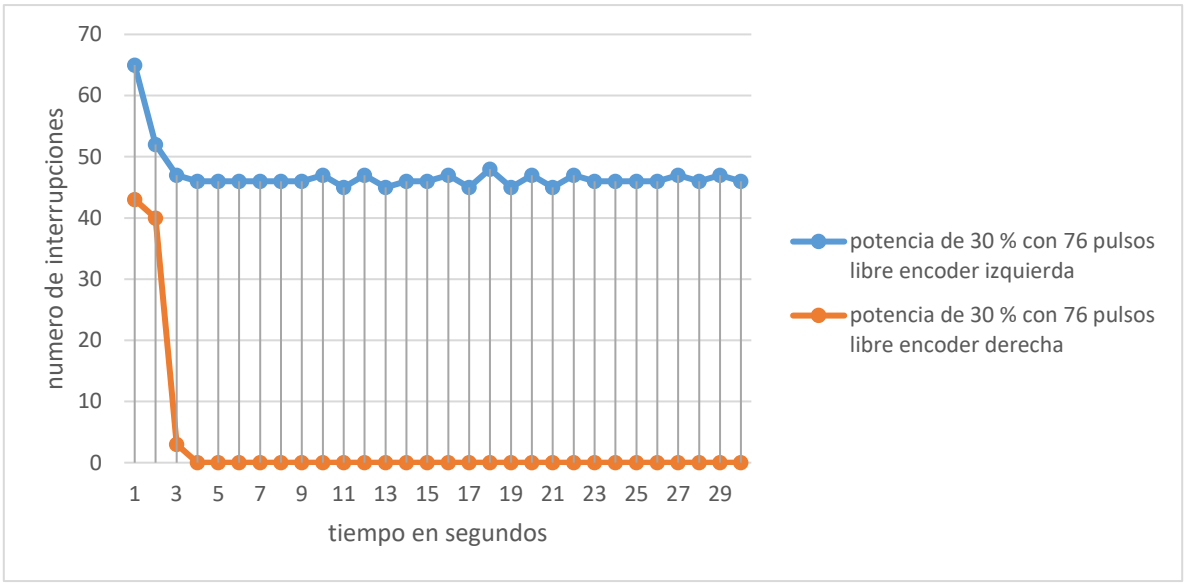


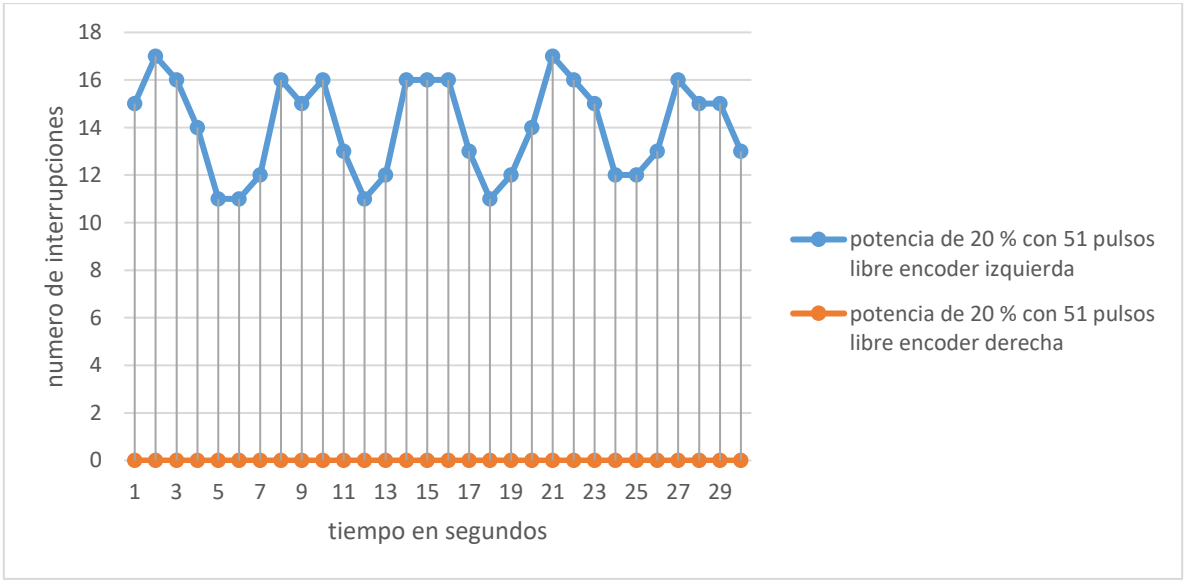

numero de interrupciones
102
100
98
96
94
92
90
88
86
84
82
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 70 % con 178 pulsos libre encoder izquierda
potencia de 70 % con 178 pulsos libre encoder derecha


numero de interrupciones
100
95
90
85
80
75
70
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 60 % con 153 pulsos libre encoder izquierda
potencia de 60 % con 153 pulsos libre encoder derecha

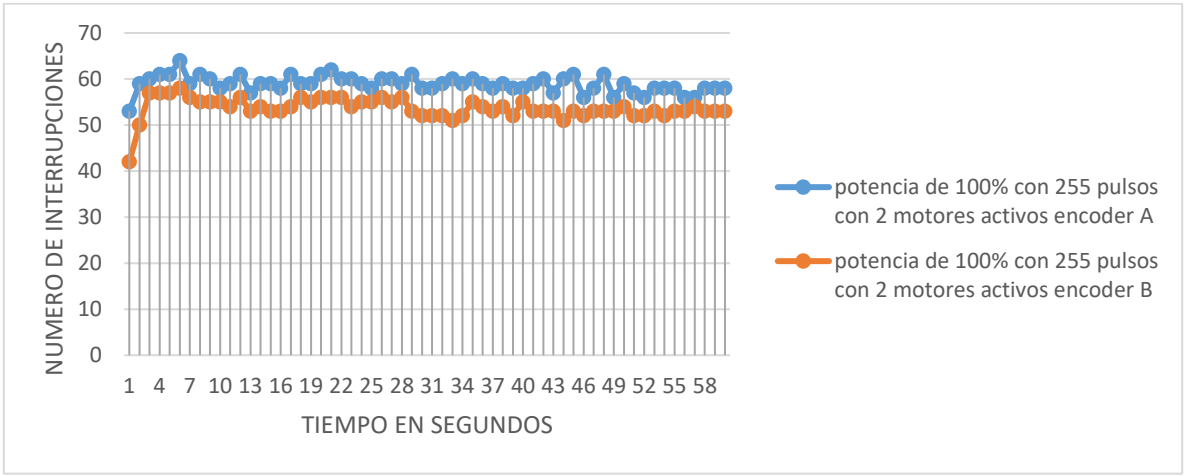

numero de interrupciones
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 50 % con 127 pulsos libre encoder izquierda
potencia de 50 % con 127 pulsos libre encoder derecha

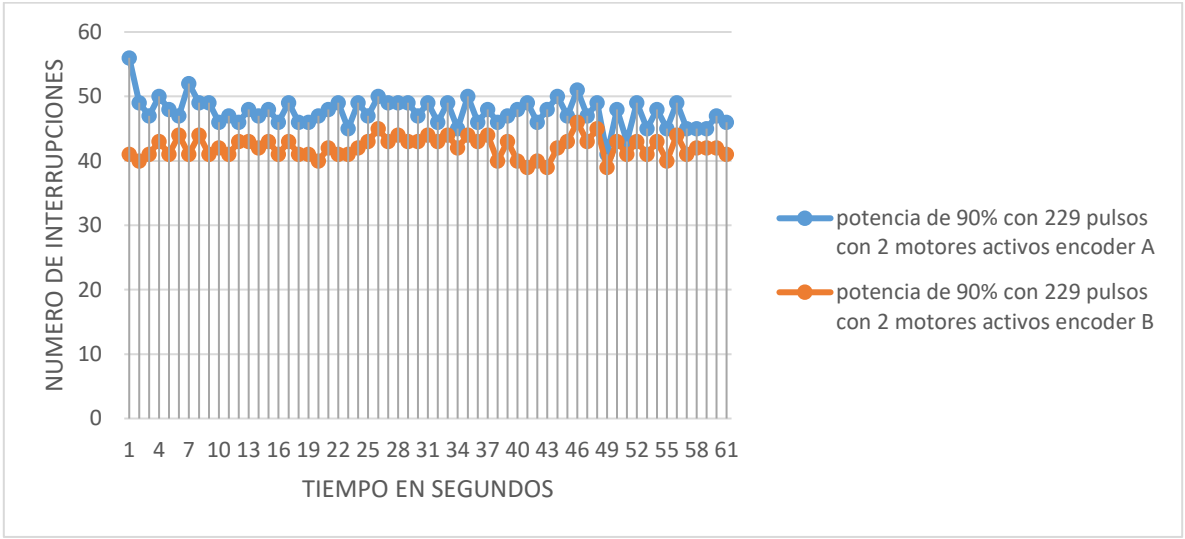

numero de interrupcines
tiempo en segundos
potencia de 40 % con 102 pulsos libre encoder izquierda
potencia de 40 % con 102 pulsos libre encoder derecha

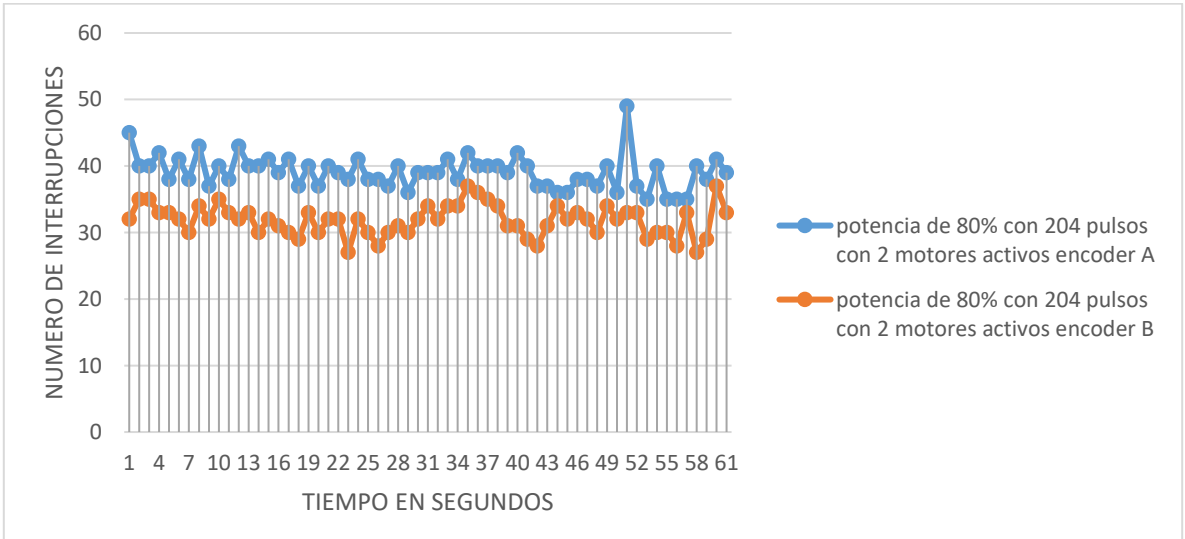

numero de interrupciones
tiempo en segundos
potencia de 30 % con 76 pulsos libre encoder izquierda
potencia de 30 % con 76 pulsos libre encoder derecha


numero de interrupciones
tiempo en segundos
potencia de 20 % con 51 pulsos libre encoder izquierda
potencia de 20 % con 51 pulsos libre encoder derecha

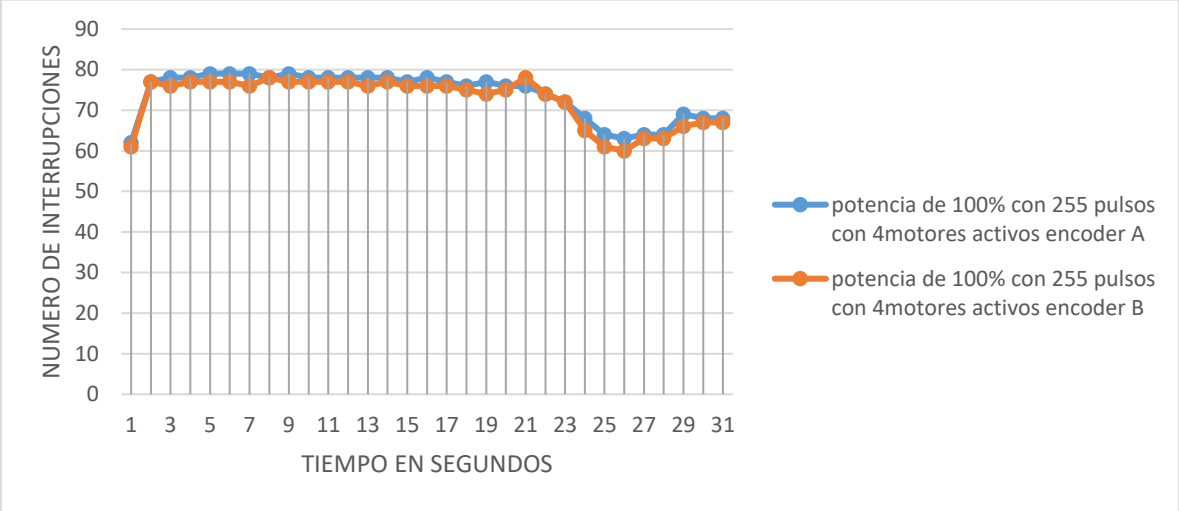
Pruebas de velocidad usando Llantas de tracción, con dos motores sin carga

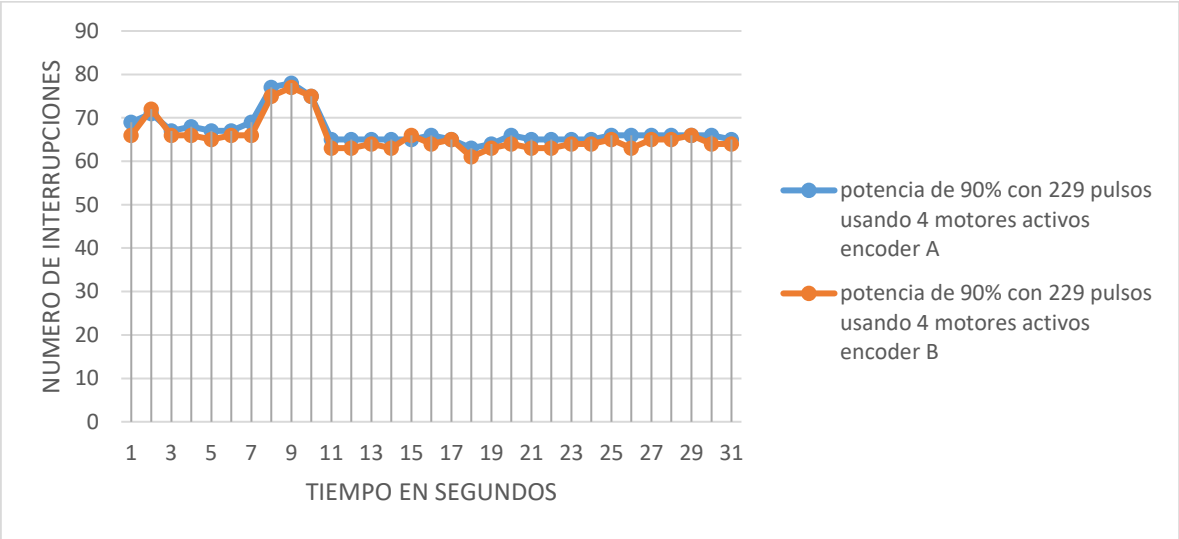


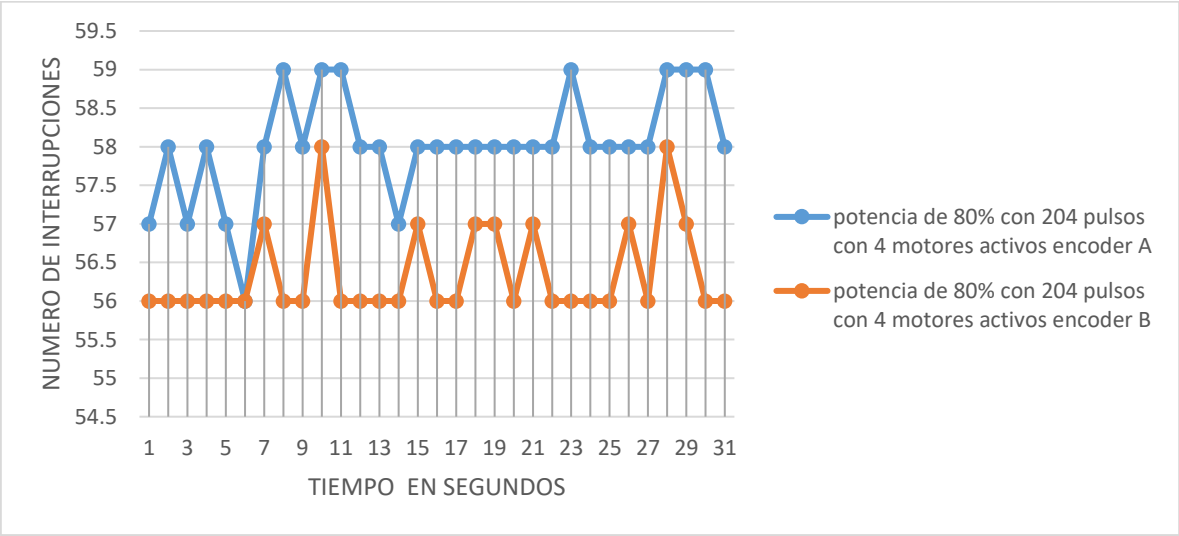


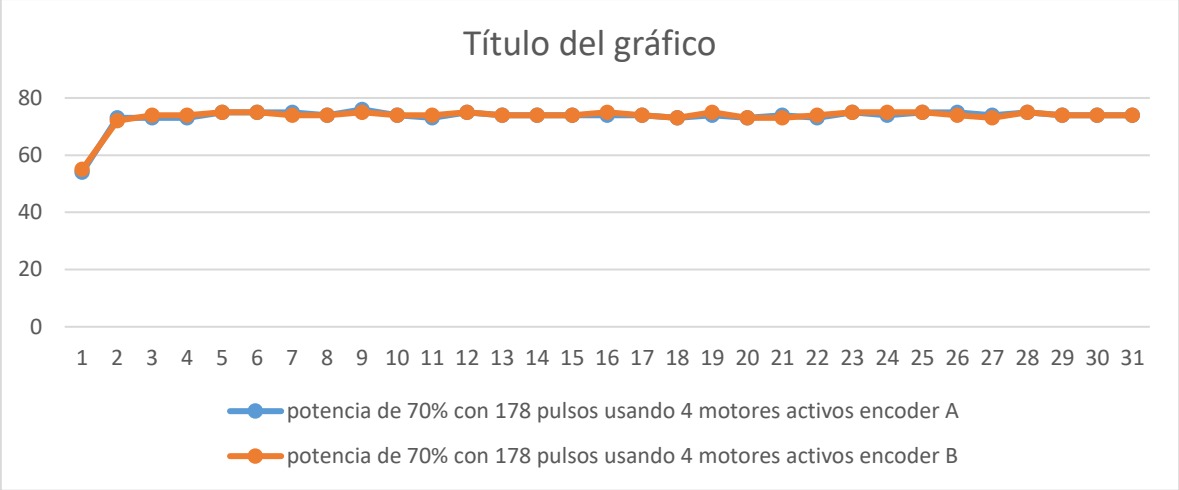


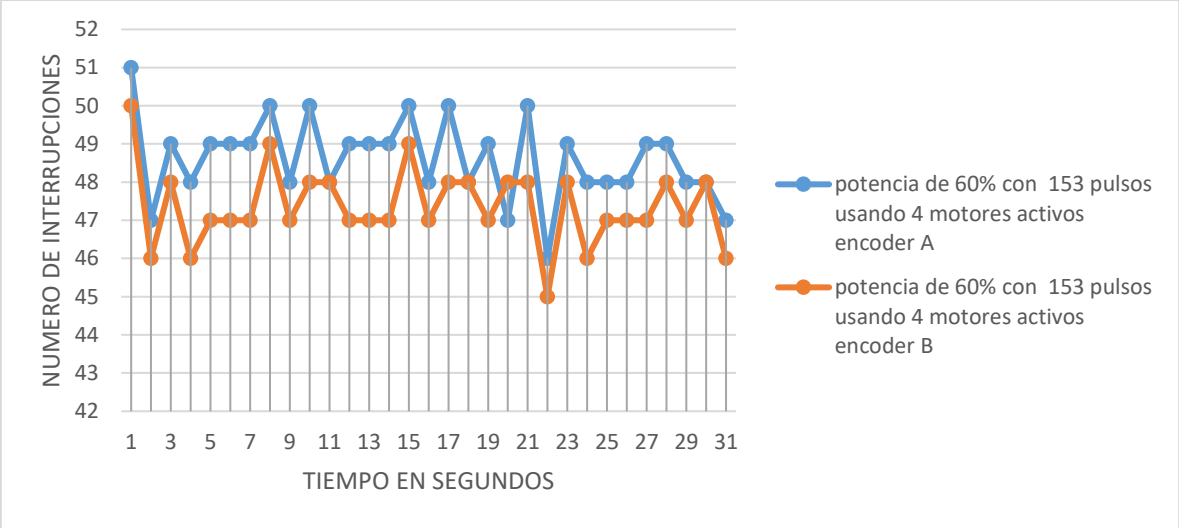
Pruebas de velocidad usando Llantas de tracción, con cuatro motores sin carga

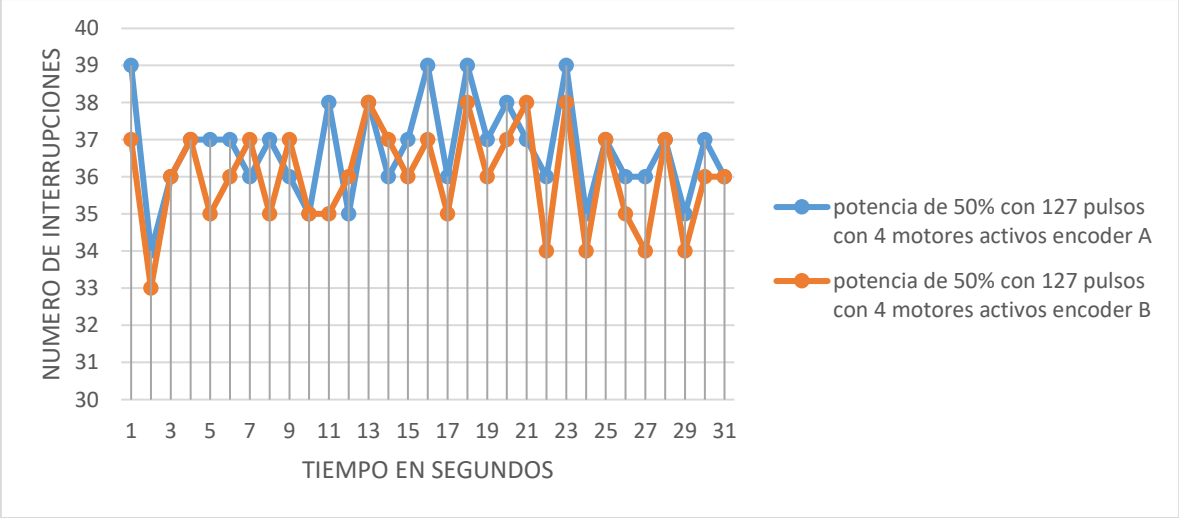


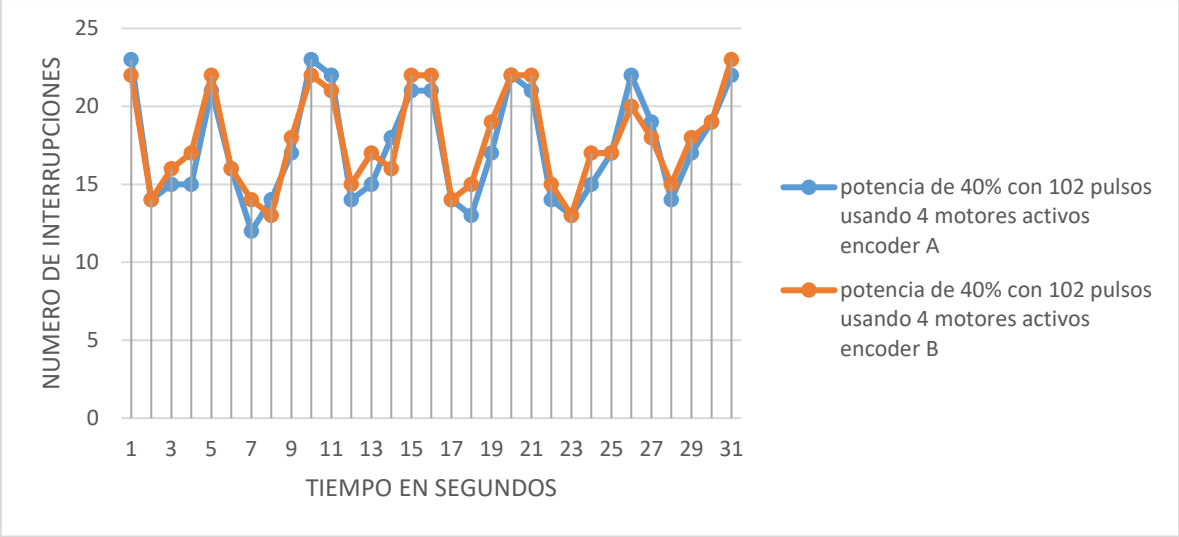




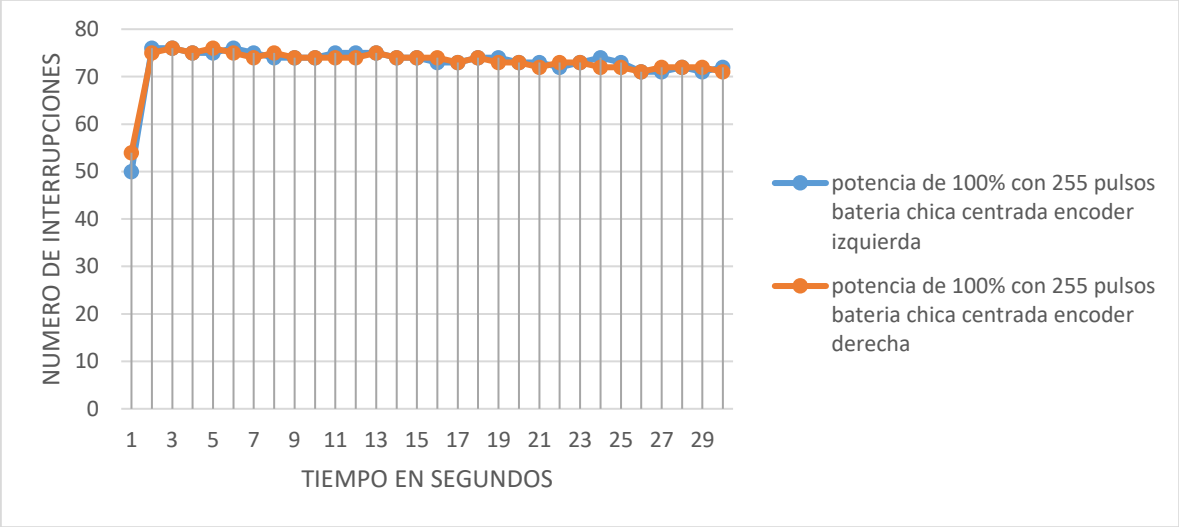

Título del gráfico
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
potencia de 70% con 178 pulsos usando 4 motores activos encoder A
potencia de 70% con 178 pulsos usando 4 motores activos encoder B

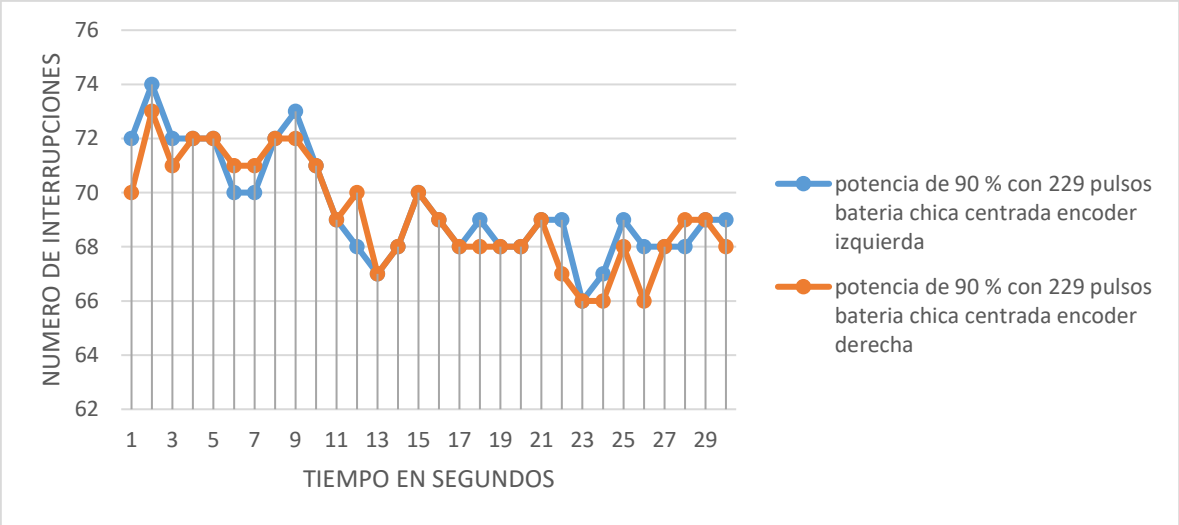

NUMERO DE INTERRUPCIONES
52
51
50
49
48
47
46
45
44
43
42
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos usando 4 motores activos encoder A
potencia de 60% con 153 pulsos usando 4 motores activos encoder B

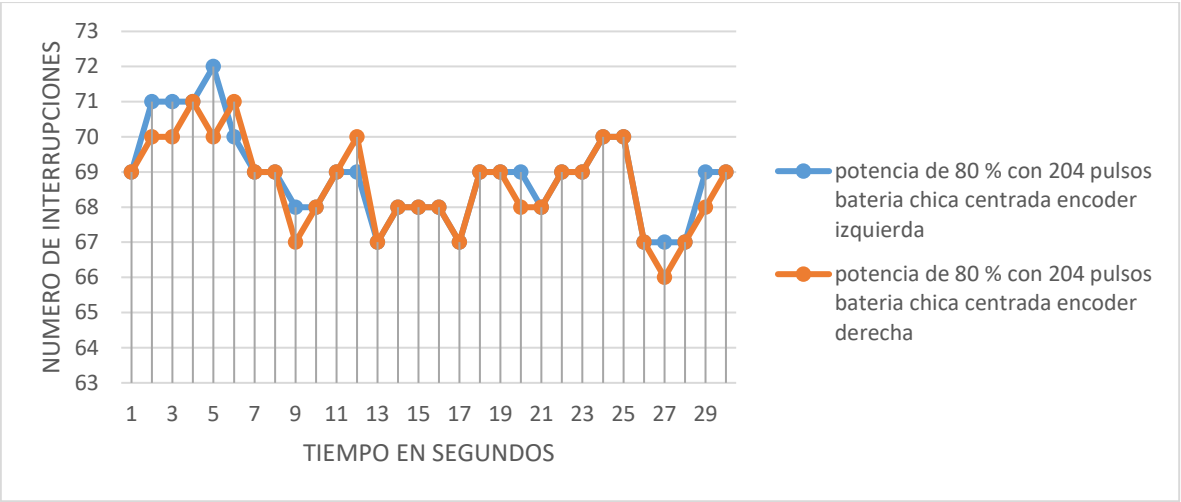

NUMERO DE INTERRUPCIONES
40
39
38
37
36
35
34
33
32
31
30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos con 4 motores activos encoder A
potencia de 50% con 127 pulsos con 4 motores activos encoder B

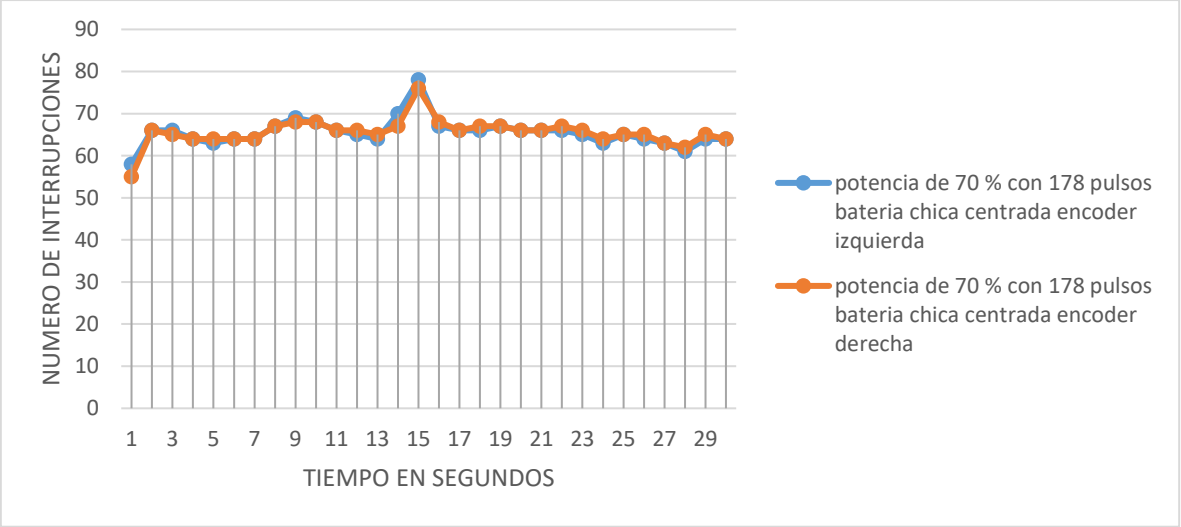


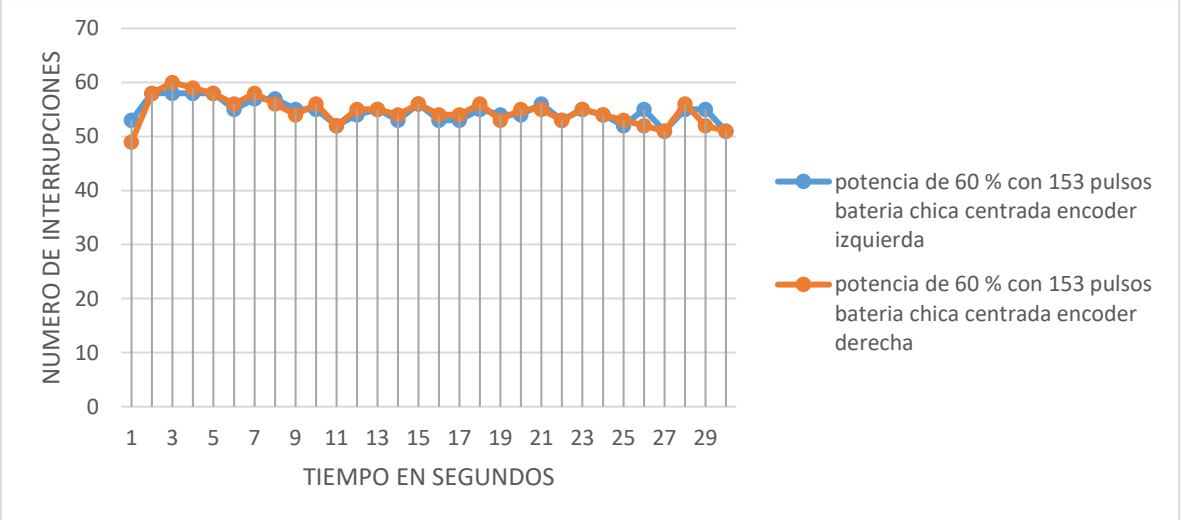
Pruebas de velocidad usando Llantas de tracción, con carga de 1.7k centrada

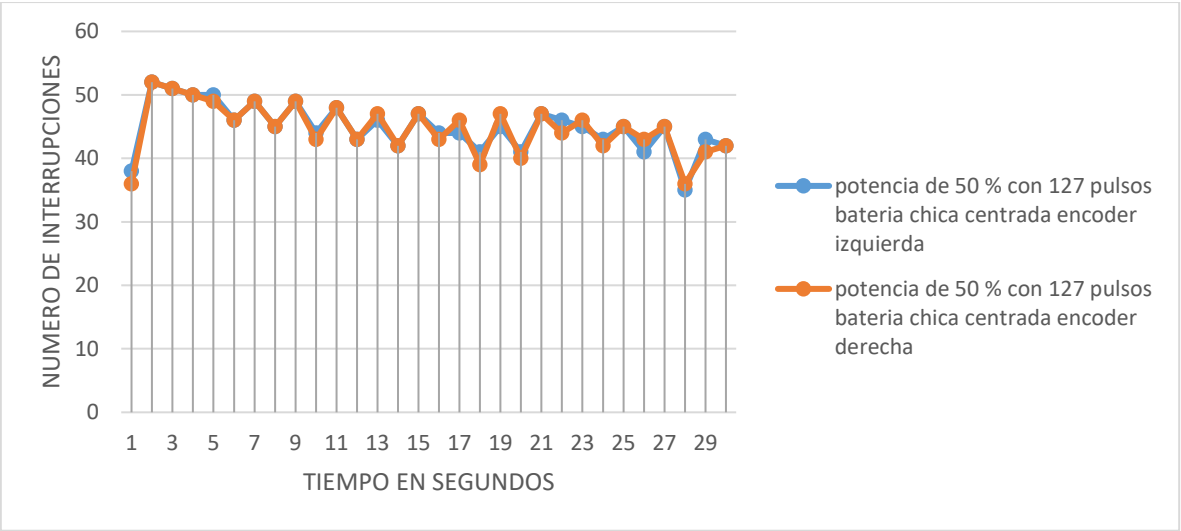


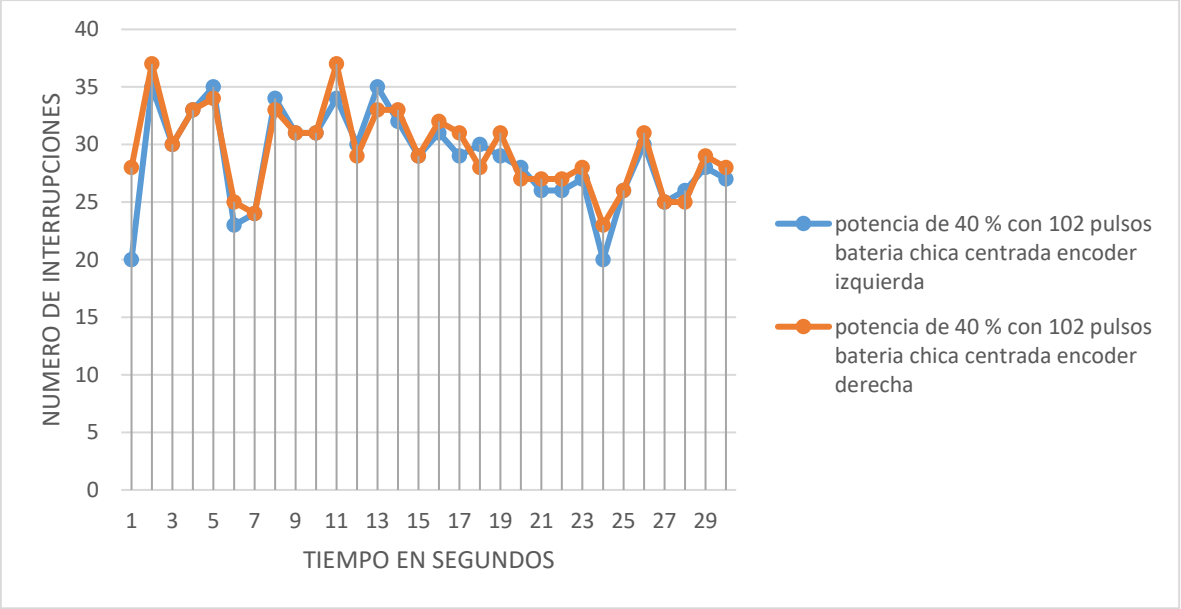


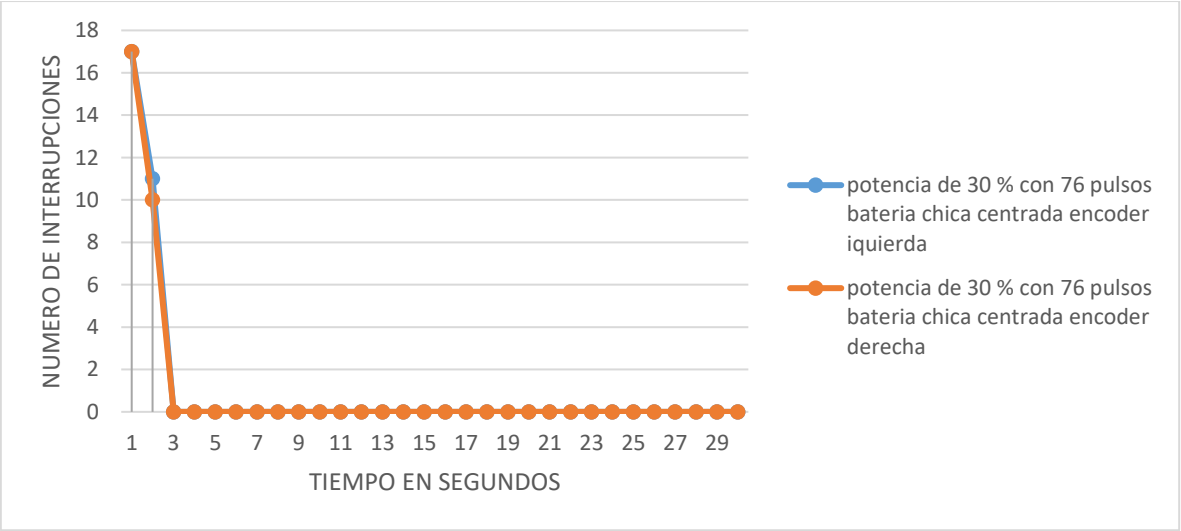

NUMERO DE INTERRUPCIONES
73
72
71
70
69
68
67
66
65
64
63
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria chica centrada encoder izquierda
potencia de 80 % con 204 pulsos bateria chica centrada encoder derecha


NUMERO DE INTERRUPCIONES
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica centrada encoder izquierda
potencia de 70 % con 178 pulsos bateria chica centrada encoder derecha

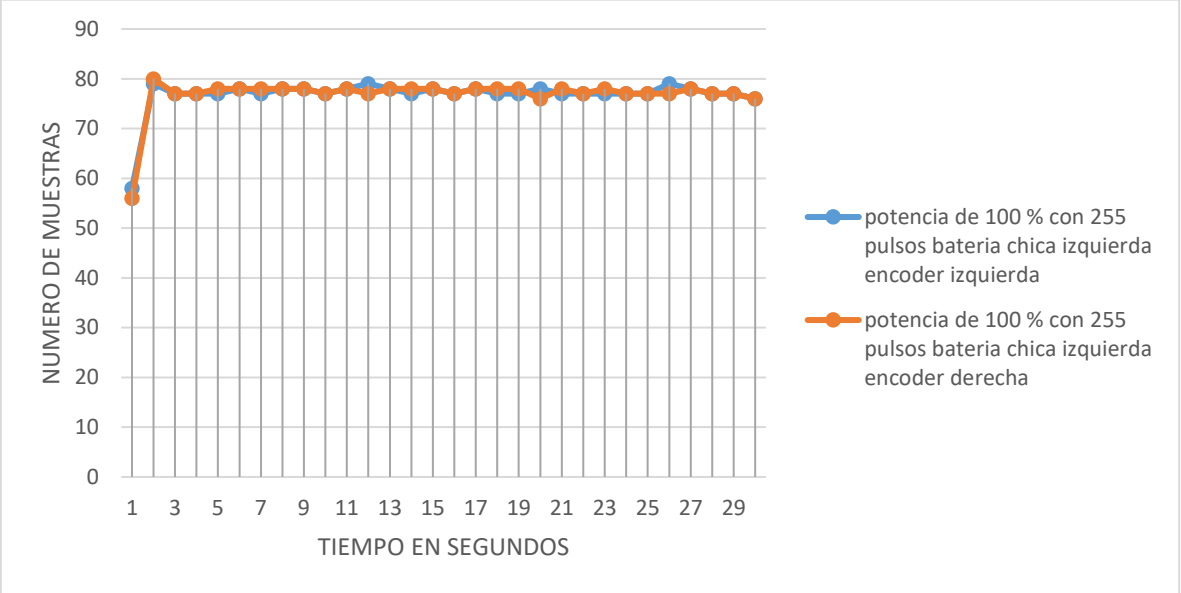

NUMERO DE INTERRUPCIONES
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica centrada encoder izquierda
potencia de 60 % con 153 pulsos bateria chica centrada encoder derecha

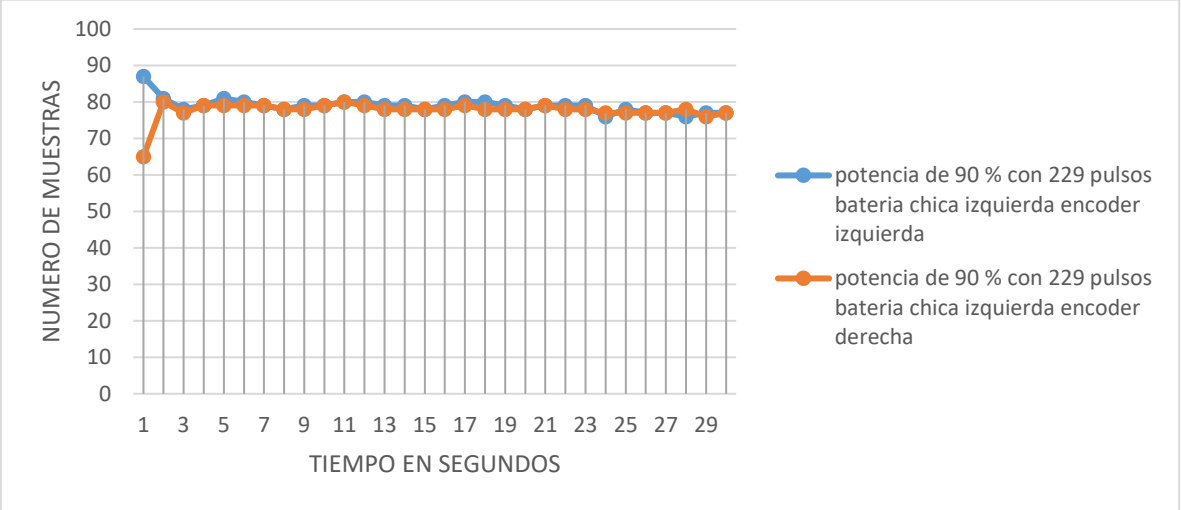

NUMERO DE INTERRUPCIONES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica centrada encoder izquierda
potencia de 50 % con 127 pulsos bateria chica centrada encoder derecha

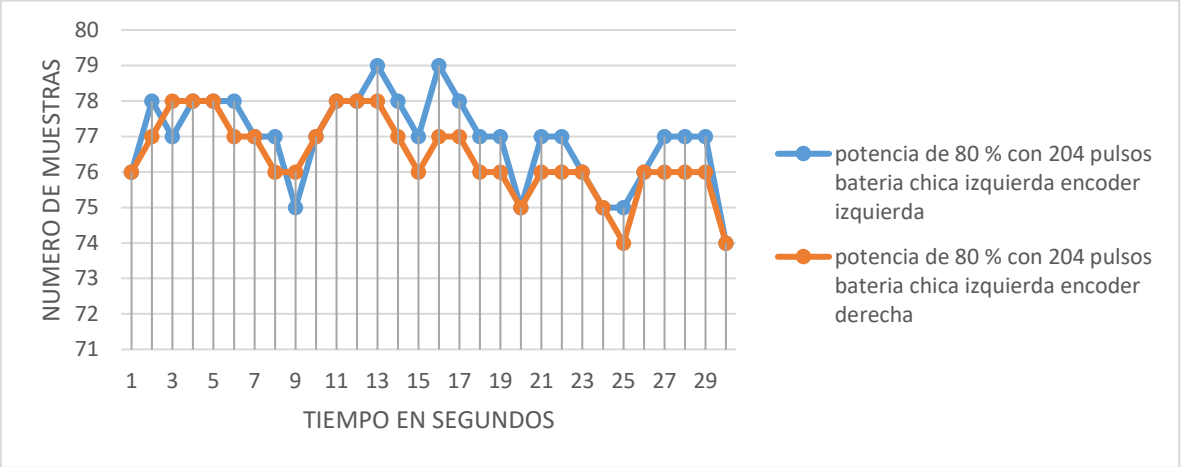

NUMERO DE INTERRUPCIONES
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria chica centrada encoder izquierda
potencia de 40 % con 102 pulsos bateria chica centrada encoder derecha

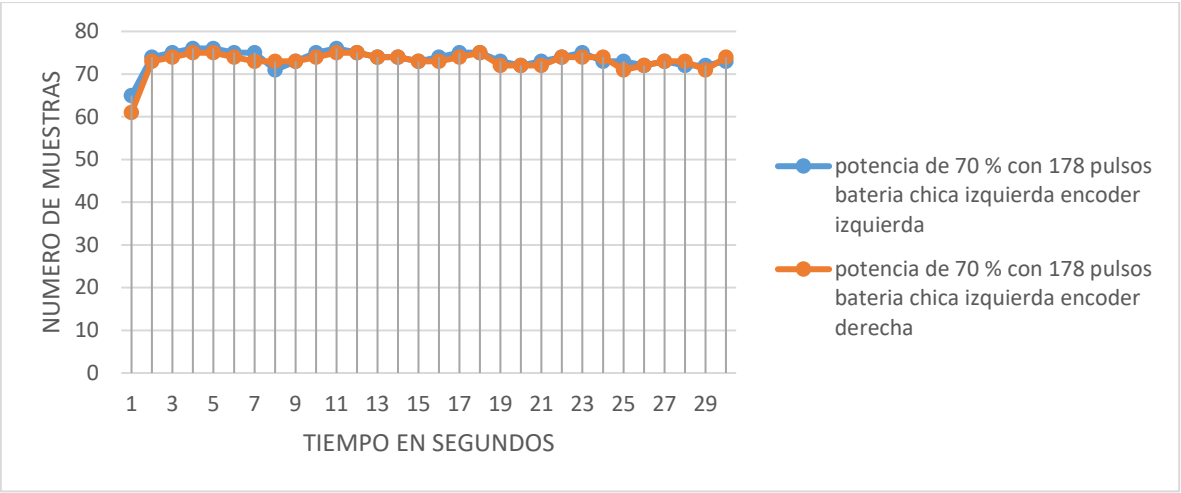

NUMERO DE INTERRUPCIONES
18
16
14
12
10
8
6
4
2
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30 % con 76 pulsos bateria chica centrada encoder iquierda
potencia de 30 % con 76 pulsos bateria chica centrada encoder derecha

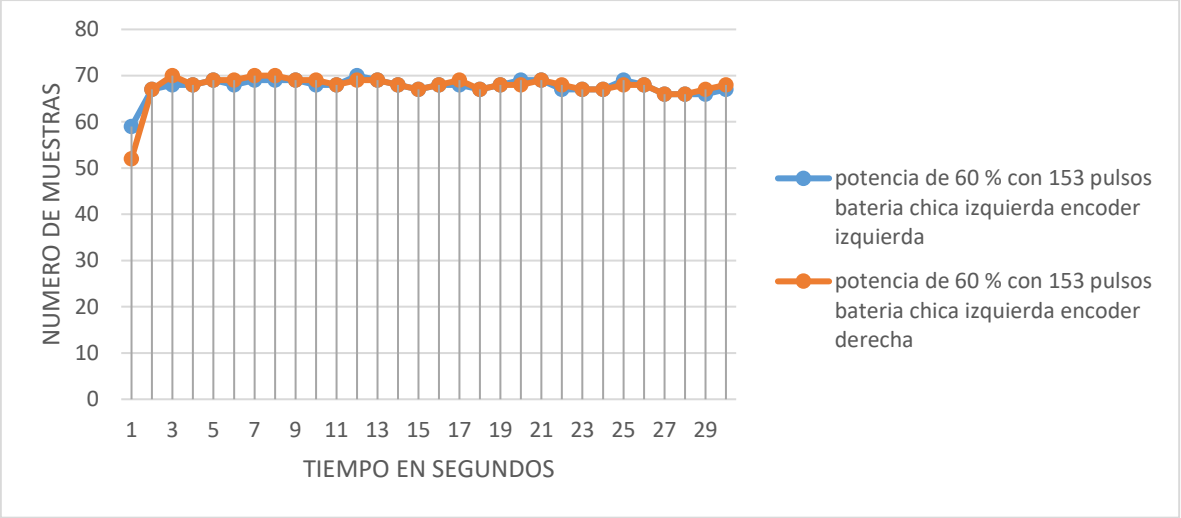
Pruebas de velocidad usando Llantas de tracción, con carga de 1.7k lado izquierdo

