

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Modelado y optimización del control de un
vehículo eléctrico usando PYOMO

Autor: Fernando Estévez García

Tutor: Alejandro Marano Marcolini

**Dpto. Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2021



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Modelado y optimización del control de un vehículo eléctrico usando PYOMO

Autor:

Fernando Estévez García

Tutor:

Alejandro Marano Marcolini

Profesor Contratado Doctor

Dpto. Ingeniería Eléctrica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Grado: Modelado y optimización del control de un vehículo eléctrico
usando PYOMO

Autor: Fernando Estévez García
Tutor: Alejandro Marano Marcolini

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

En el momento que tomé la decisión de introducirme en la rama eléctrica de la ingeniería, no tardé en darme cuenta de lo acertado que estuve. Por seguir el ejemplo de los ingenieros industriales eléctricos presentes en mi familia y por acrecentar la motivación y las ganas de aprender que tengo gracias a ellos. También me gustaría agradecer de manera especial a Alejandro Marano Marcolini, por su atención constante y dedicación para llevar adelante este trabajo. Además de su capacidad de análisis para hacer fácil lo que pueda parecer difícil, he podido trabajar codo con codo con una persona agradable y amable, lo que es de agradecer.

Fernando Estévez García
Sevilla, 2021

Resumen

En este documento se desarrollará el modelo del sistema eléctrico de tracción de un vehículo eléctrico, el Walk Car. Más adelante, ese módulo se usará para optimizar determinados aspectos del funcionamiento del vehículo, utilizando la herramienta Pyomo. Se llevarán a cabo distintas simulaciones y se extraerán los resultados pertinentes para comprender el funcionamiento del modelo. Estos estudios servirán para ejemplificar el funcionamiento de la herramienta Pyomo, paquete de optimización basado en el lenguaje de programación Python, que está experimentando un crecimiento notable en los últimos años en formulación de modelos de optimización.

Abstract

In this document, the model of the electric traction system of an electric vehicle will be developed, the Walk Car. Later, this module will be used to optimize certain aspects of the vehicle's operation, using the Pyomo tool. Different simulations will be carried out and the relevant results will be extracted to understand the operation of the model. These studies will serve to exemplify the operation of the Pyomo tool, an optimization package based on the Python programming language, which has been experiencing notable growth in recent years in the formulation of optimization models.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notación</i>	IX
1 Introducción y Motivación	1
2 El Walk Car	5
2.1 Modos de Funcionamiento	6
2.1.1 Modo de Asistencia para Caminar	7
2.1.2 Modo Normal	7
2.1.3 Modo Deportivo	7
3 Modelos y Simulaciones	9
3.1 Motor de Corriente Continua en Pyomo	9
3.1.1 Modelo Eléctrico y Mecánico	10
Modelo Eléctrico	10
Modelo mecánico	11
3.1.2 Descripción del Modelo e Implementación en Pyomo	11
Librerías	12
Datos y Modelo	12
Variables	13
Ecuaciones Diferenciales	13
Constraints	14
Condiciones Iniciales y Finales, Objetivo y Solver	14
3.1.3 Simulaciones	15
Gráficas y Resultados	15
3.2 Modelo Físico Completo	18
3.2.1 Modelo y Ecuaciones	18
Fuerza de las Ruedas	19
Fuerza Gravitatoria	20
Fuerza de Arrastre Aerodinámico (Drag)	20
Fuerza de Rozamiento	20
3.2.2 Primera Simulación. Tres modos en movimiento rectilíneo	21
Modo de Asistencia para Caminar	22
Modo Normal	25
Modo Deportivo	28

3.2.3 Simulación del efecto de la inclinación	30
3.3 Modelo Físico con Giros y Aceleración Lateral	31
3.3.1 Modelo Matemático y Ejes de Coordinadas	31
3.3.2 Primera Simulación	32
3.3.3 Segunda Simulación	36
3.3.4 Tercera simulación	38
3.3.5 Cuarta Simulación	44
4 Conclusiones y Posibles Ampliaciones	49
4.1 Resultados	49
4.2 Walk Car	50
4.3 Pyomo	50
4.4 Modelo	51
<i>Índice de Figuras</i>	53
<i>Índice de Tablas</i>	55
<i>Bibliografía</i>	57