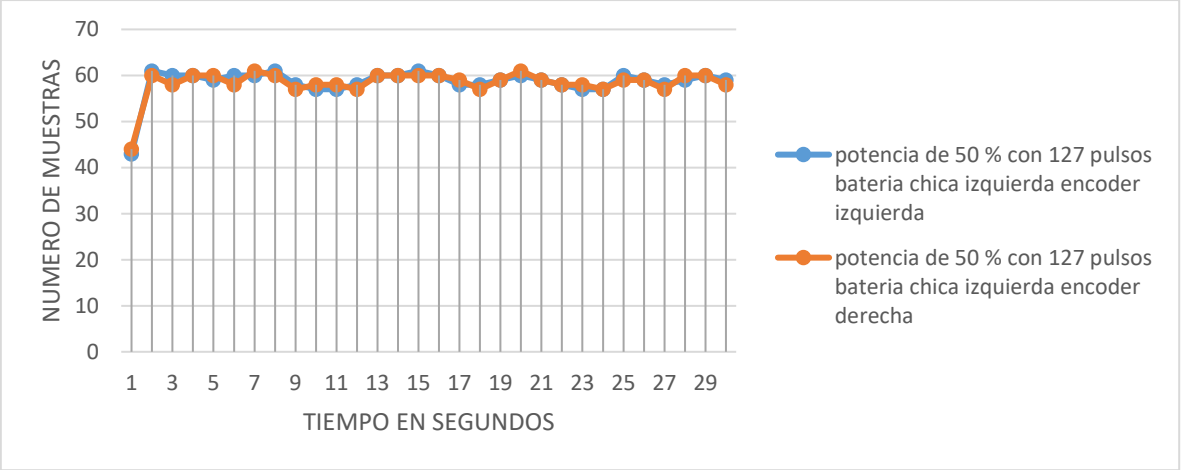


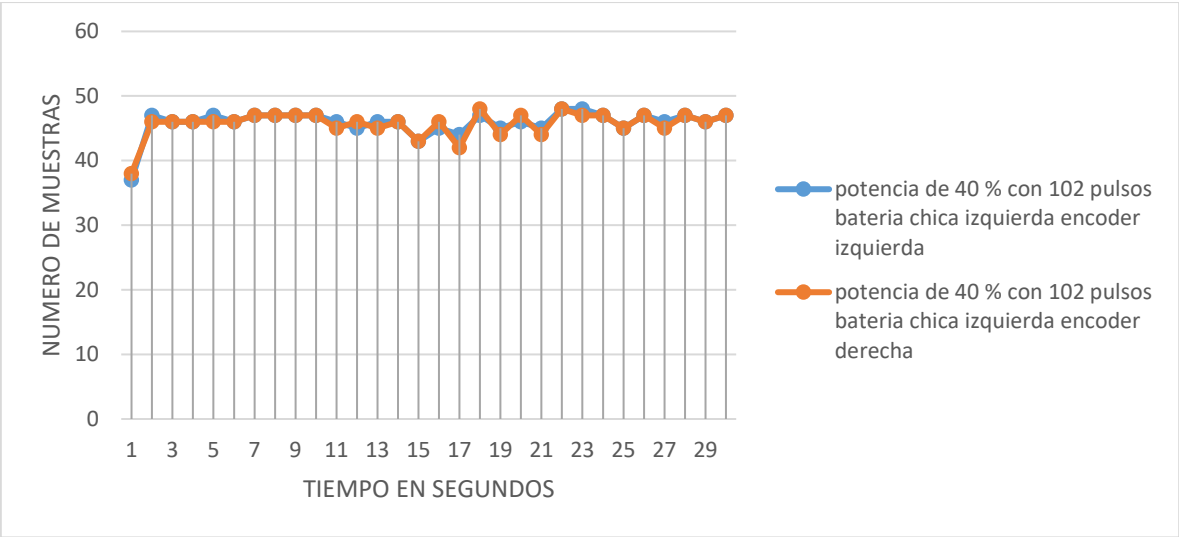


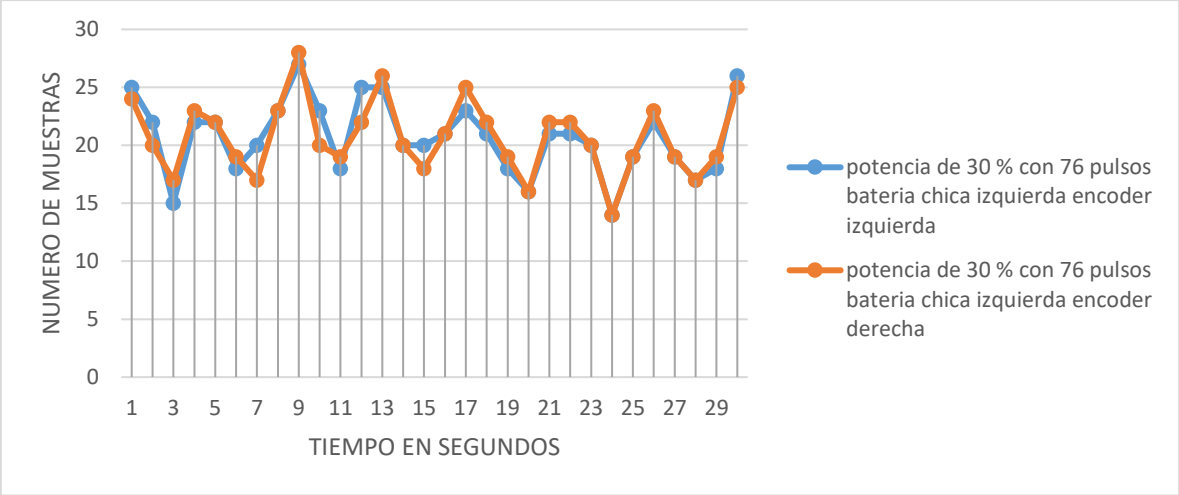


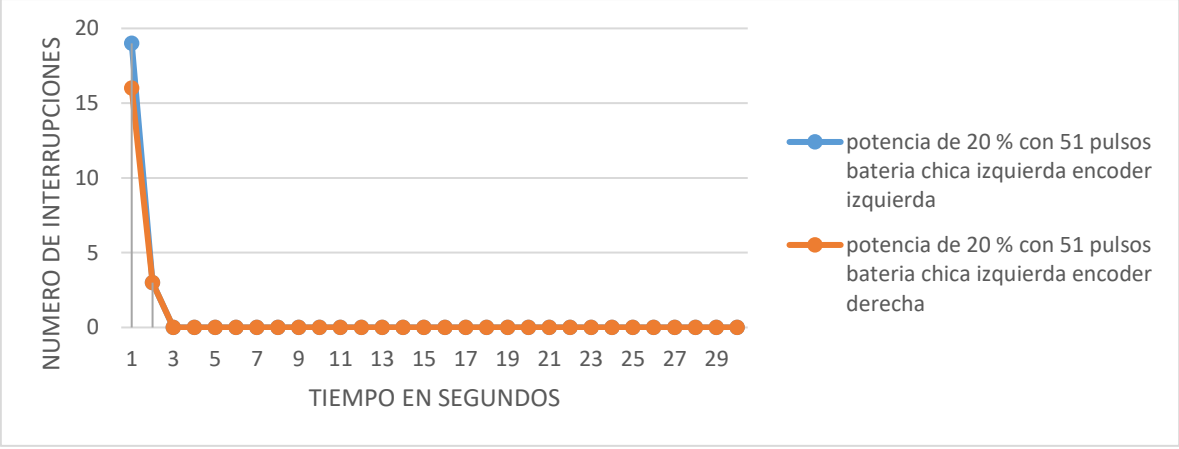

NUMERO DE MUESTRAS
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica izquierda encoder izquierda
potencia de 70 % con 178 pulsos bateria chica izquierda encoder derecha


NUMERO DE MUESTRAS
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica izquierda encoder izquierda
potencia de 60 % con 153 pulsos bateria chica izquierda encoder derecha

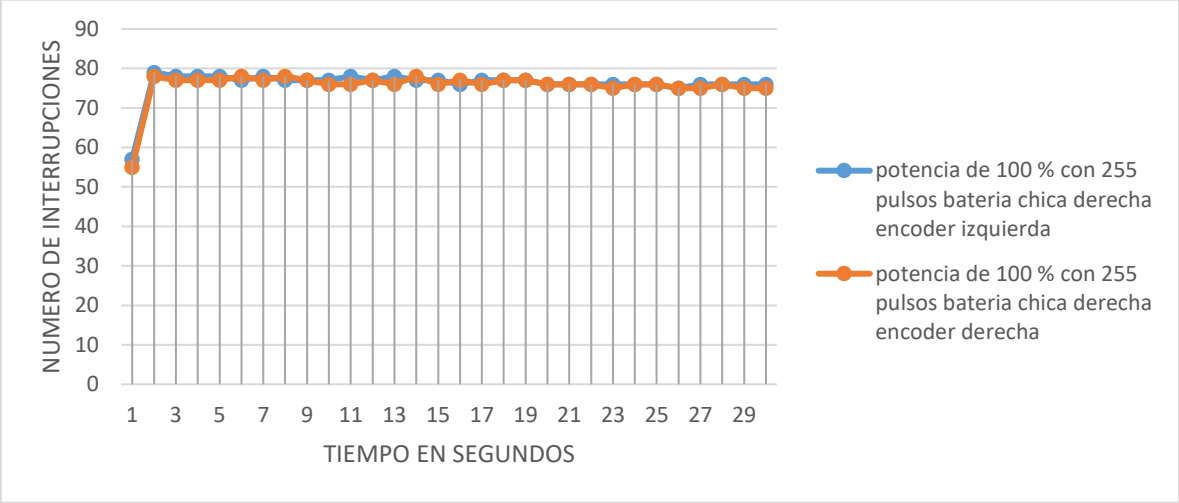

NUMERO DE MUESTRAS
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica izquierda encoder izquierda
potencia de 50 % con 127 pulsos bateria chica izquierda encoder derecha

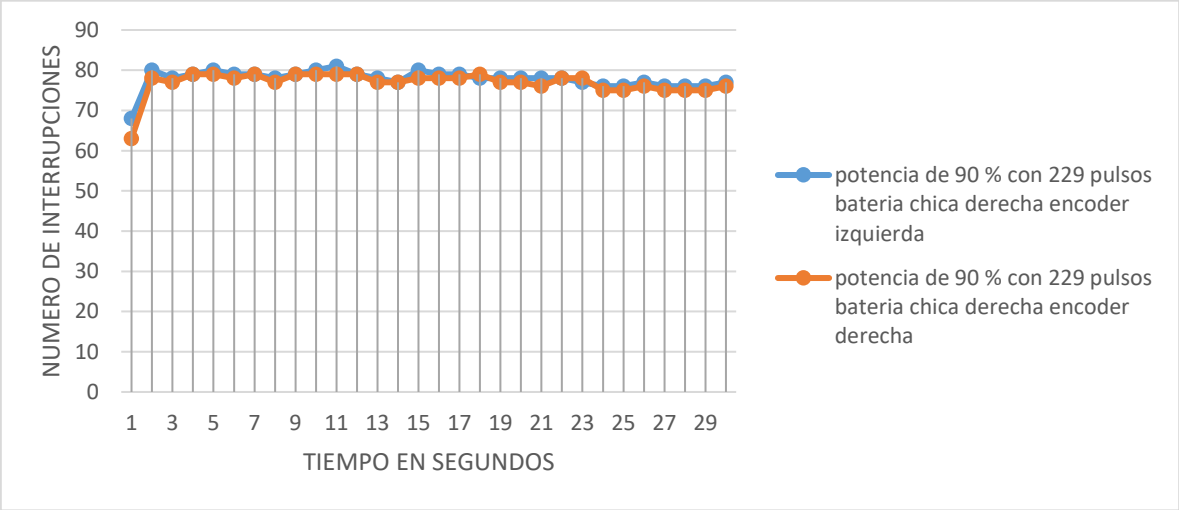

60
50
40
30
20
10
0
NUMERO DE MUESTRAS
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria chica izquierda encoder izquierda
potencia de 40 % con 102 pulsos bateria chica izquierda encoder derecha

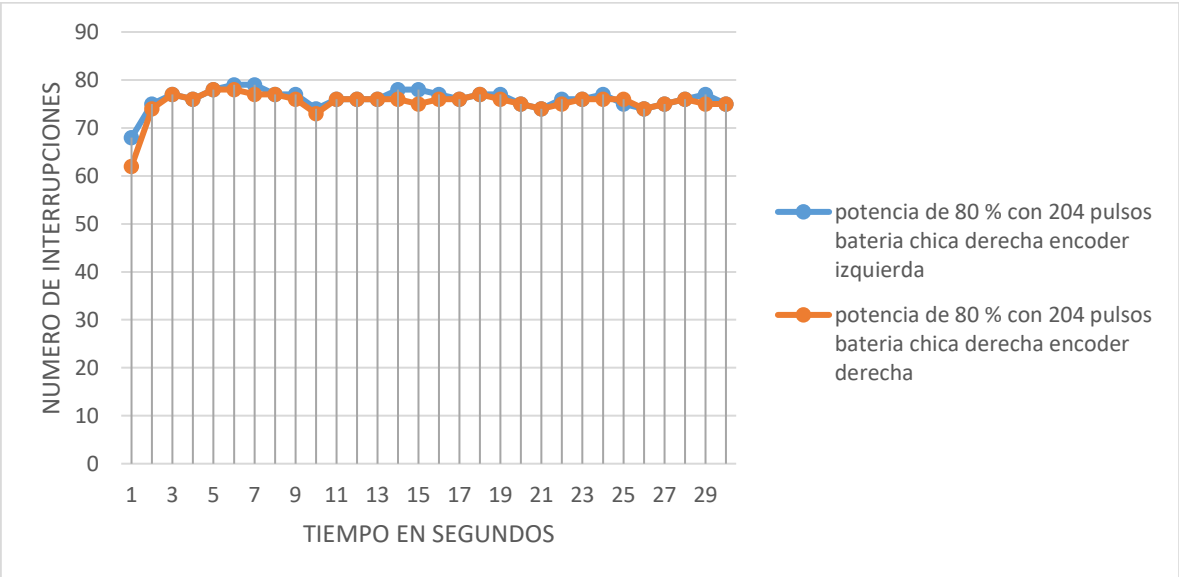

30
25
20
15
10
5
0
NUMERO DE MUESTRAS
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30 % con 76 pulsos bateria chica izquierda encoder izquierda
potencia de 30 % con 76 pulsos bateria chica izquierda encoder derecha

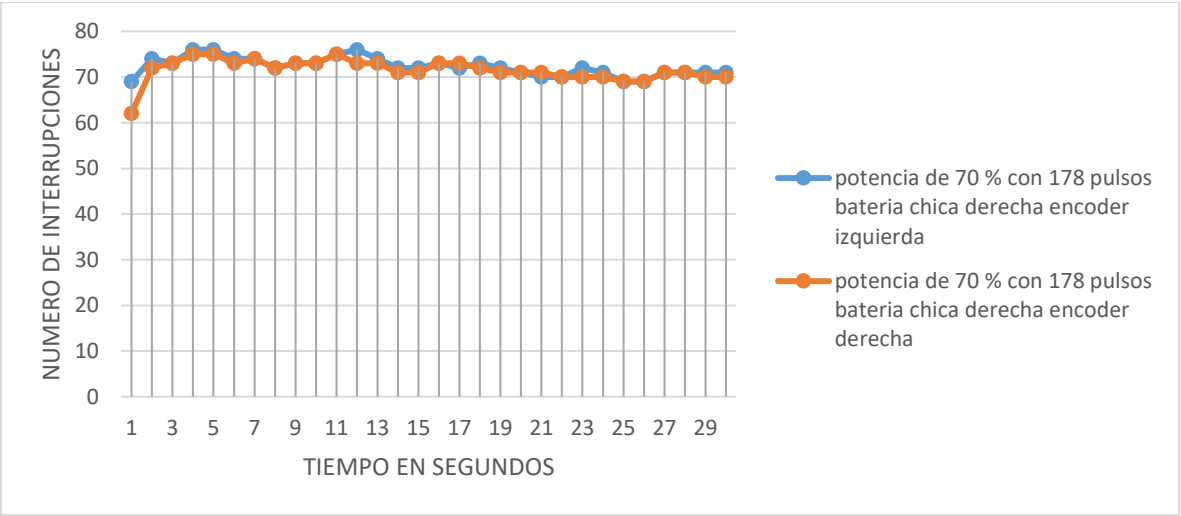

20
15
10
5
0
NUMERO DE INTERRUPCIONES
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 20 % con 51 pulsos bateria chica izquierda encoder izquierda
potencia de 20 % con 51 pulsos bateria chica izquierda encoder derecha

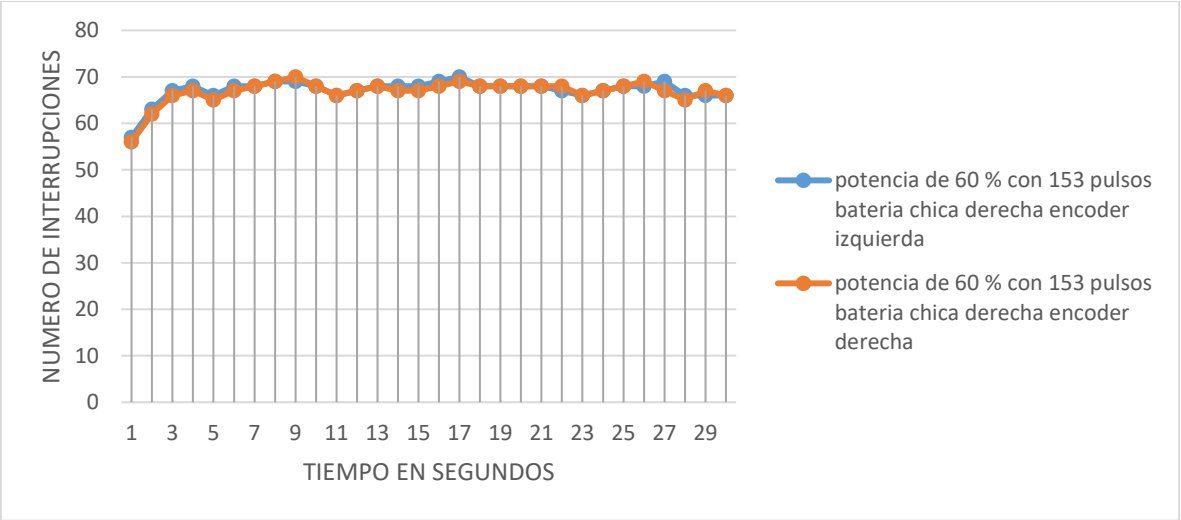
Pruebas de velocidad usando Llantas de tracción, con carga de 1.7k lado derecho

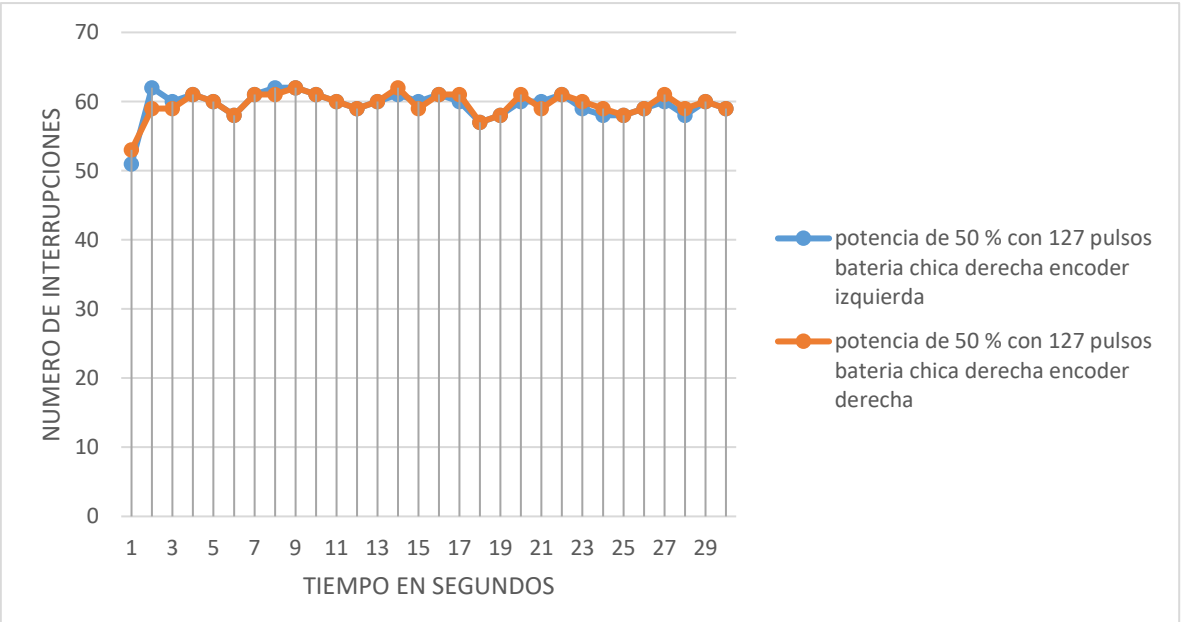


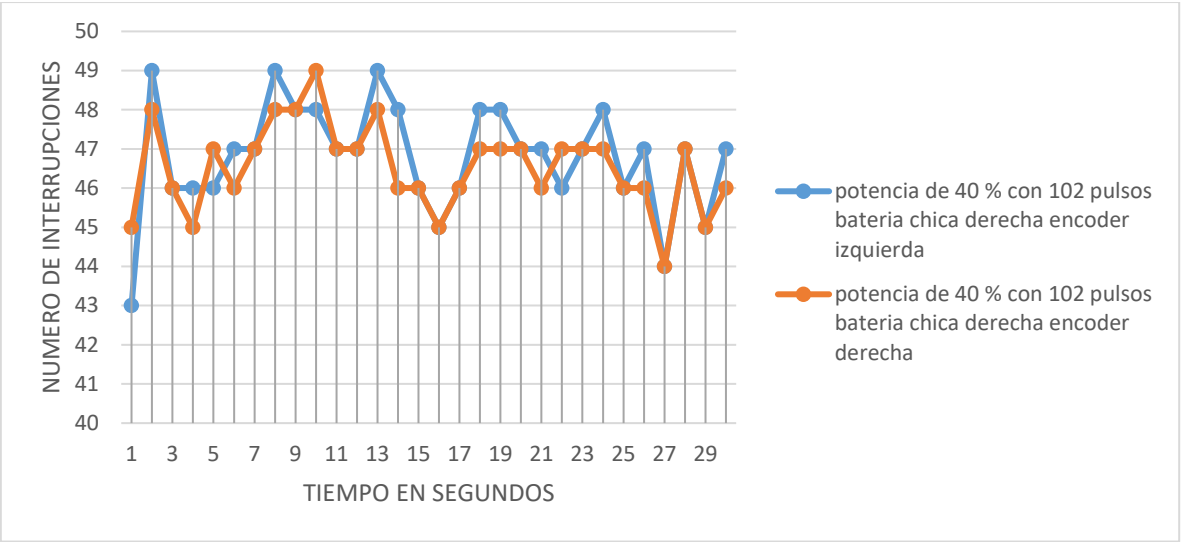


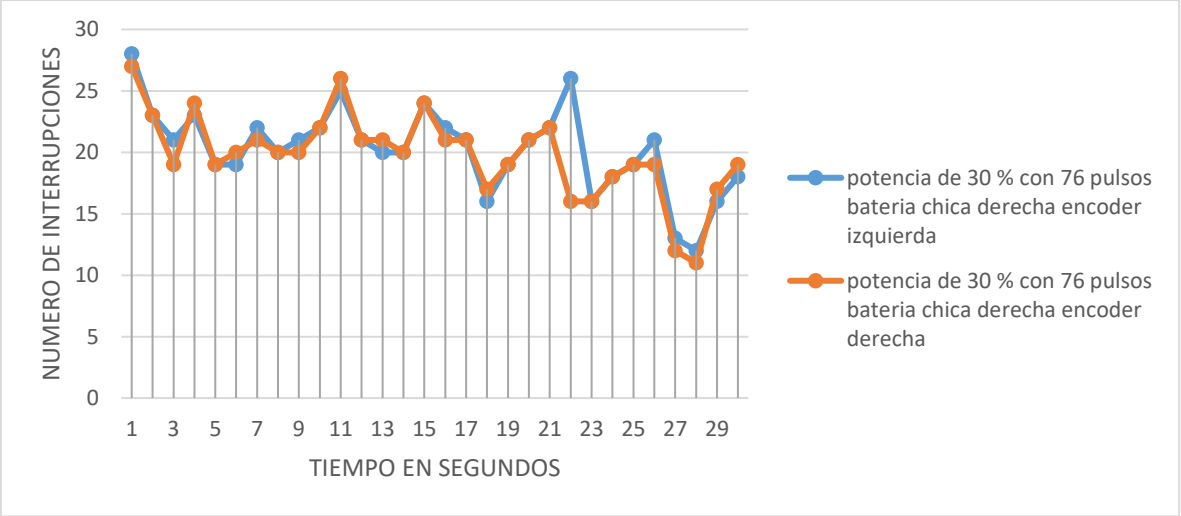


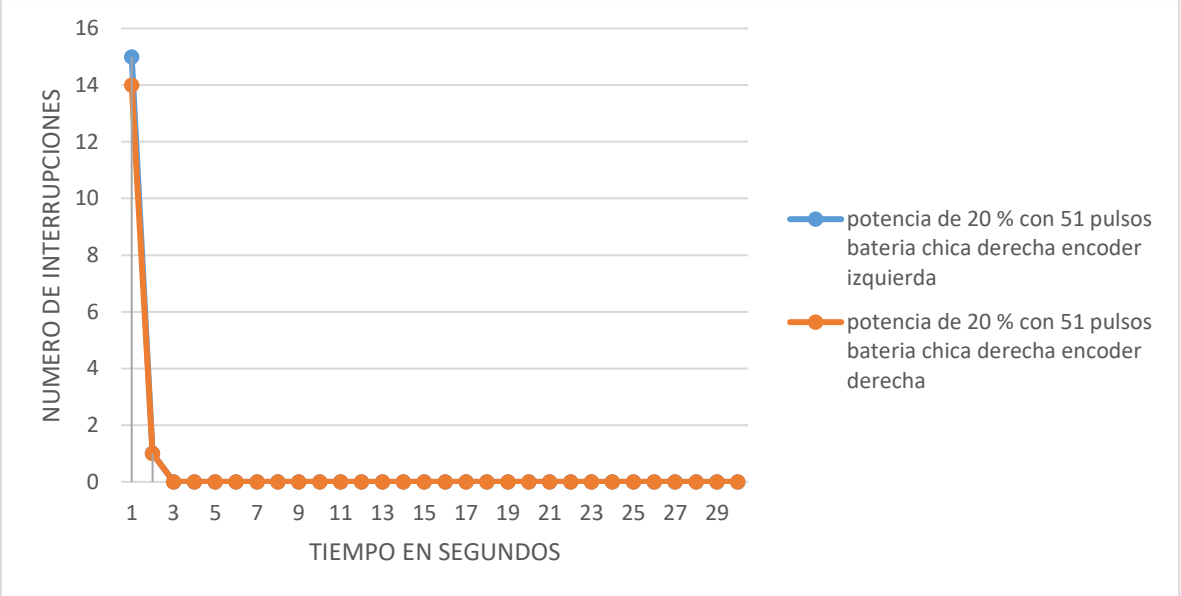

NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica derecha encoder izquierda
potencia de 70 % con 178 pulsos bateria chica derecha encoder derecha


NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica derecha encoder izquierda
potencia de 60 % con 153 pulsos bateria chica derecha encoder derecha

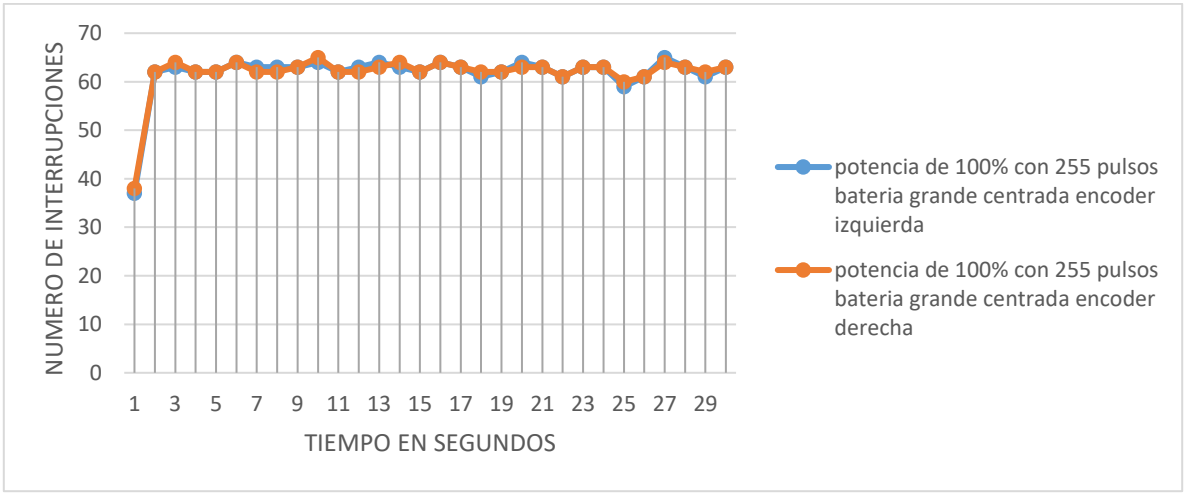

NUMERO DE INTERRUPCIONES
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica derecha encoder izquierda
potencia de 50 % con 127 pulsos bateria chica derecha encoder derecha

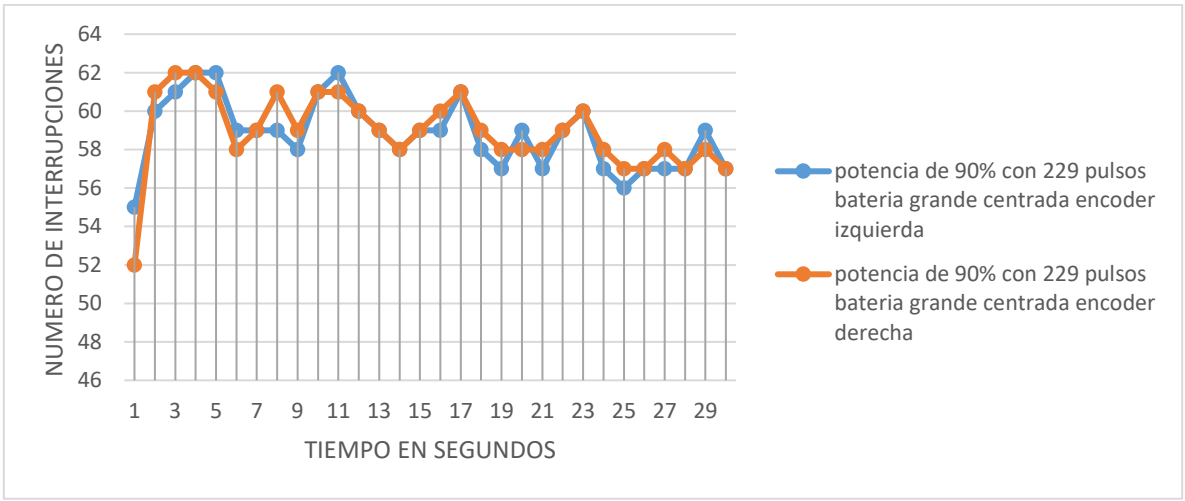

NUMERO DE INTERRUPCIONES
50
49
48
47
46
45
44
43
42
41
40
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria chica derecha encoder izquierda
potencia de 40 % con 102 pulsos bateria chica derecha encoder derecha

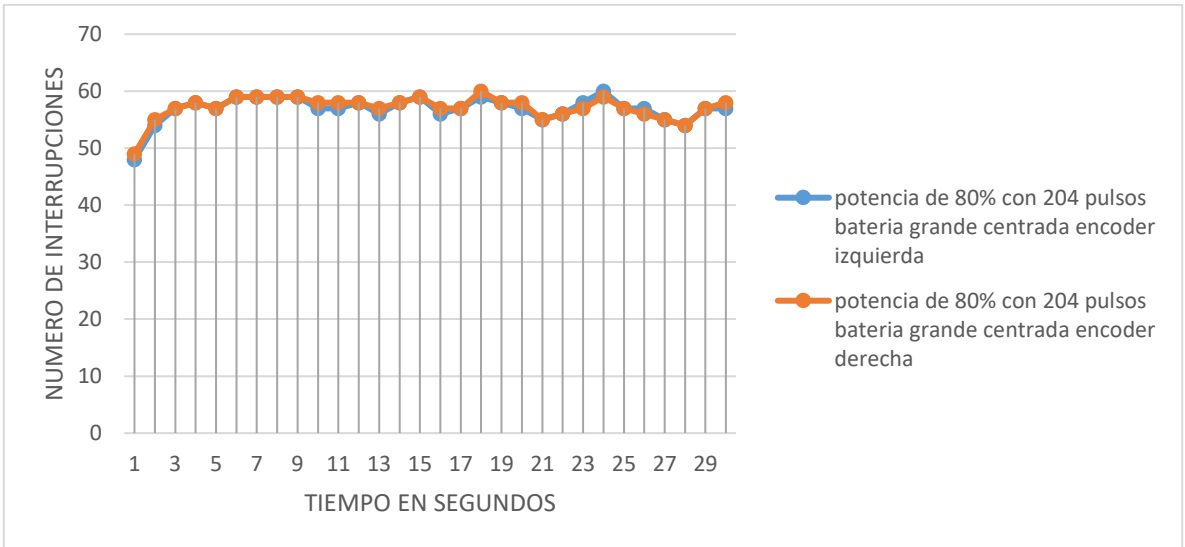

NUMERO DE INTERRUPCIONES
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30 % con 76 pulsos bateria chica derecha encoder izquierda
potencia de 30 % con 76 pulsos bateria chica derecha encoder derecha

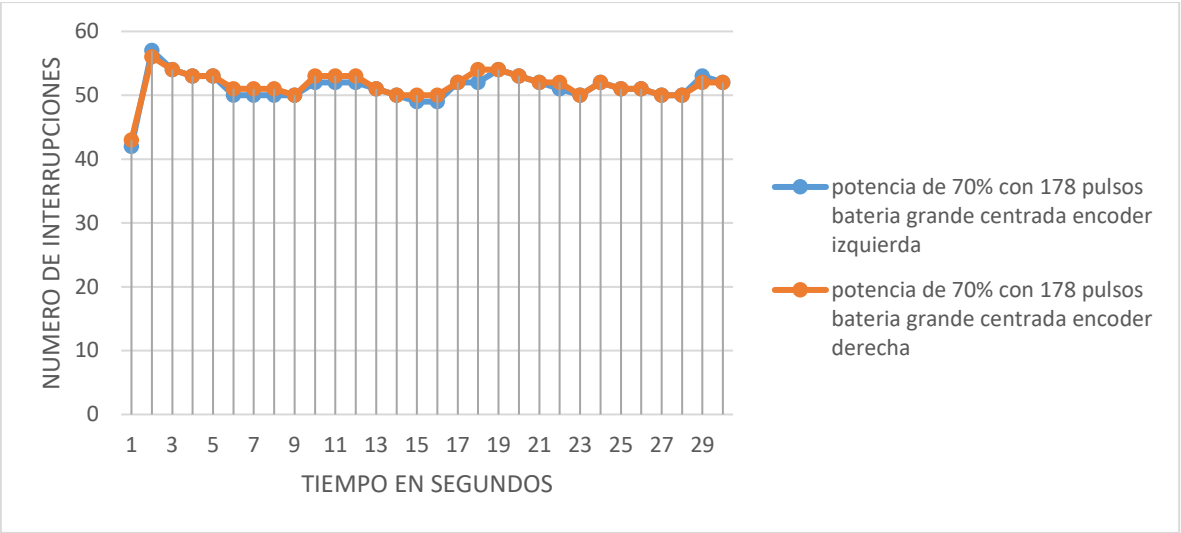

NUMERO DE INTERRUPCIONES
16
14
12
10
8
6
4
2
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 20 % con 51 pulsos bateria chica derecha encoder izquierda
potencia de 20 % con 51 pulsos bateria chica derecha encoder derecha

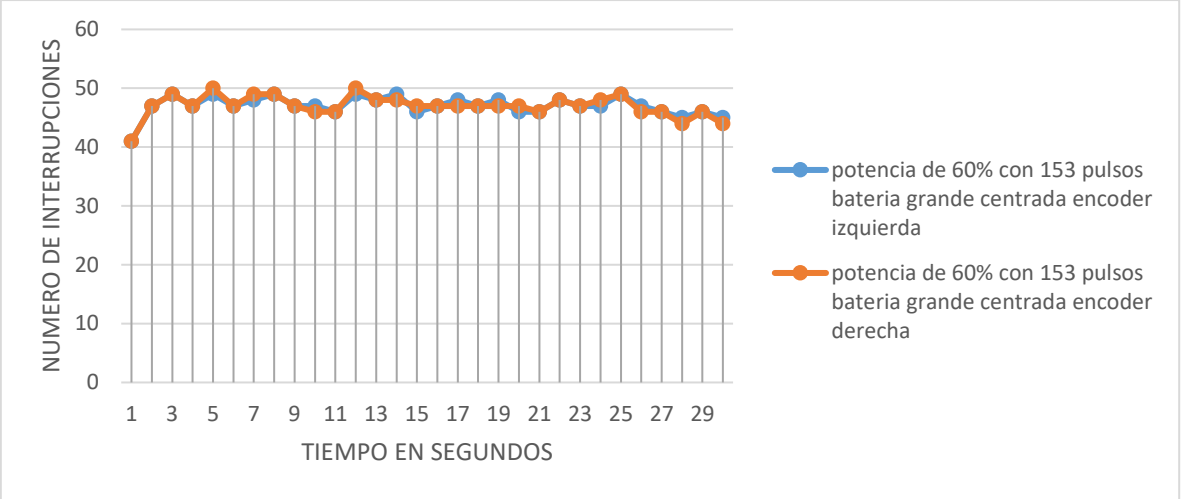
Pruebas de velocidad usando Llantas de tracción, con carga de 2.5k centrada

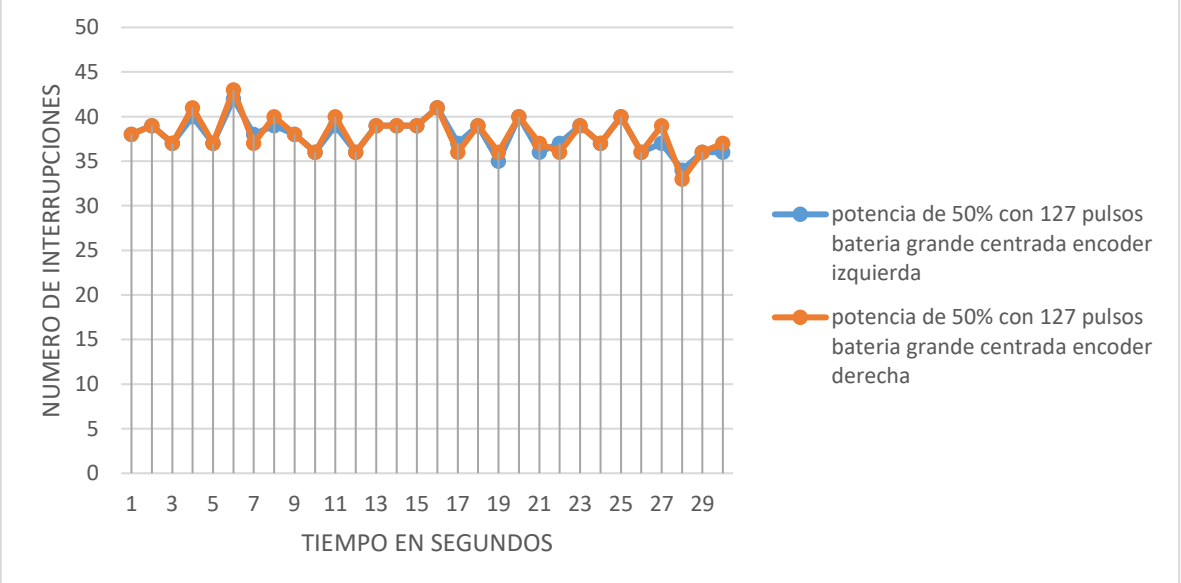


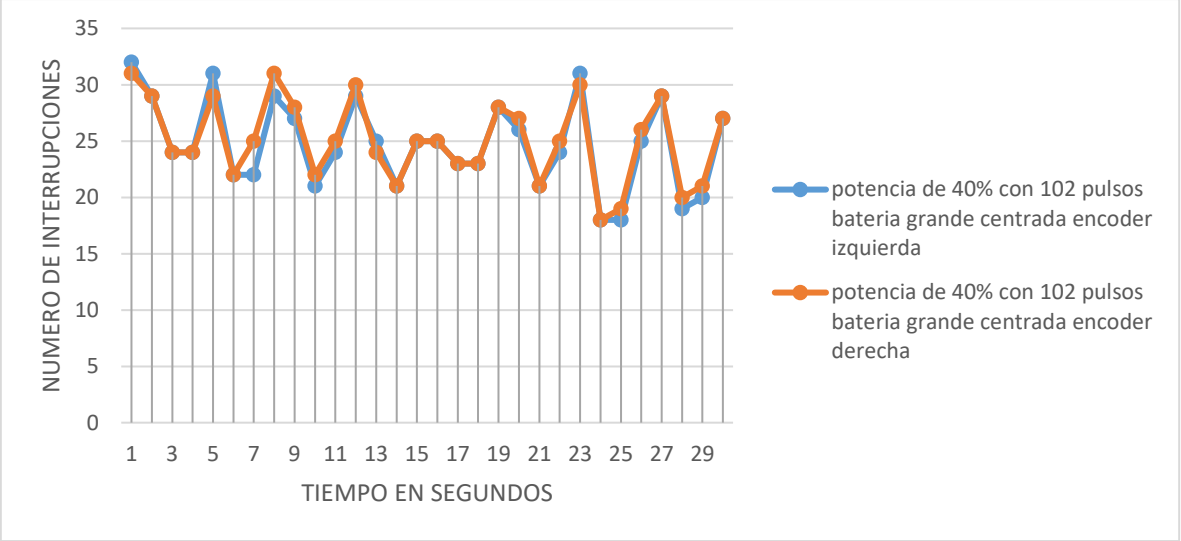


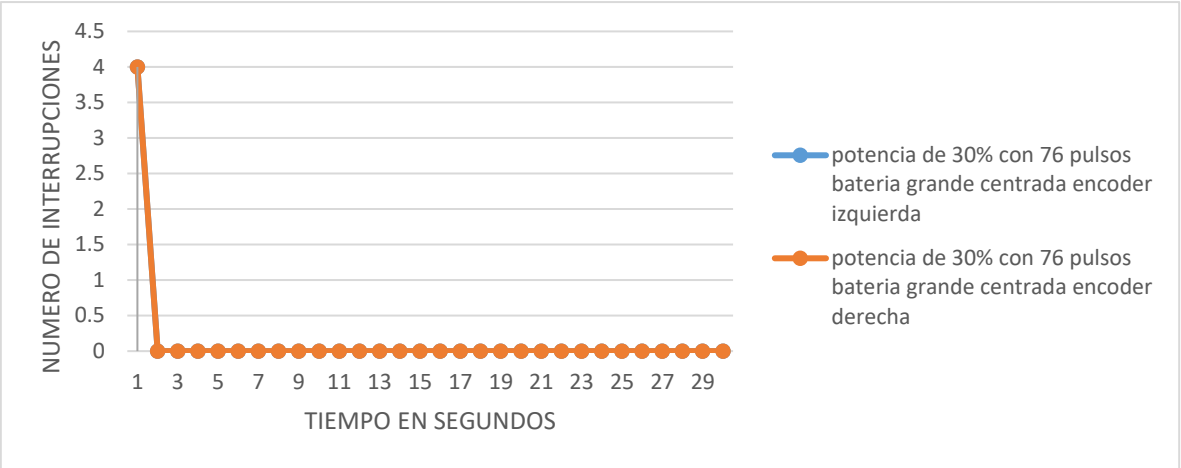



NUMERO DE INTERRUPCIONES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos bateria grande centrada encoder izquierda
potencia de 70% con 178 pulsos bateria grande centrada encoder derecha

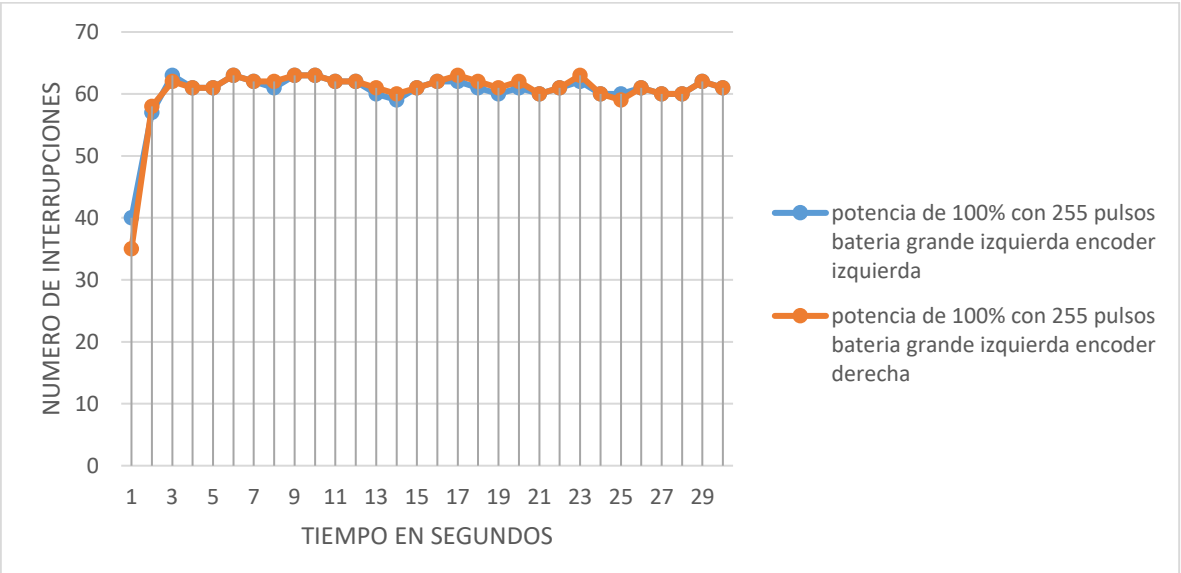

NUMERO DE INTERRUPCIONES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos bateria grande centrada encoder izquierda
potencia de 60% con 153 pulsos bateria grande centrada encoder derecha

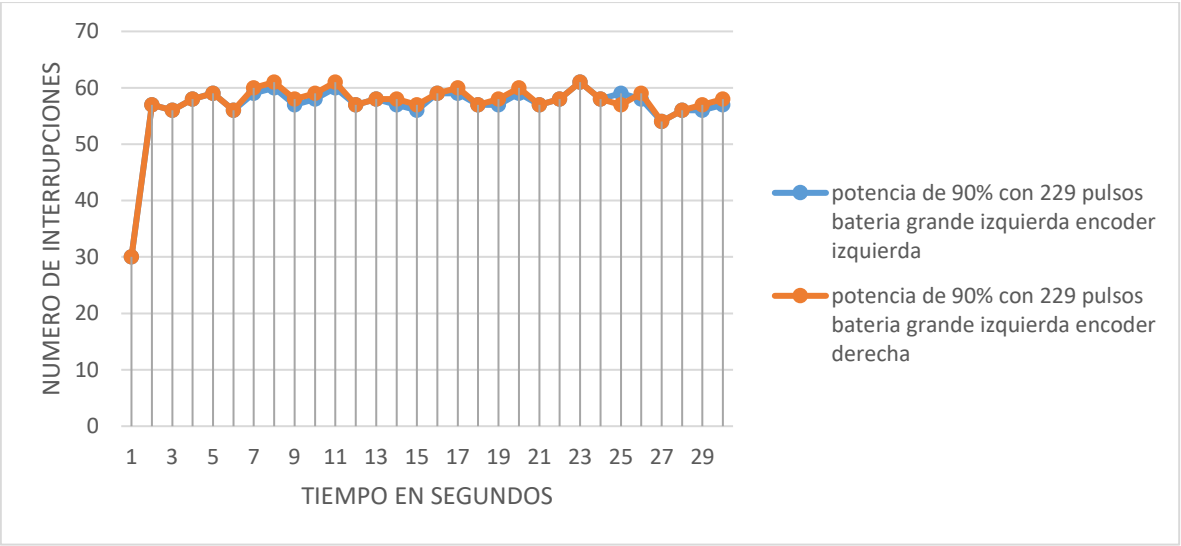

NUMERO DE INTERRUPCIONES
50
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos bateria grande centrada encoder izquierda
potencia de 50% con 127 pulsos bateria grande centrada encoder derecha

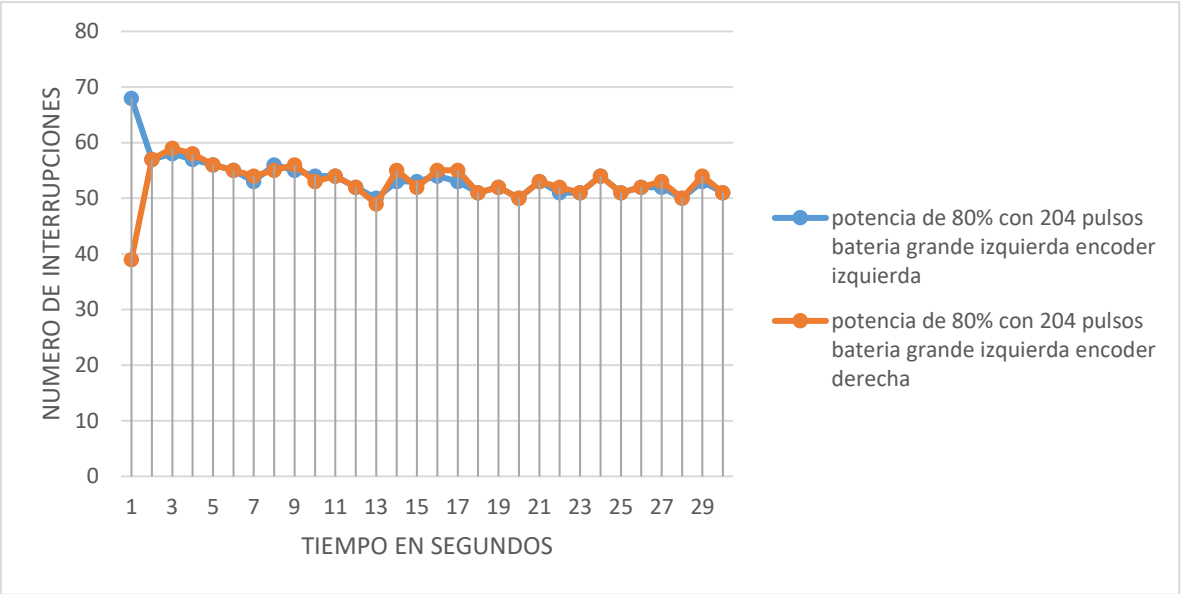


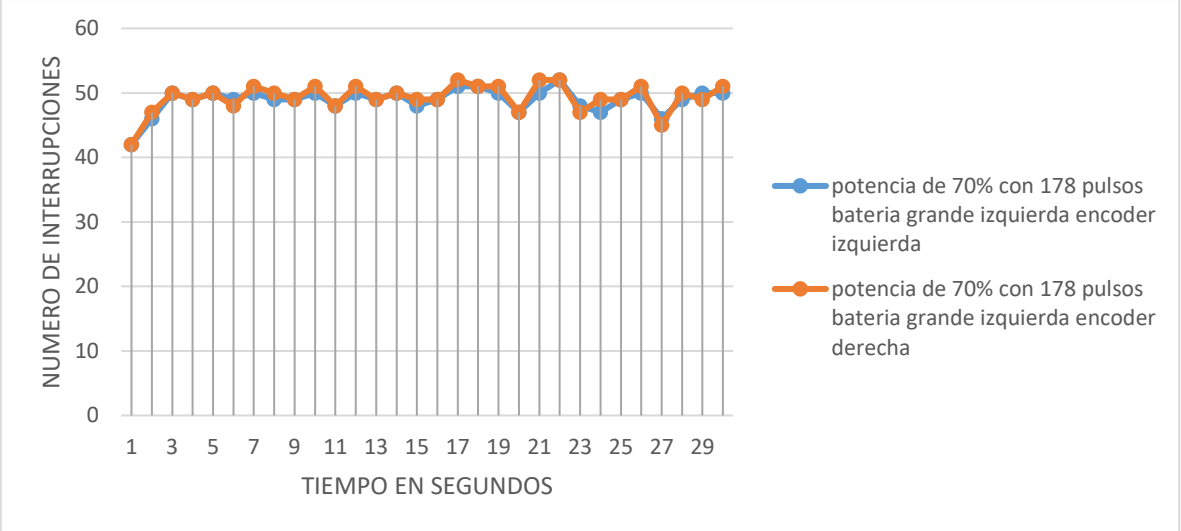


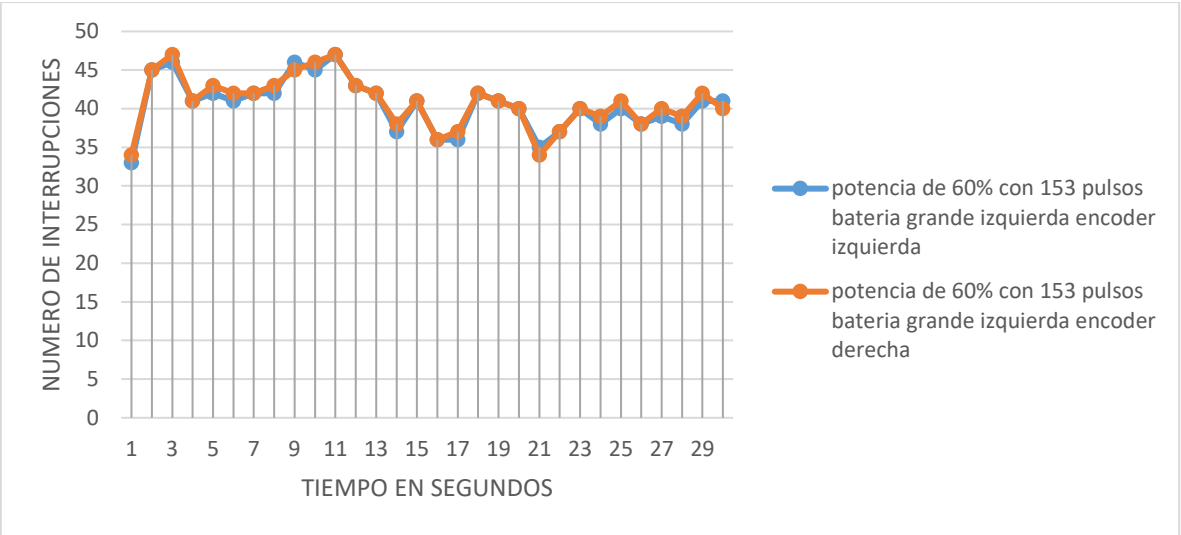
Pruebas de velocidad usando Llantas de tracción, con carga de 2.5k lado izquierdo

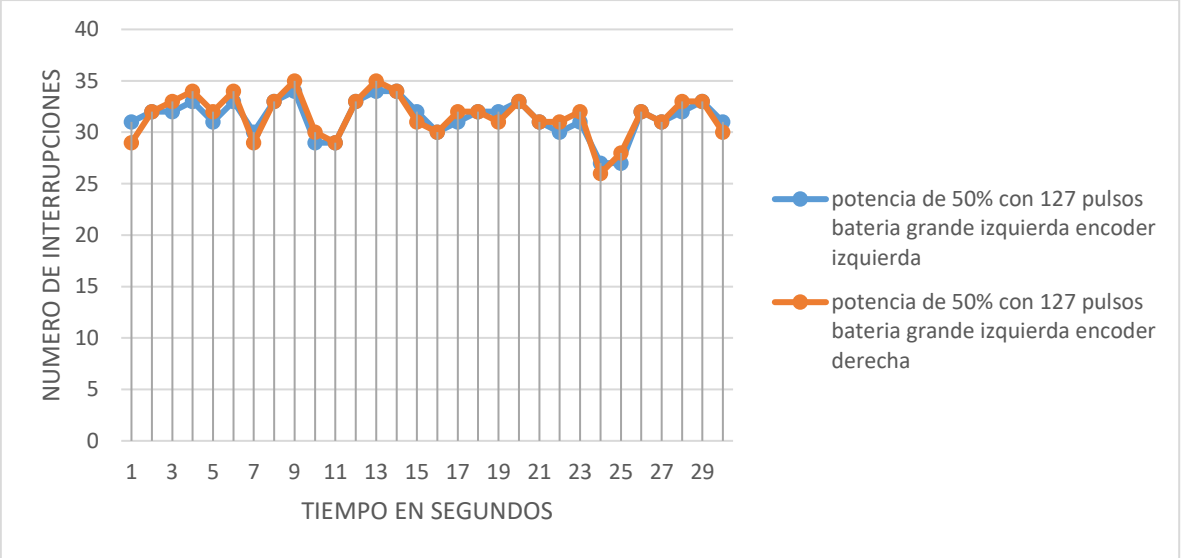


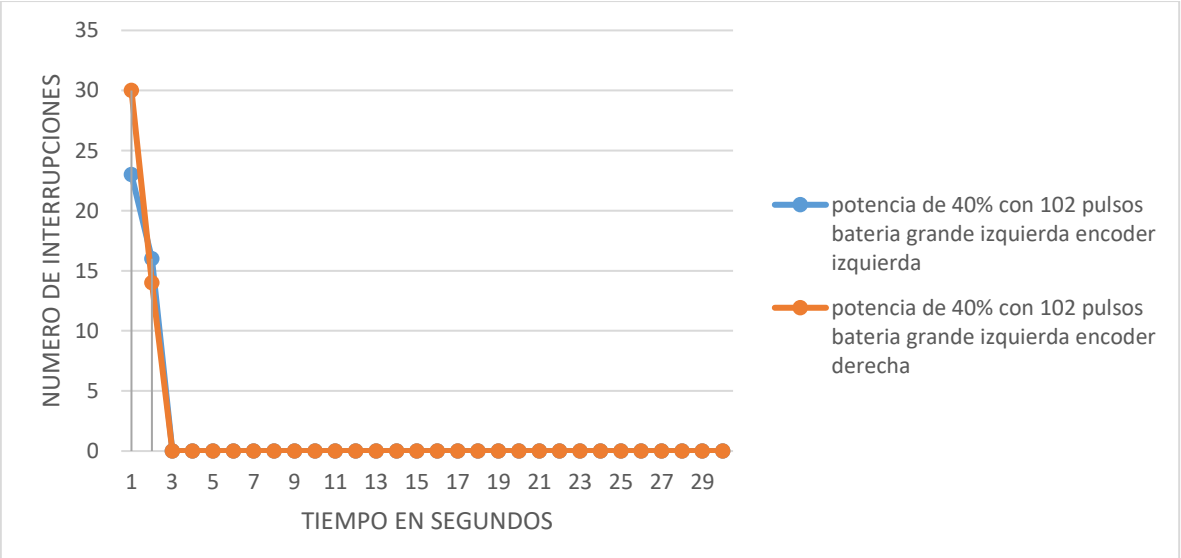

NUMERO DE INTERRUPCIONES
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 90% con 229 pulsos bateria grande izquierda encoder izquierda
potencia de 90% con 229 pulsos bateria grande izquierda encoder derecha


NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80% con 204 pulsos bateria grande izquierda encoder izquierda
potencia de 80% con 204 pulsos bateria grande izquierda encoder derecha

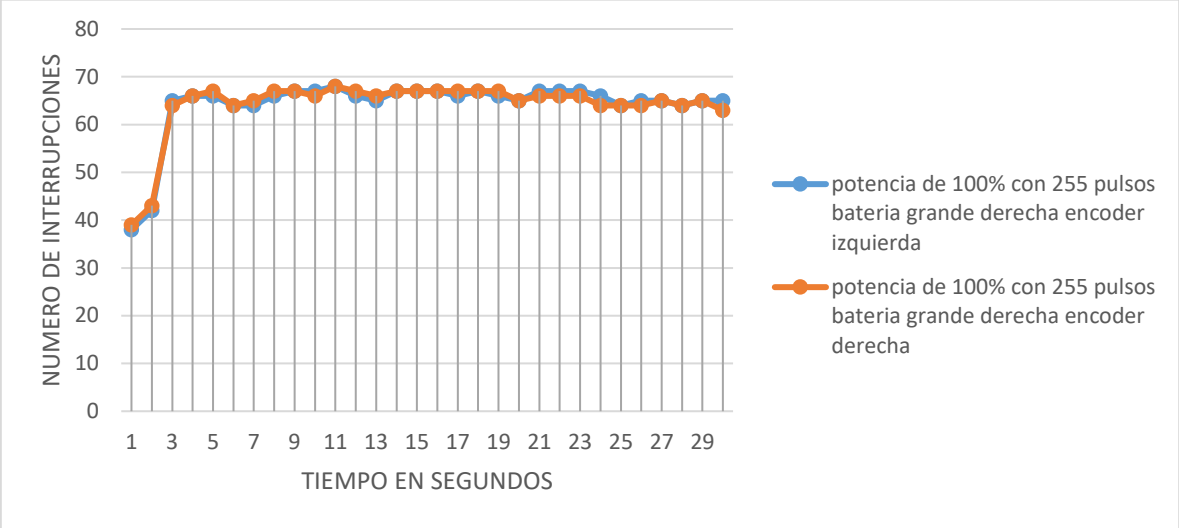

NUMERO DE INTERRUPCIONES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos bateria grande izquierda encoder izquierda
potencia de 70% con 178 pulsos bateria grande izquierda encoder derecha

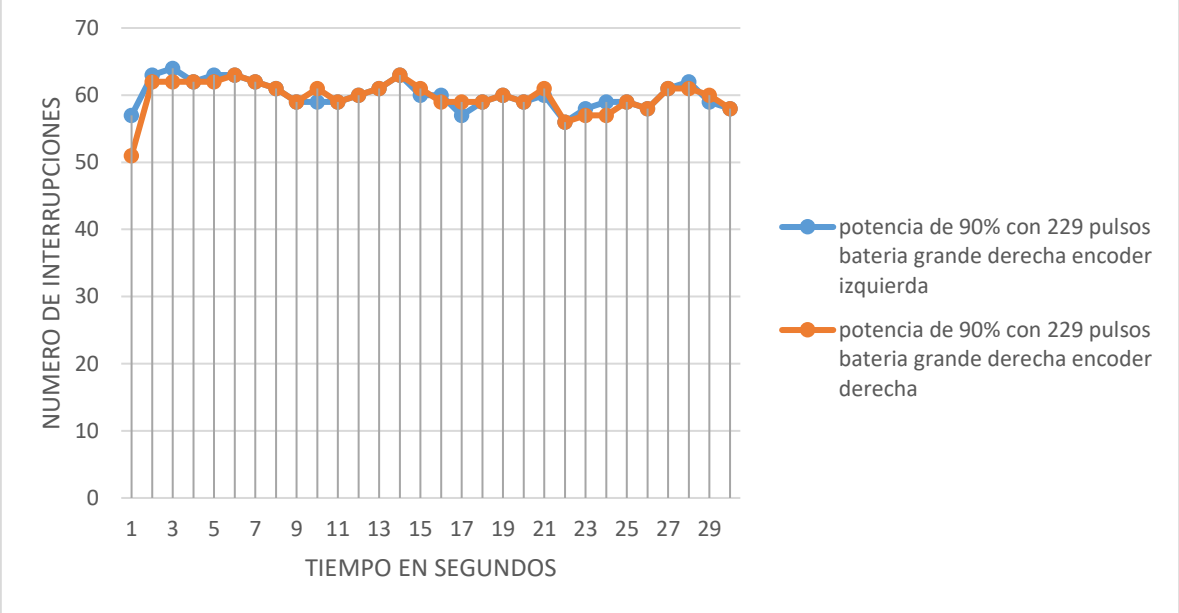

NUMERO DE INTERRUPCIONES
50
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos bateria grande izquierda encoder izquierda
potencia de 60% con 153 pulsos bateria grande izquierda encoder derecha

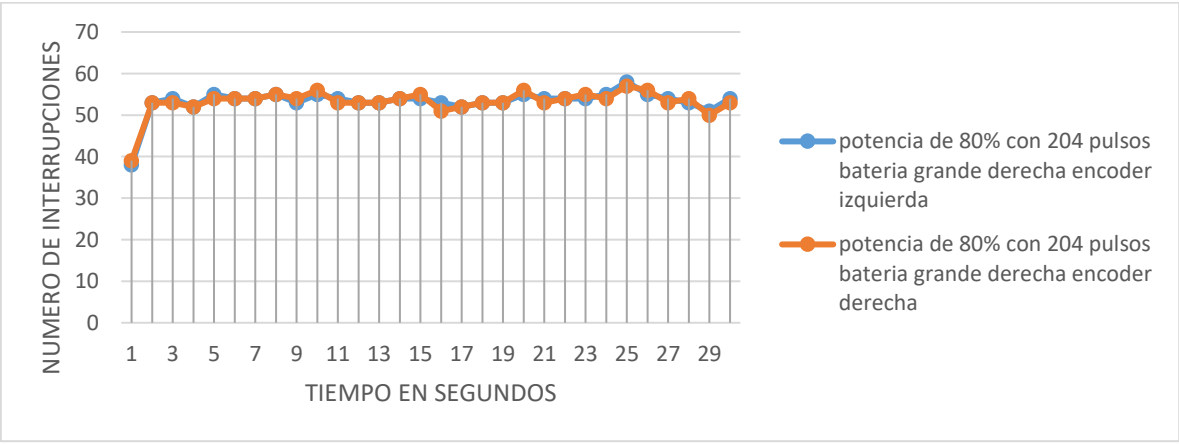

NUMERO DE INTERRUPCIONES
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos bateria grande izquierda encoder izquierda
potencia de 50% con 127 pulsos bateria grande izquierda encoder derecha

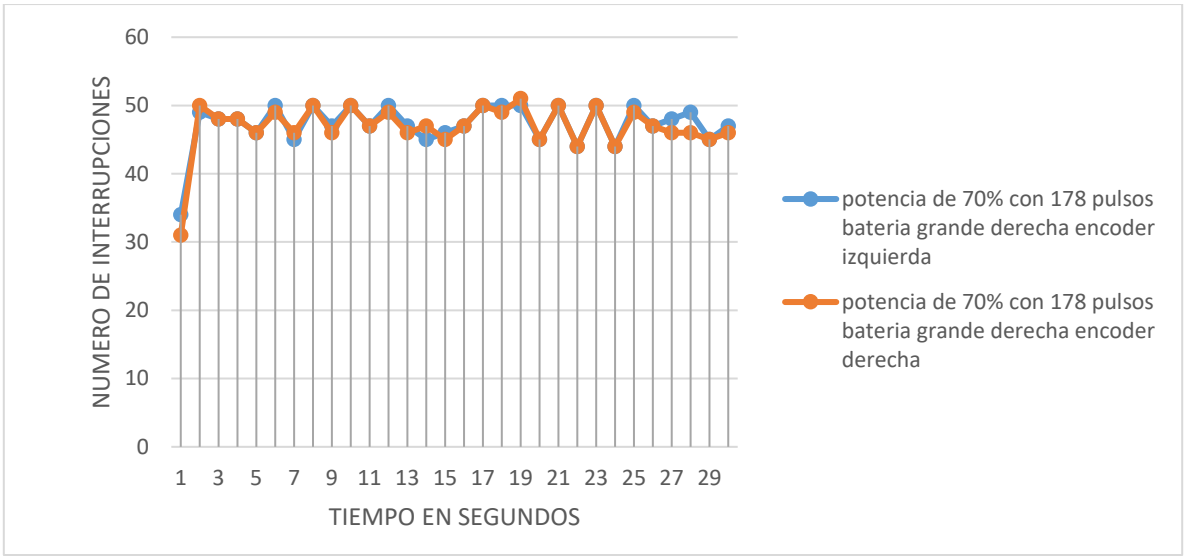

NUMERO DE INTERRUPCIONES
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40% con 102 pulsos bateria grande izquierda encoder izquierda
potencia de 40% con 102 pulsos bateria grande izquierda encoder derecha

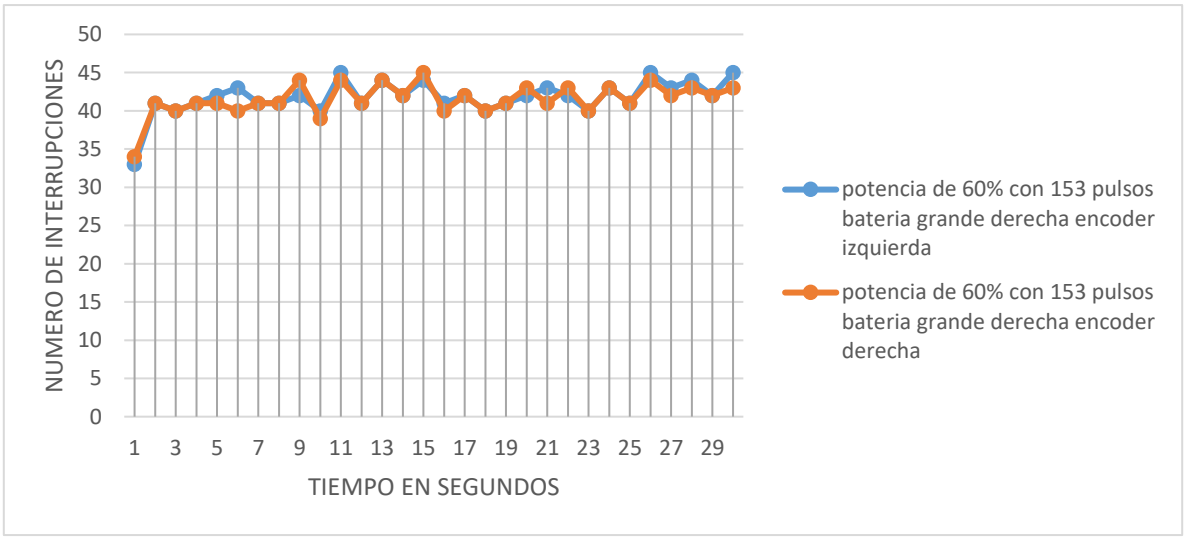
Pruebas de velocidad usando Llantas de tracción, con carga de 2.5k lado derecho

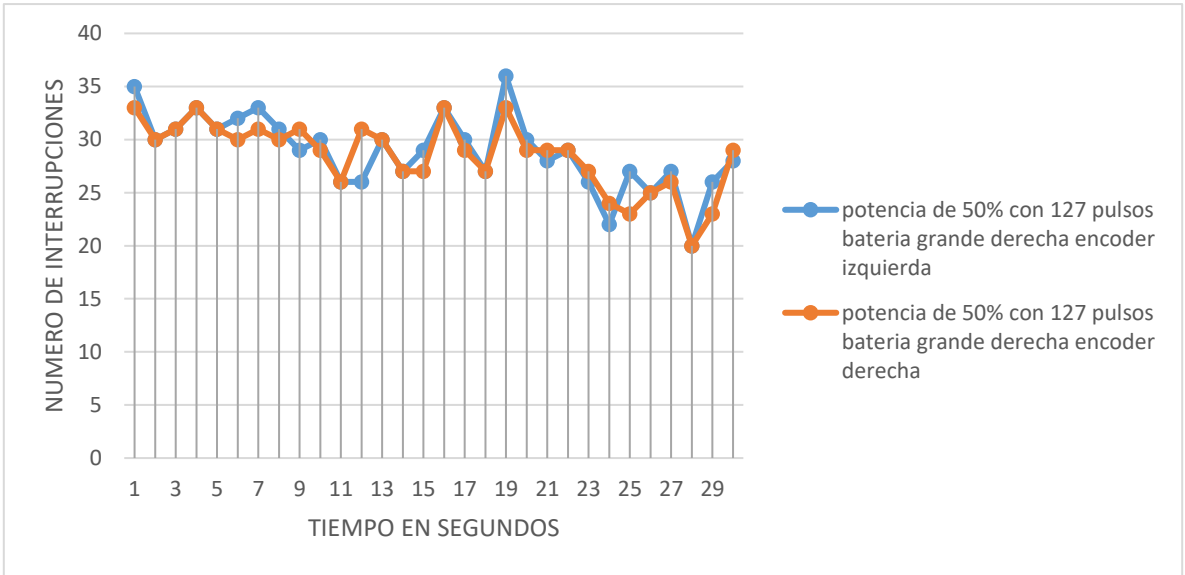




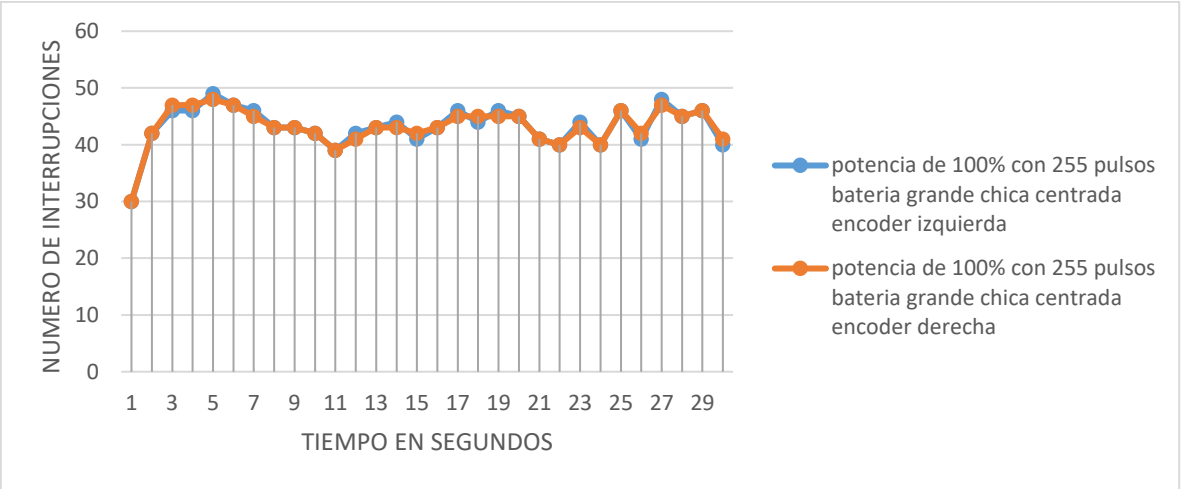


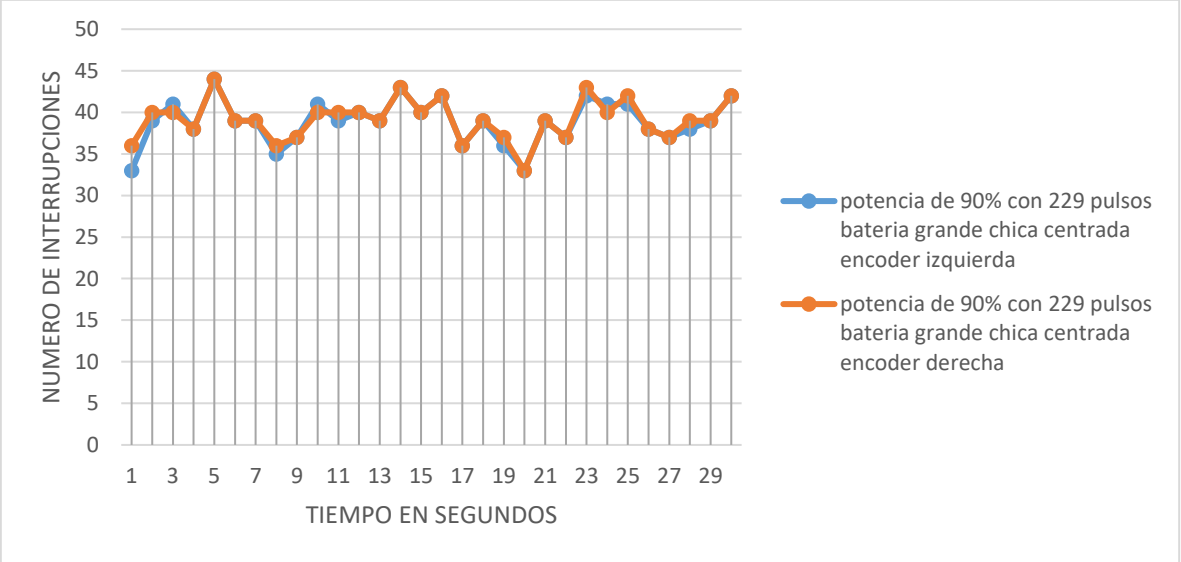

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos bateria grande derecha encoder izquierda
potencia de 70% con 178 pulsos bateria grande derecha encoder derecha

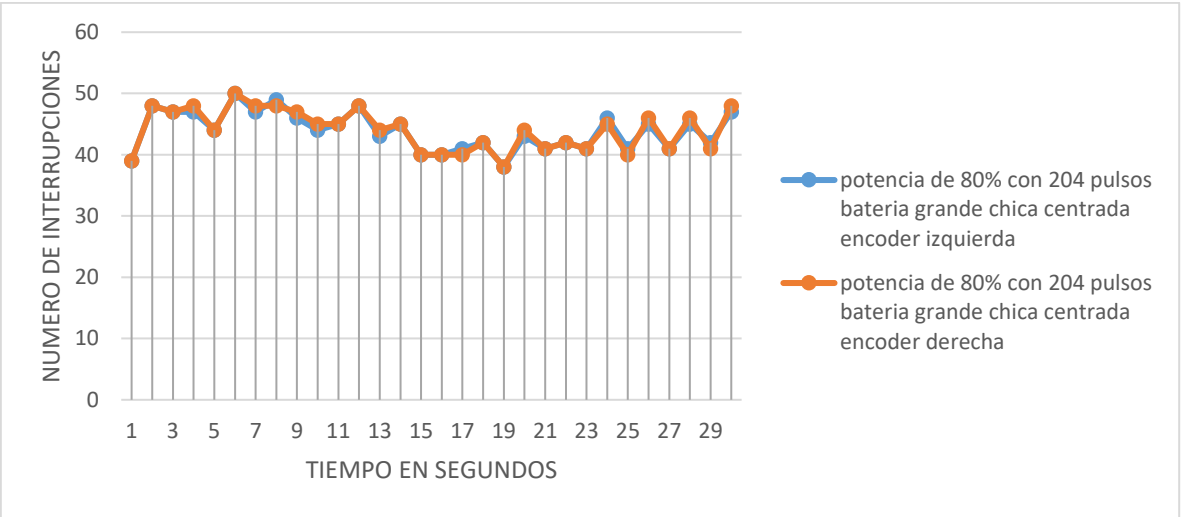

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos bateria grande derecha encoder izquierda
potencia de 60% con 153 pulsos bateria grande derecha encoder derecha

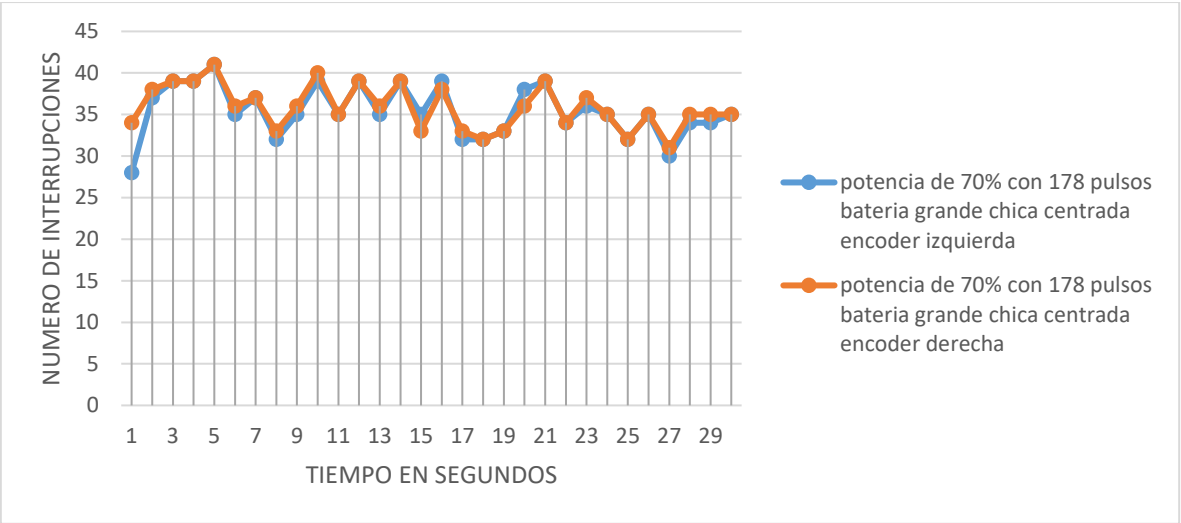

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos bateria grande derecha encoder izquierda
potencia de 50% con 127 pulsos bateria grande derecha encoder derecha

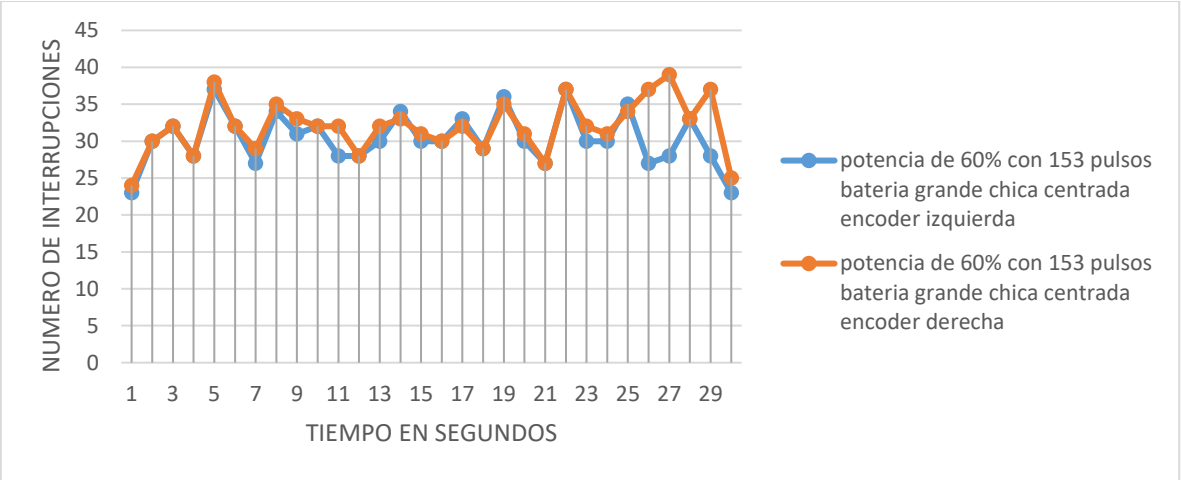
Pruebas de velocidad usando Llantas de tracción, con carga de 4.2k centrada



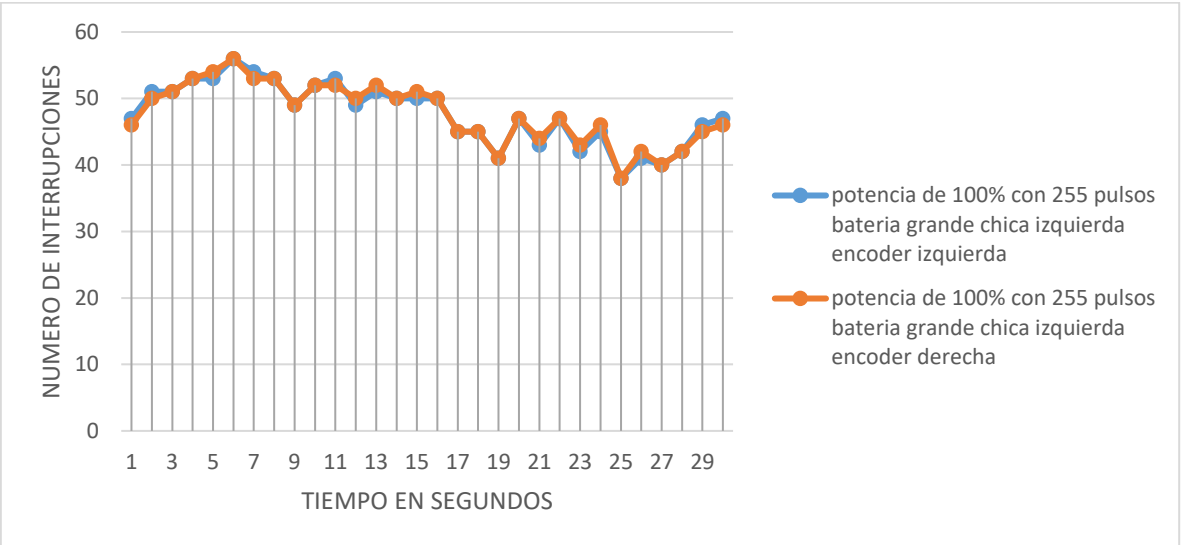


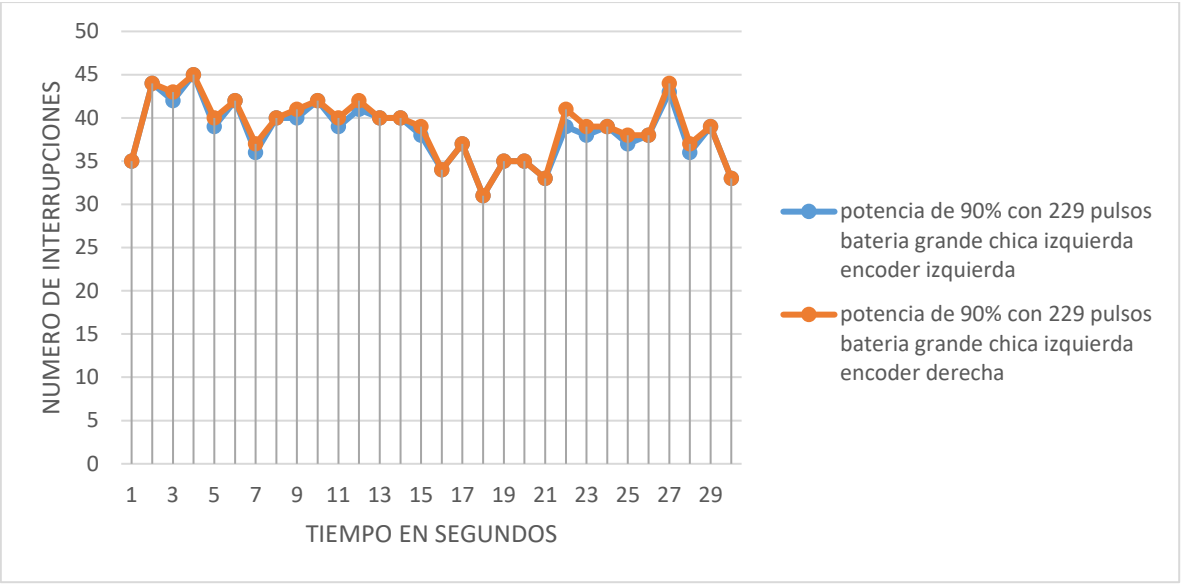


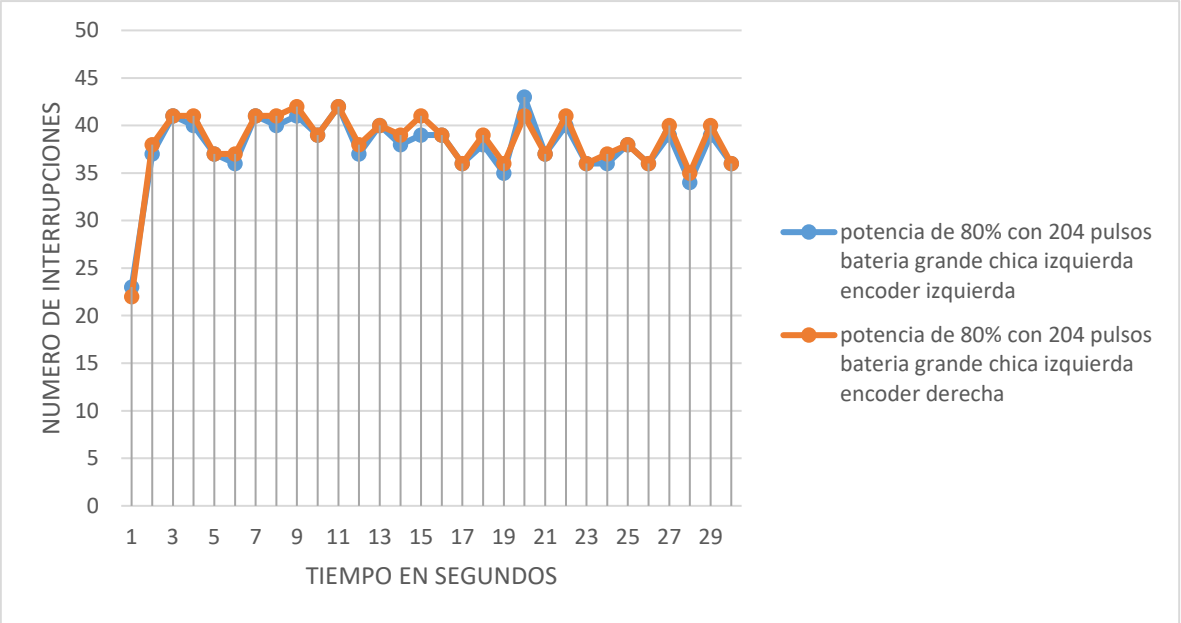


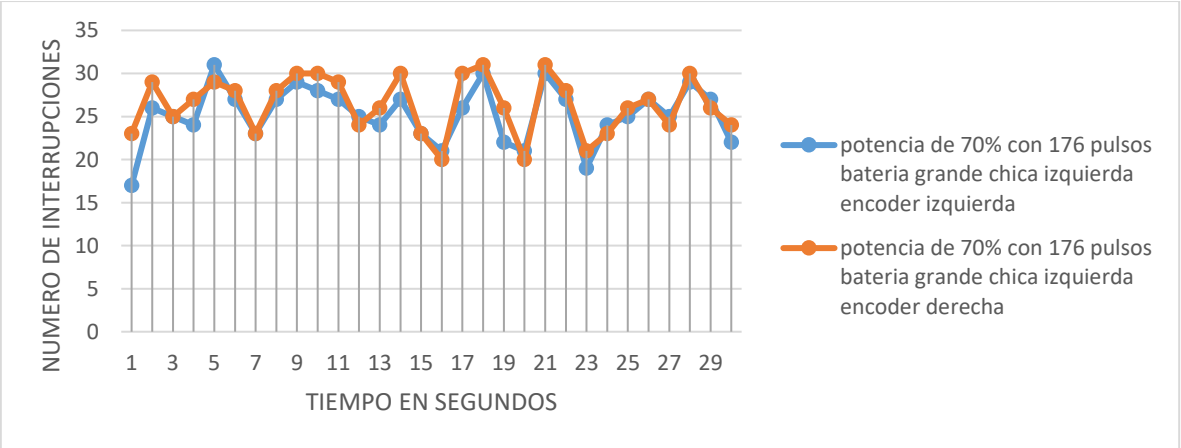


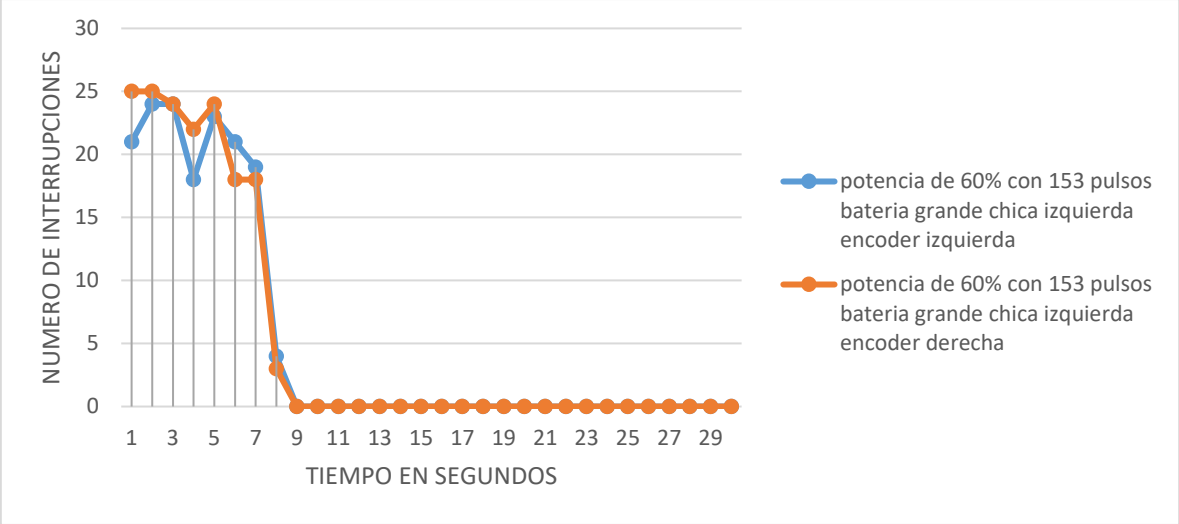
Pruebas de velocidad usando Llantas de tracción, con carga de 4.2k lado izquierdo



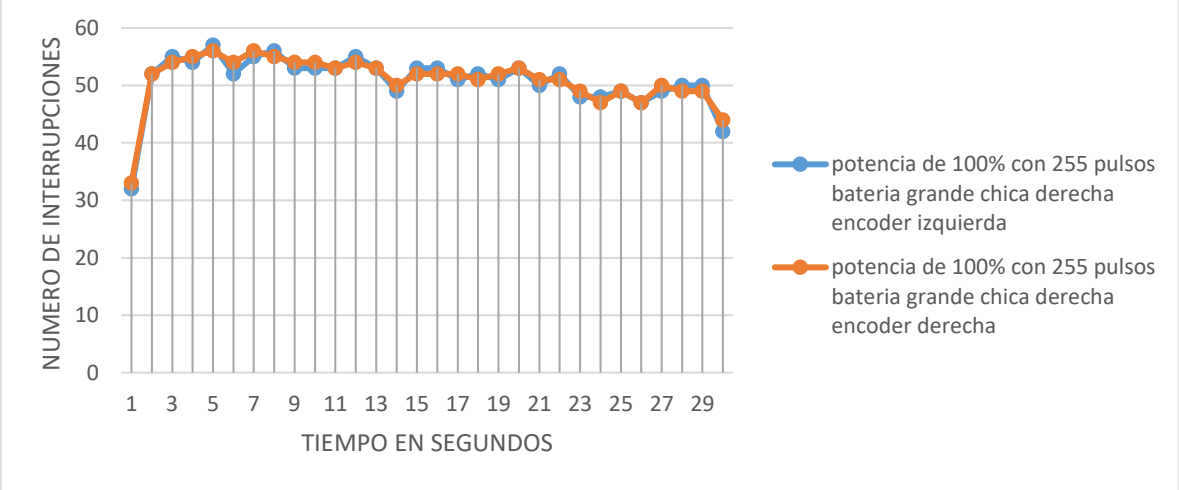

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 90% con 229 pulsos bateria grande chica izquierda encoder izquierda
potencia de 90% con 229 pulsos bateria grande chica izquierda encoder derecha

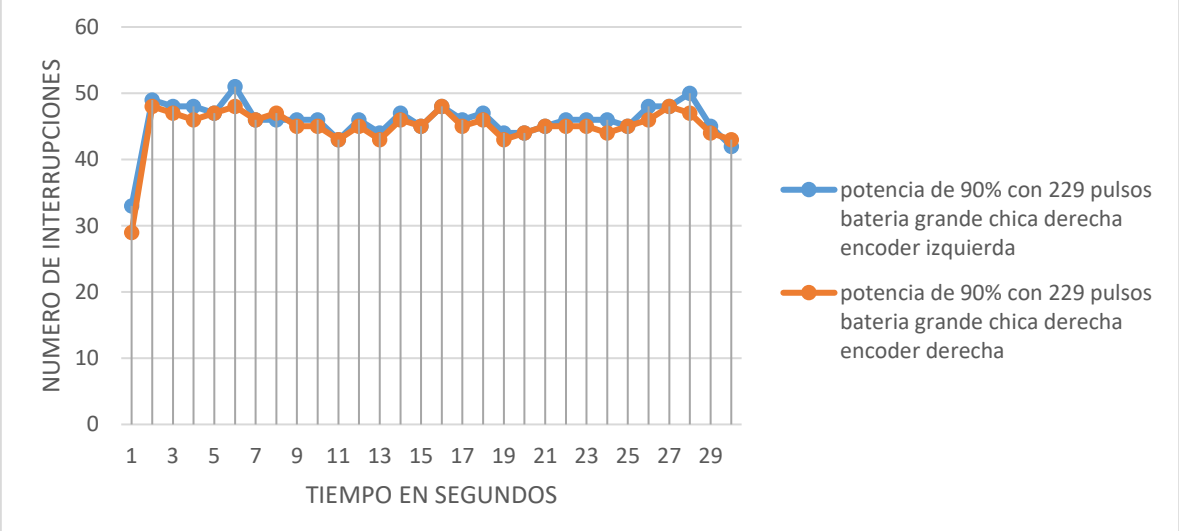

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 80% con 204 pulsos bateria grande chica izquierda encoder izquierda
potencia de 80% con 204 pulsos bateria grande chica izquierda encoder derecha

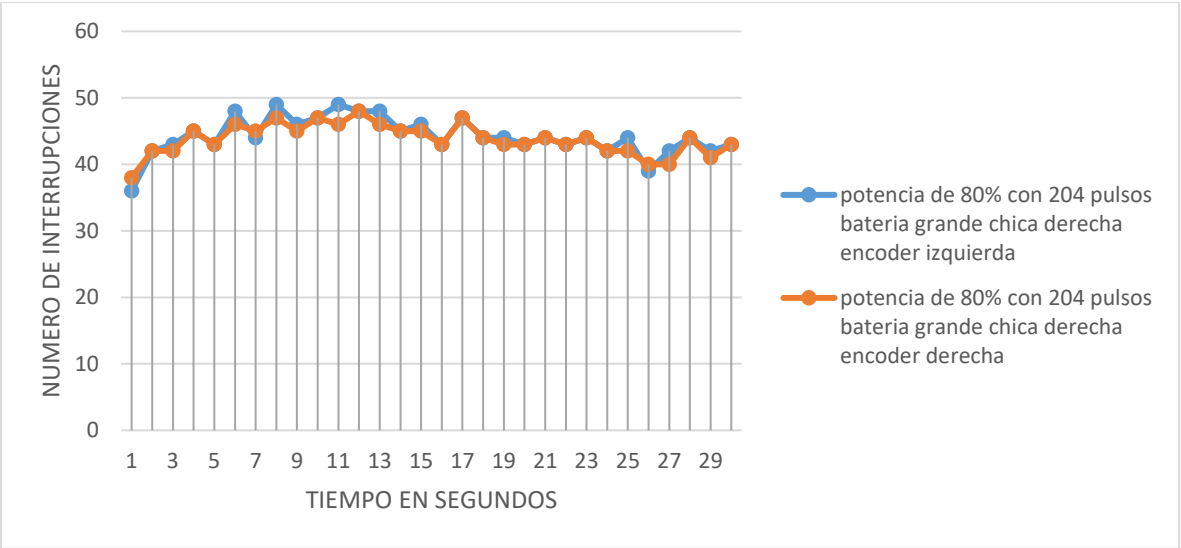

NUMERO DE INTERRUPCIONES
TIEMPO EN SEGUNDOS
potencia de 70% con 176 pulsos bateria grande chica izquierda encoder izquierda
potencia de 70% con 176 pulsos bateria grande chica izquierda encoder derecha

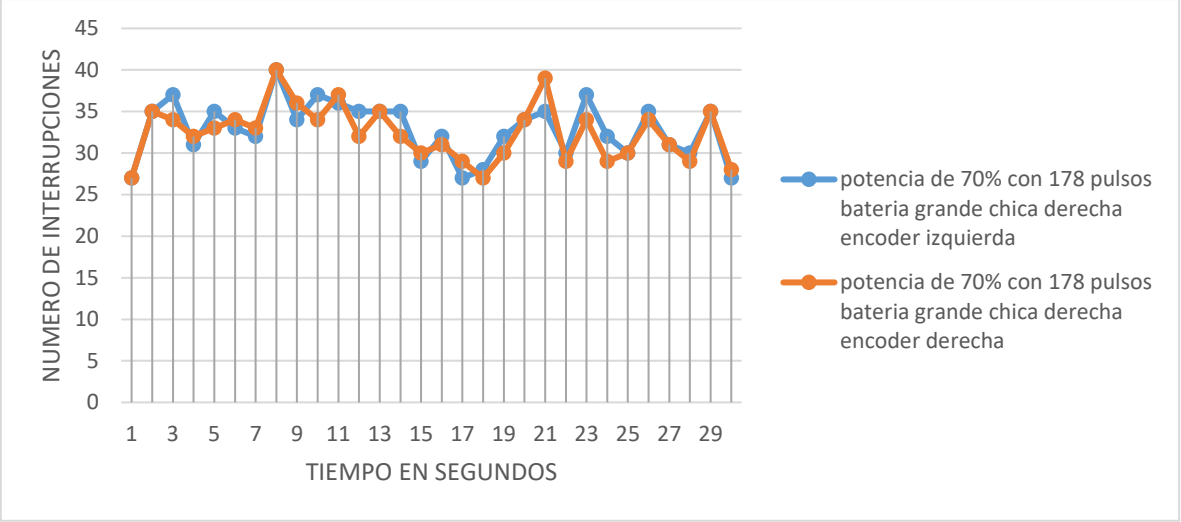


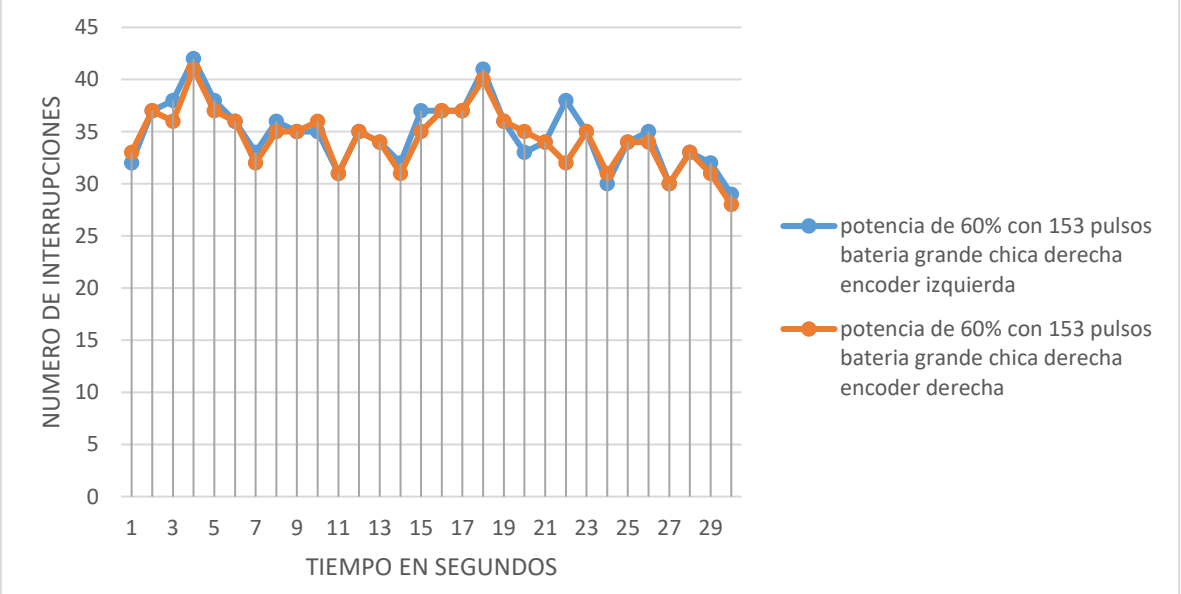
Pruebas de velocidad usando Llantas de tracción, con carga de 4.2k lado derecho



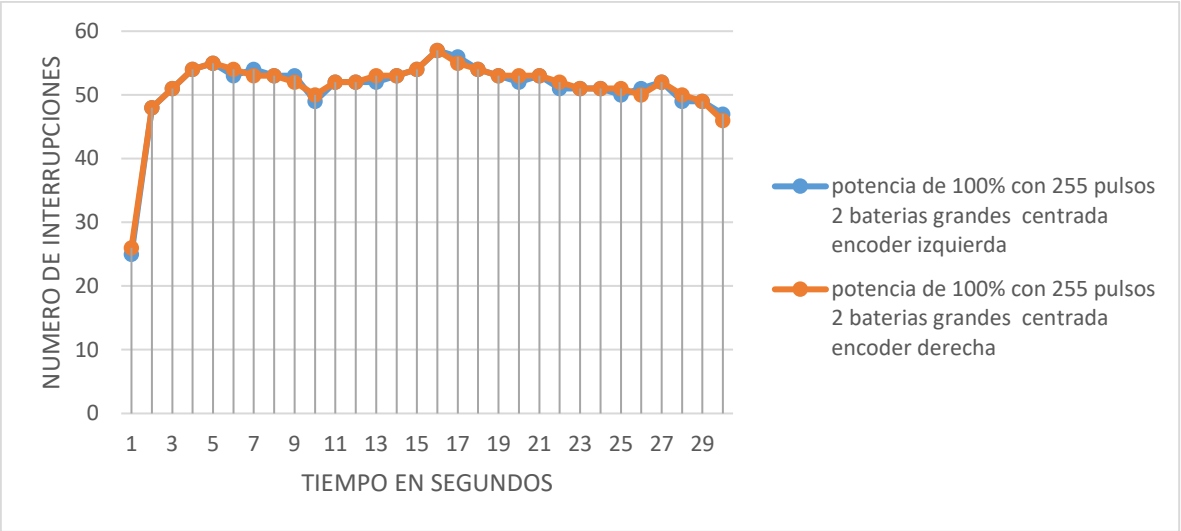


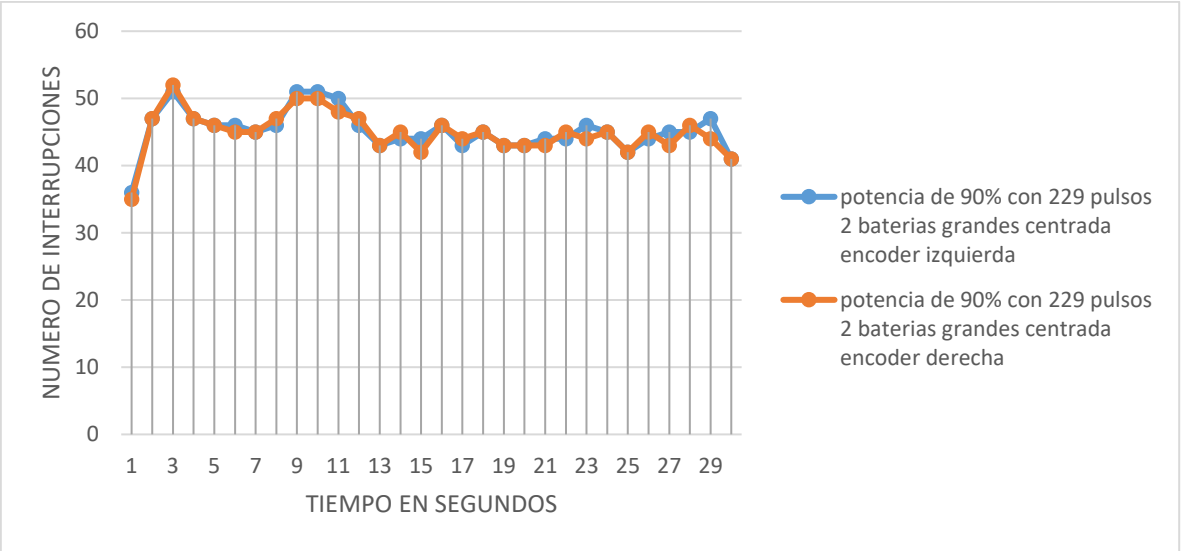

NUMERO DE INTERRUPCIONES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80% con 204 pulsos bateria grande chica derecha encoder izquierda
potencia de 80% con 204 pulsos bateria grande chica derecha encoder derecha