Notación

e	número e
sen	Función seno
tg	Función tangente
arc tg	Función arco tangente
$\frac{\partial y}{\partial x}$	Derivada parcial de y respecto a x
x°	Notación de grado, x grados.
dx	Diferencial de x
\leq	Menor o igual
\geq	Mayor o igual
i_a	Intensidad de armadura
L_a	Inductancia de armadura
v_a	Tensión de armadura
R_a	Resistencia de armadura
L_{af}	Inductancia mutua
ω_e	Frecuencia eléctrica
ω_r	Frecuencia mecánica
i_f	Intensidad de campo
L_f	Inductancia de campo
V_f	Tensión de campo
R_f	Resistencia de campo
J_r	Constante de inercia
τ_r	Par electromagnético
τ_f	Par de fricción
τ_l	Par de carga
K_f	Constante de fricción
M	Masa total
R_w	Radio de la rueda
K_w	Constante de transmisión
f_d	Fuerza drag
f_β	Fuerza gravitatoria
f_r	Fuerza de rozamiento
f_w	Fuerza de rueda
ω_w	Velocidad de rueda
v	Velocidad absoluta
τ_w	Par de rueda
g	Constante gravitacional

ρ_{aire}	Densidad del aire
S_f	Sección frontal
C_x	Coefficiente aerodinámico
C_{rr}	Coefficiente de rozamiento
x	Coordenada horizontal
y	Coordenada vertical
θ	Ángulo de inclinación con respecto a la horizontal
ϕ	Ángulo de inclinación de las ruedas
a_r	Aceleración lateral
p	Potencia
v	Velocidad
a	Aceleración

1 Introducción y Motivación

Es imposible obviar el enorme desarrollo que está viviendo la movilidad urbana gracias a las necesidades de los ciudadanos, nuevos dispositivos eléctricos salen a la luz con el fin de mejorar los trayectos diarios y a raíz de este hecho la competitividad de las marcas hace que el desarrollo de estos dispositivos sea constante, por lo que su estudio tiene un efecto muy interesante. Resulta evidente que nuestros trayectos diarios están cambiando, la demanda y la necesidad de reducir el tiempo de viaje al trabajo, a la universidad o al colegio han provocado que sistemas como el patinete eléctrico, la bicicleta eléctrica o distintos vehículos de transporte ligero eléctricos birrueda copen el mercado y sean cada vez más populares.



Figura 1.1 Distintos dispositivos de movilidad eléctrica urbana.

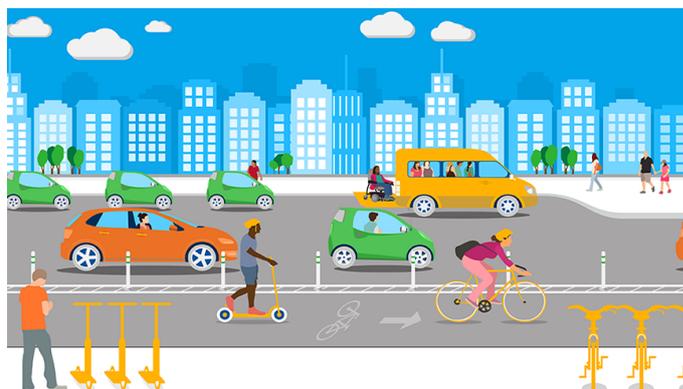


Figura 1.2 Modelo de una ciudad actual que integra distintos sistemas de movilidad.

Uno de estos dispositivos, el Walk Car, está expandiéndose de manera vertiginosa en Asia, y seguramente en los próximos años lo veamos ya por las calles en Europa. Desarrollado por Cocoa

Motors [1], una startup japonesa, fue lanzado en el año 2015 como la solución final en un país con grandes problemas de movilidad. Acapararon la atención de todos presentándolo como la solución de movilidad eléctrica más compacta del mundo. Se ha elegido este dispositivo de movilidad y no otro por el desafío que implica, por su inmensa popularidad y por la motivación que supone estudiar un modelo que han llevado a cabo estudiantes de una universidad japonesa, que gracias a su dedicación decidieron crear una empresa y tener un impacto en la sociedad, tratando de mejorar la movilidad urbana y dar solución a los problemas que tienen los seres humanos, lo que es básicamente la Ingeniería. De esta forma, estudiaremos el Walk Car en todas sus variantes, con diferentes objetivos para cada simulación en nuestro estudio, sacando los resultados y conclusiones pertinentes para entender cómo funciona el que está llamado a cambiar la forma en la que nos desplazamos.



Figura 1.3 Walk Car.



Figura 1.4 Logotipo de Cocoa Motors.