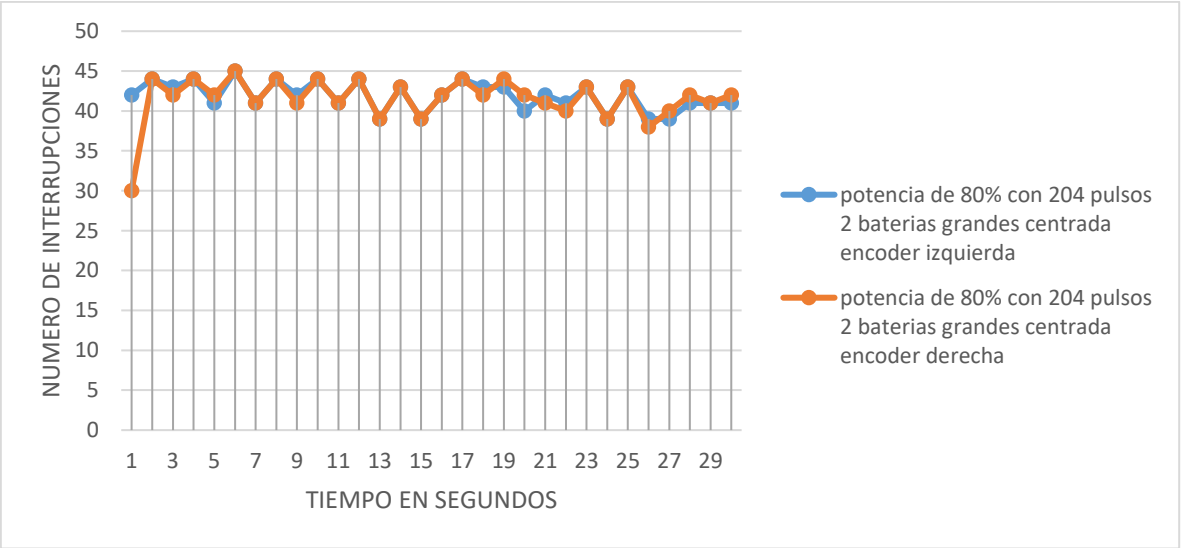

NUMERO DE INTERRUPCIONES
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos bateria grande chica derecha encoder izquierda
potencia de 70% con 178 pulsos bateria grande chica derecha encoder derecha

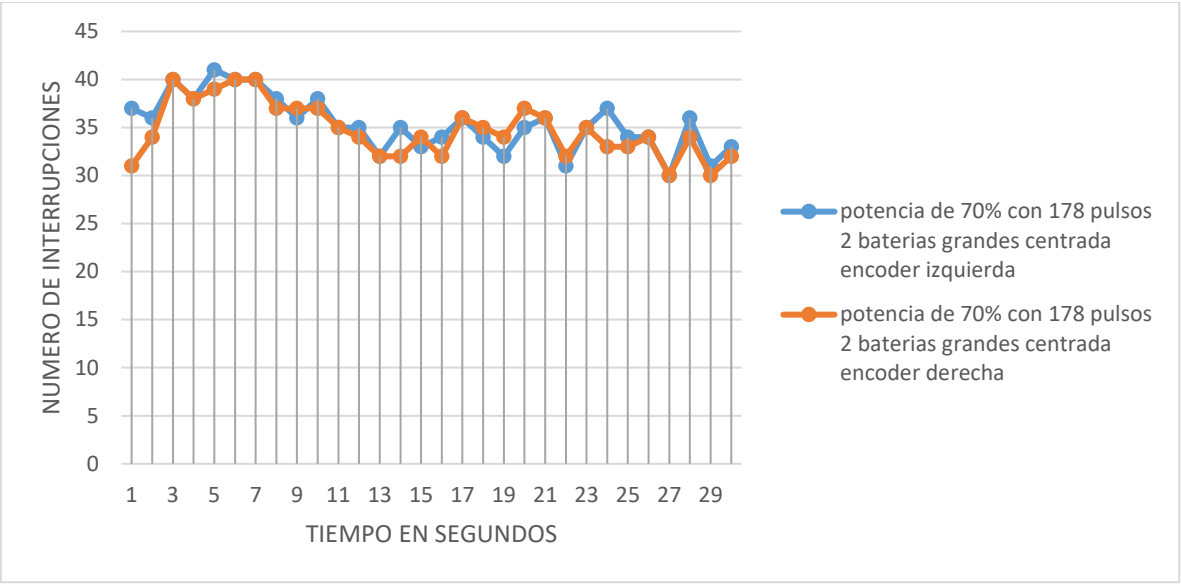

NUMERO DE INTERRUPCIONES
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos bateria grande chica derecha encoder izquierda
potencia de 60% con 153 pulsos bateria grande chica derecha encoder derecha

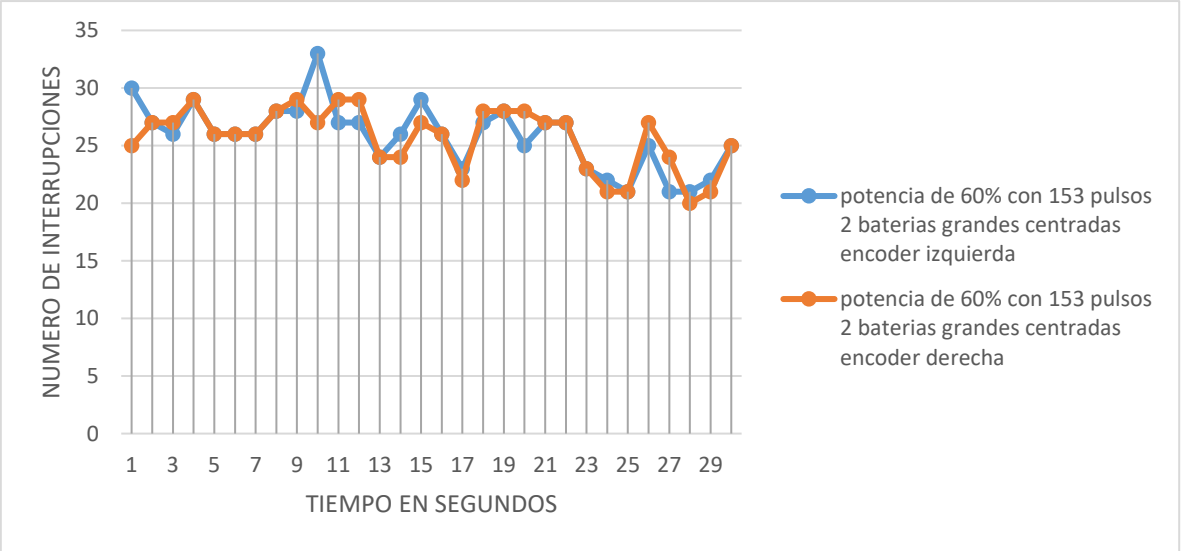
Pruebas de velocidad usando Llantas de tracción, con carga de 5k centrada

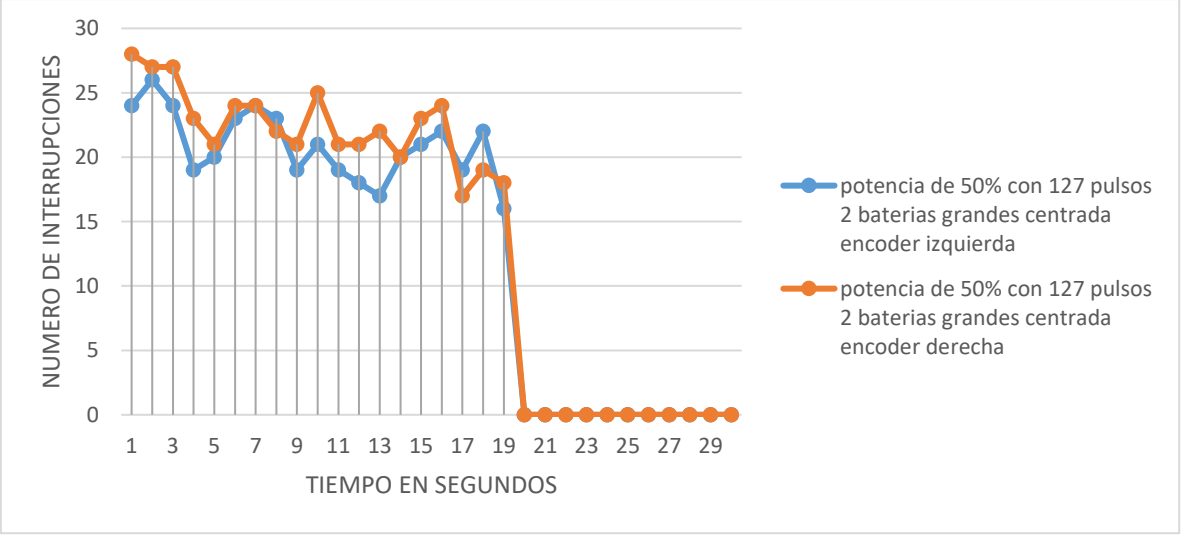




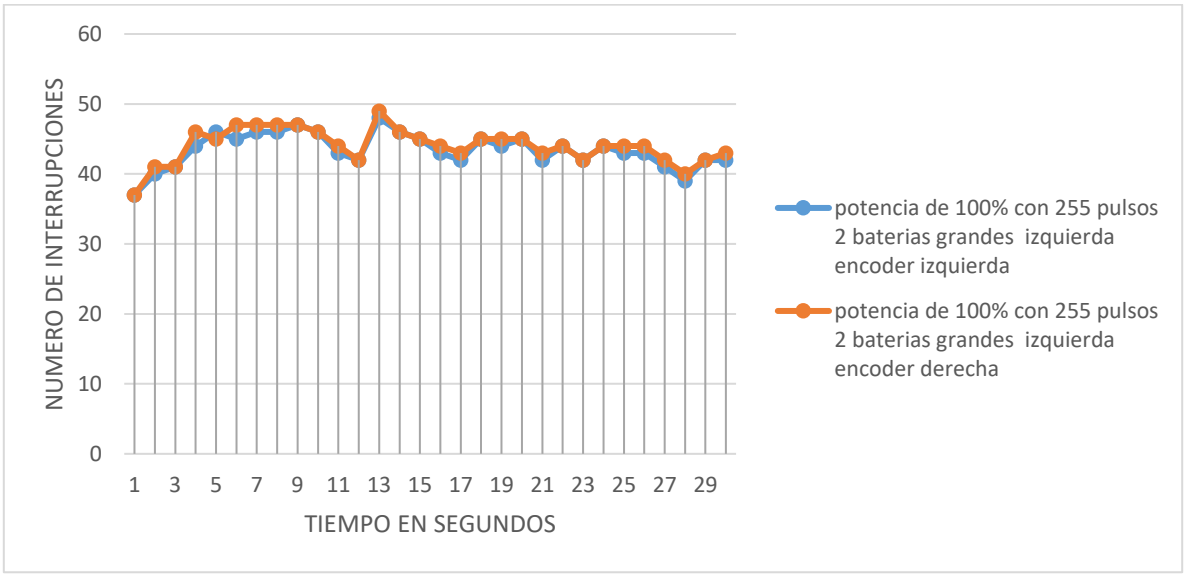


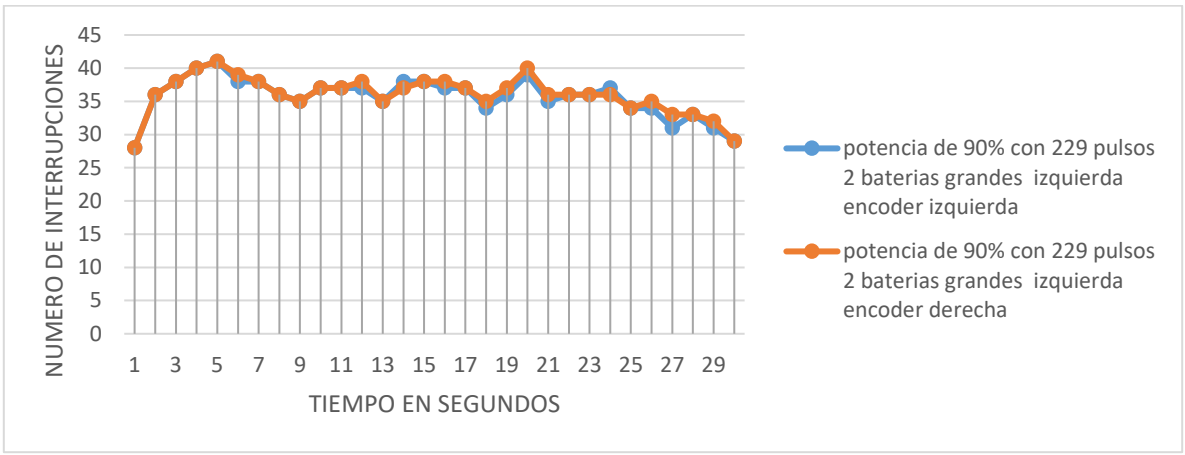

NUMERO DE INTERRUPCIONES
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos 2 baterias grandes centrada encoder izquierda
potencia de 70% con 178 pulsos 2 baterias grandes centrada encoder derecha

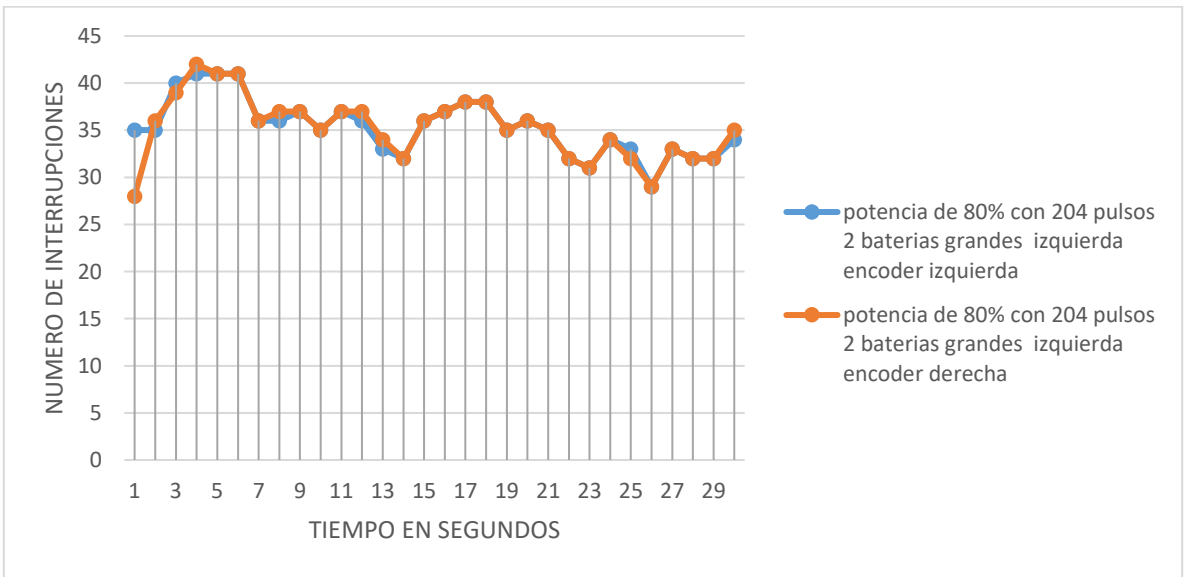

NUMERO DE INTERRUPCIONES
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos 2 baterias grandes centradas encoder izquierda
potencia de 60% con 153 pulsos 2 baterias grandes centradas encoder derecha


NUMERO DE INTERRUPCIONES
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos 2 baterias grandes centrada encoder izquierda
potencia de 50% con 127 pulsos 2 baterias grandes centrada encoder derecha

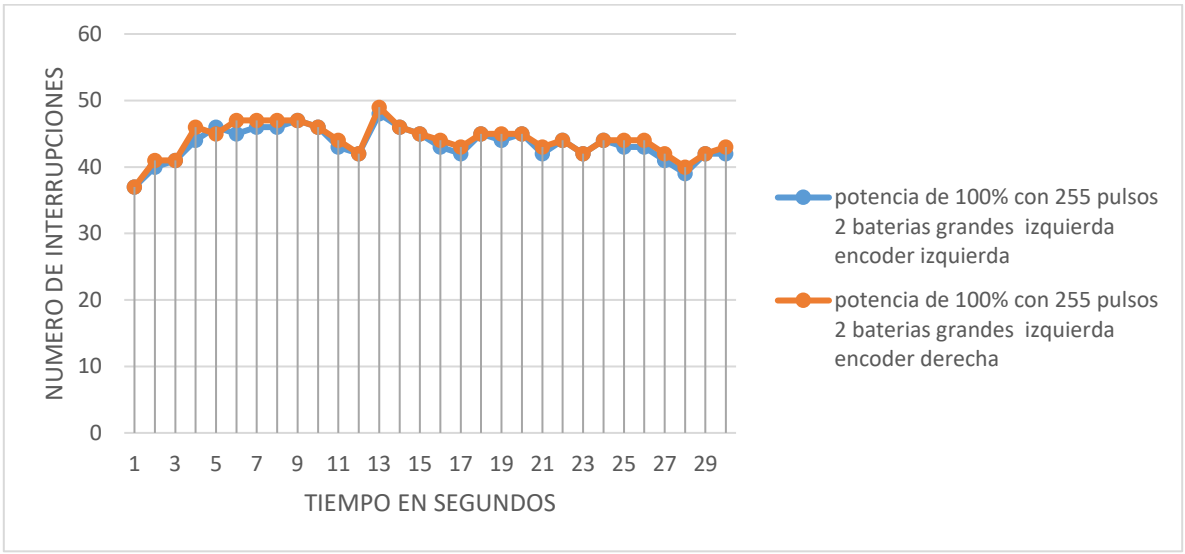
Pruebas de velocidad usando Llantas de tracción, con carga de 5k lado izquierdo

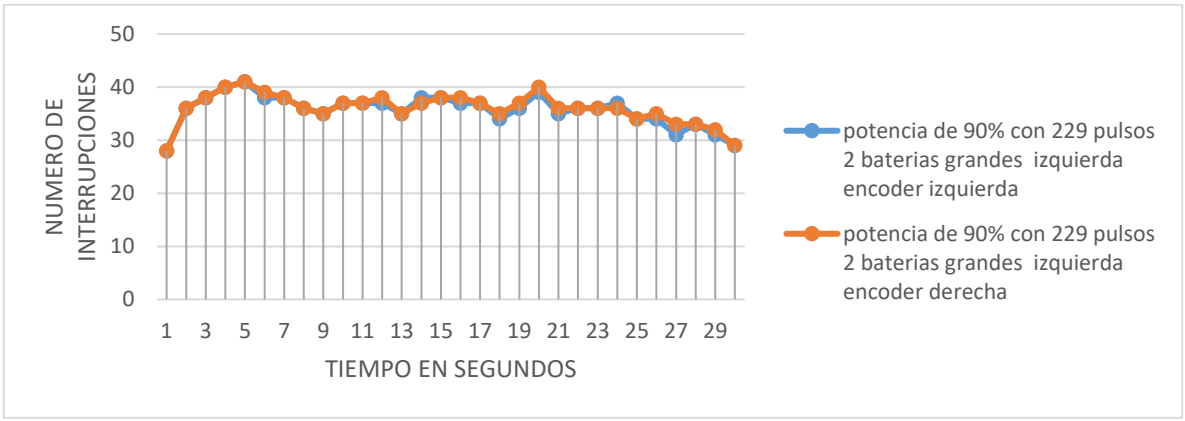


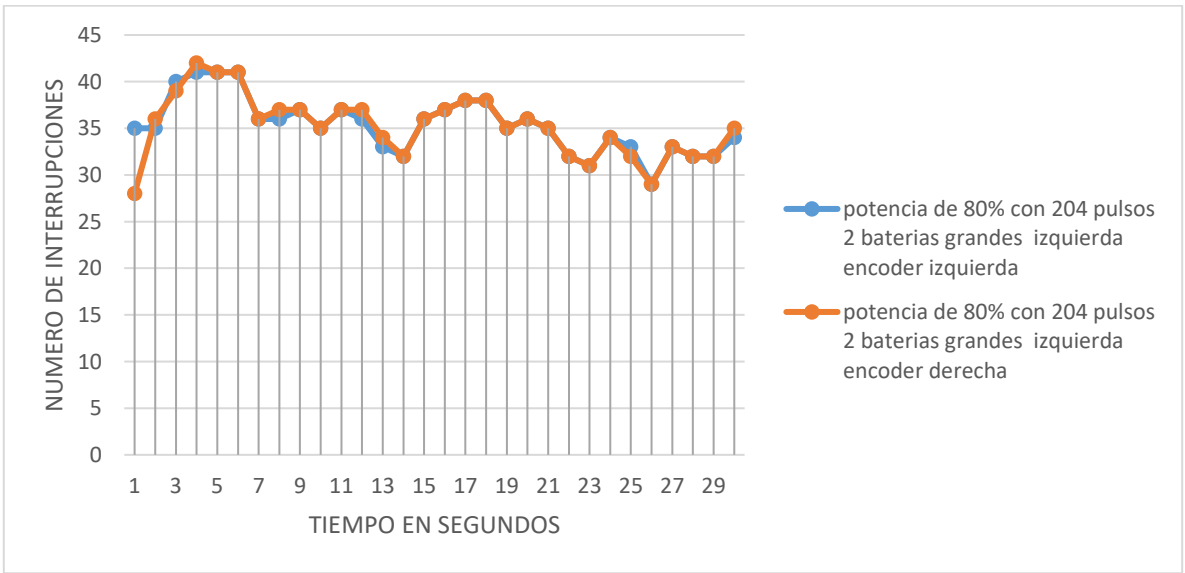




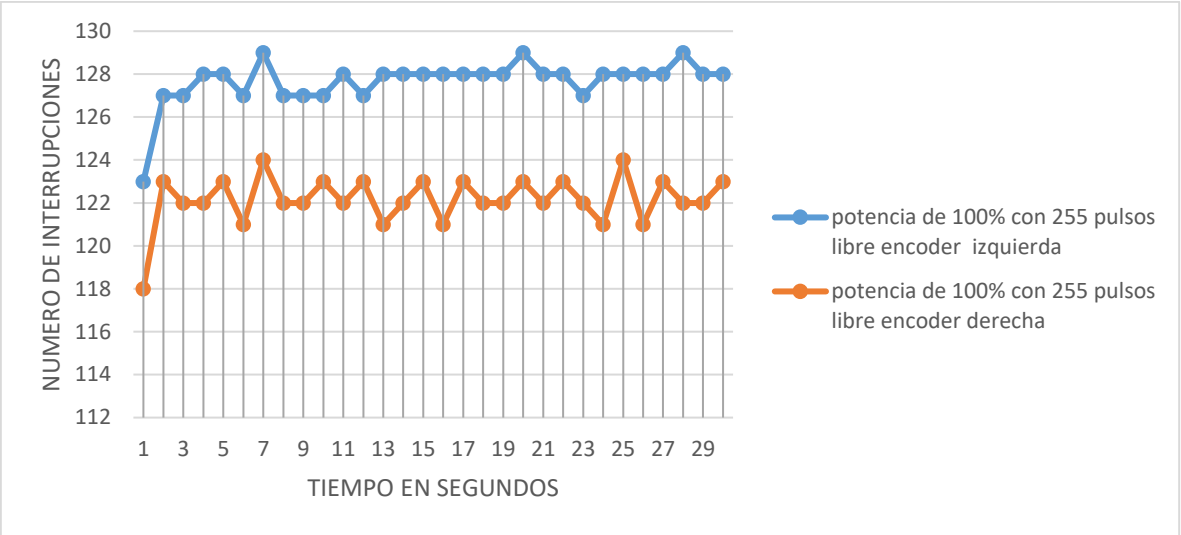
Pruebas de velocidad usando Llantas de tracción, con carga de 5k lado derecho

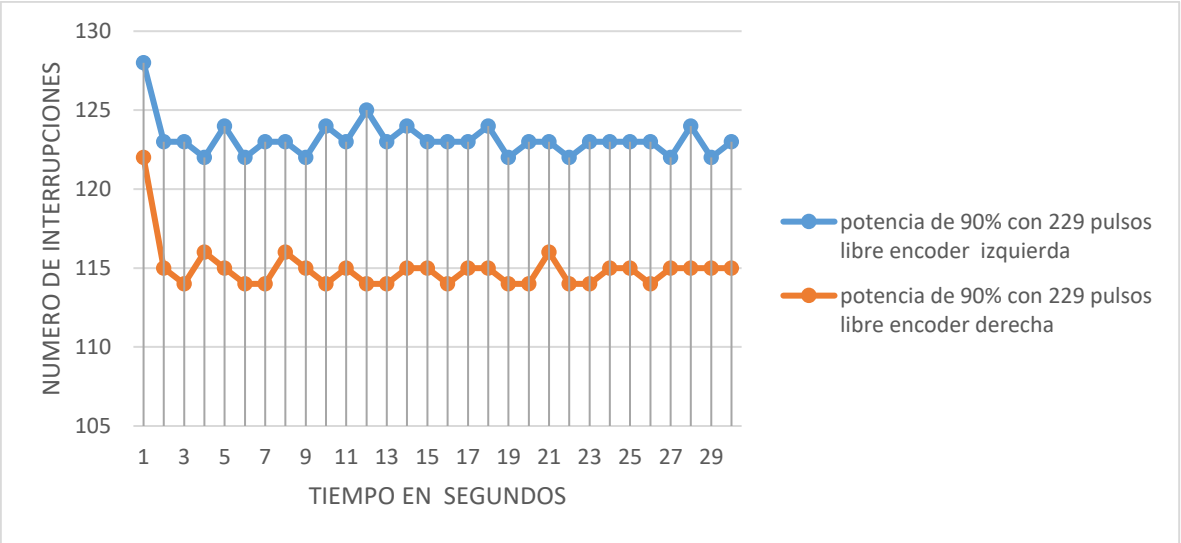


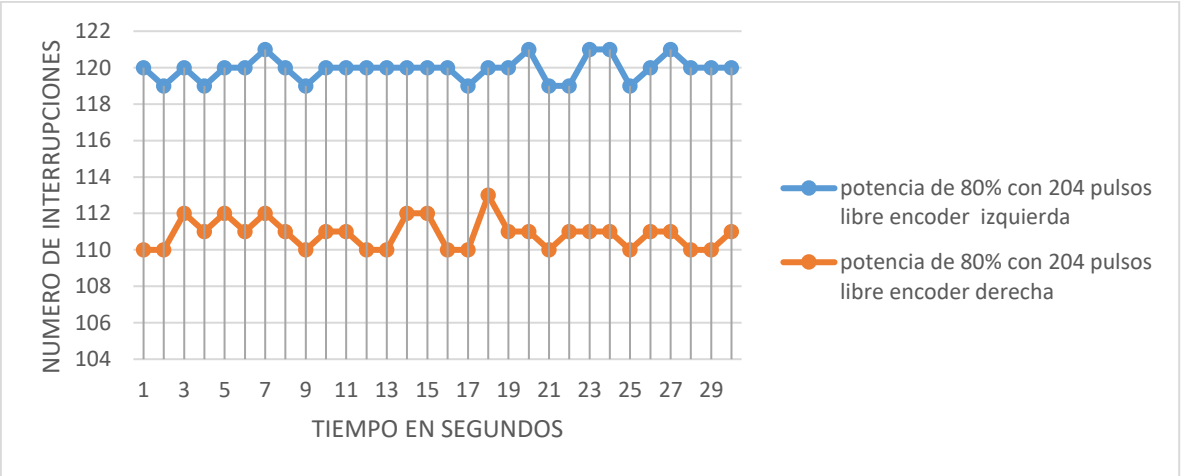


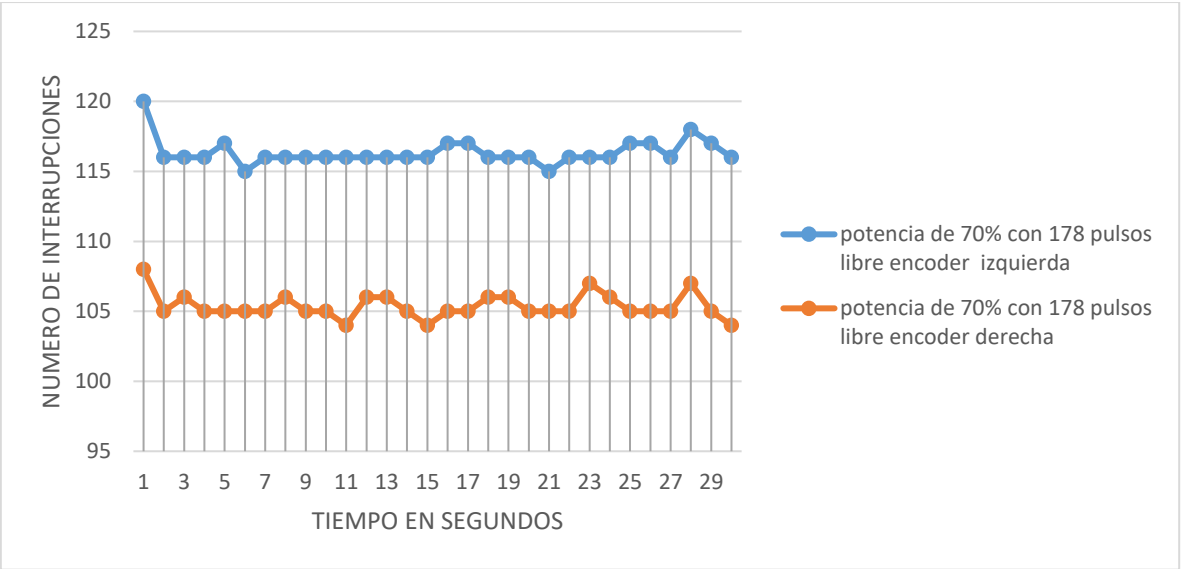


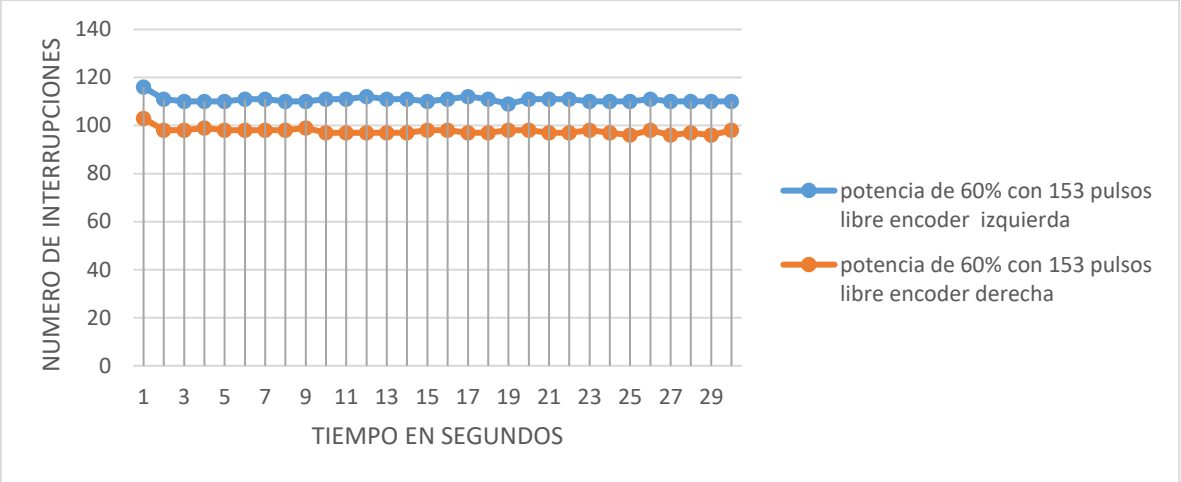
Pruebas de velocidad usando Llantas omnidireccional, sin fricción

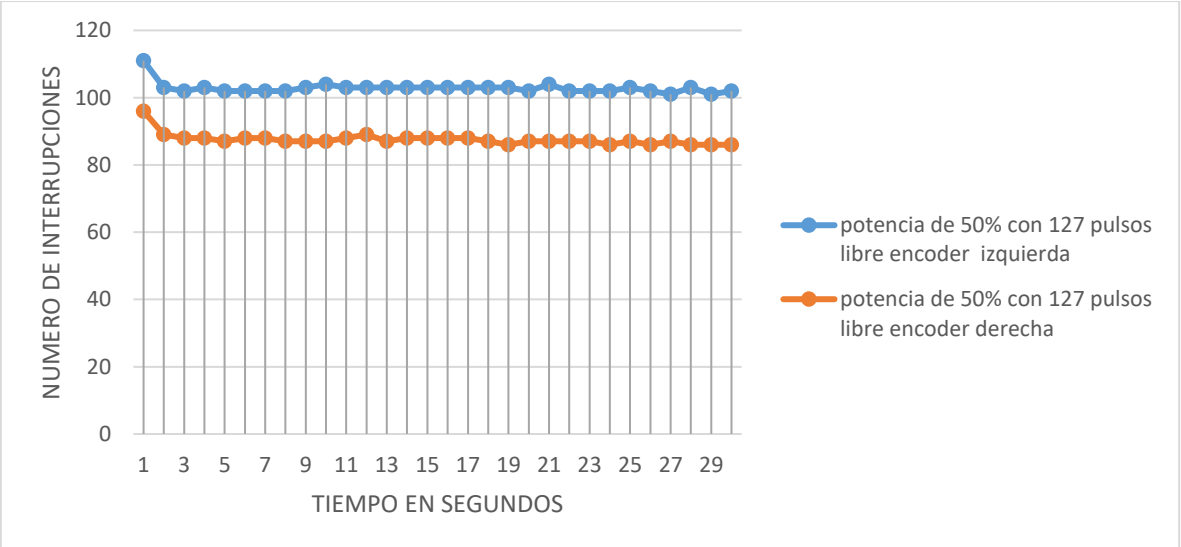


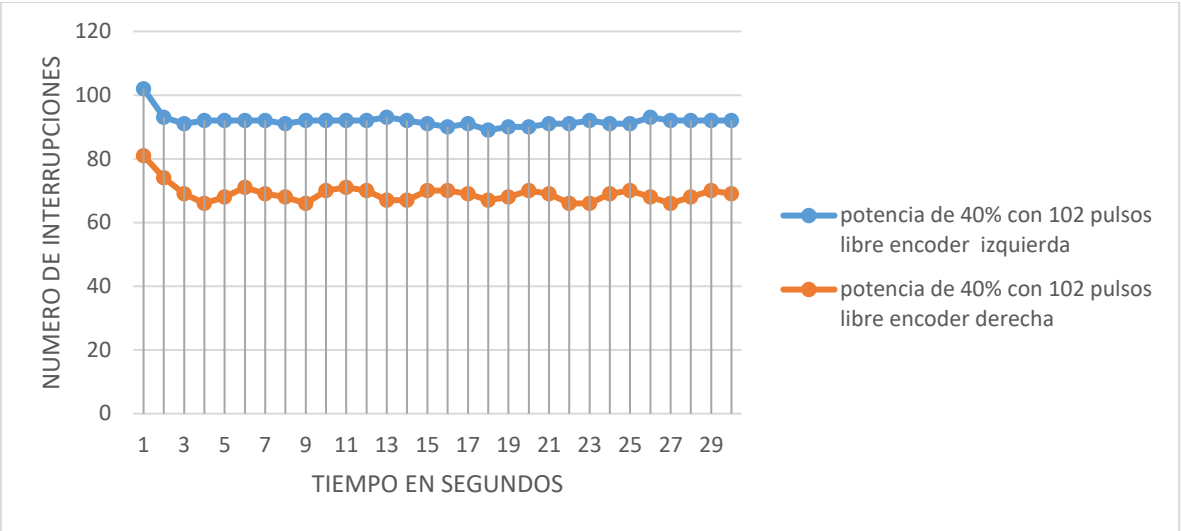


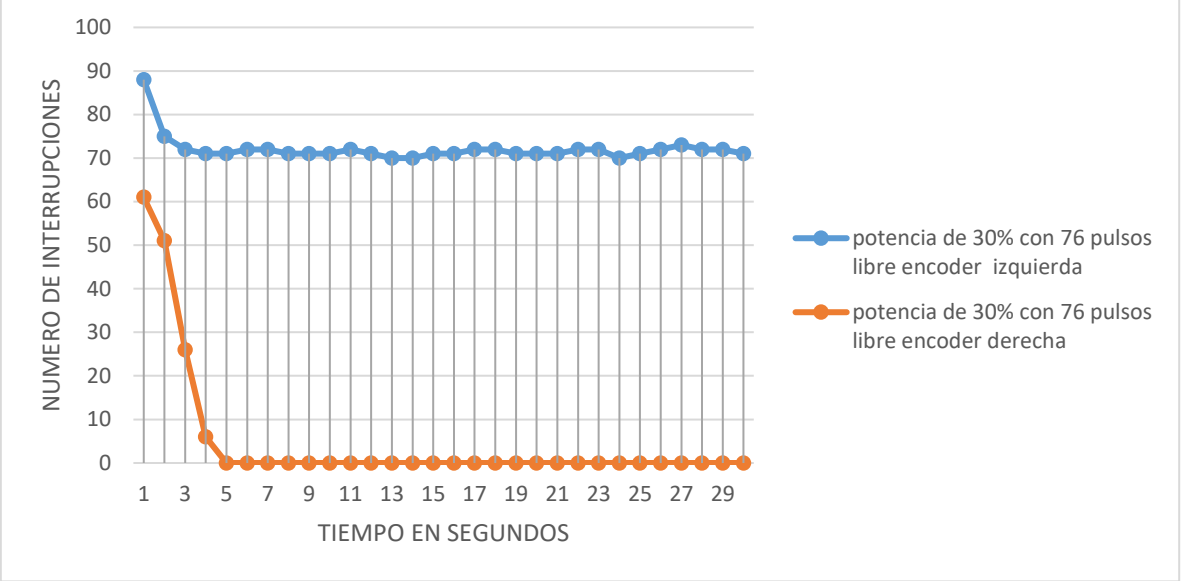


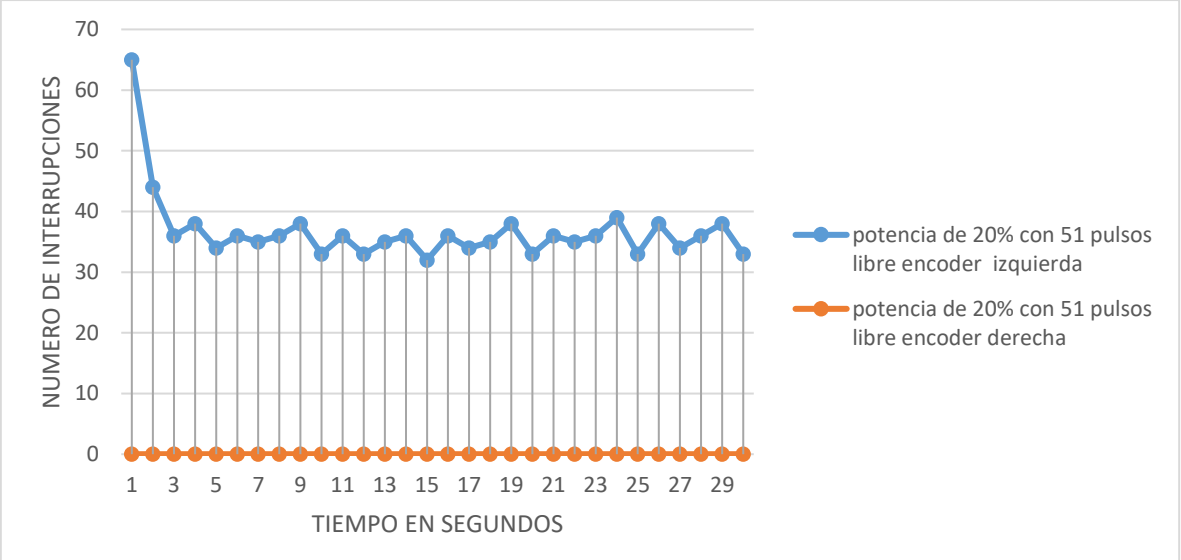

NUMERO DE INTERRUPCIONES
125
120
115
110
105
100
95
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos libre encoder izquierda
potencia de 70% con 178 pulsos libre encoder derecha


NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos libre encoder izquierda
potencia de 60% con 153 pulsos libre encoder derecha

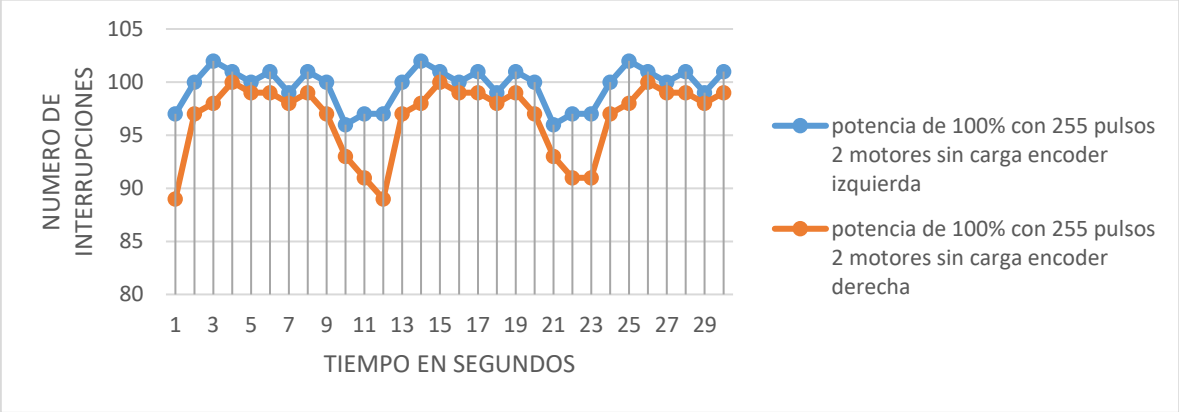

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos libre encoder izquierda
potencia de 50% con 127 pulsos libre encoder derecha

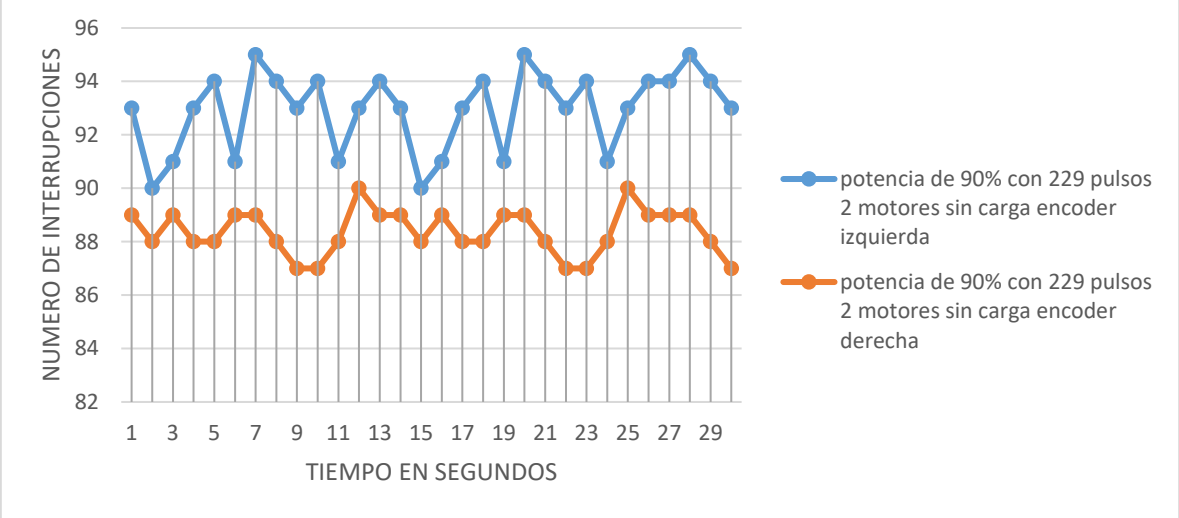

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40% con 102 pulsos libre encoder izquierda
potencia de 40% con 102 pulsos libre encoder derecha

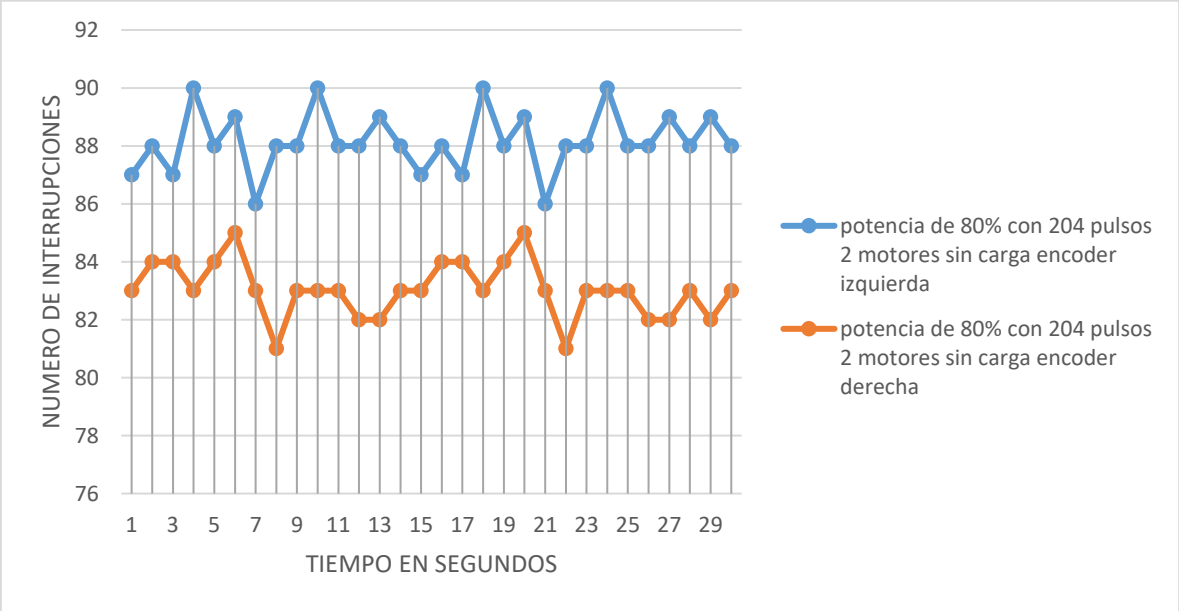

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30% con 76 pulsos libre encoder izquierda
potencia de 30% con 76 pulsos libre encoder derecha

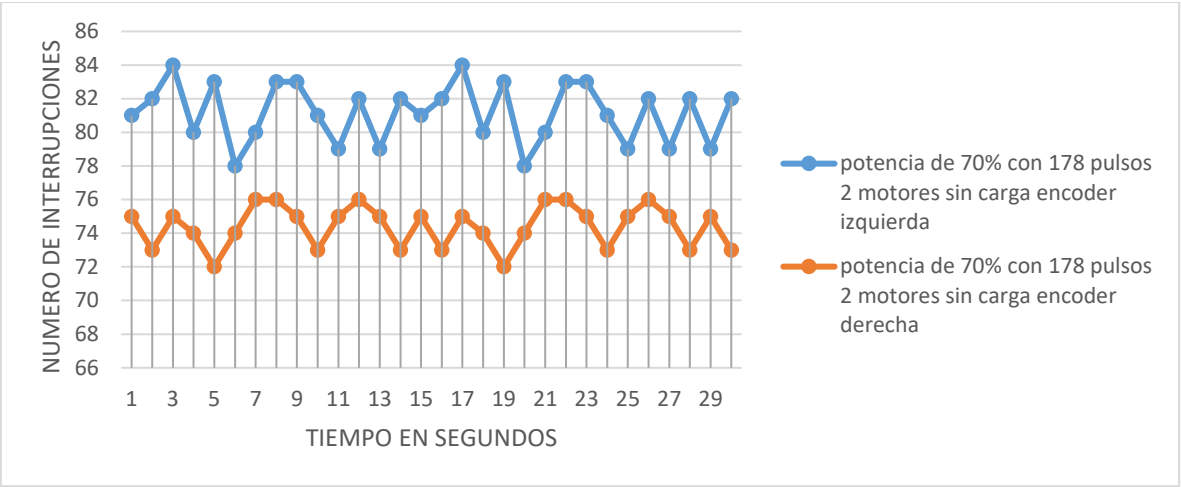

NUMERO DE INTERRUPCIONES
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 20% con 51 pulsos libre encoder izquierda
potencia de 20% con 51 pulsos libre encoder derecha

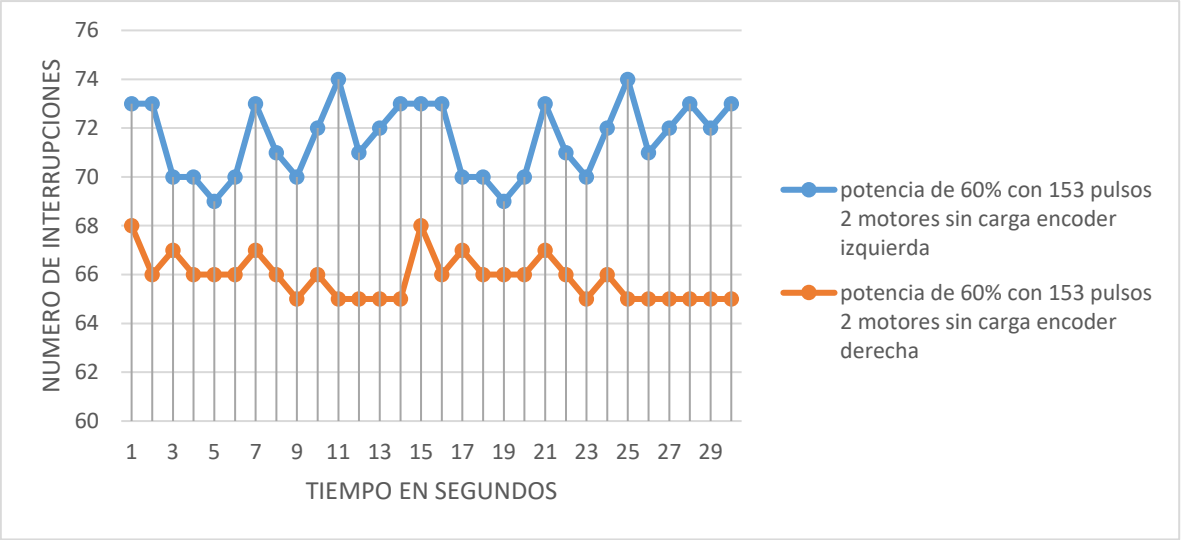
Pruebas de velocidad usando Llantas omnidireccional, con dos motores sin carga

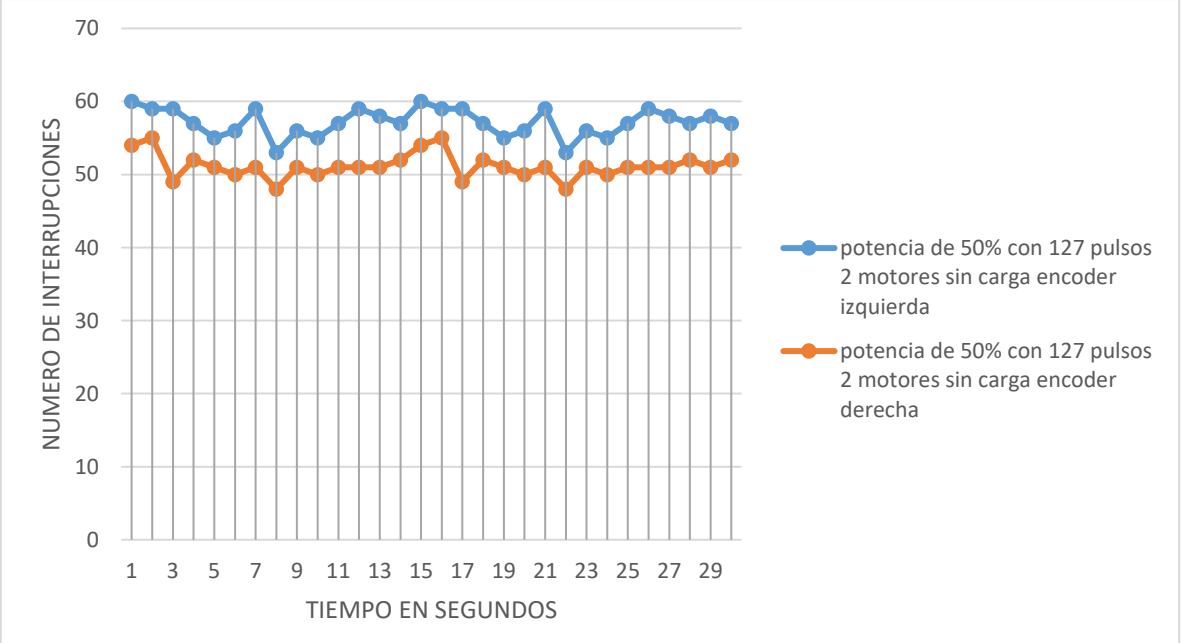


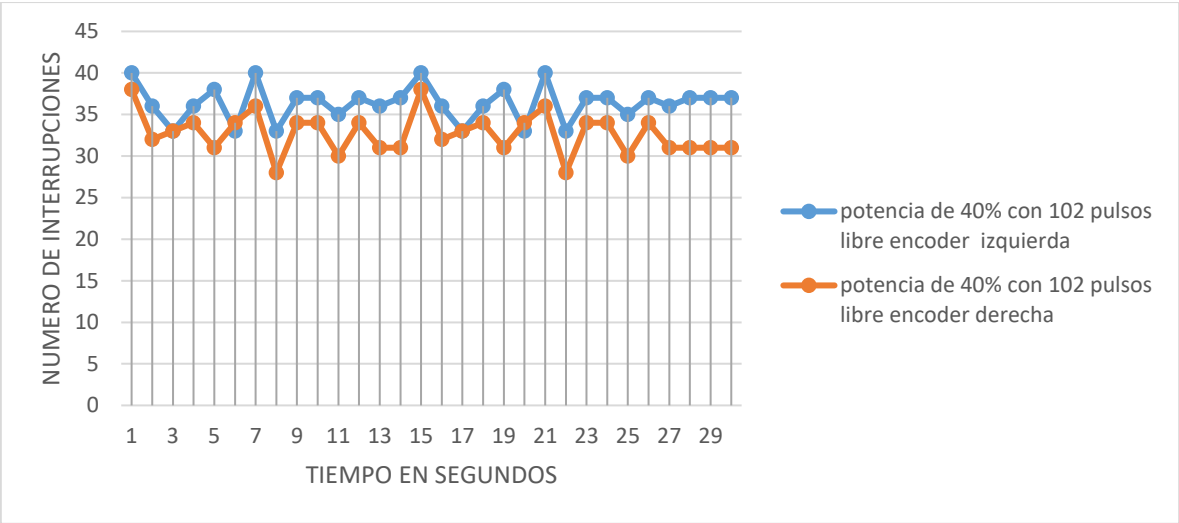




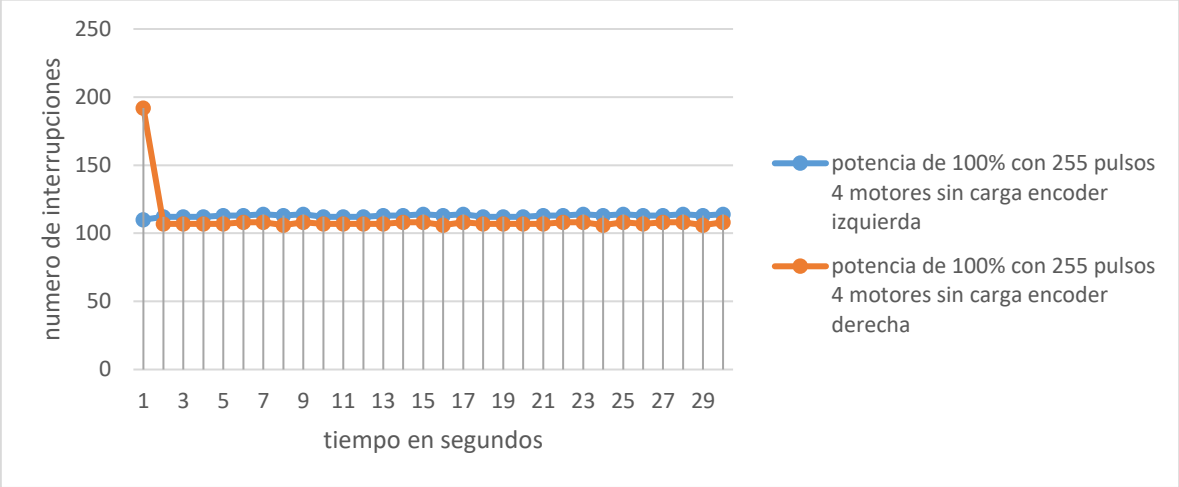

NUMERO DE INTERRUPCIONES
86
84
82
80
78
76
74
72
70
68
66
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70% con 178 pulsos 2 motores sin carga encoder izquierda
potencia de 70% con 178 pulsos 2 motores sin carga encoder derecha

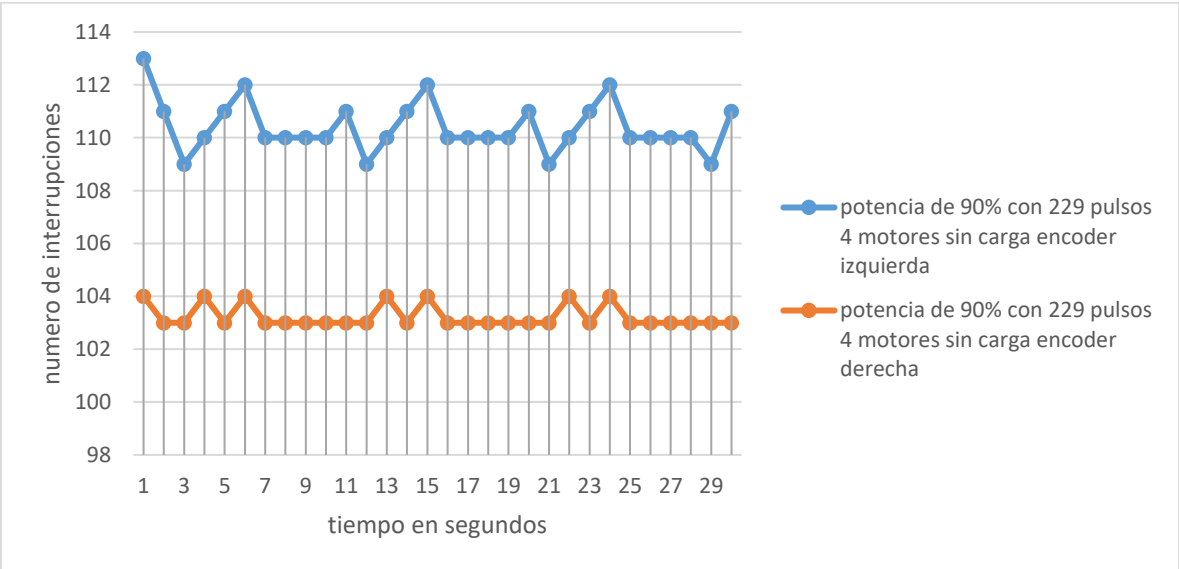

NUMERO DE INTERRUPCIONES
76
74
72
70
68
66
64
62
60
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60% con 153 pulsos 2 motores sin carga encoder izquierda
potencia de 60% con 153 pulsos 2 motores sin carga encoder derecha

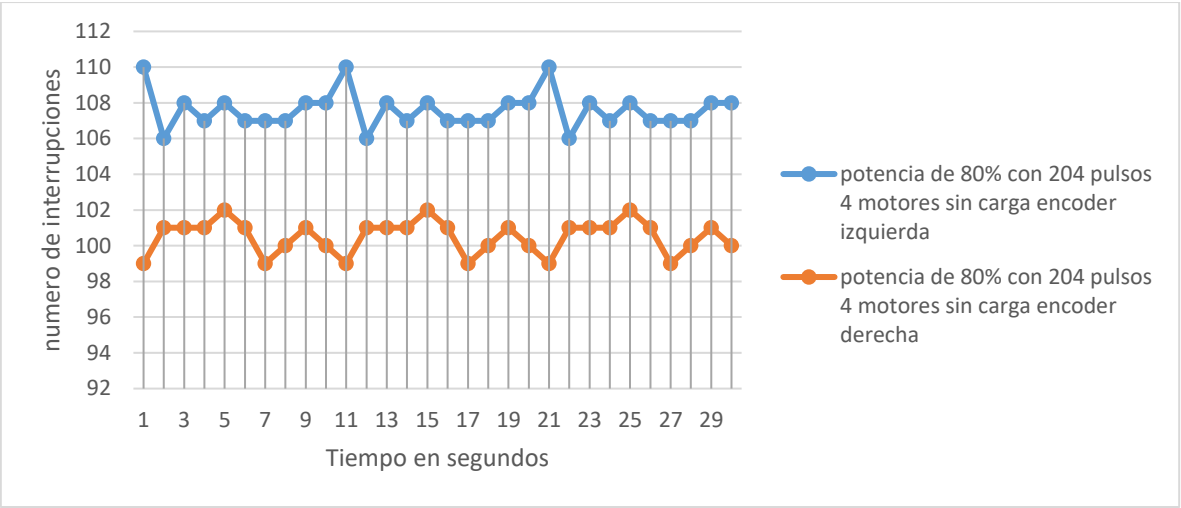

NUMERO DE INTERRUPCIONES
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50% con 127 pulsos 2 motores sin carga encoder izquierda
potencia de 50% con 127 pulsos 2 motores sin carga encoder derecha

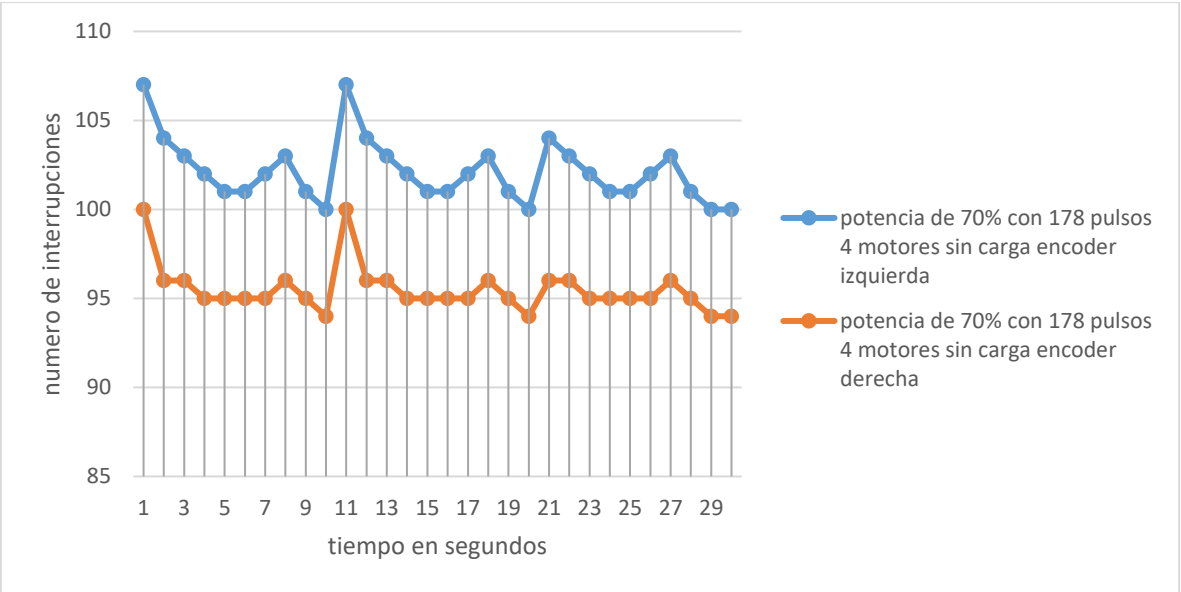


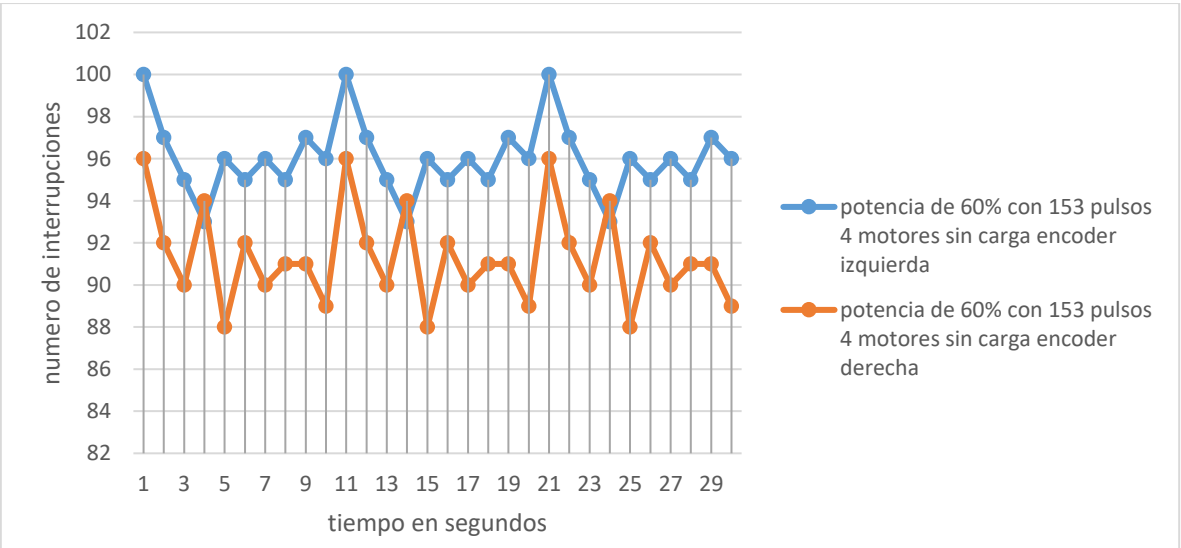
Pruebas de velocidad usando Llantas omnidireccional, con cuatro motores sin carga

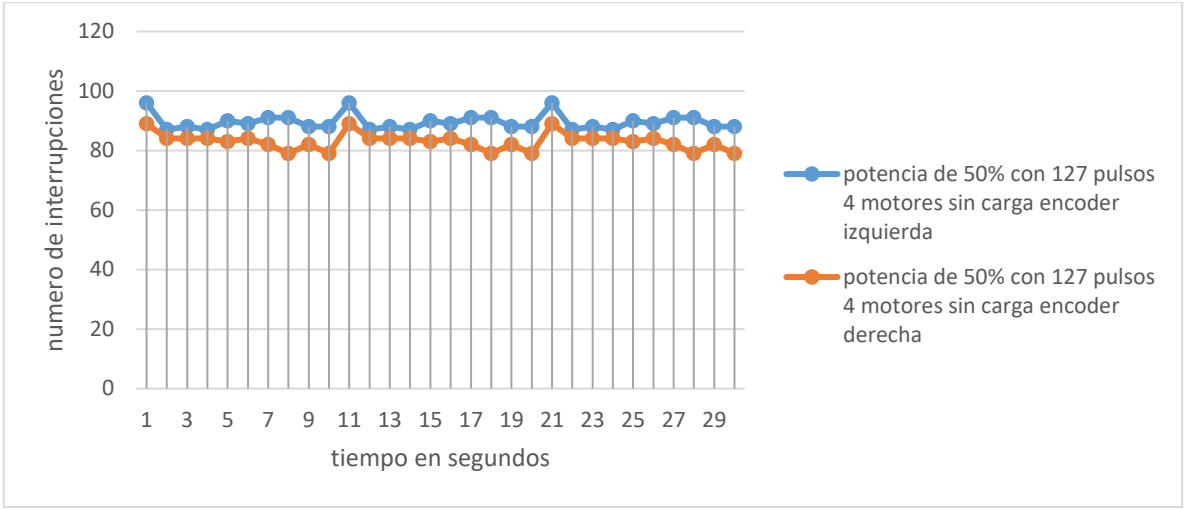


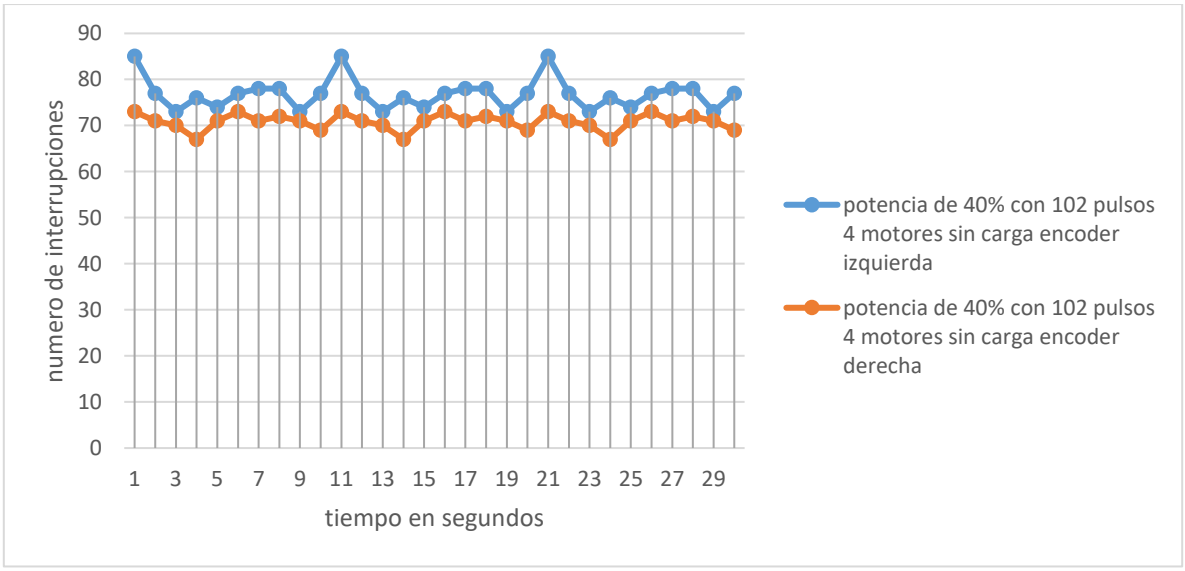


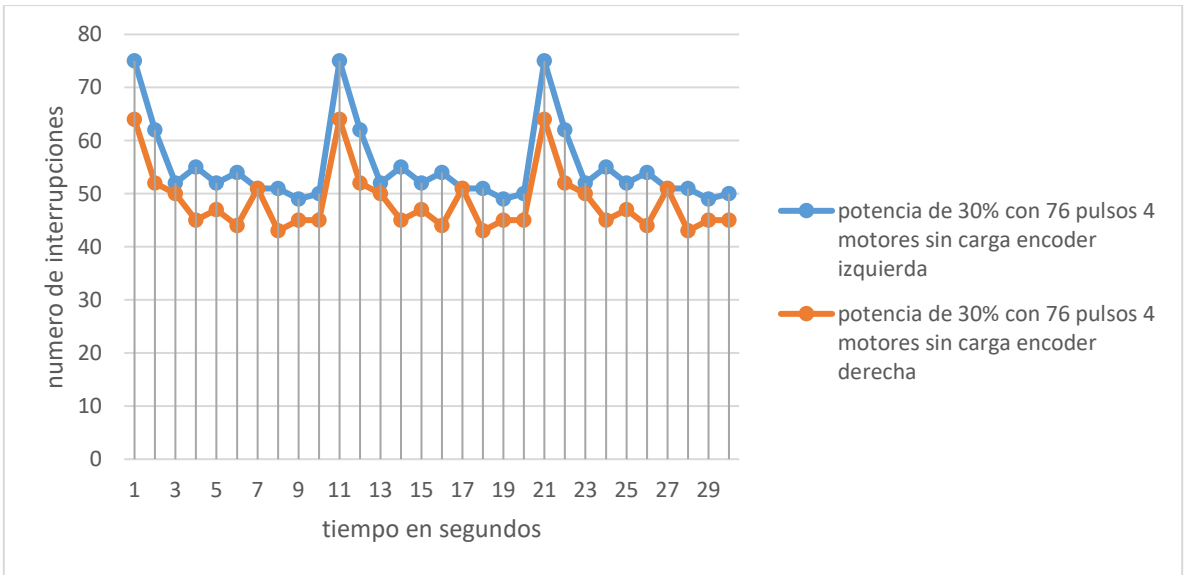

numero de interrupciones
112
110
108
106
104
102
100
98
96
94
92
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Tiempo en segundos
potencia de 80% con 204 pulsos 4 motores sin carga encoder izquierda
potencia de 80% con 204 pulsos 4 motores sin carga encoder derecha


numero de interrupciones
110
105
100
95
90
85
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 70% con 178 pulsos 4 motores sin carga encoder izquierda
potencia de 70% con 178 pulsos 4 motores sin carga encoder derecha

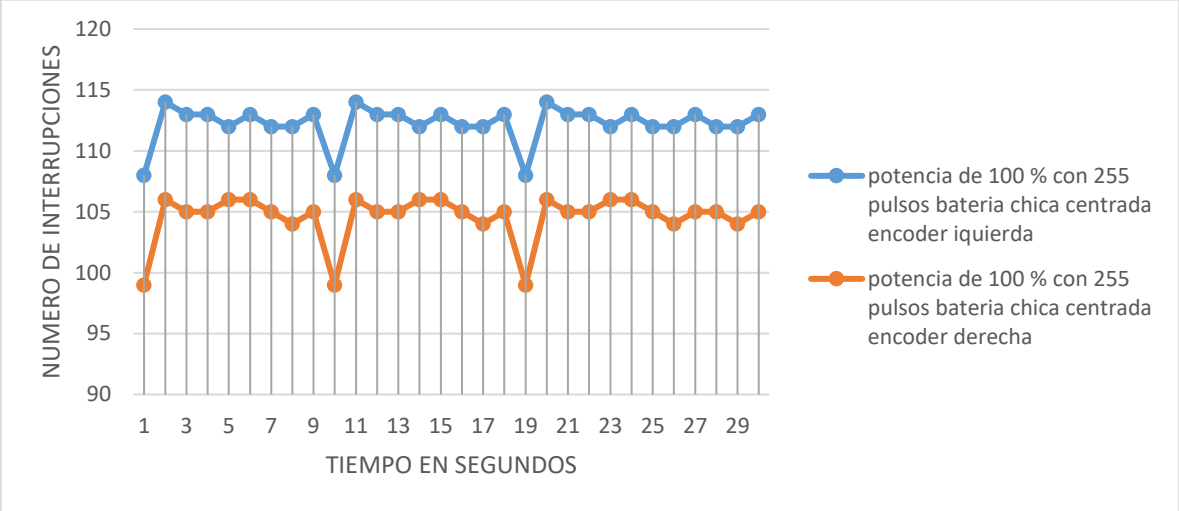

numero de interrupciones
102
100
98
96
94
92
90
88
86
84
82
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 60% con 153 pulsos 4 motores sin carga encoder izquierda
potencia de 60% con 153 pulsos 4 motores sin carga encoder derecha

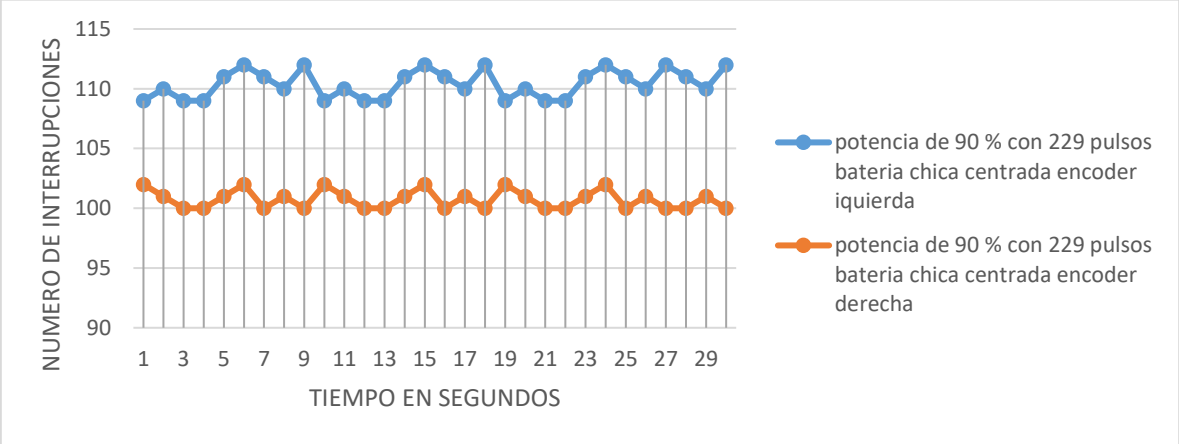

120
100
80
60
40
20
0
numero de interrupciones
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 50% con 127 pulsos 4 motores sin carga encoder izquierda
potencia de 50% con 127 pulsos 4 motores sin carga encoder derecha

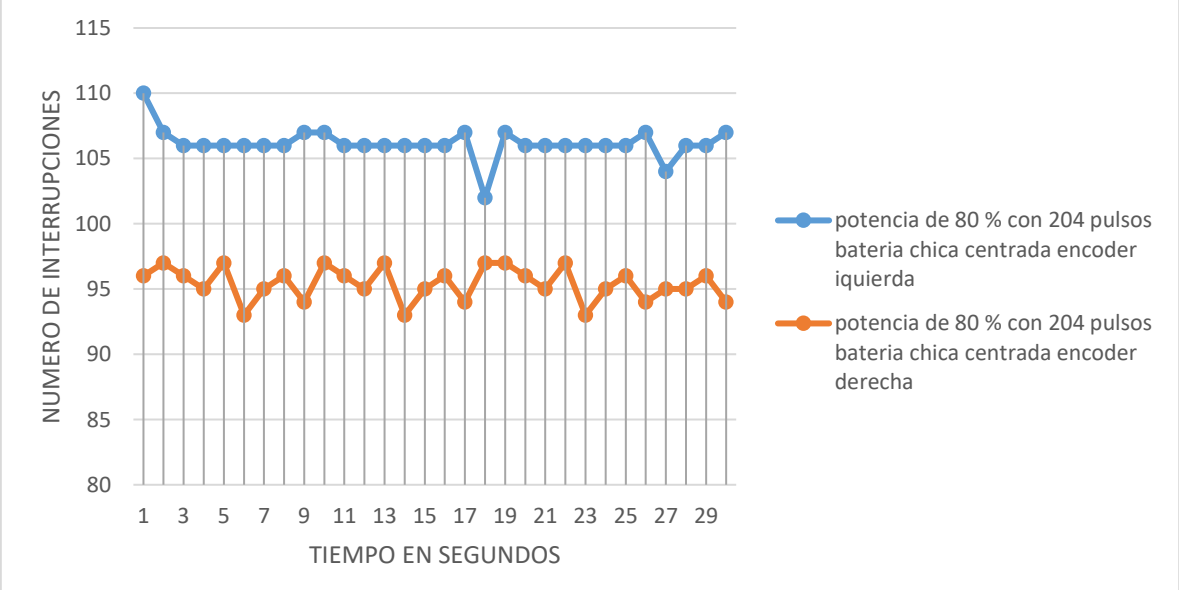

90
80
70
60
50
40
30
20
10
0
numero de interrupciones
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 40% con 102 pulsos 4 motores sin carga encoder izquierda
potencia de 40% con 102 pulsos 4 motores sin carga encoder derecha

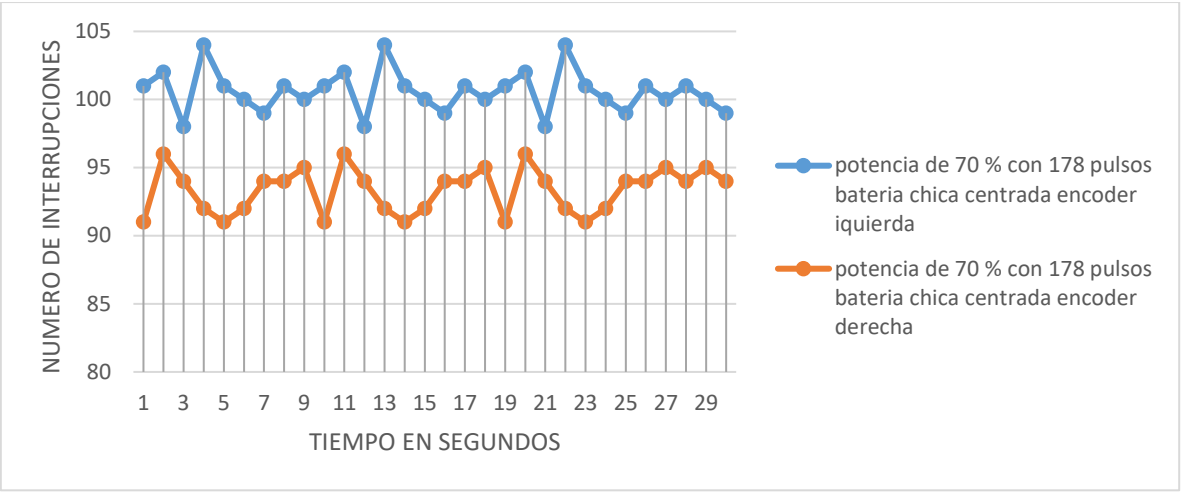

80
70
60
50
40
30
20
10
0
numero de interrupciones
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
tiempo en segundos
potencia de 30% con 76 pulsos 4 motores sin carga encoder izquierda
potencia de 30% con 76 pulsos 4 motores sin carga encoder derecha

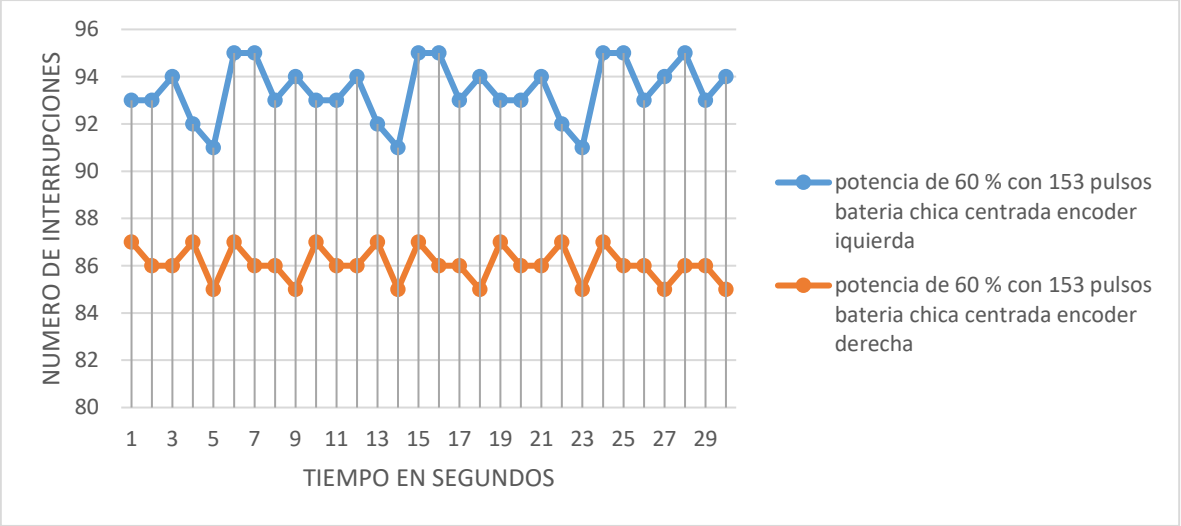
Pruebas de velocidad usando Llantas omnidireccional, con carga de 1.7k centrada

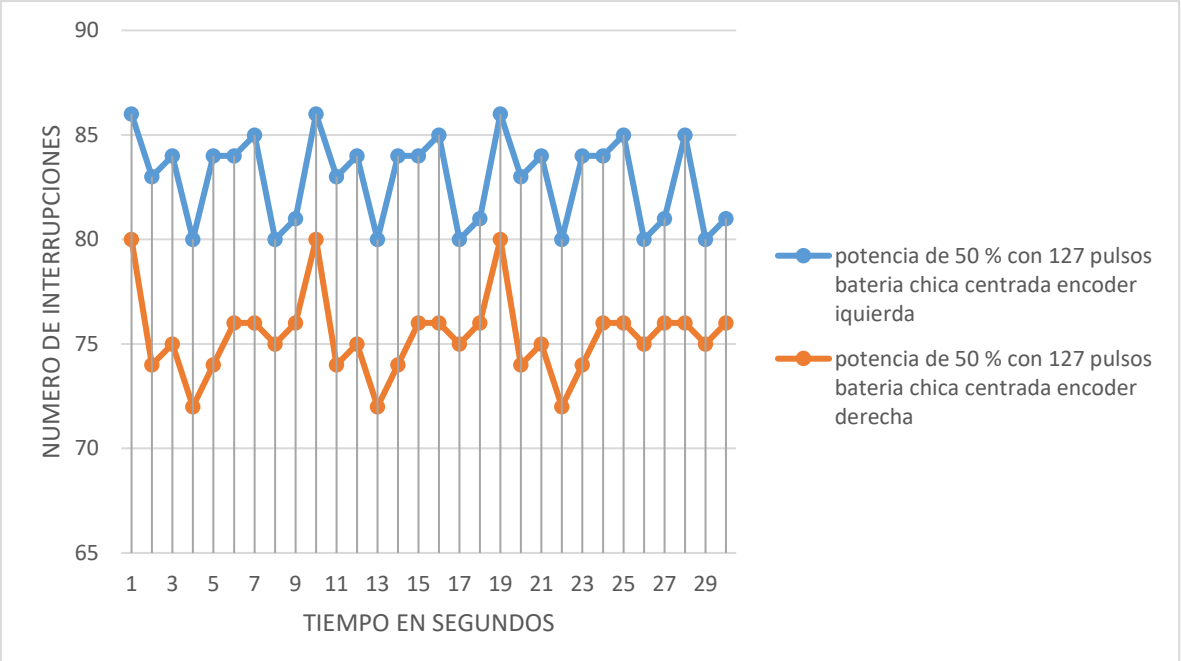


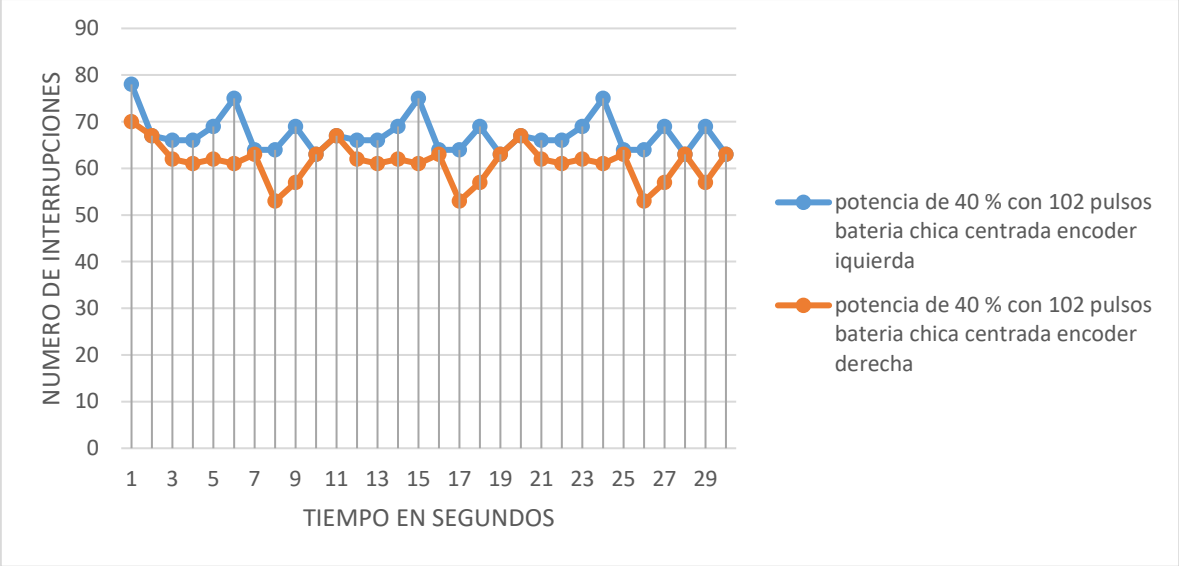


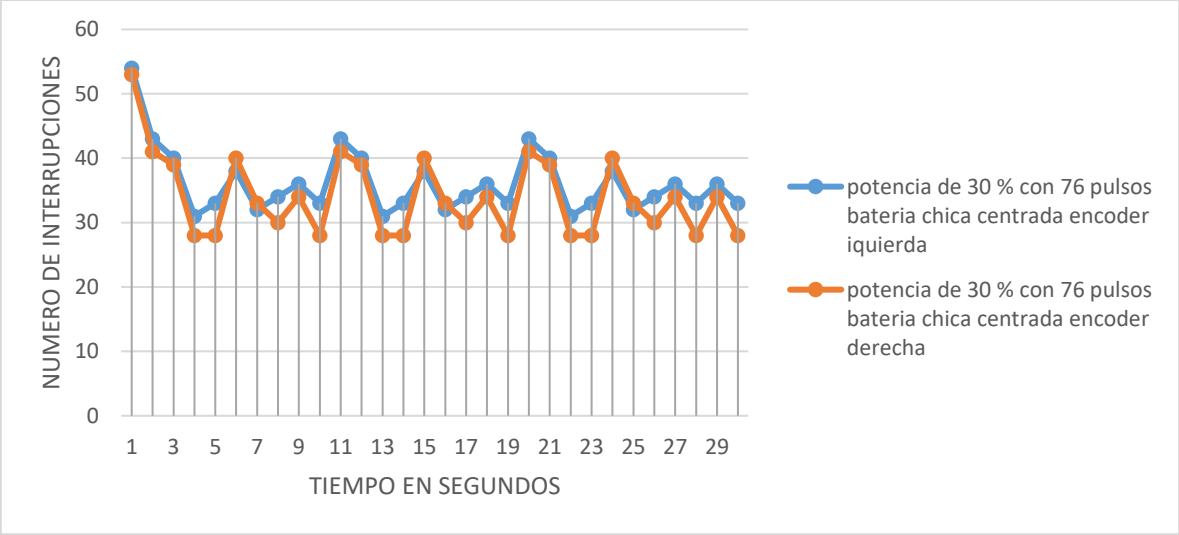



NUMERO DE INTERRUPCIONES
105
100
95
90
85
80
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica centrada encoder iquierda
potencia de 70 % con 178 pulsos bateria chica centrada encoder derecha

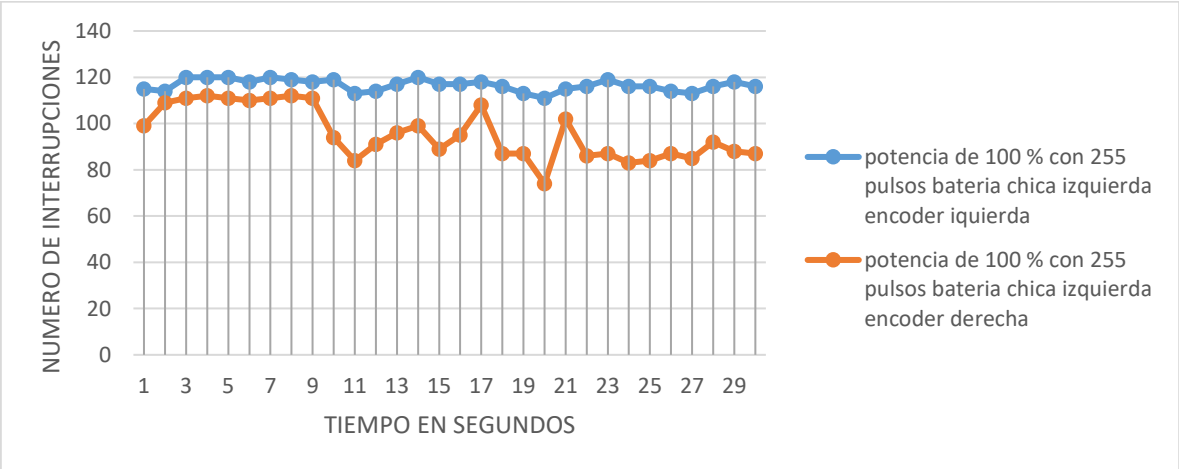

NUMERO DE INTERRUPCIONES
96
94
92
90
88
86
84
82
80
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica centrada encoder iquierda
potencia de 60 % con 153 pulsos bateria chica centrada encoder derecha

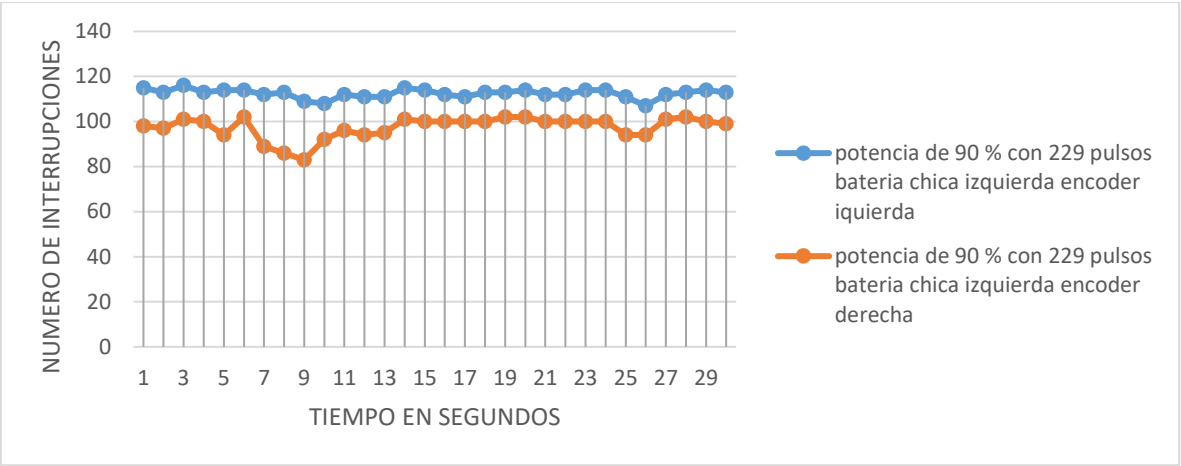

NUMERO DE INTERRUPCIONES
90
85
80
75
70
65
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica centrada encoder iquierda
potencia de 50 % con 127 pulsos bateria chica centrada encoder derecha

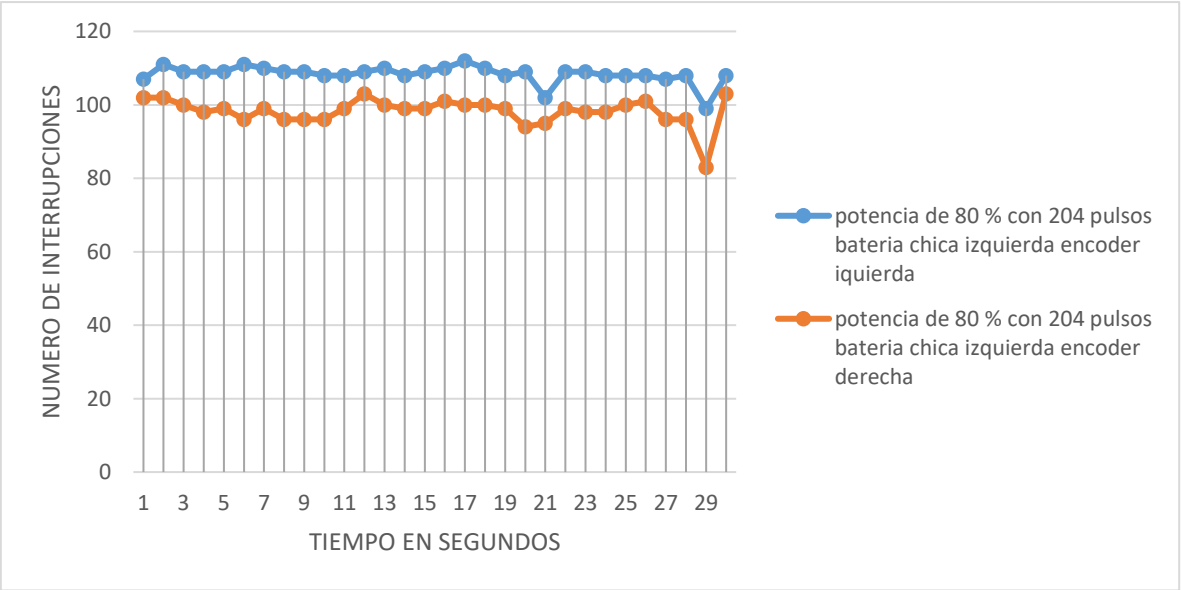


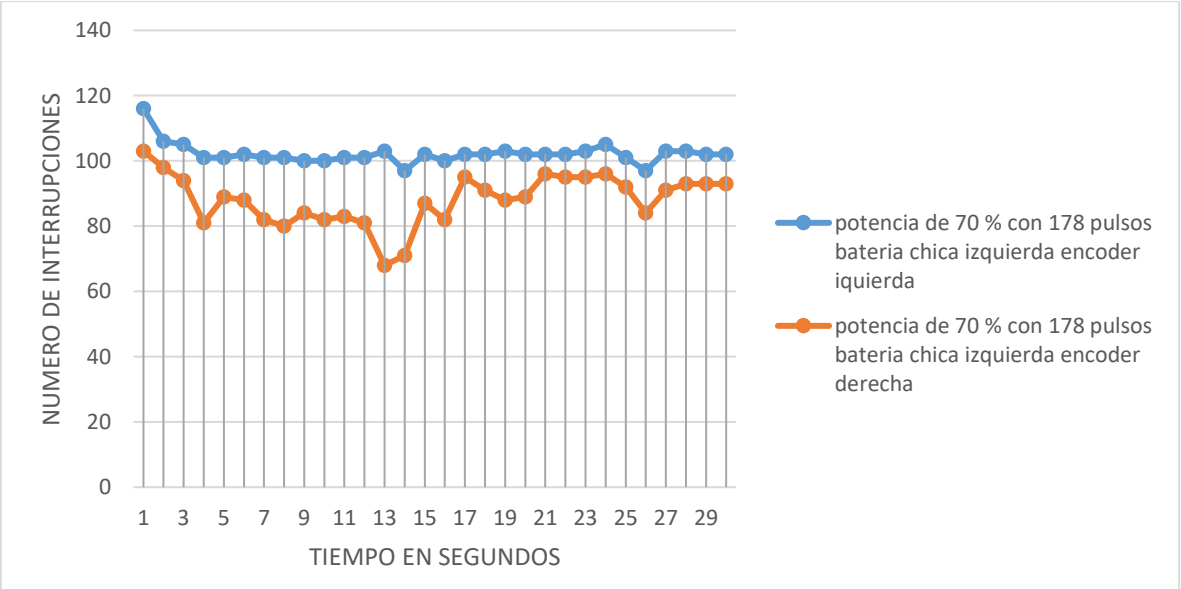


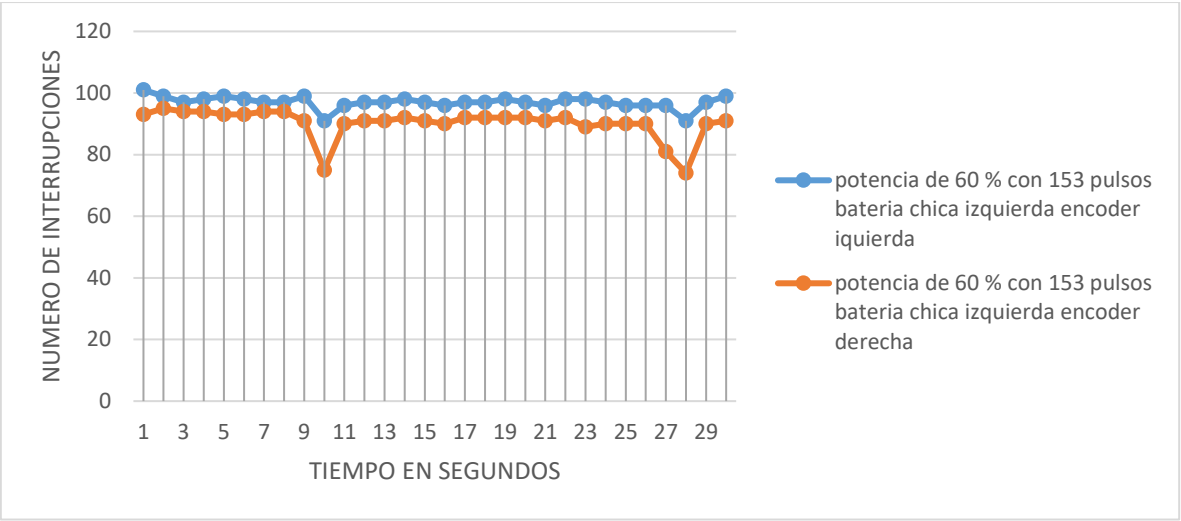
Pruebas de velocidad usando Llantas omnidireccional, con carga de 1.7k lado izquierdo

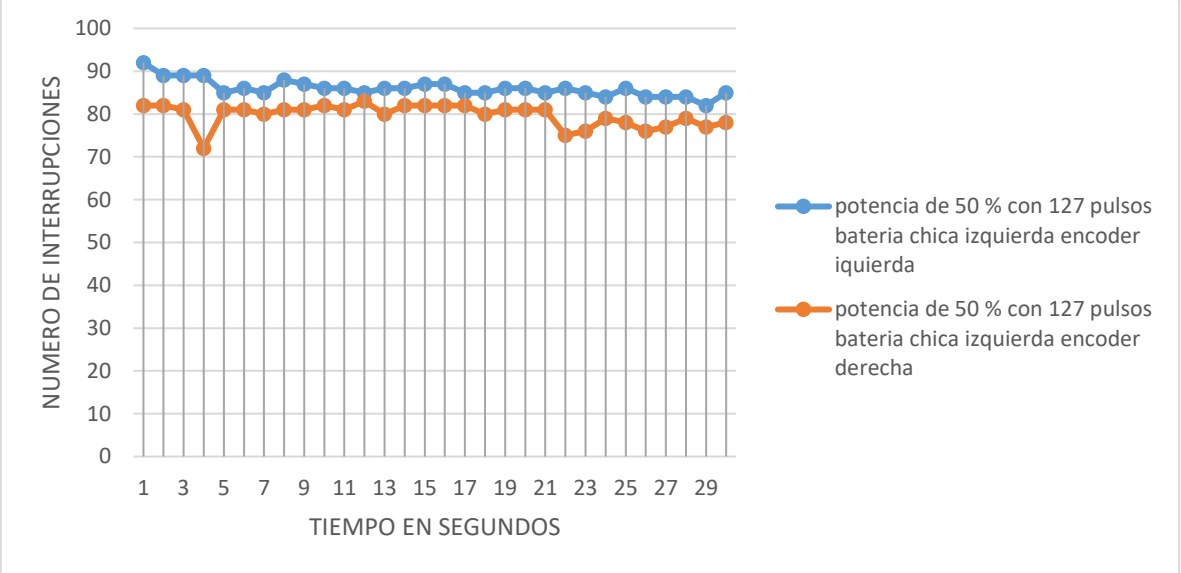


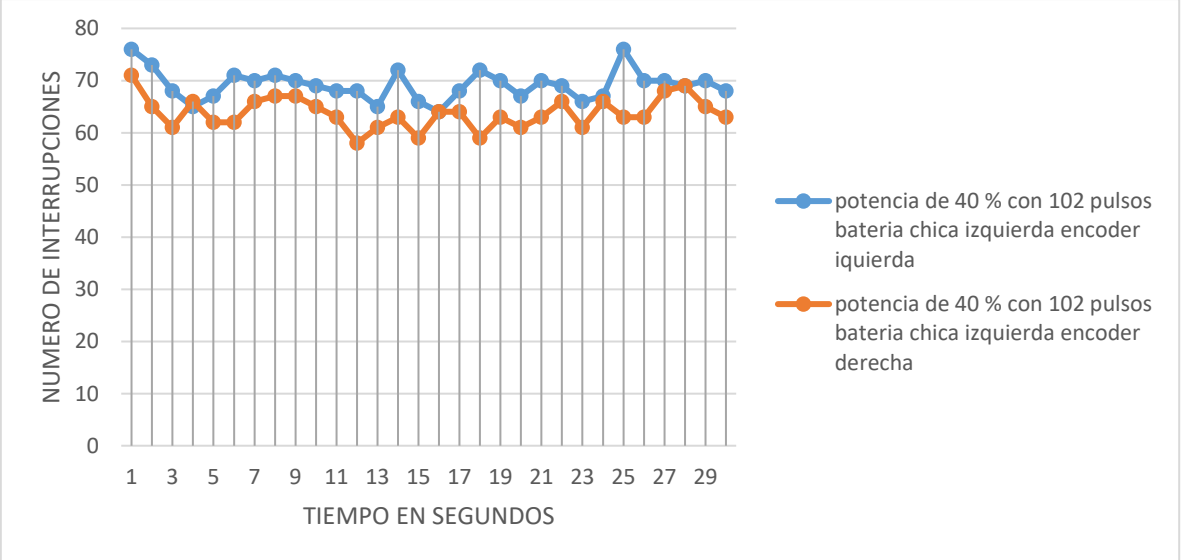

NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 90 % con 229 pulsos bateria chica izquierda encoder iquierda
potencia de 90 % con 229 pulsos bateria chica izquierda encoder derecha

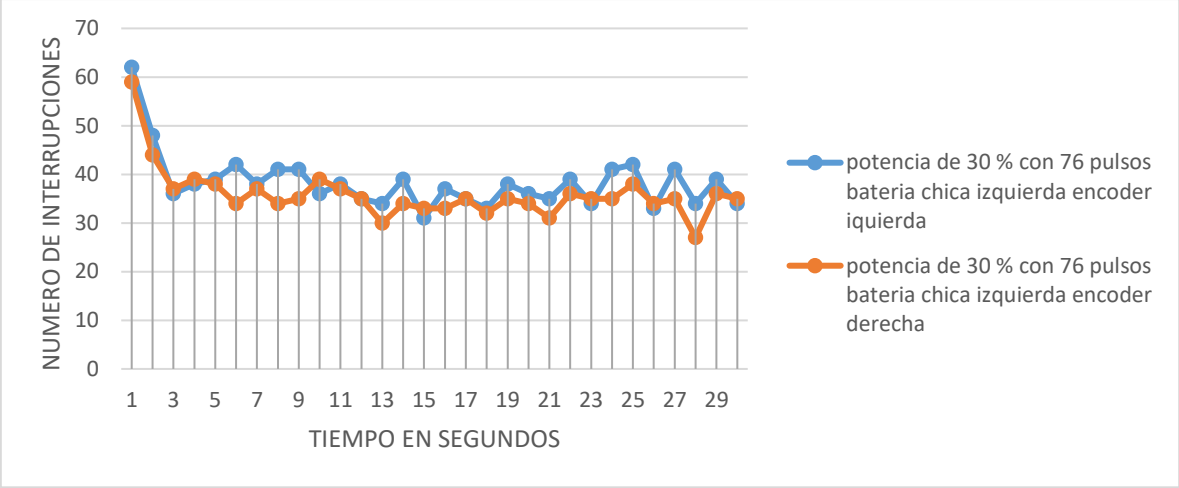

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria chica izquierda encoder iquierda
potencia de 80 % con 204 pulsos bateria chica izquierda encoder derecha


NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica izquierda encoder iquierda
potencia de 70 % con 178 pulsos bateria chica izquierda encoder derecha

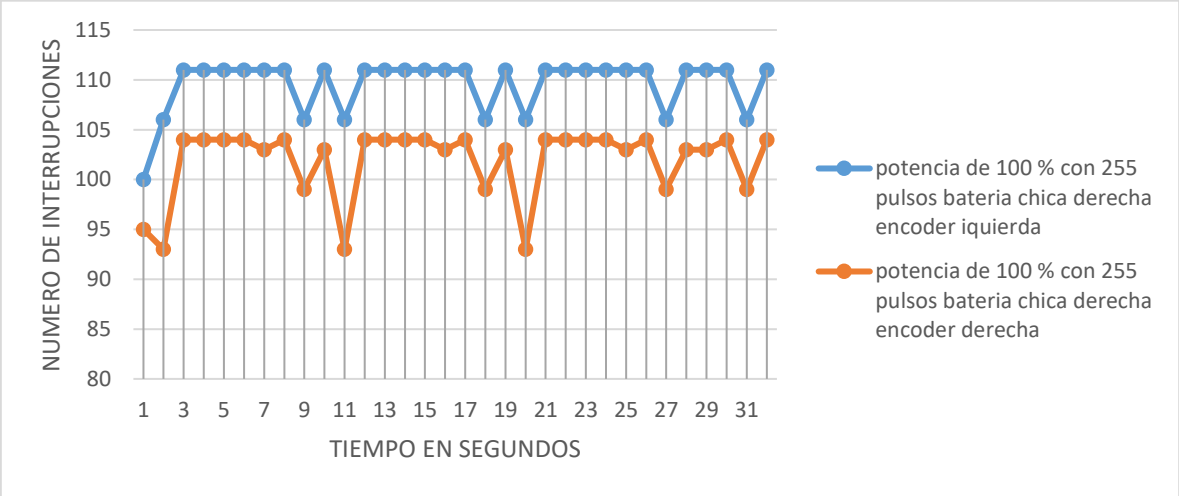

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica izquierda encoder iquierda
potencia de 60 % con 153 pulsos bateria chica izquierda encoder derecha

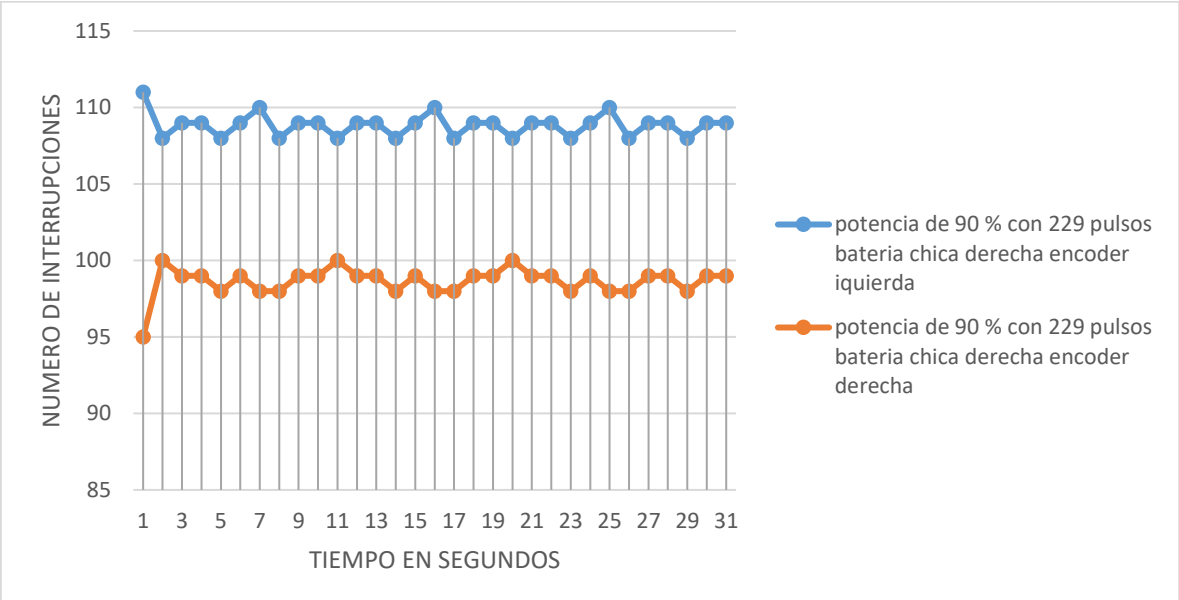

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica izquierda encoder iquierda
potencia de 50 % con 127 pulsos bateria chica izquierda encoder derecha

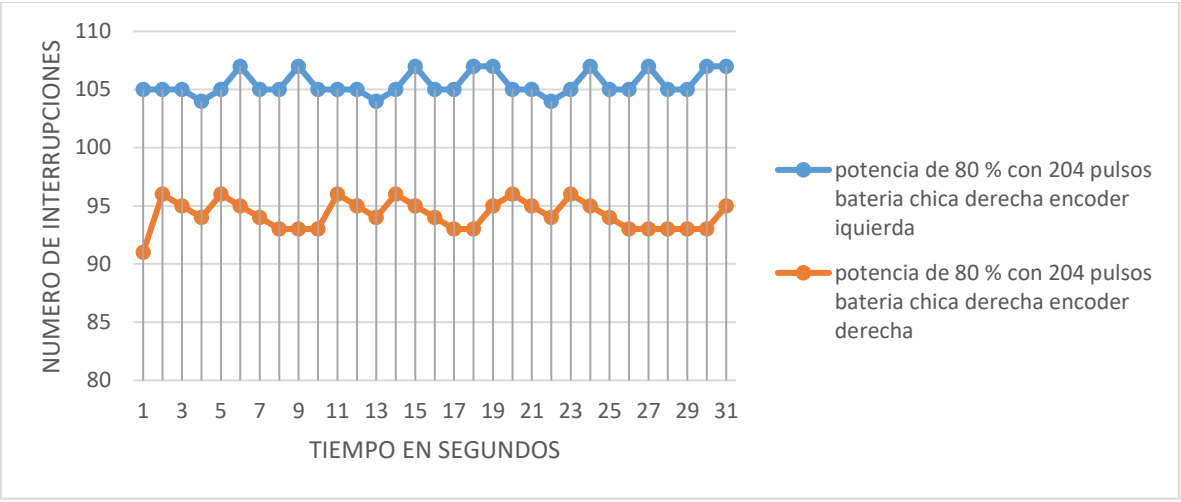

NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria chica izquierda encoder iquierda
potencia de 40 % con 102 pulsos bateria chica izquierda encoder derecha

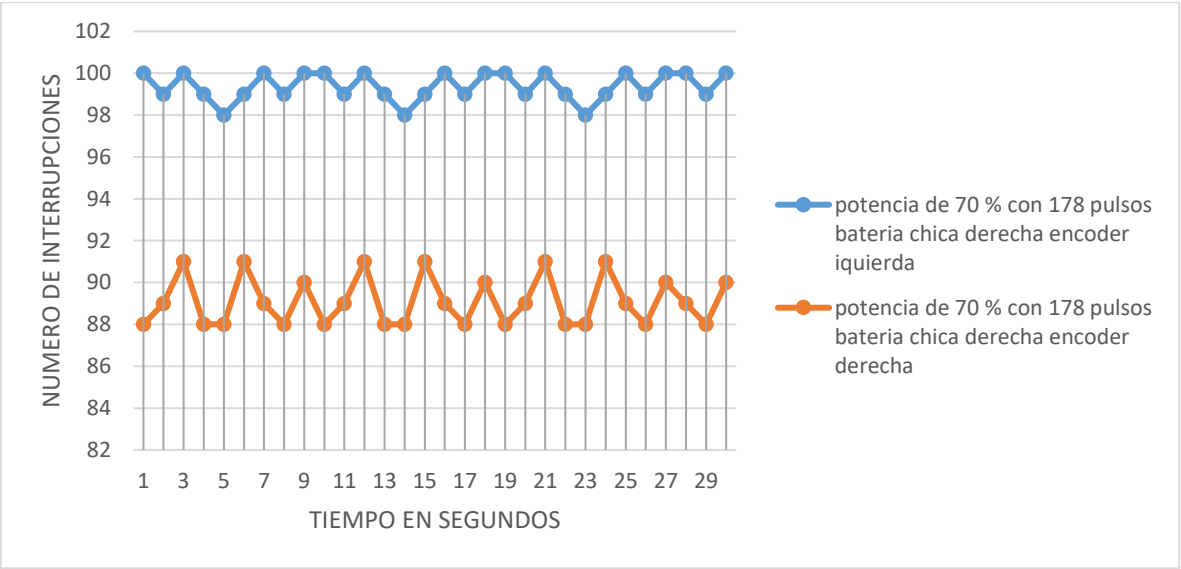


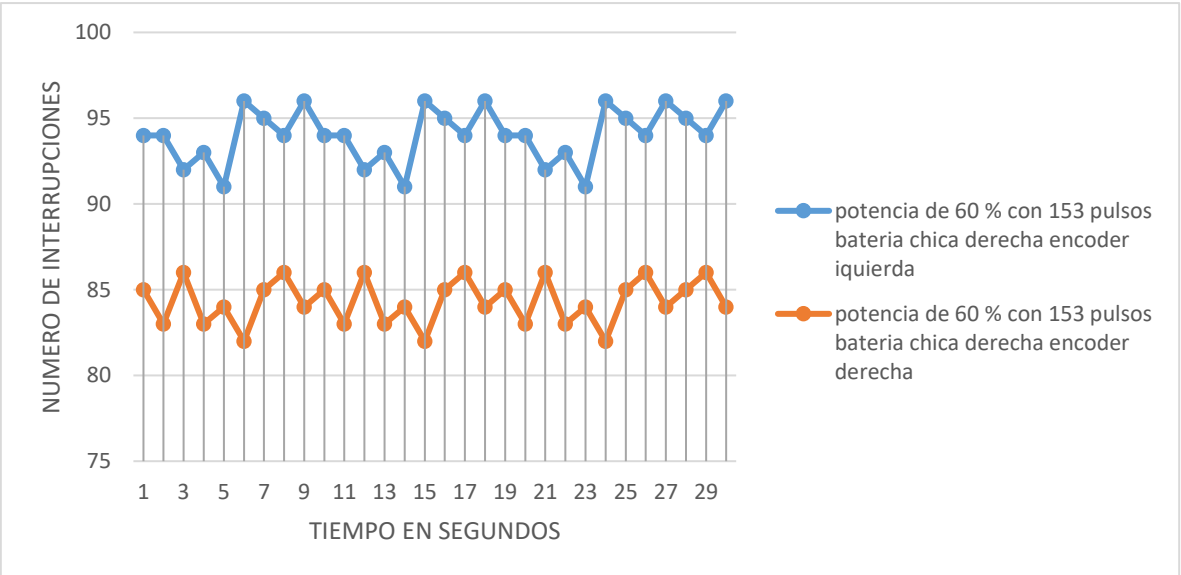
Pruebas de velocidad usando Llantas omnidireccional, con carga de 1.7k lado derecho

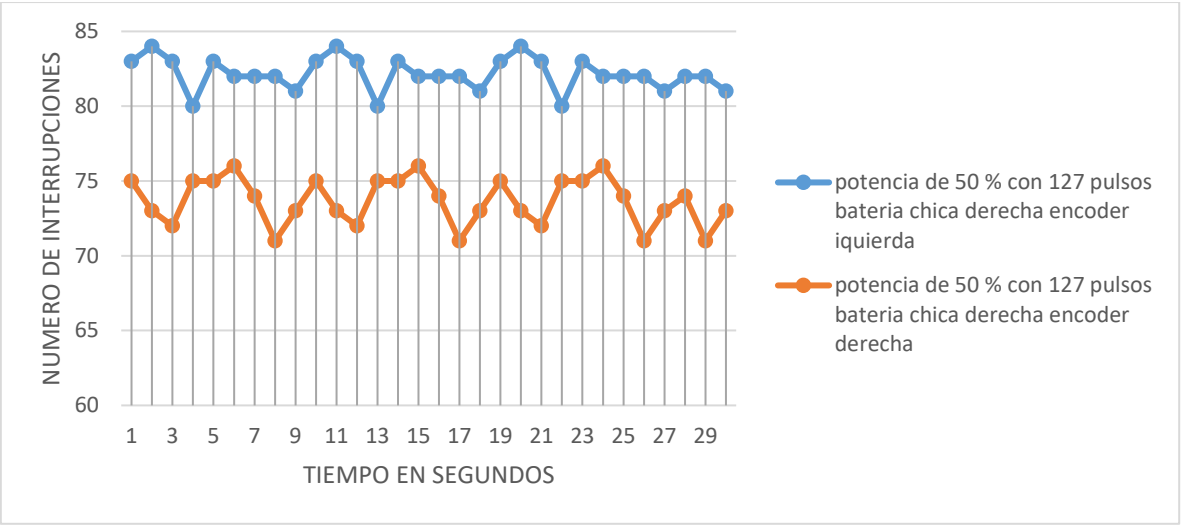


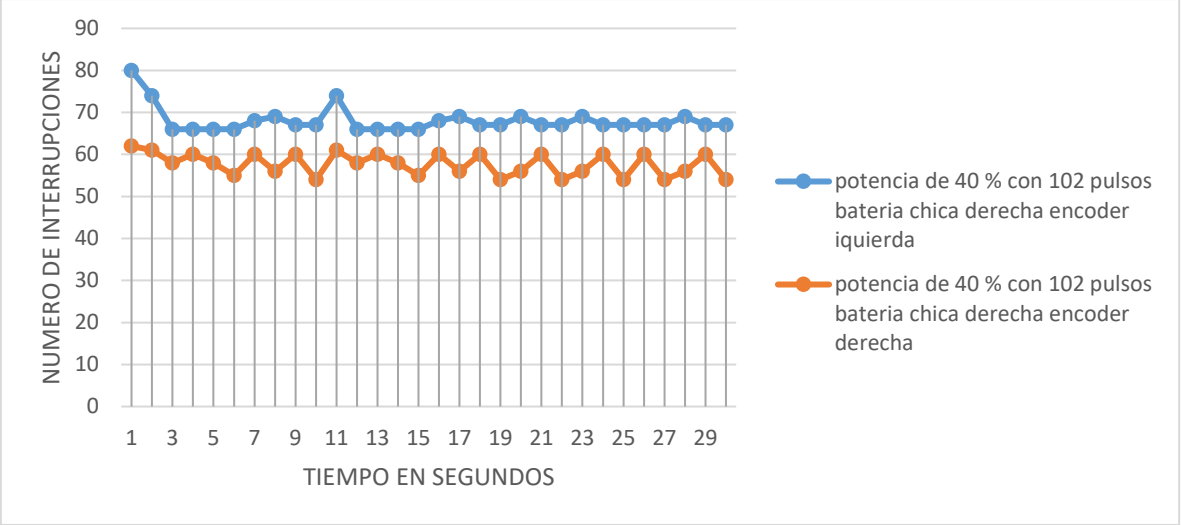


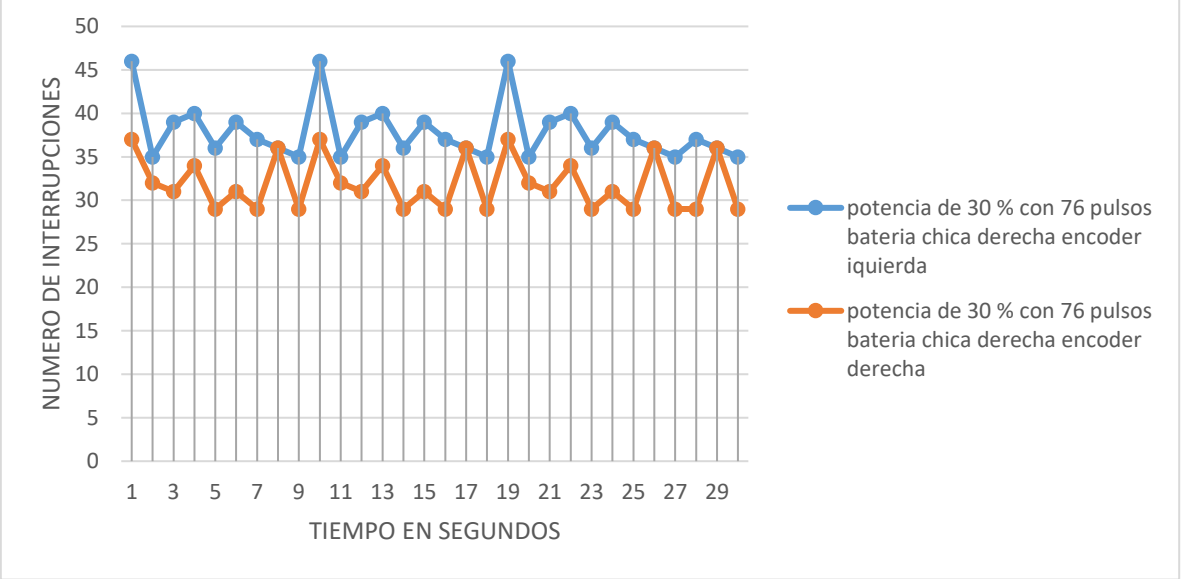

NUMERO DE INTERRUPCIONES
110
105
100
95
90
85
80
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria chica derecha encoder iquierda
potencia de 80 % con 204 pulsos bateria chica derecha encoder derecha


NUMERO DE INTERRUPCIONES
102
100
98
96
94
92
90
88
86
84
82
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria chica derecha encoder iquierda
potencia de 70 % con 178 pulsos bateria chica derecha encoder derecha

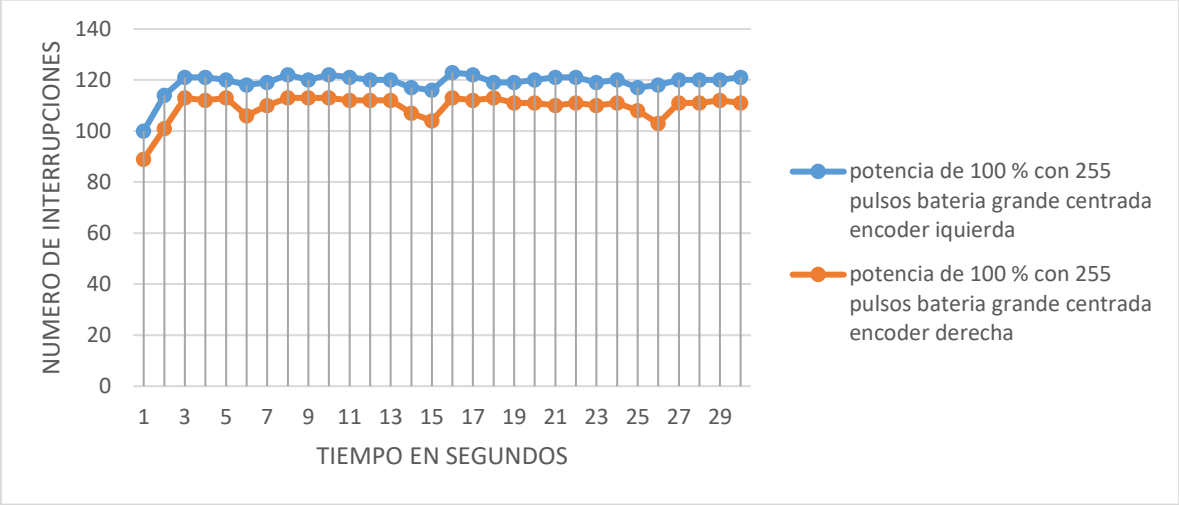

NUMERO DE INTERRUPCIONES
100
95
90
85
80
75
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria chica derecha encoder iquierda
potencia de 60 % con 153 pulsos bateria chica derecha encoder derecha

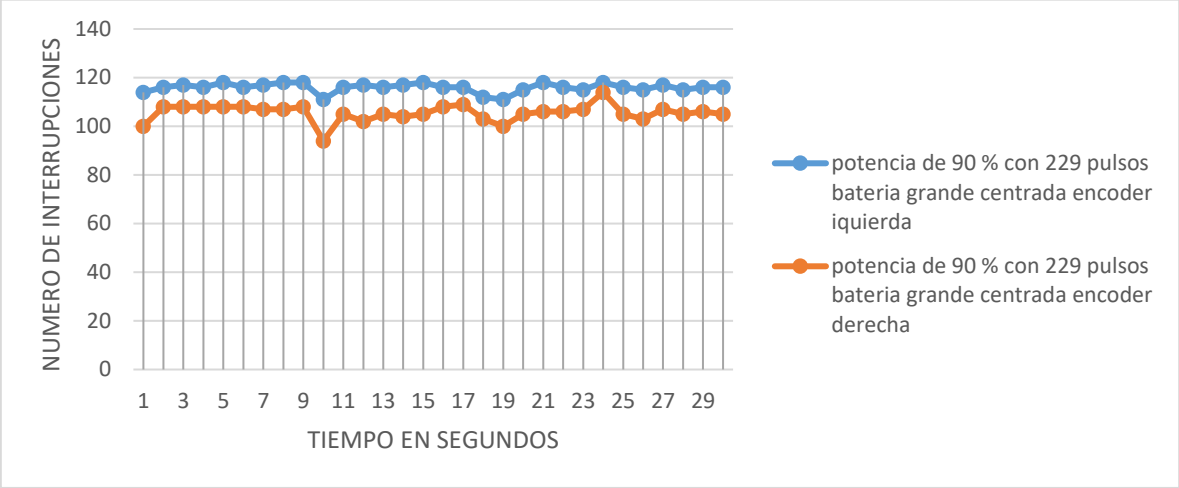

NUMERO DE INTERRUPCIONES
85
80
75
70
65
60
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria chica derecha encoder iquierda
potencia de 50 % con 127 pulsos bateria chica derecha encoder derecha

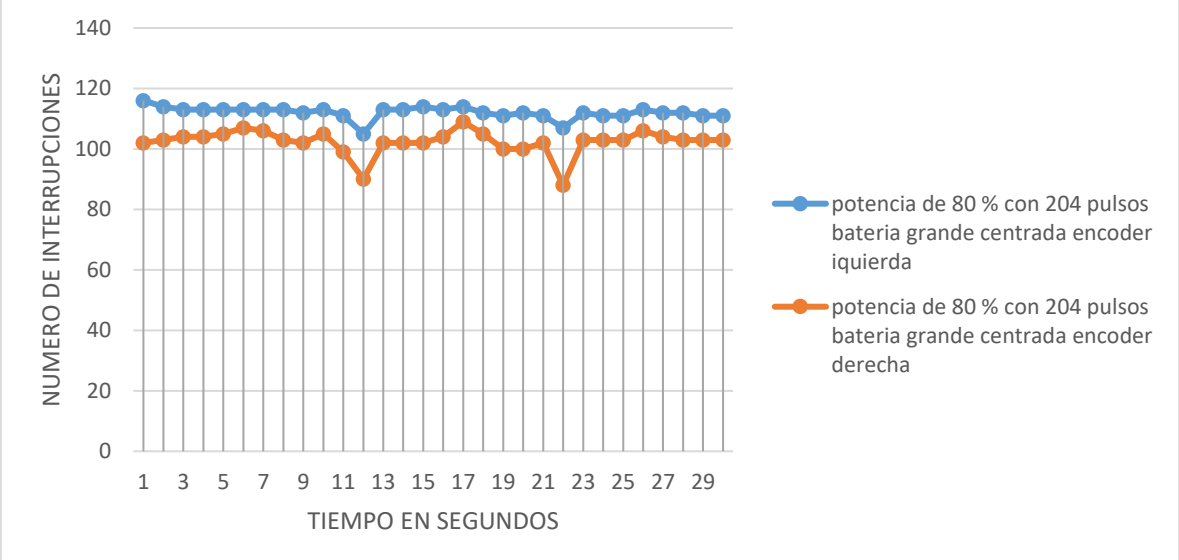

NUMERO DE INTERRUPCIONES
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria chica derecha encoder iquierda
potencia de 40 % con 102 pulsos bateria chica derecha encoder derecha

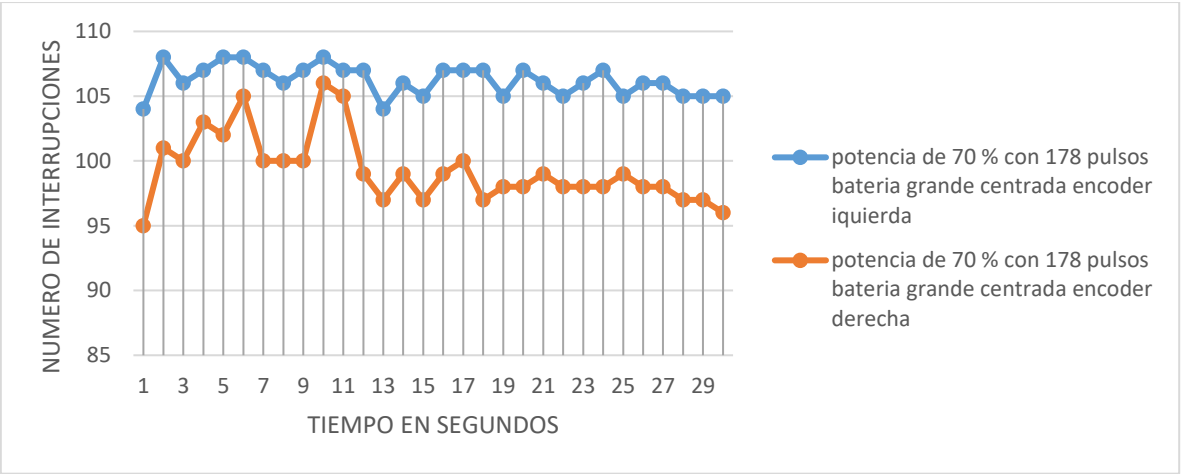

NUMERO DE INTERRUPCIONES
50
45
40
35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30 % con 76 pulsos bateria chica derecha encoder iquierda
potencia de 30 % con 76 pulsos bateria chica derecha encoder derecha

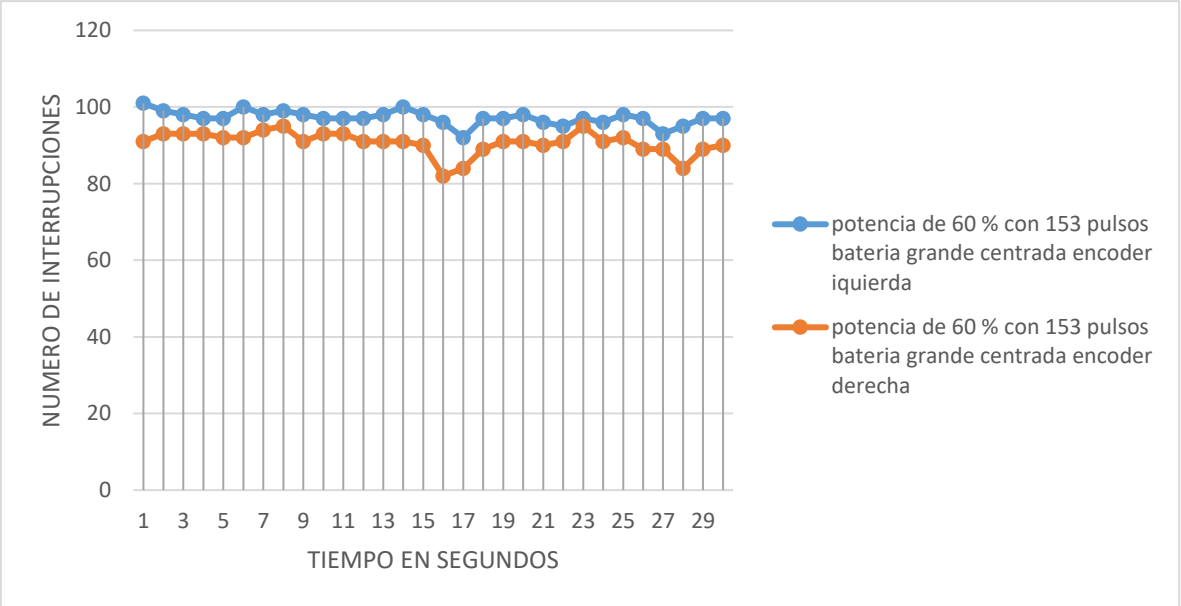
Pruebas de velocidad usando Llantas omnidireccional, con carga de 2.5k centrada

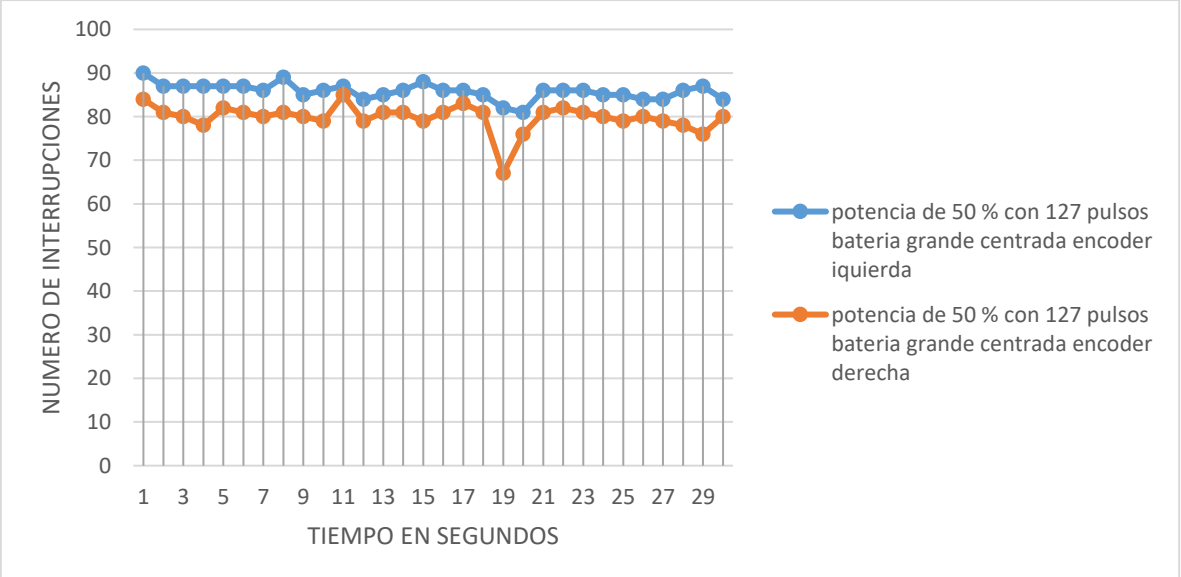


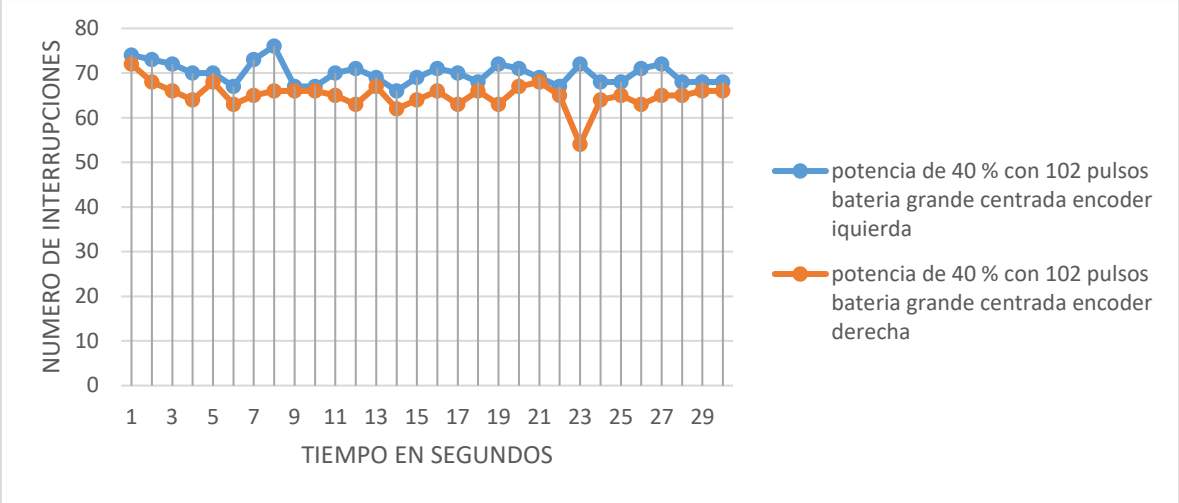


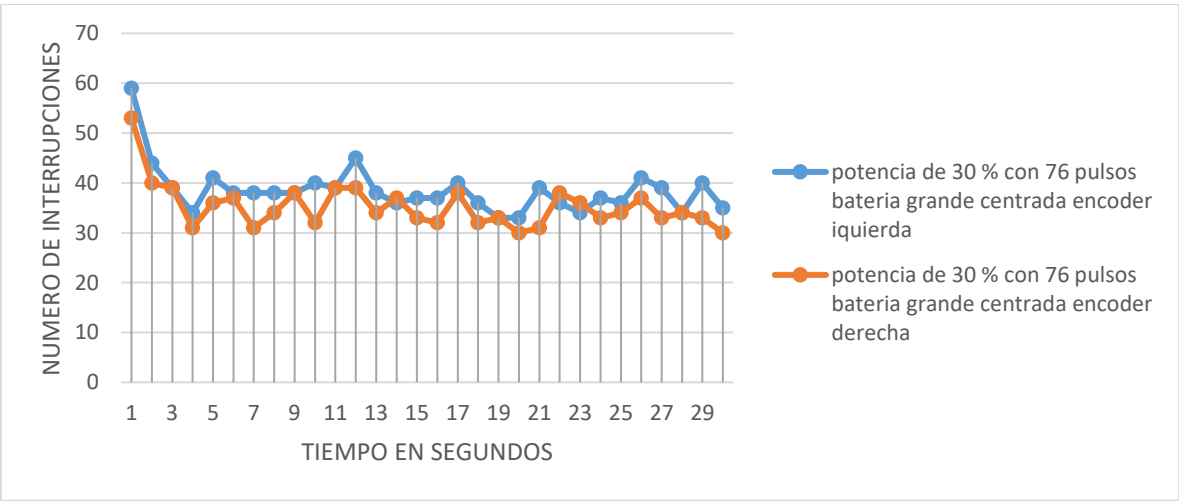



NUMERO DE INTERRUPCIONES
110
105
100
95
90
85
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande centrada encoder iquierda
potencia de 70 % con 178 pulsos bateria grande centrada encoder derecha

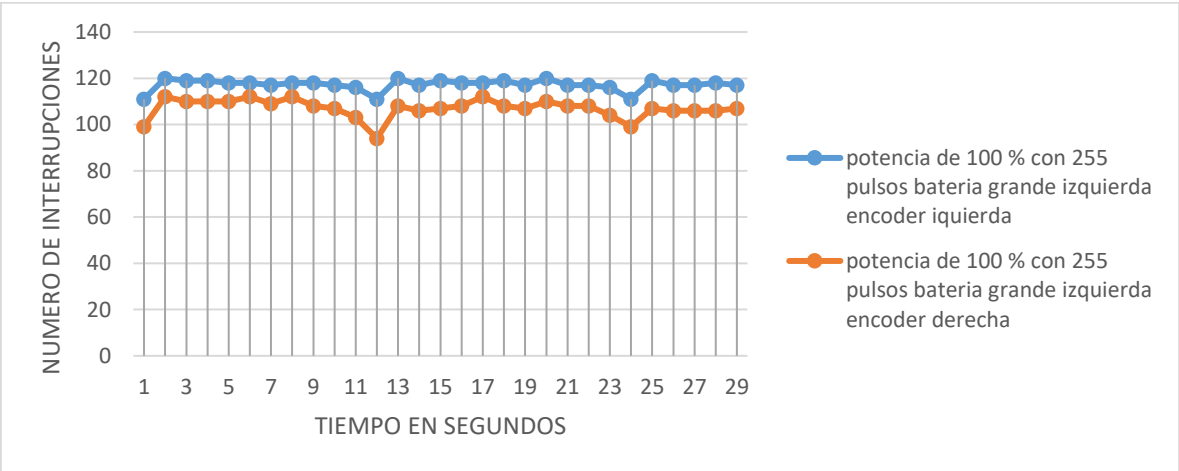

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande centrada encoder iquierda
potencia de 60 % con 153 pulsos bateria grande centrada encoder derecha

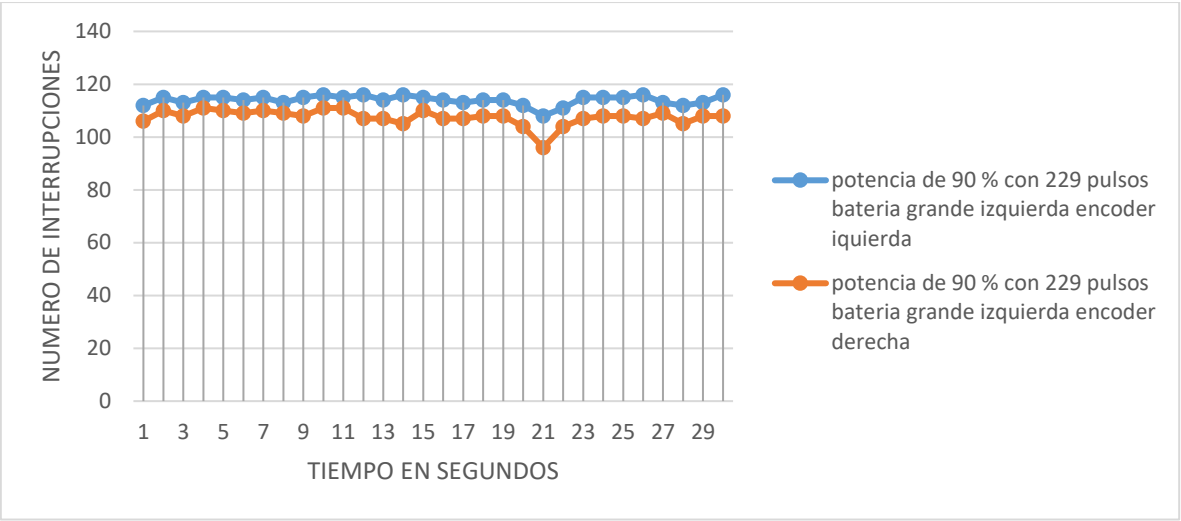

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria grande centrada encoder iquierda
potencia de 50 % con 127 pulsos bateria grande centrada encoder derecha

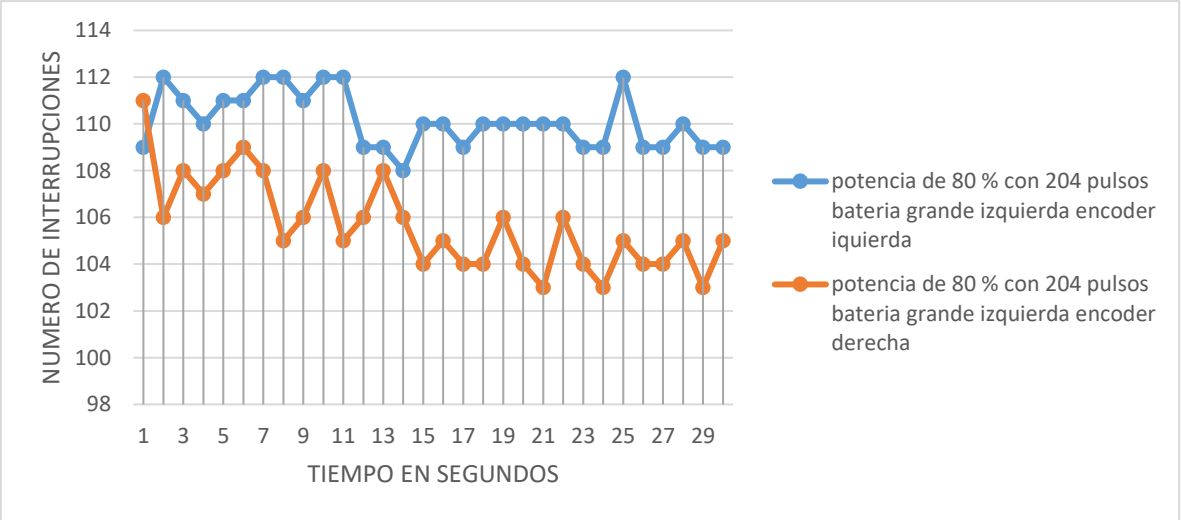


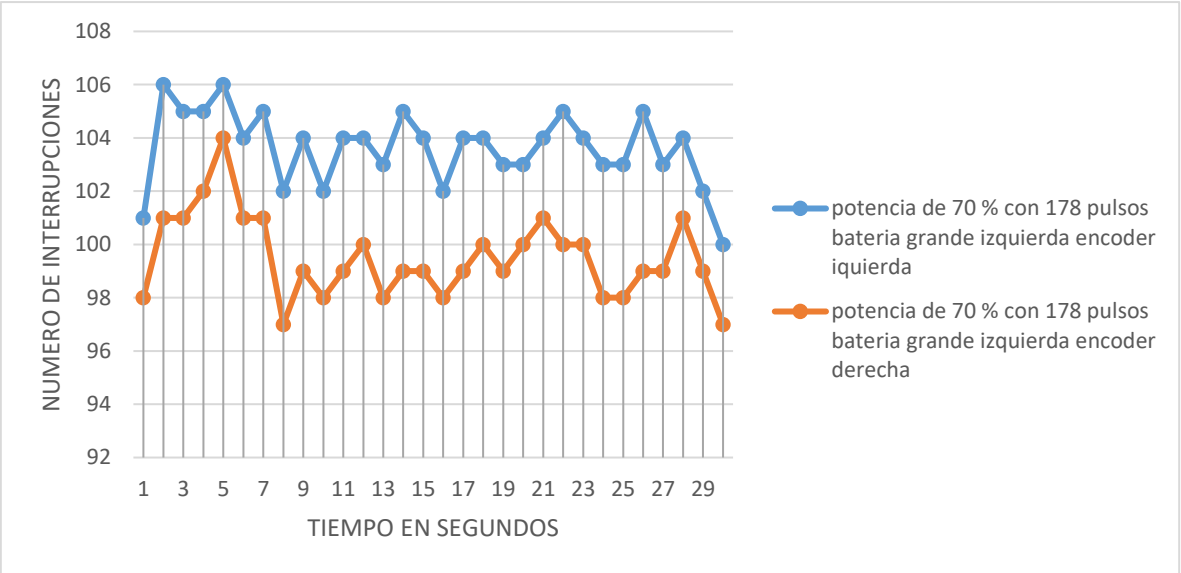


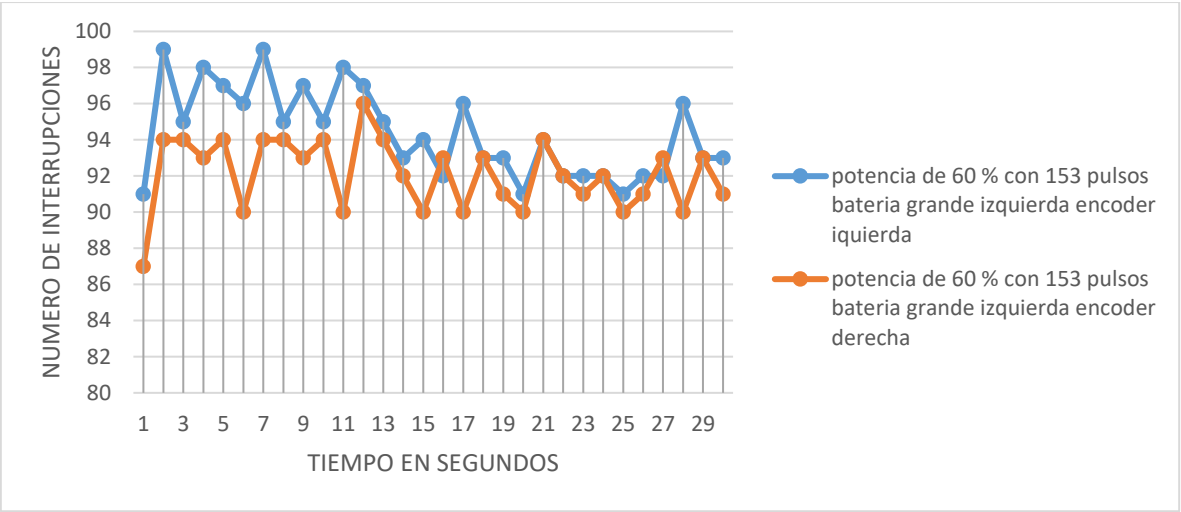
Pruebas de velocidad usando Llantas omnidireccional, con carga de 2.5k lado izquierdo

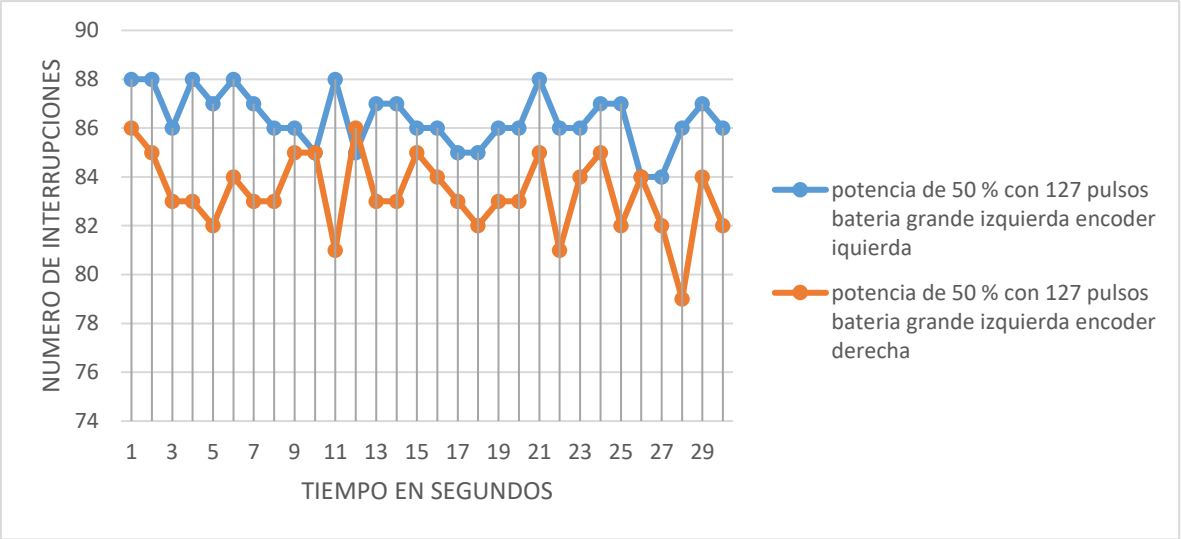


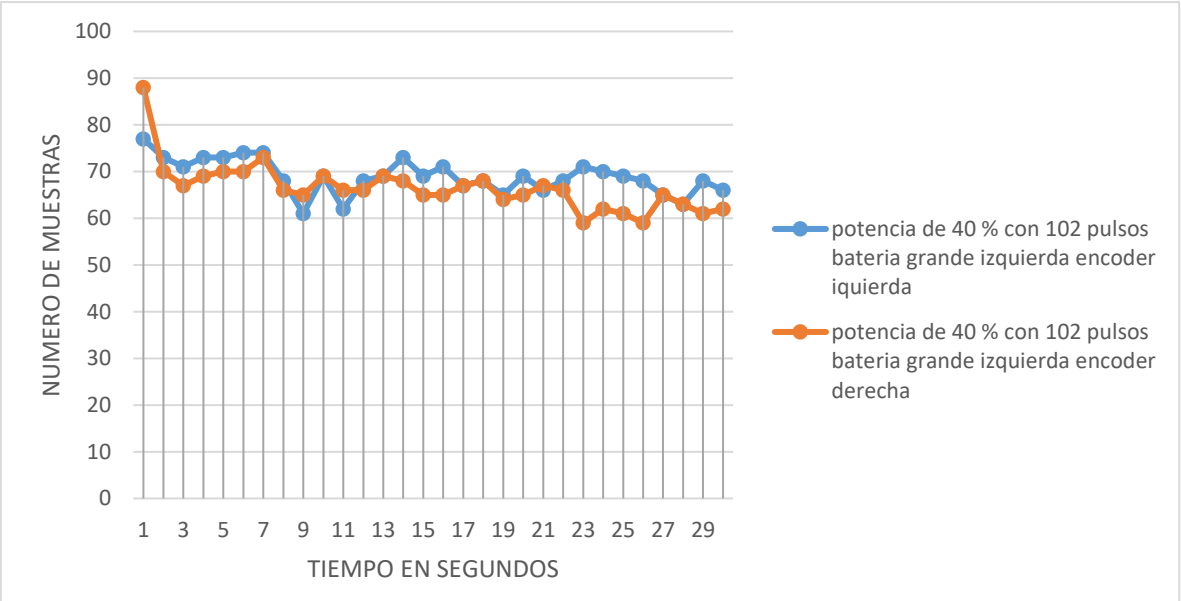

NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 90 % con 229 pulsos bateria grande izquierda encoder iquierda
potencia de 90 % con 229 pulsos bateria grande izquierda encoder derecha

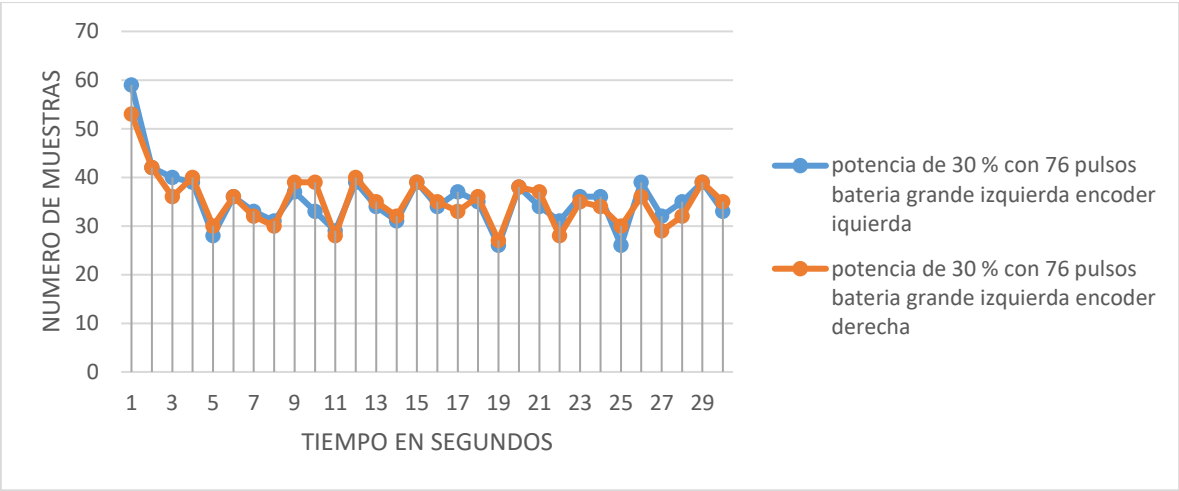

NUMERO DE INTERRUPCIONES
114
112
110
108
106
104
102
100
98
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria grande izquierda encoder iquierda
potencia de 80 % con 204 pulsos bateria grande izquierda encoder derecha


NUMERO DE INTERRUPCIONES
108
106
104
102
100
98
96
94
92
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande izquierda encoder iquierda
potencia de 70 % con 178 pulsos bateria grande izquierda encoder derecha

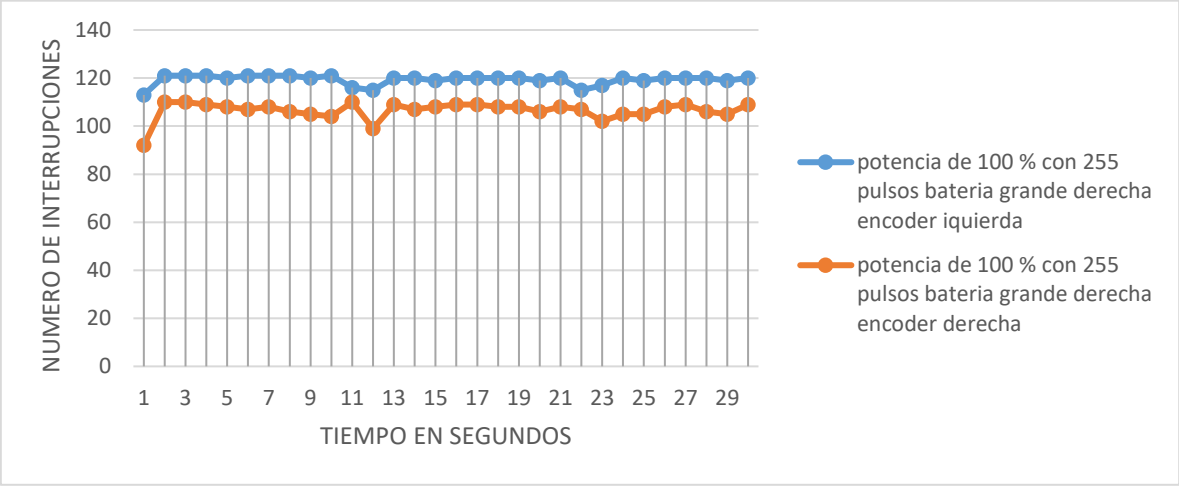

NUMERO DE INTERRUPCIONES
100
98
96
94
92
90
88
86
84
82
80
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande izquierda encoder iquierda
potencia de 60 % con 153 pulsos bateria grande izquierda encoder derecha

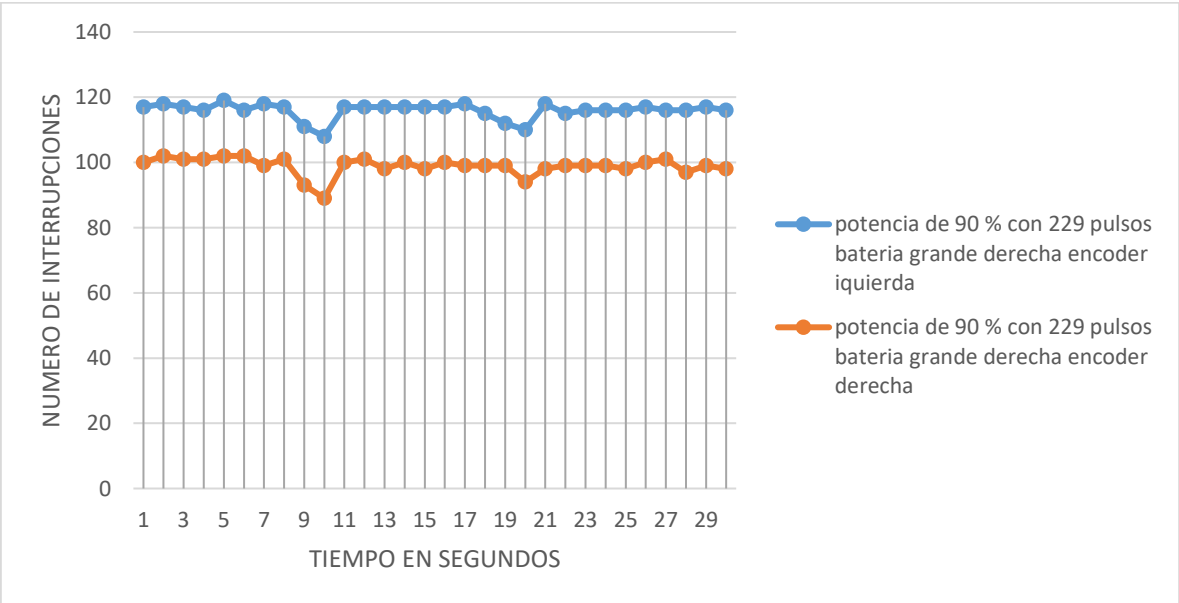

NUMERO DE INTERRUPCIONES
90
88
86
84
82
80
78
76
74
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria grande izquierda encoder iquierda
potencia de 50 % con 127 pulsos bateria grande izquierda encoder derecha

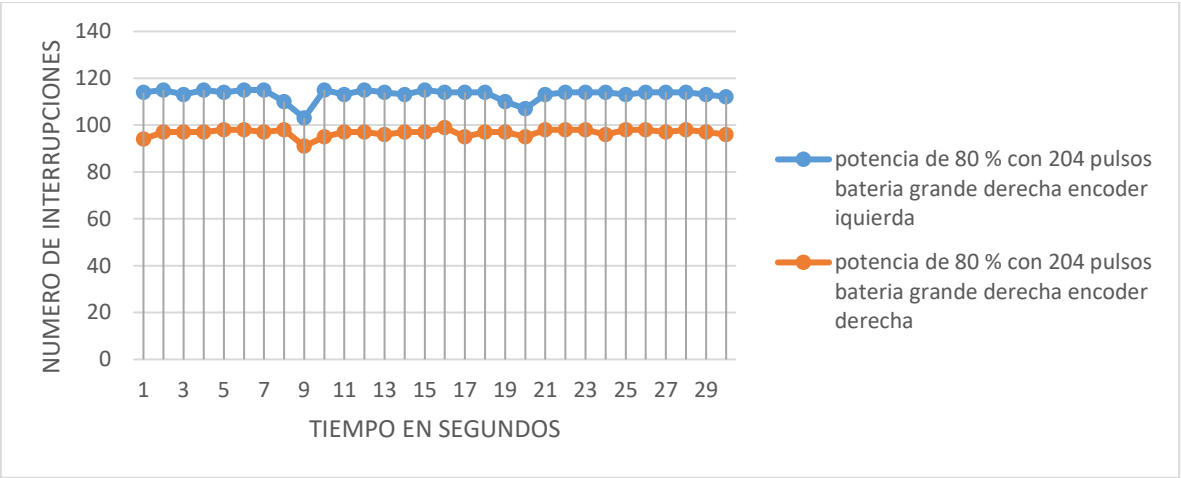

NUMERO DE MUESTRAS
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria grande izquierda encoder iquierda
potencia de 40 % con 102 pulsos bateria grande izquierda encoder derecha

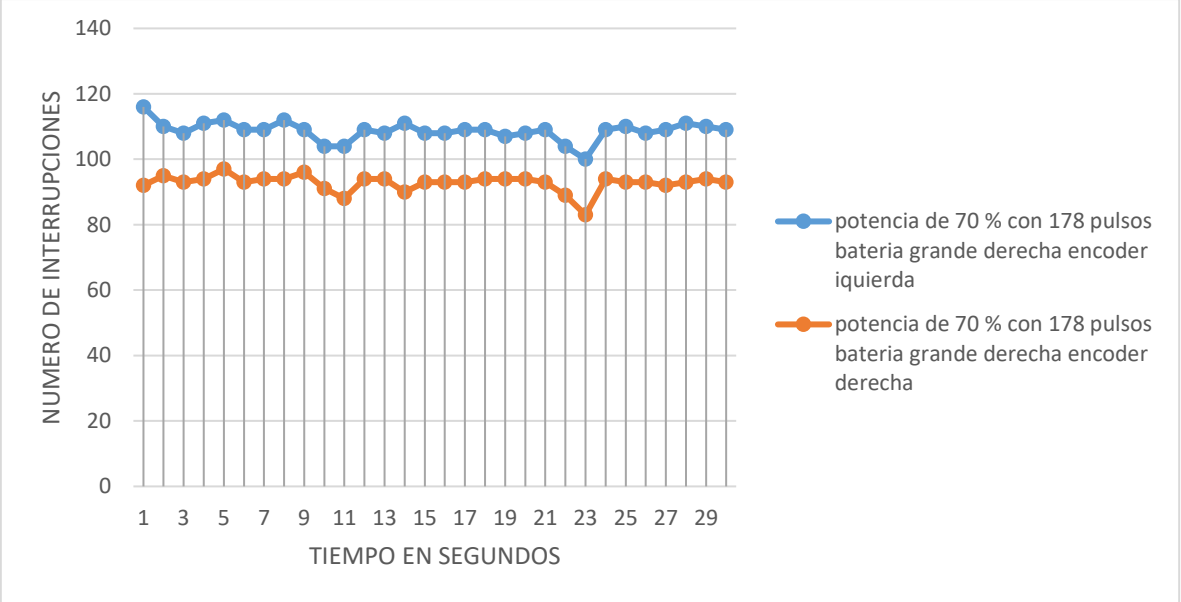


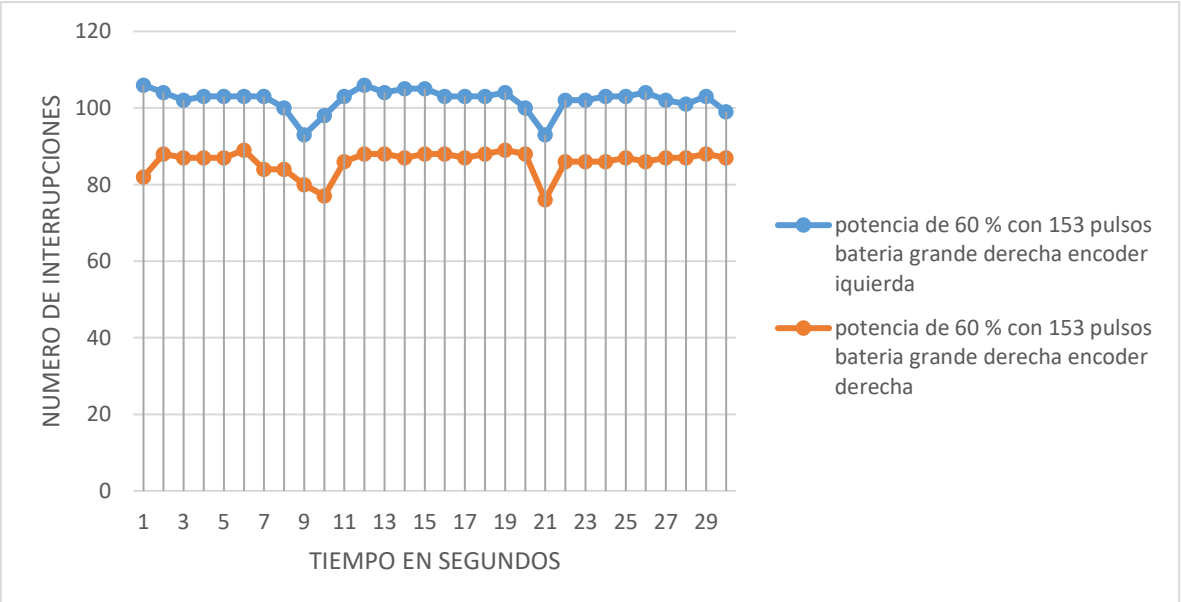
Pruebas de velocidad usando Llantas omnidireccional, con carga de 2.5k lado derecho

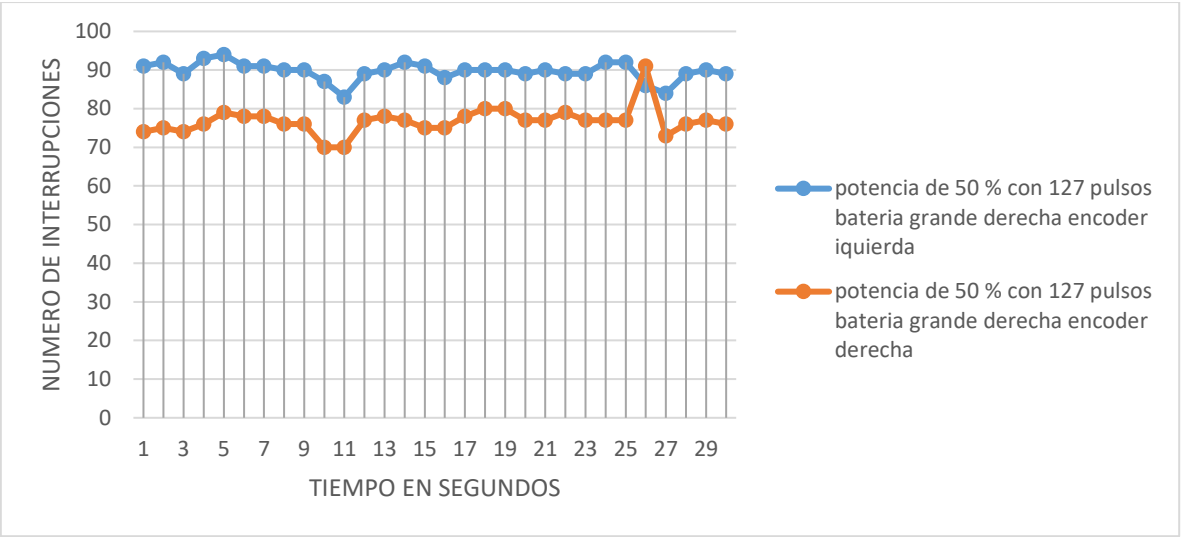


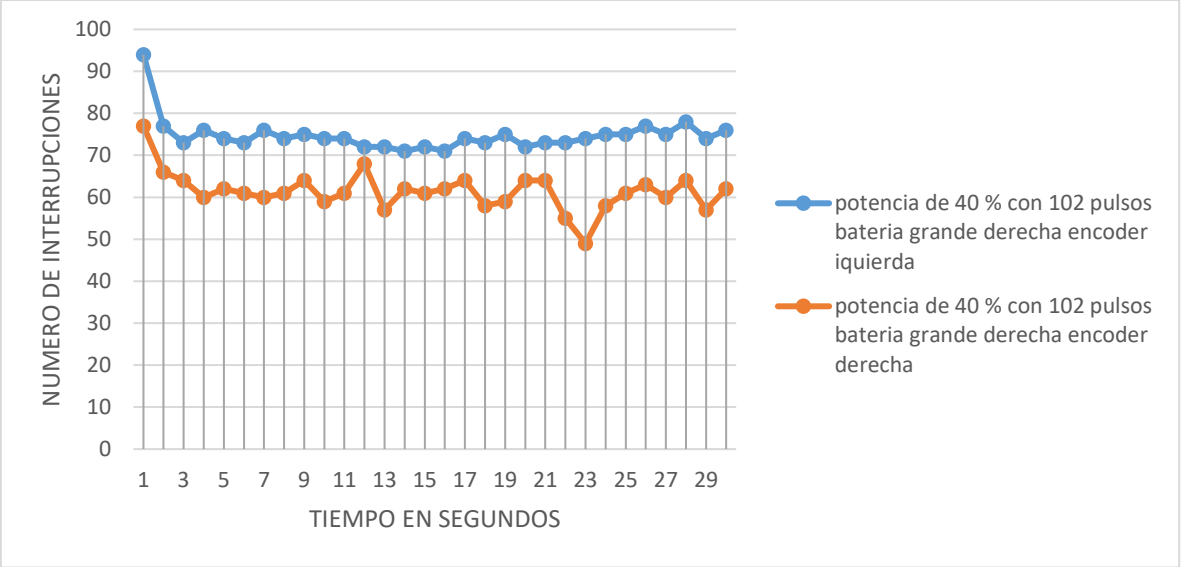


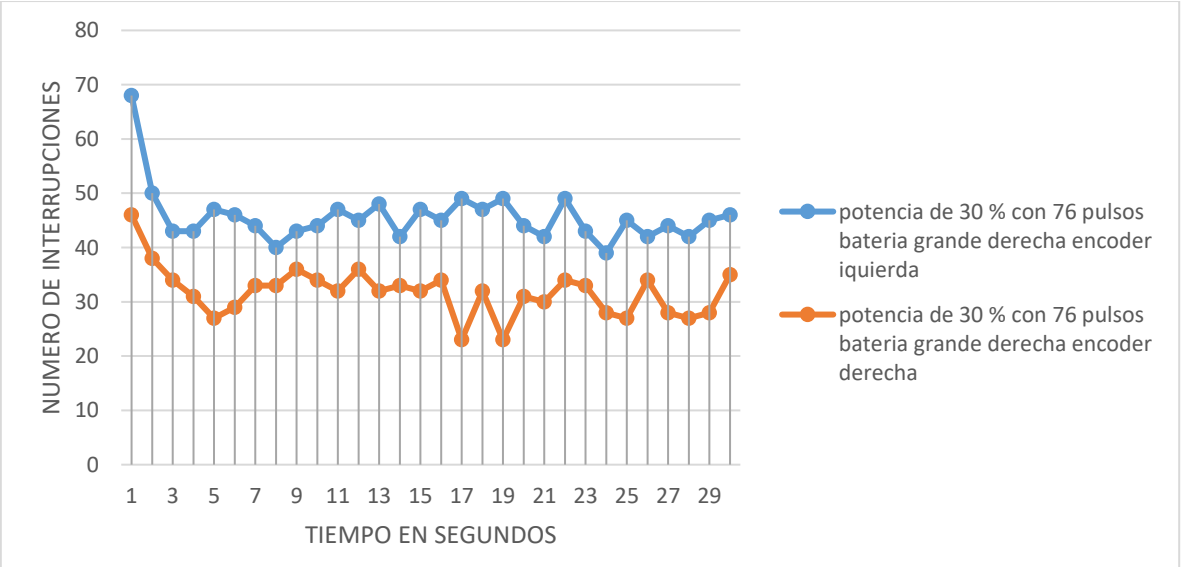

NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria grande derecha encoder iquierda
potencia de 80 % con 204 pulsos bateria grande derecha encoder derecha


NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande derecha encoder iquierda
potencia de 70 % con 178 pulsos bateria grande derecha encoder derecha

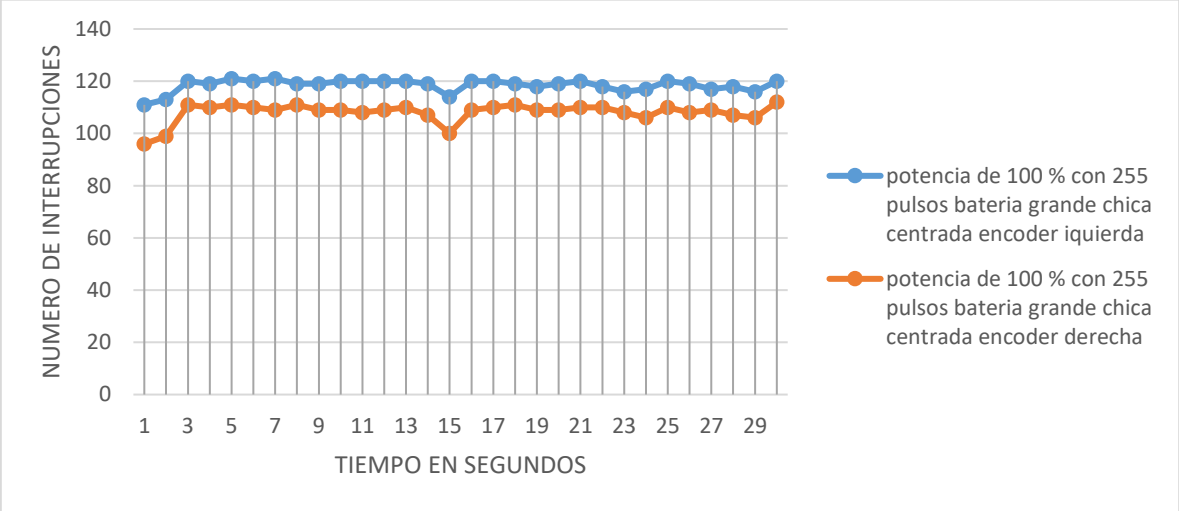

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande derecha encoder iquierda
potencia de 60 % con 153 pulsos bateria grande derecha encoder derecha

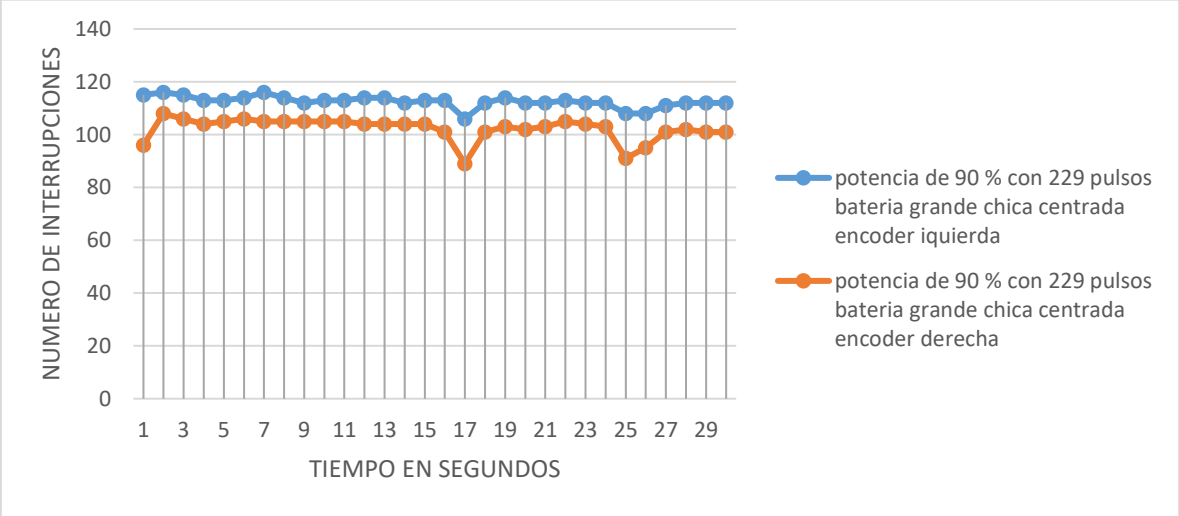

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria grande derecha encoder iquierda
potencia de 50 % con 127 pulsos bateria grande derecha encoder derecha

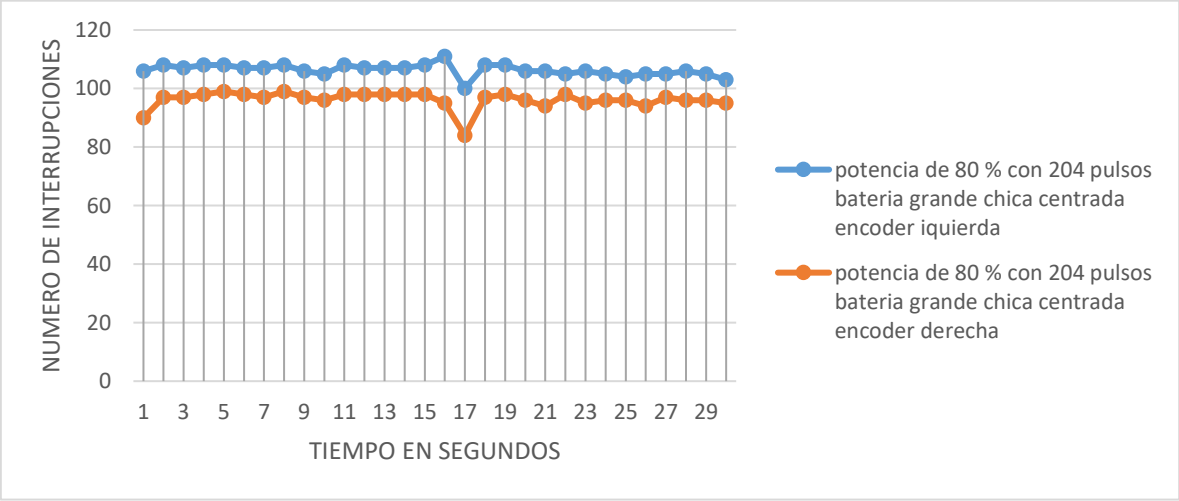

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria grande derecha encoder iquierda
potencia de 40 % con 102 pulsos bateria grande derecha encoder derecha

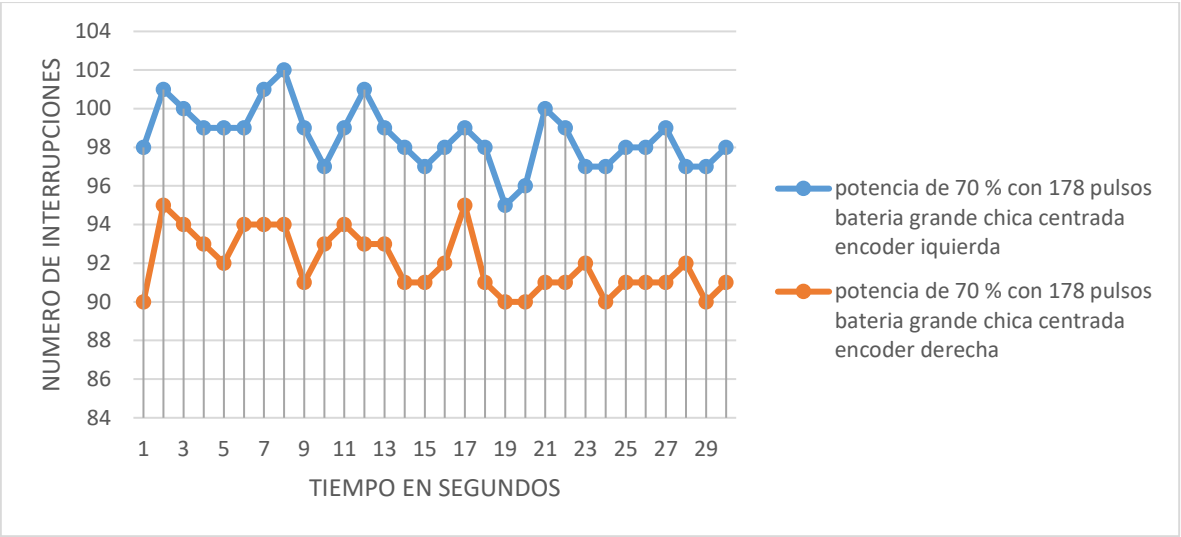

NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 30 % con 76 pulsos bateria grande derecha encoder iquierda
potencia de 30 % con 76 pulsos bateria grande derecha encoder derecha

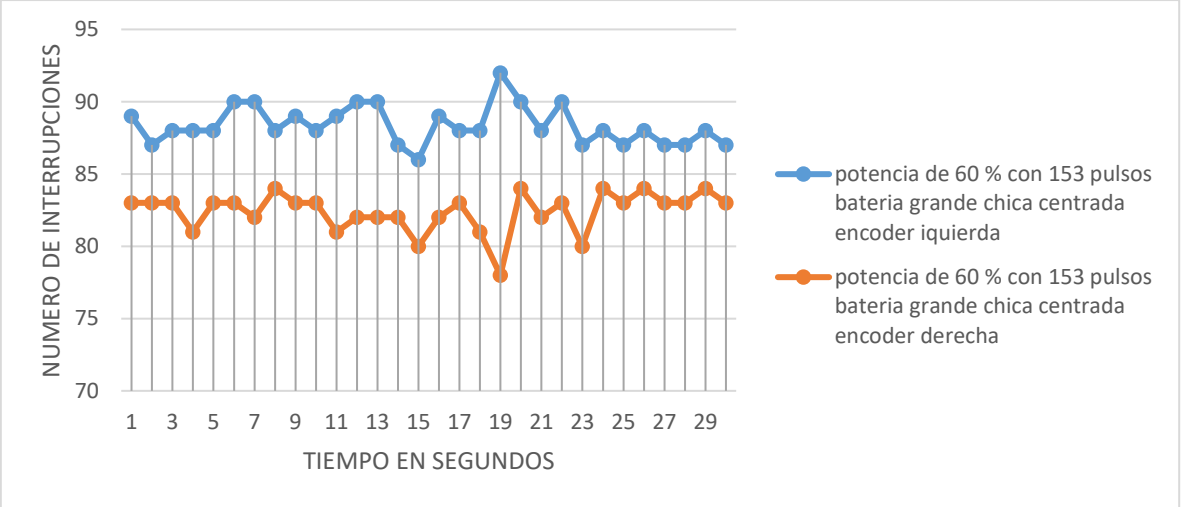
Pruebas de velocidad usando Llantas omnidireccional, con carga de 4.2k centrada

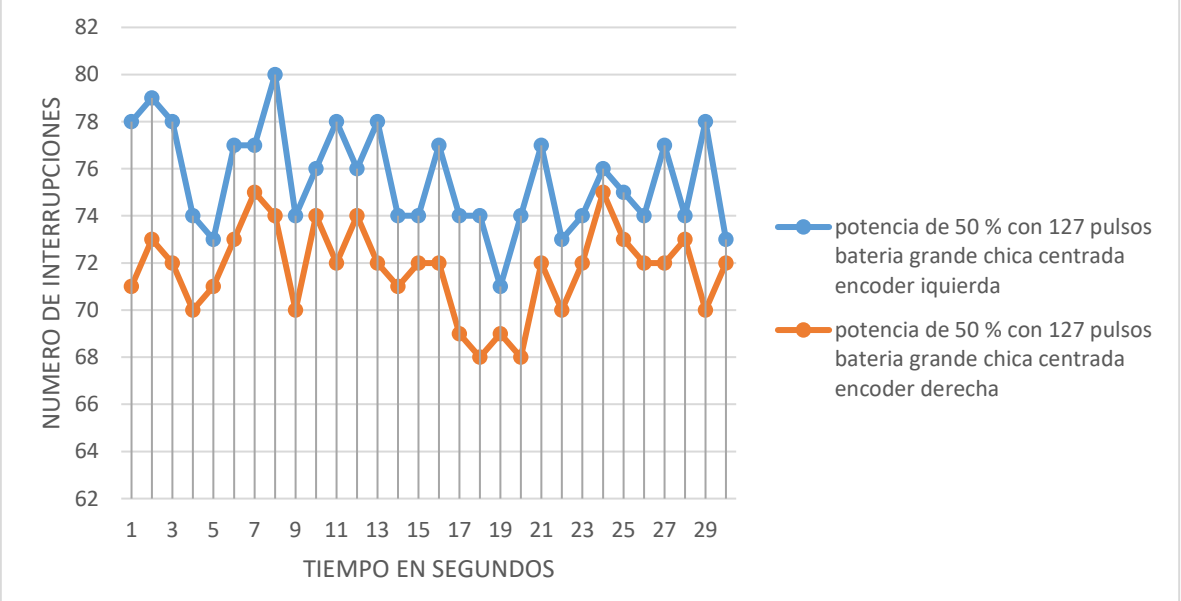


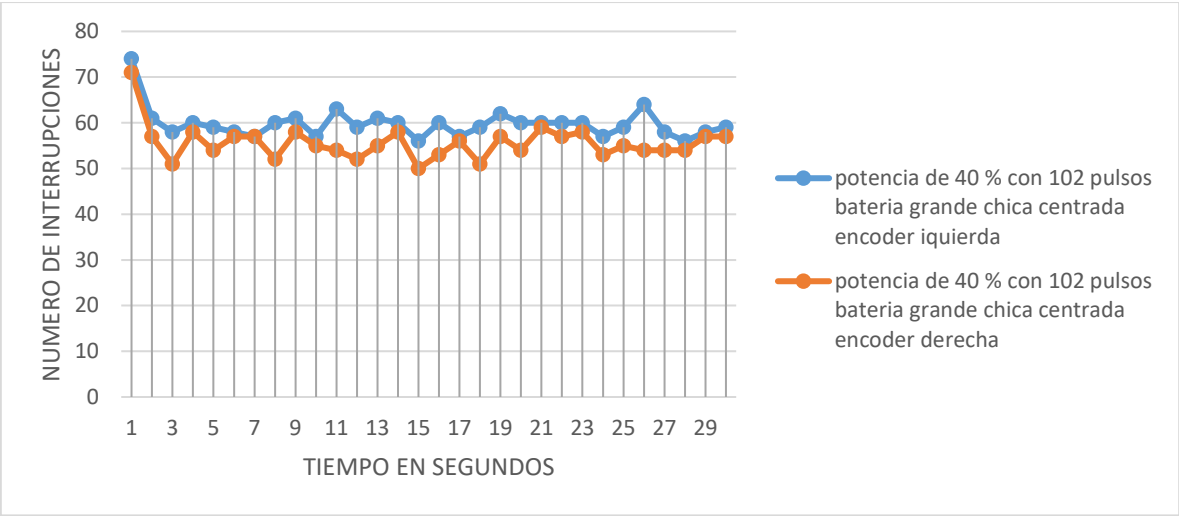




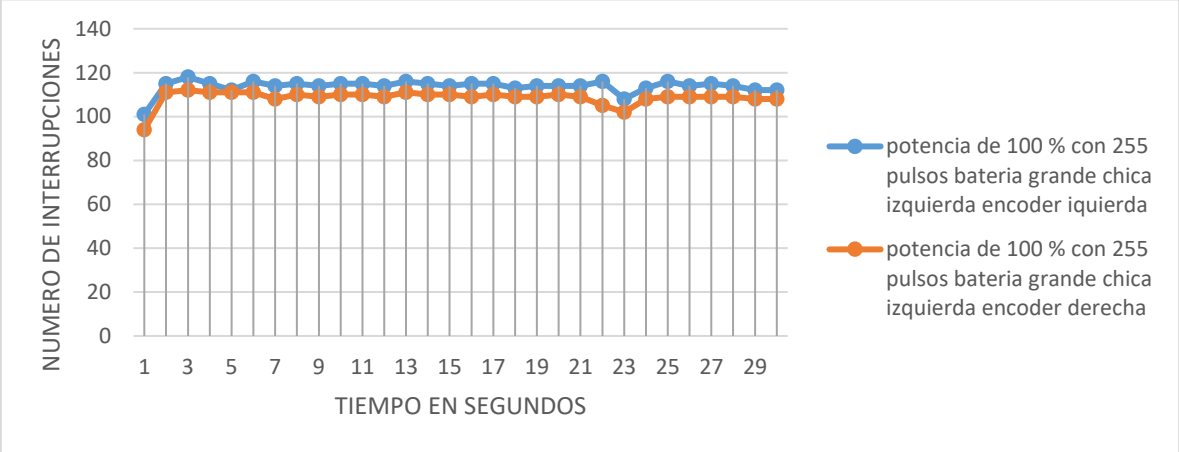

NUMERO DE INTERRUPCIONES
104
102
100
98
96
94
92
90
88
86
84
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande chica centrada encoder iquierda
potencia de 70 % con 178 pulsos bateria grande chica centrada encoder derecha

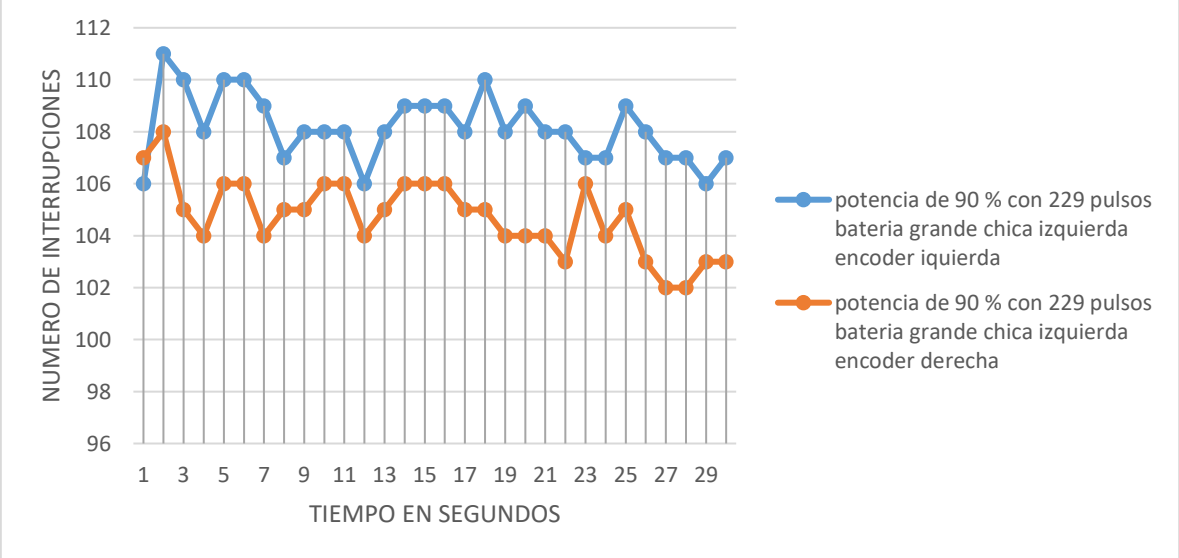

NUMERO DE INTERRUPCIONES
95
90
85
80
75
70
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande chica centrada encoder iquierda
potencia de 60 % con 153 pulsos bateria grande chica centrada encoder derecha

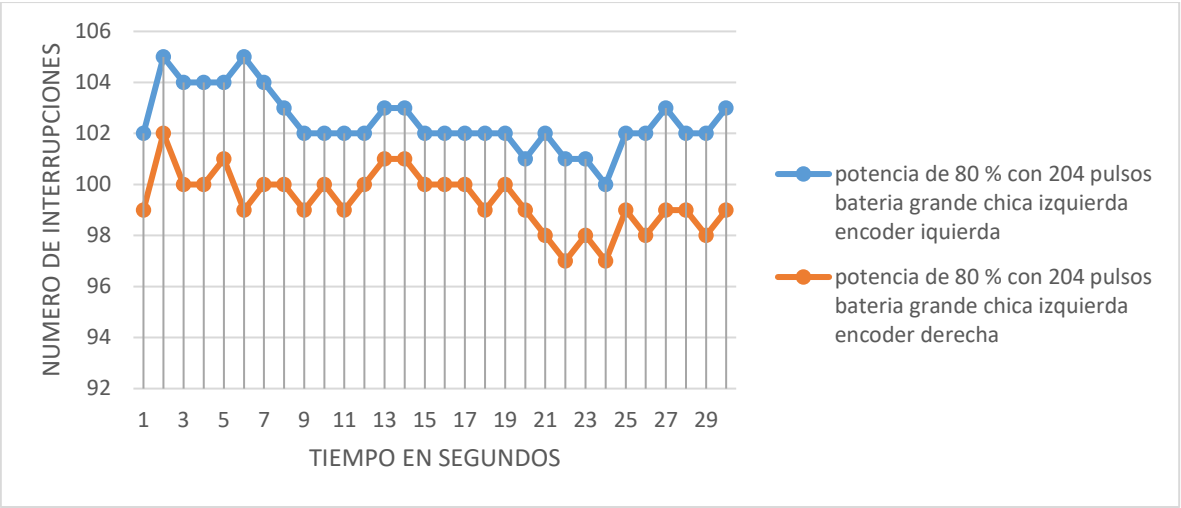

NUMERO DE INTERRUPCIONES
82
80
78
76
74
72
70
68
66
64
62
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria grande chica centrada encoder iquierda
potencia de 50 % con 127 pulsos bateria grande chica centrada encoder derecha

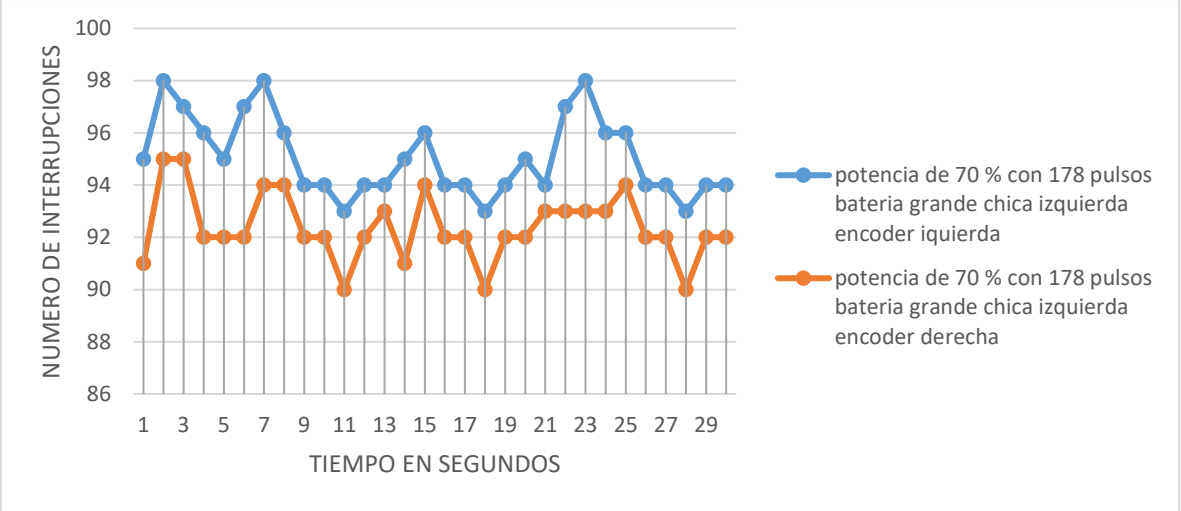


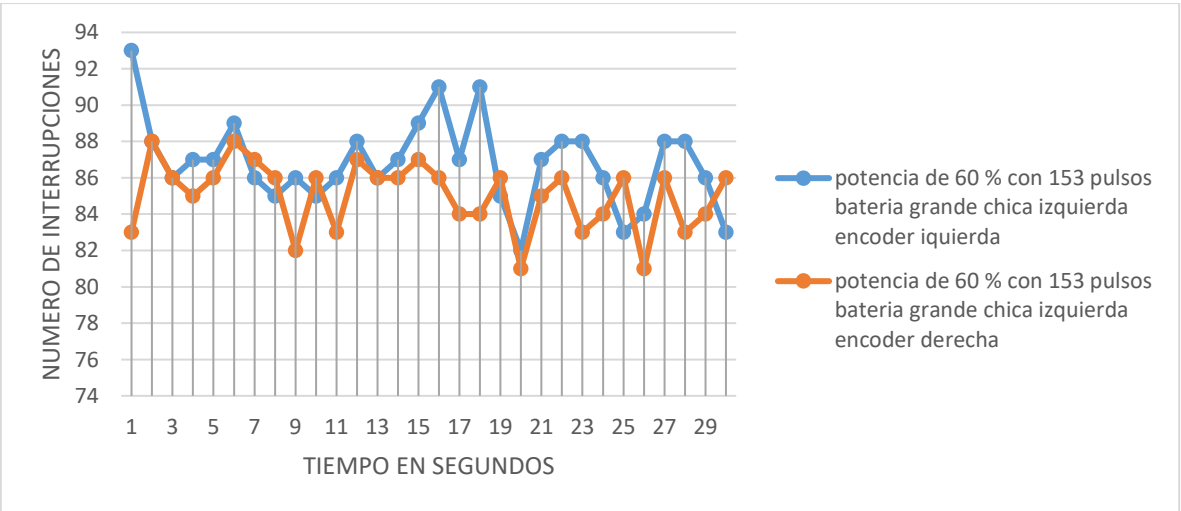
Pruebas de velocidad usando Llantas omnidireccional, con carga de 4.2k lado izquierdo

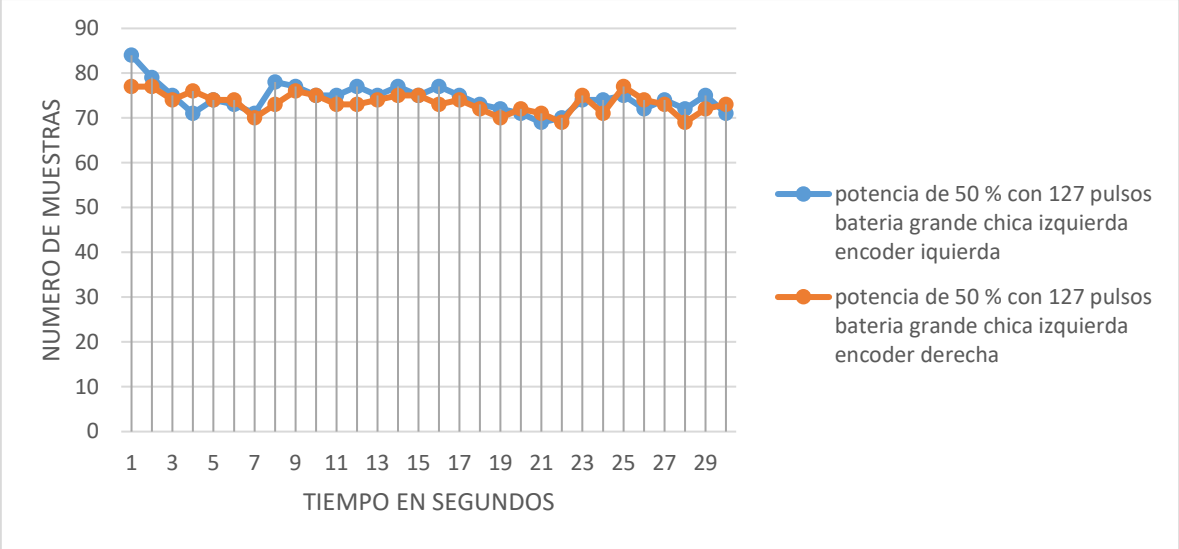


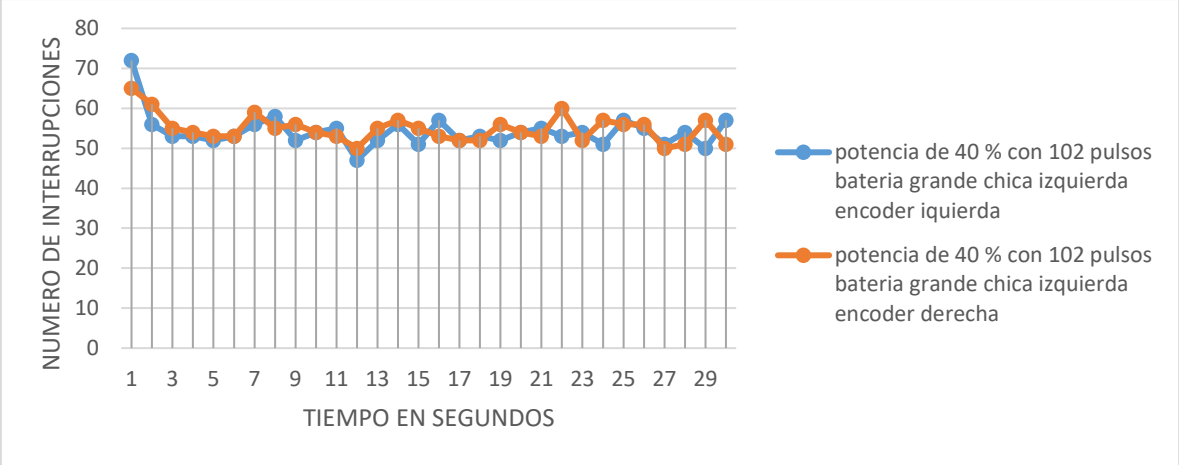



NUMERO DE INTERRUPCIONES
106
104
102
100
98
96
94
92
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria grande chica izquierda encoder iquierda
potencia de 80 % con 204 pulsos bateria grande chica izquierda encoder derecha

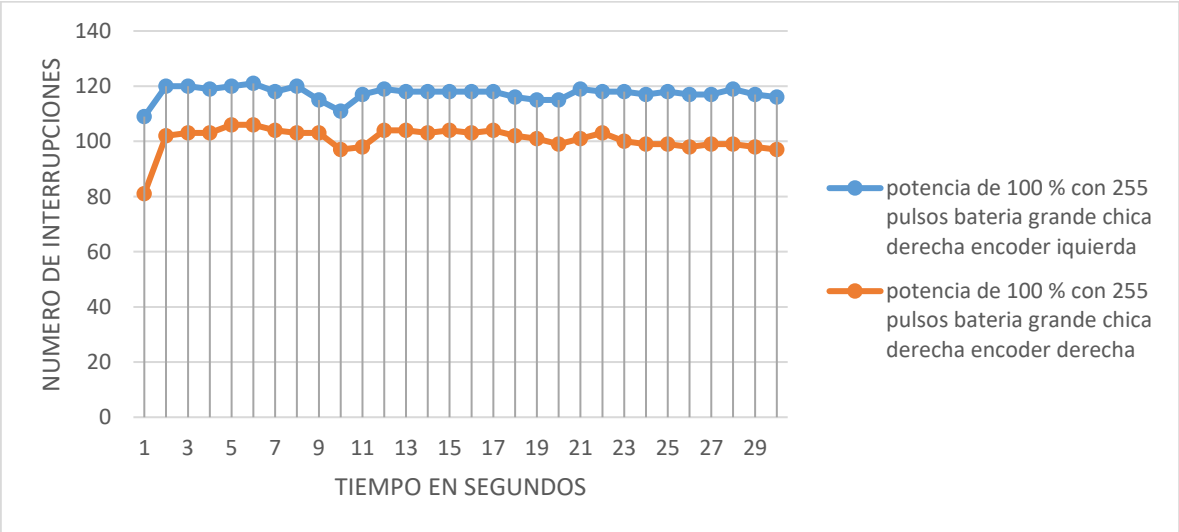

NUMERO DE INTERRUPCIONES
100
98
96
94
92
90
88
86
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande chica izquierda encoder iquierda
potencia de 70 % con 178 pulsos bateria grande chica izquierda encoder derecha

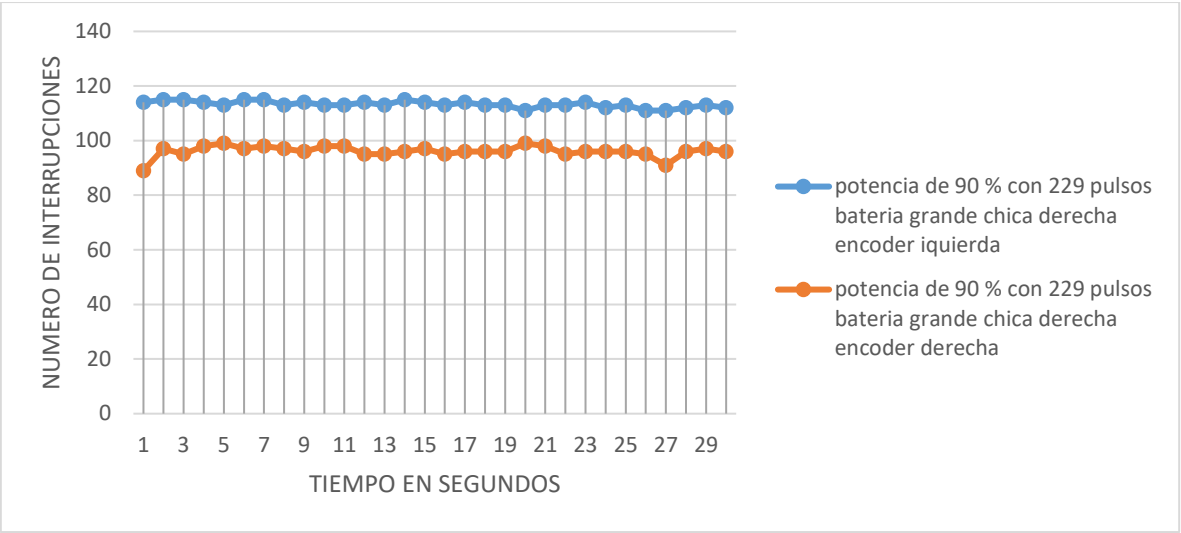

NUMERO DE INTERRUPCIONES
94
92
90
88
86
84
82
80
78
76
74
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande chica izquierda encoder iquierda
potencia de 60 % con 153 pulsos bateria grande chica izquierda encoder derecha

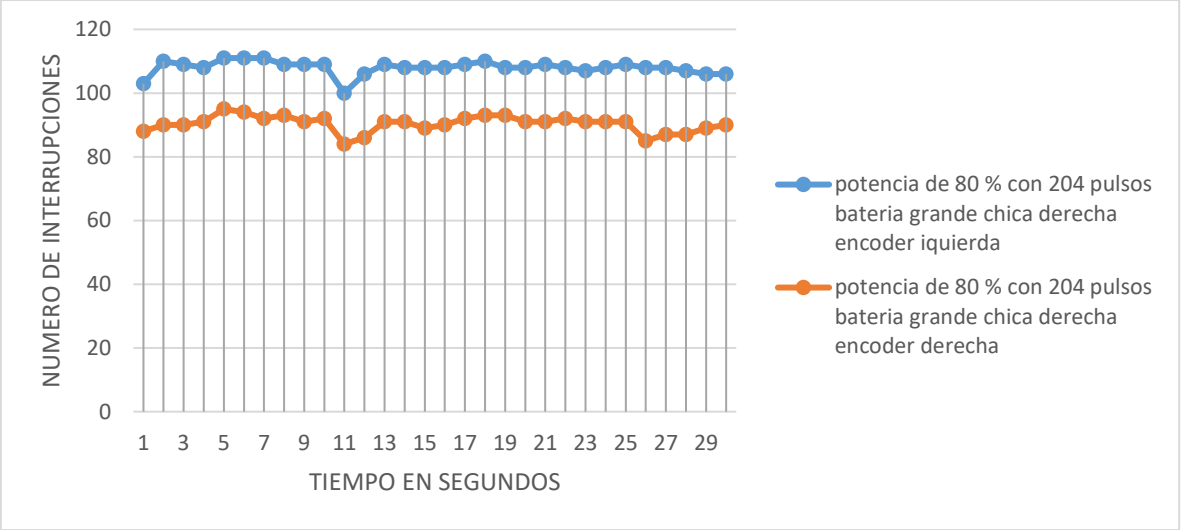


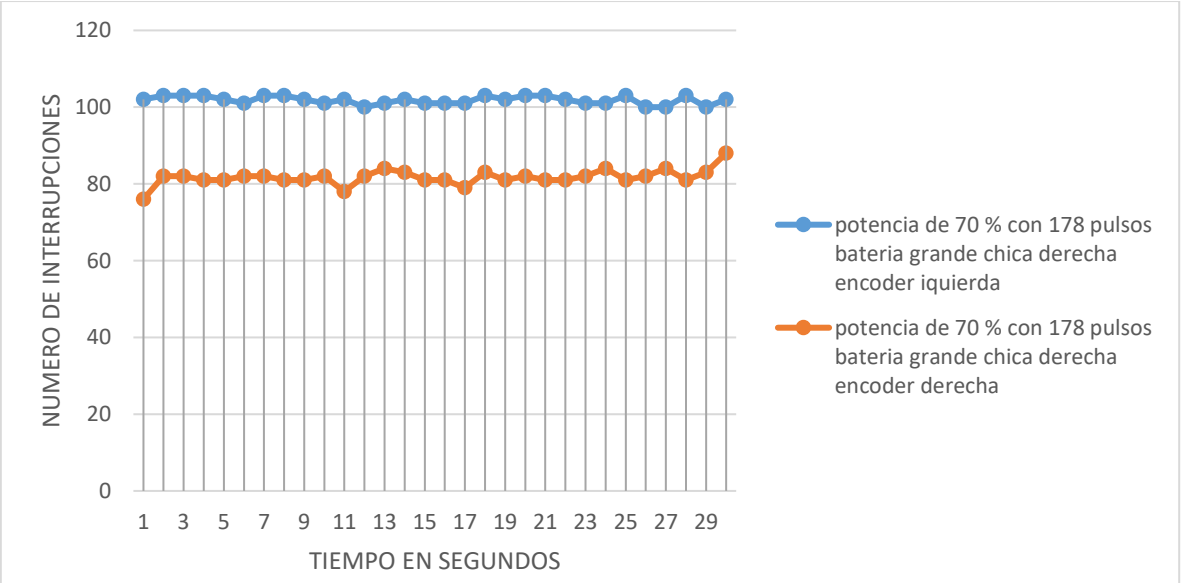


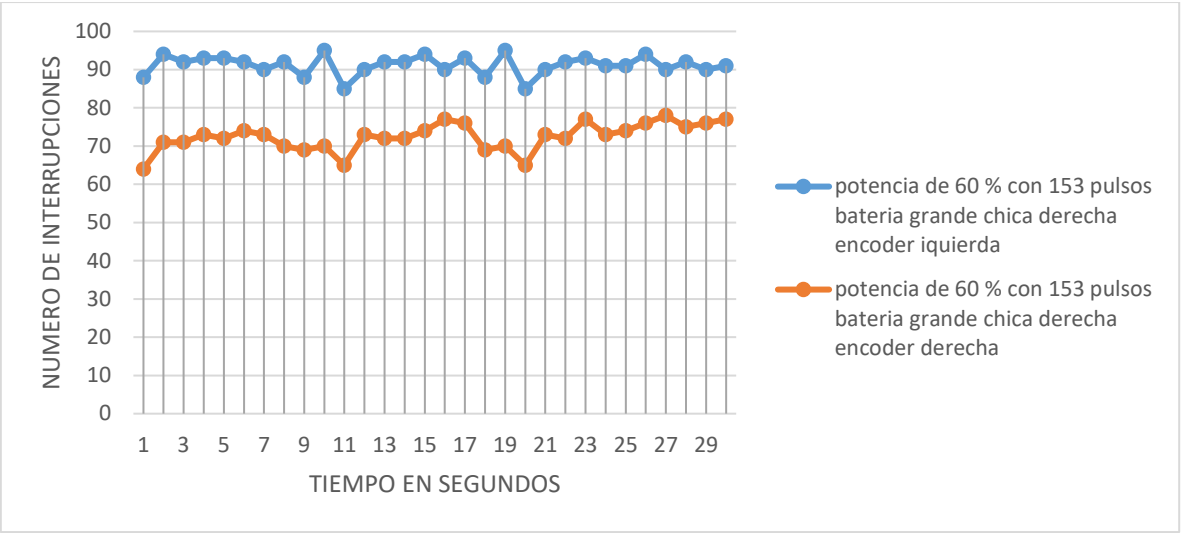
Pruebas de velocidad usando Llantas omnidireccional, con carga de 4.2k lado derecho

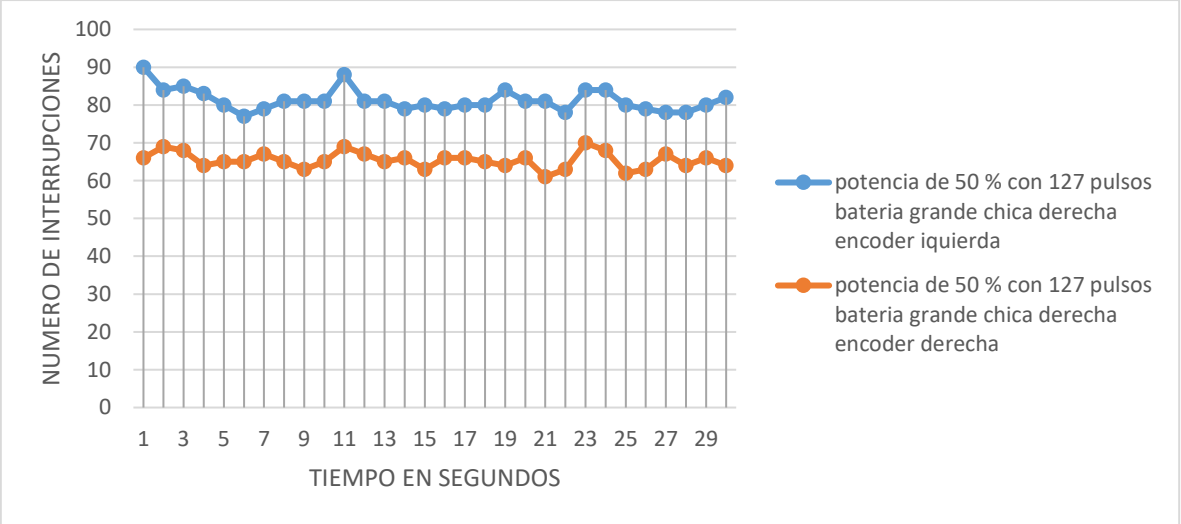


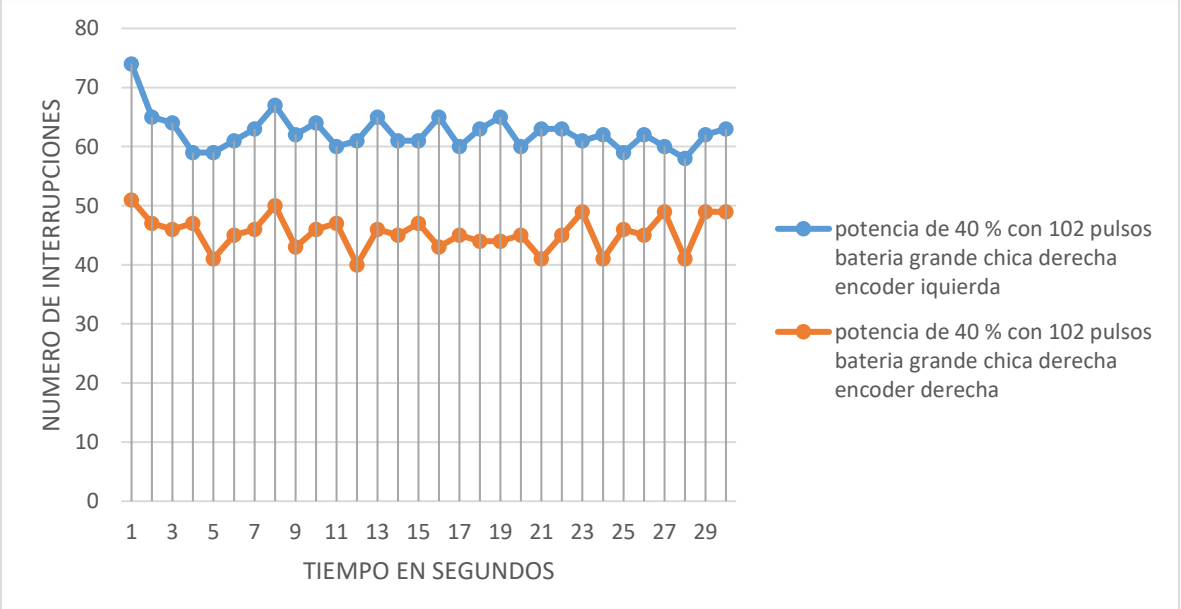

NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 90 % con 229 pulsos bateria grande chica derecha encoder iquierda
potencia de 90 % con 229 pulsos bateria grande chica derecha encoder derecha


NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos bateria grande chica derecha encoder iquierda
potencia de 80 % con 204 pulsos bateria grande chica derecha encoder derecha

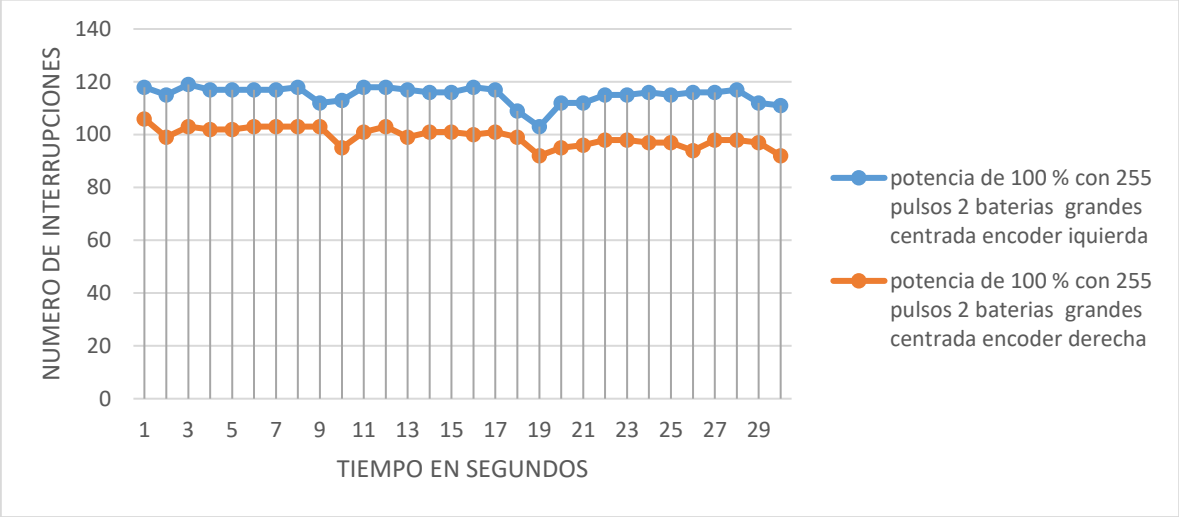

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos bateria grande chica derecha encoder iquierda
potencia de 70 % con 178 pulsos bateria grande chica derecha encoder derecha

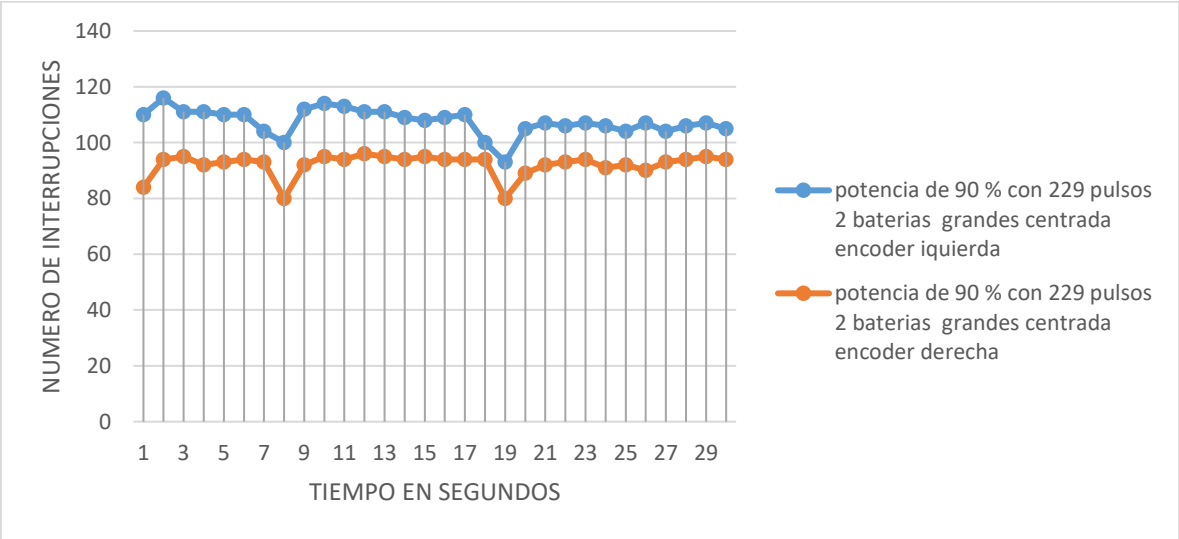

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos bateria grande chica derecha encoder iquierda
potencia de 60 % con 153 pulsos bateria grande chica derecha encoder derecha

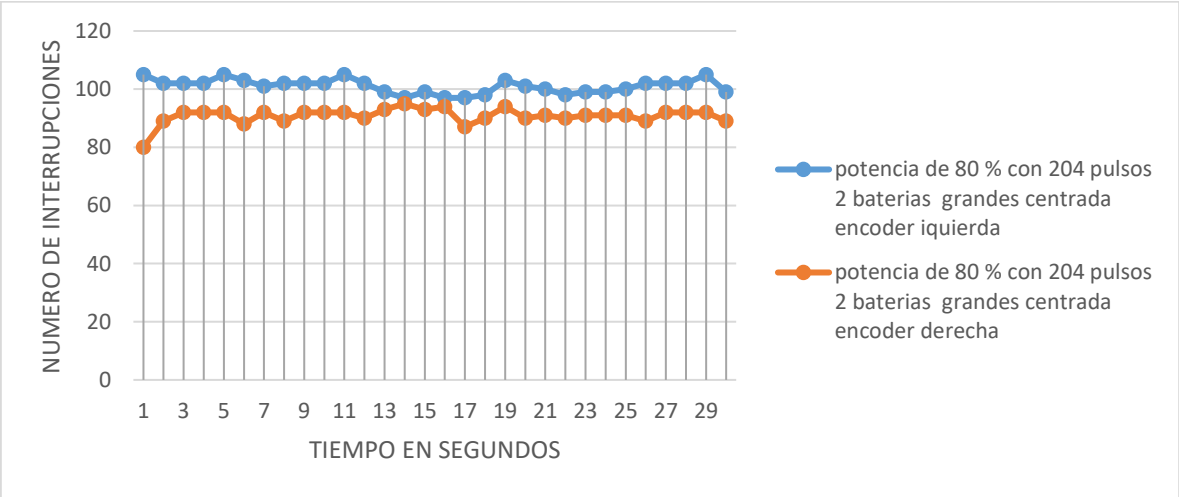

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos bateria grande chica derecha encoder iquierda
potencia de 50 % con 127 pulsos bateria grande chica derecha encoder derecha

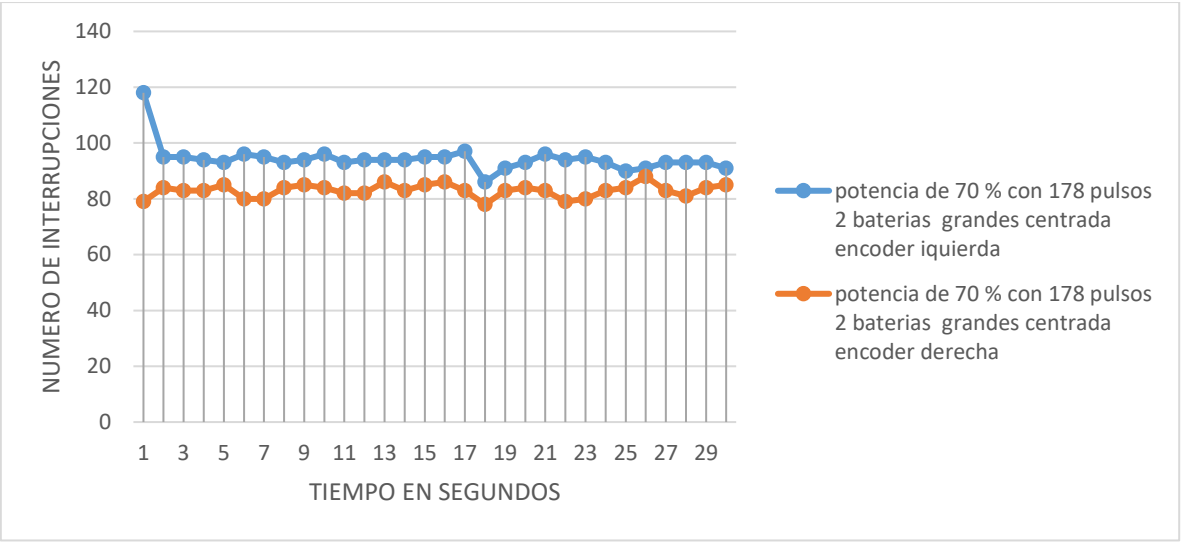

NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos bateria grande chica derecha encoder iquierda
potencia de 40 % con 102 pulsos bateria grande chica derecha encoder derecha

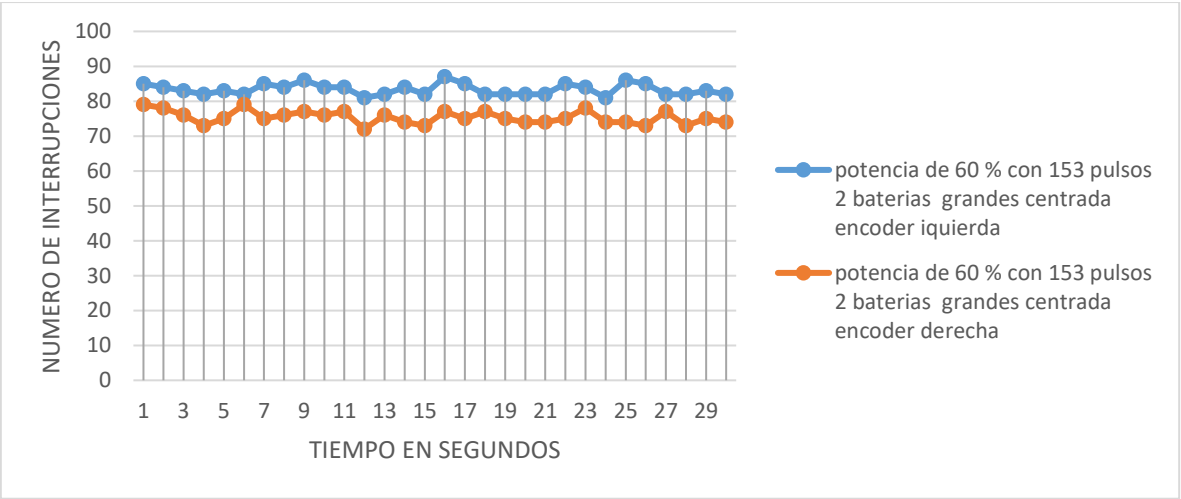
Pruebas de velocidad usando Llantas omnidireccional, con carga de 5k centrada

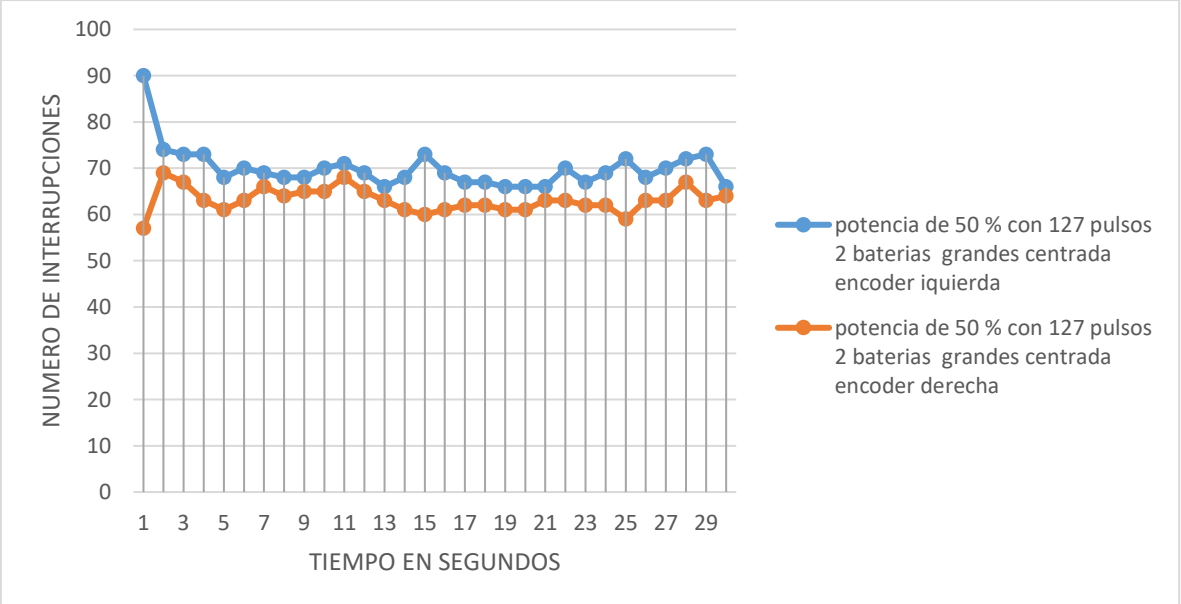


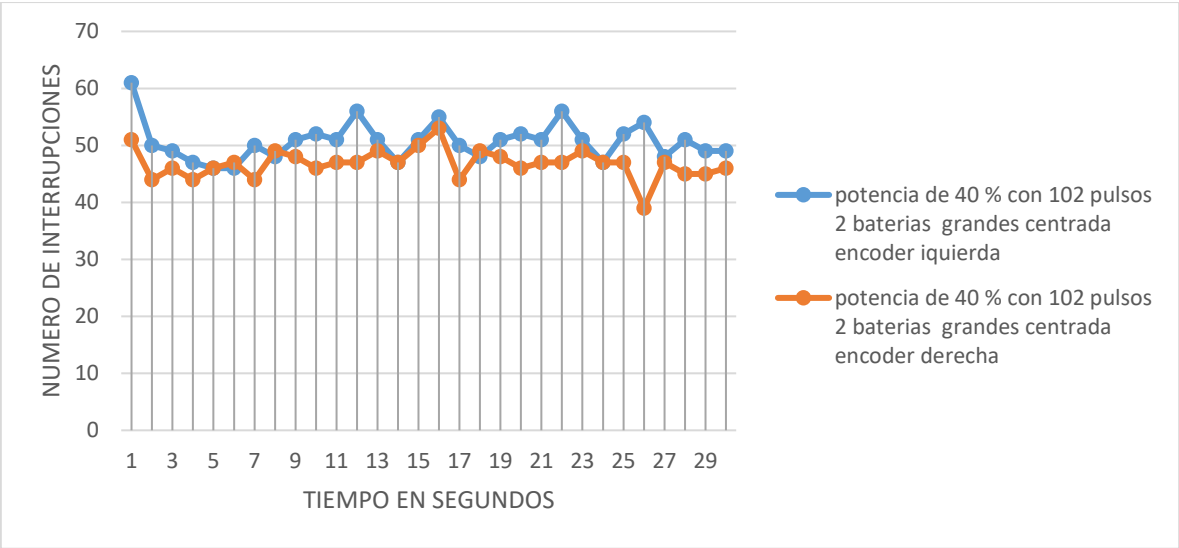




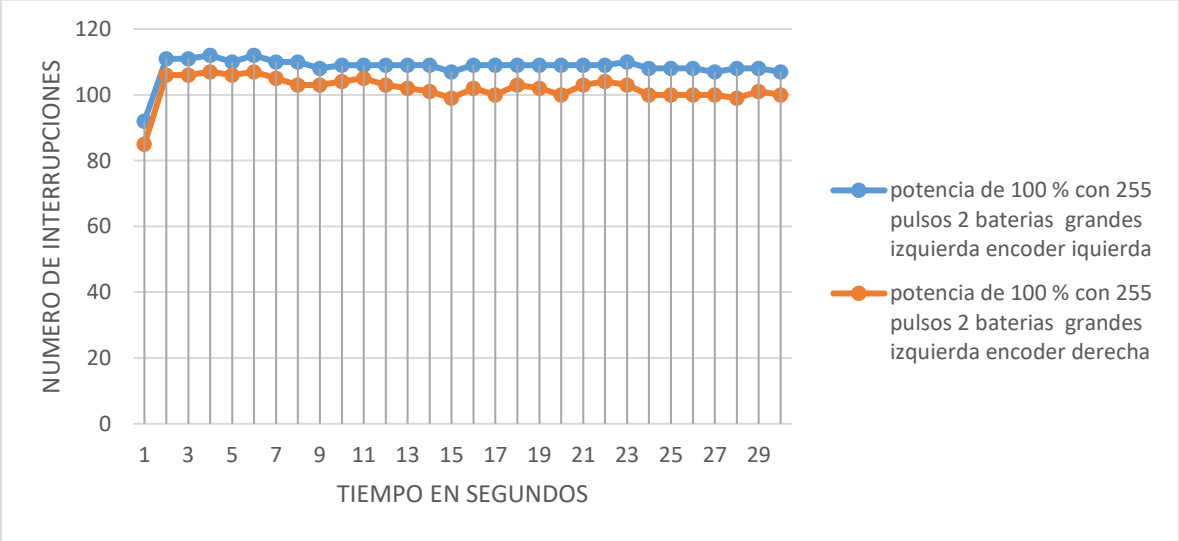

NUMERO DE INTERRUPCIONES
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos 2 baterias grandes centrada encoder iquierda
potencia de 70 % con 178 pulsos 2 baterias grandes centrada encoder derecha

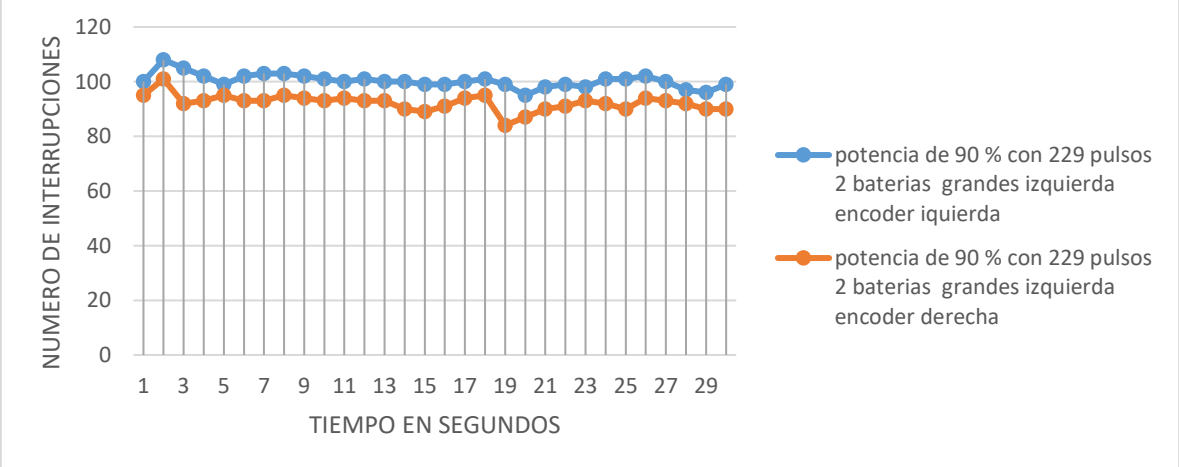

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos 2 baterias grandes centrada encoder iquierda
potencia de 60 % con 153 pulsos 2 baterias grandes centrada encoder derecha

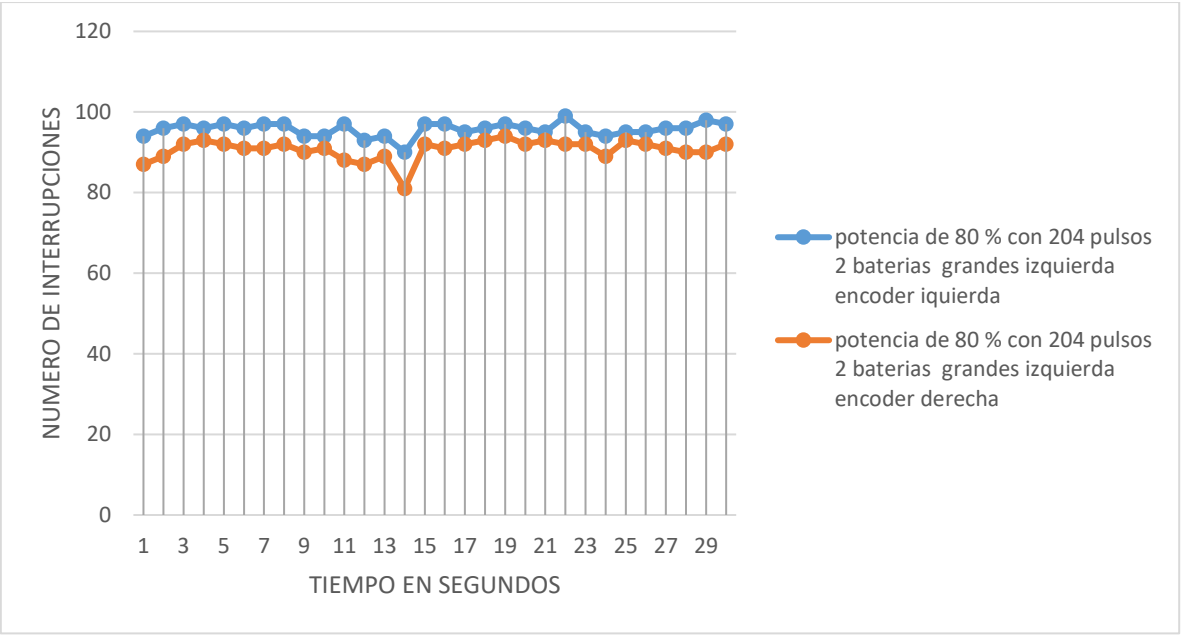

NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos 2 baterias grandes centrada encoder iquierda
potencia de 50 % con 127 pulsos 2 baterias grandes centrada encoder derecha

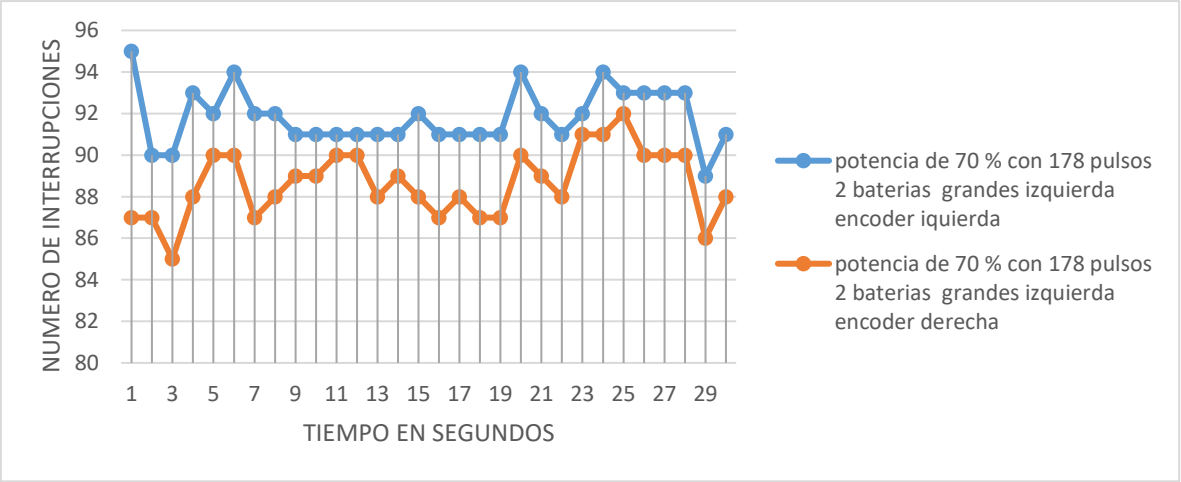


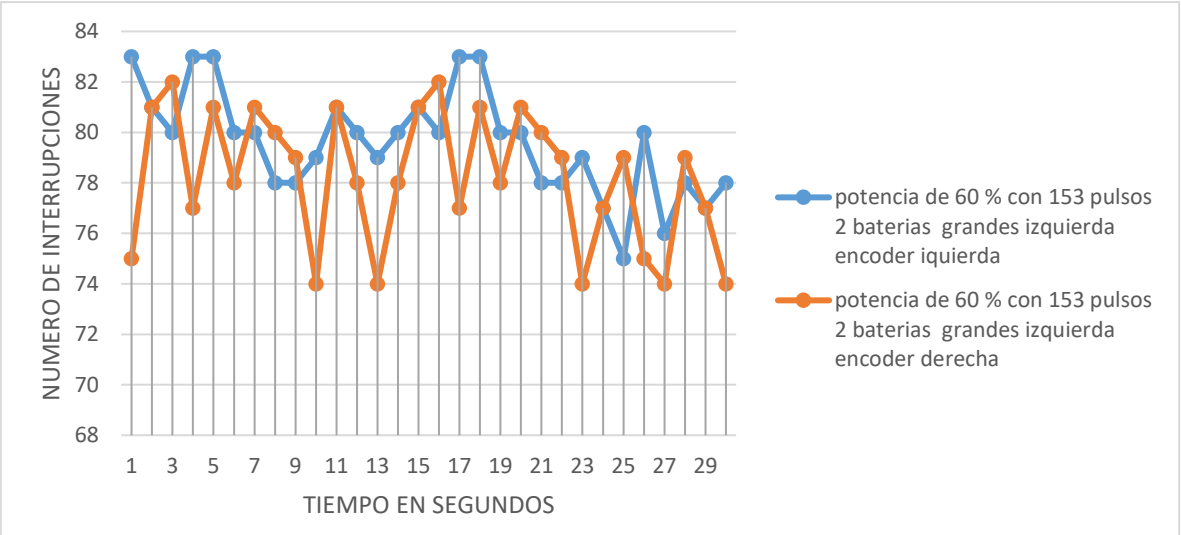
Pruebas de velocidad usando Llantas omnidireccional, con carga de 5k lado izquierdo

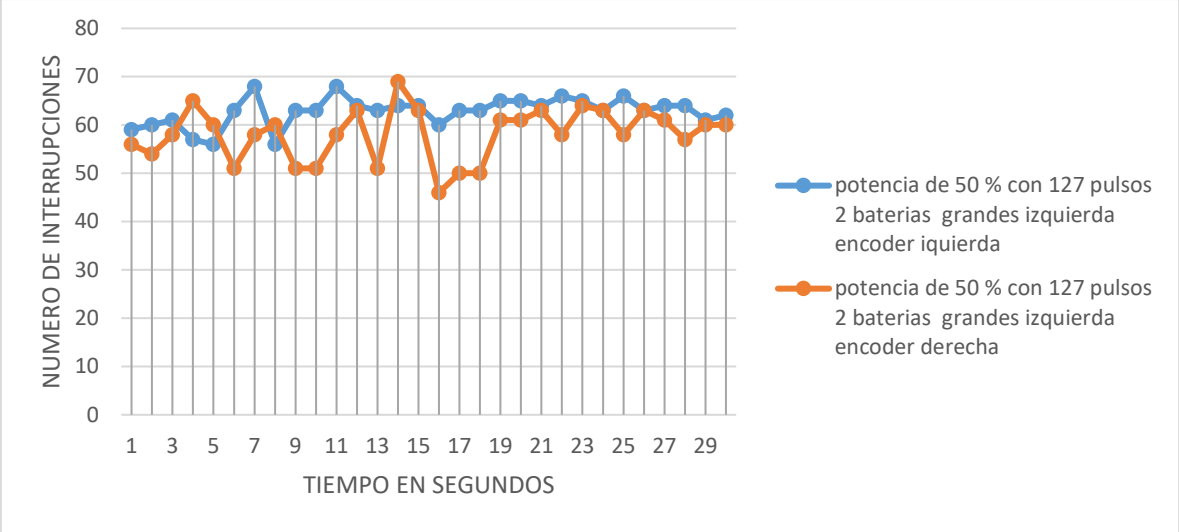


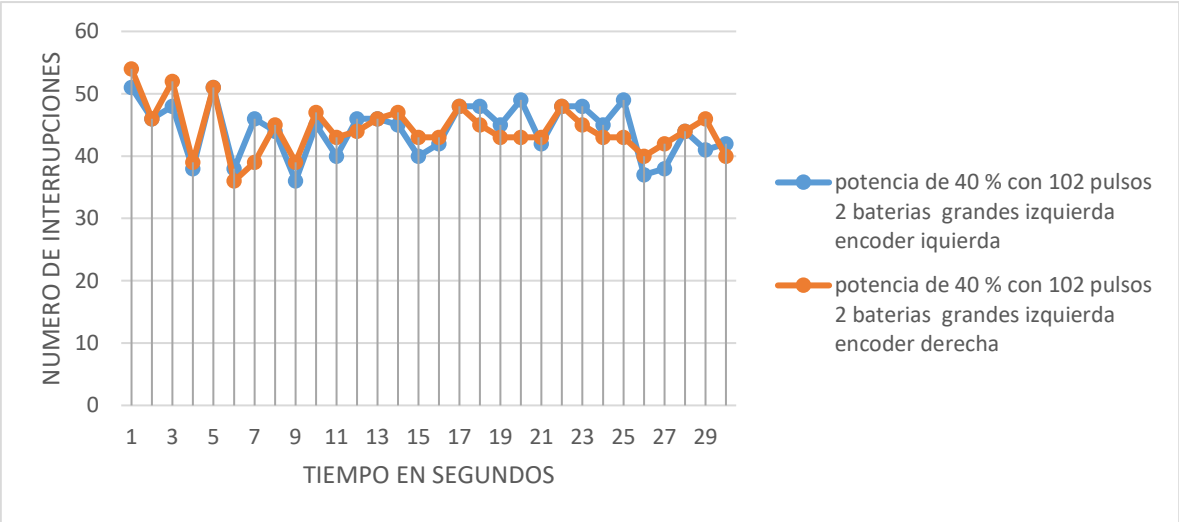



120
100
80
60
40
20
0
NUMERO DE INTERRUPCIONES
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos 2 baterias grandes izquierda encoder iquierda
potencia de 80 % con 204 pulsos 2 baterias grandes izquierda encoder derecha

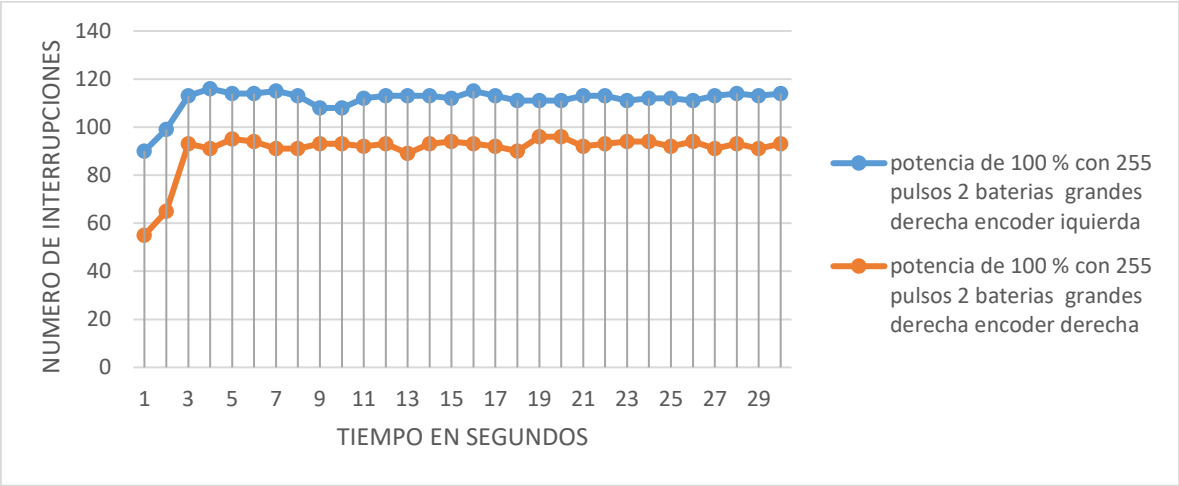

96
94
92
90
88
86
84
82
80
NUMERO DE INTERRUPCIONES
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos 2 baterias grandes izquierda encoder iquierda
potencia de 70 % con 178 pulsos 2 baterias grandes izquierda encoder derecha

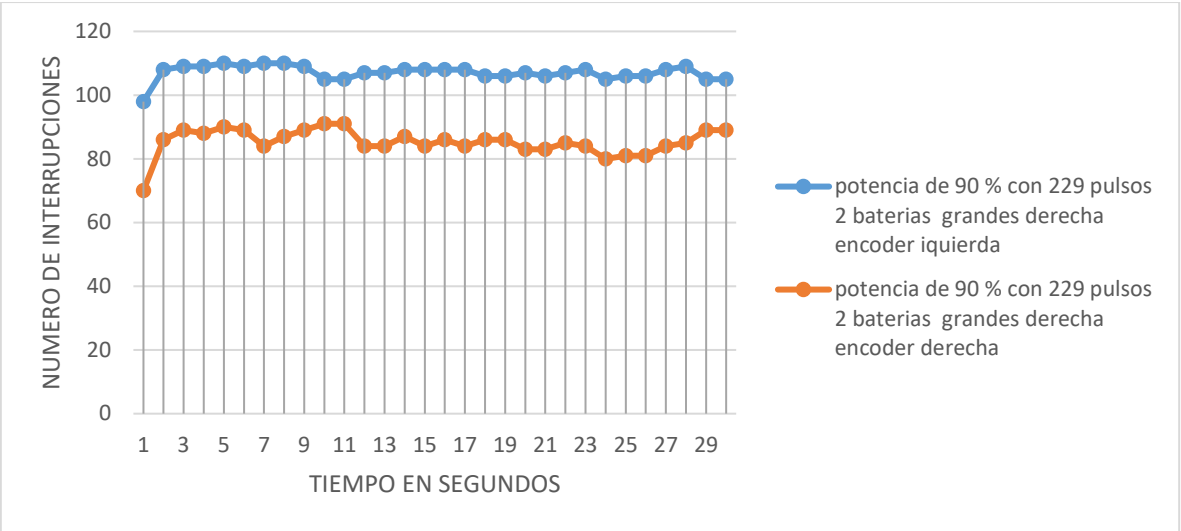

84
82
80
78
76
74
72
70
68
NUMERO DE INTERRUPCIONES
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos 2 baterias grandes izquierda encoder iquierda
potencia de 60 % con 153 pulsos 2 baterias grandes izquierda encoder derecha

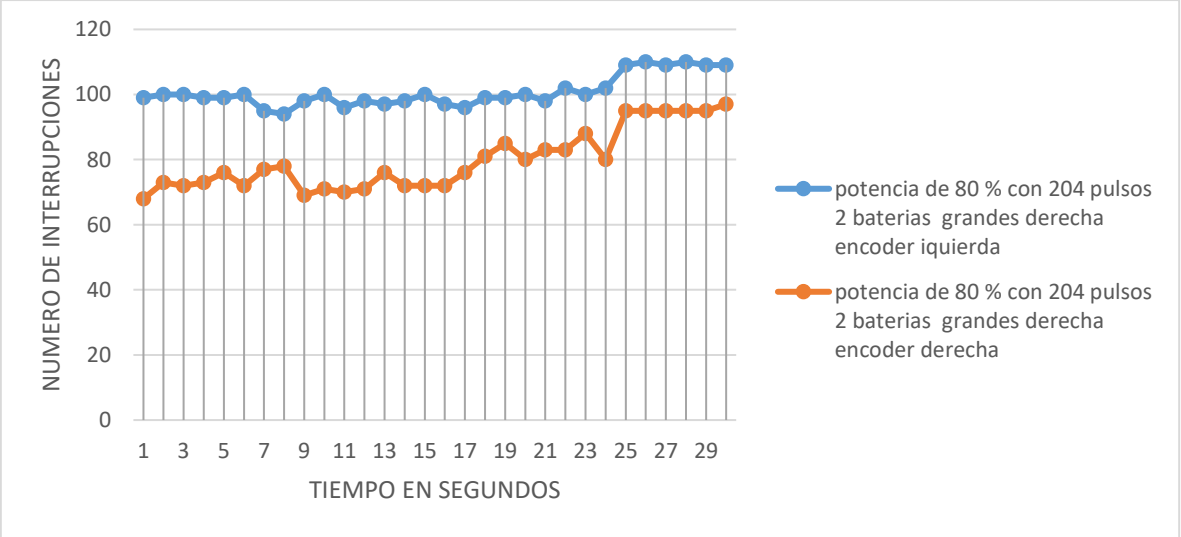


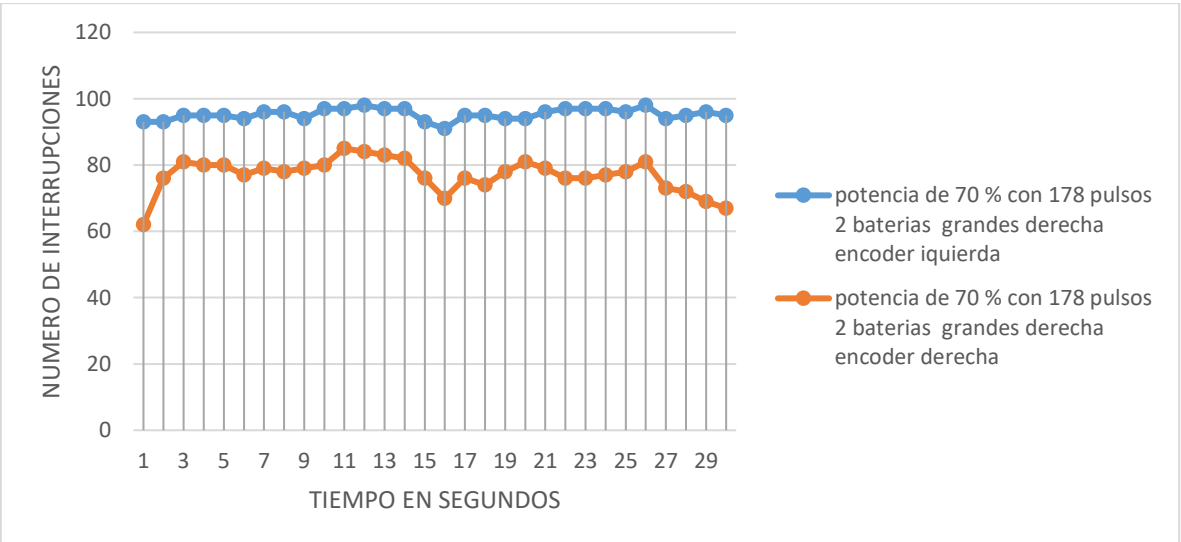


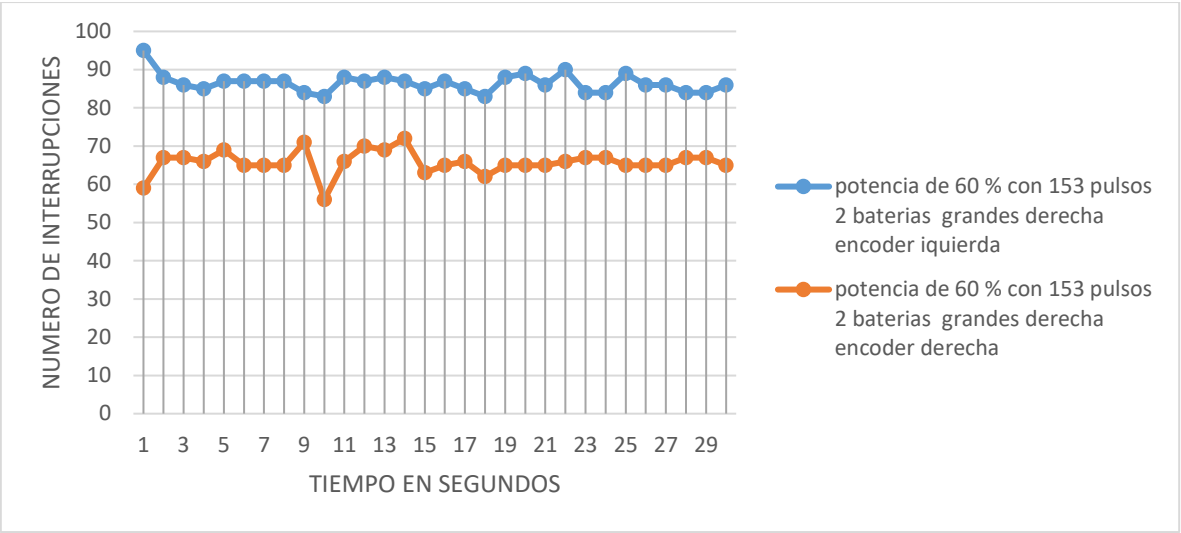
Pruebas de velocidad usando Llantas omnidireccional, con carga de 5k lado derecho

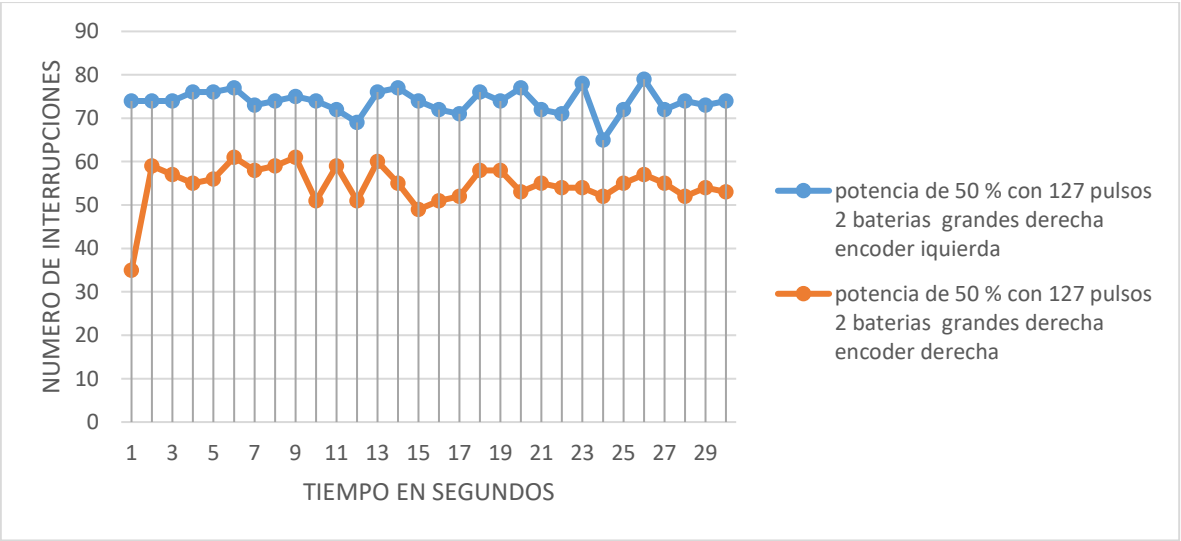


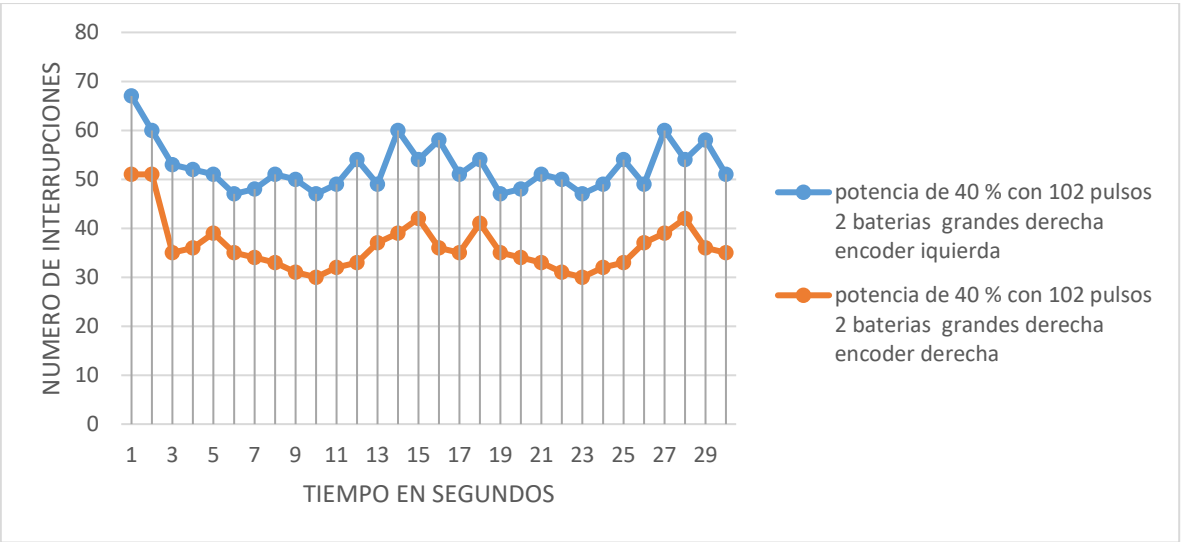

NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 90 % con 229 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 90 % con 229 pulsos 2 baterias grandes derecha encoder derecha


NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 80 % con 204 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 80 % con 204 pulsos 2 baterias grandes derecha encoder derecha


NUMERO DE INTERRUPCIONES
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 70 % con 178 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 70 % con 178 pulsos 2 baterias grandes derecha encoder derecha


NUMERO DE INTERRUPCIONES
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 60 % con 153 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 60 % con 153 pulsos 2 baterias grandes derecha encoder derecha


NUMERO DE INTERRUPCIONES
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 50 % con 127 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 50 % con 127 pulsos 2 baterias grandes derecha encoder derecha


NUMERO DE INTERRUPCIONES
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
TIEMPO EN SEGUNDOS
potencia de 40 % con 102 pulsos 2 baterias grandes derecha encoder iquierda
potencia de 40 % con 102 pulsos 2 baterias grandes derecha encoder derecha