Surge entonces la pregunta de qué herramienta vamos a utilizar para analizar y desgranar el Walk Car, para comprenderlo y sacar resultados. La respuesta la encontramos en el lenguaje de programación Python, que con su programación funcional, su orientación a objetos y su carácter multiparadigma ha revolucionado el mundo tal y como lo conocemos hoy día. Creado por el holandés Guido van Rossum a finales de los años ochenta, le dió el nombre por su afición al grupo humorístico británico Monty Python. Hoy en día, es utilizado en todos los campos de inteligencia artificial y machine learning. Resulta curioso descubrir que Python consta de una filosofía muy concreta, que fue enunciada en una serie de postulados por el desarrollador de Python Tim Peters, que bien podría ser aplicada a cualquier campo de la Ingeniería [3].

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.

- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente, a menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una, y preferiblemente solo una, manera obvia de hacerlo. Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.
- Ahora es mejor que nunca, aunque nunca es a menudo mejor que ya mismo.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (namespaces) son una gran idea ¡Hagamos más de esas cosas!

Dentro de Python utilizaremos el entorno Spyder, y Pyomo será la herramienta que nos permitirá realizar nuestro análisis. Desarrollada por Sandia National Laboratories y la Universidad de California, es una colección de paquetes de software para implementar modelos de optimización como el nuestro. Se desarrolla como parte del proyecto COIN-OR (Computational Infrastructure for Operations Research). ¿Quién utiliza Pyomo? Se viene utilizando mayoritariamente por investigadores para resolver problemas complejos del mundo real, ya que es una herramienta que está en constante desarrollo. Pyomo integra un módulo que permite modelar ecuaciones diferenciales en modelos de optimización. Este módulo, denominado DAE, es la herramienta en la que se basa el trabajo [5].



Figura 1.5 Logotipo de Pyomo DAE.

Entre los objetivos de este trabajo, destacan estudiar la viabilidad, efectividad y rapidez de Pyomo para resolver los distintos modelos que se van a plantear, así como la precisión del solver utilizado. Para determinar la efectividad de Pyomo, se observará cómo consigue esta herramienta alcanzar la función objetivo para cada simulación, optimizando. Además, trataremos de entender el comportamiento del Walk Car para los diferentes modelos, con sus limitaciones y modos de funcionamiento.

En el primer capítulo se realizará una introducción sobre nuestro dispositivo, el Walk Car. En el se describirán sus características y aspectos técnicos y se desarrollarán de forma teórica sus modos de funcionamiento. Más adelante, en el segundo capítulo se describirán los modelos matemáticos empleados para llevar a cabo los distintos modelos y optimizaciones de los mismos. Finalmente, en el último capítulo se desarrollarán las conclusiones sobre los resultados obtenidos y se tratará de comentar una serie de posibles mejoras o ampliaciones para este trabajo.

2 El Walk Car

En este capítulo profundizaremos sobre los aspectos físicos y técnicos de nuestro dispositivo, tales como sus dimensiones, materiales, potencia y funcionalidades. Salvando las diferencias, bien podría decirse que el Walk Car es algo así como un skateboard eléctrico con la forma de un ordenador portátil de 13 pulgadas. Su reducido tamaño y masa, de tan sólo 2.9 kg, permiten llevarlo fácilmente en el interior de un bolso o una mochila. De ahí su tan destacado carácter compacto. Sus dimensiones, 346 mm de largo, 215 mm de ancho y 74 mm de altura, permiten alcanzar un nivel de portabilidad óptimo[1].

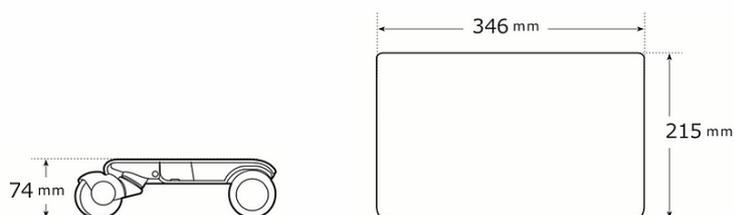


Figura 2.1 Dimensiones del Walk Car.

En cuanto a los materiales empleados para su construcción, para la superficie de contacto, donde el individuo sitúa sus pies se utiliza un material de policarbonato, lo que aporta una gran resistencia al impacto. Además, está recubierta con una pintura que permite evitar rayones y mejorar el agarre. Se emplea el carbono, con un espesor de 3 mm, con el propósito de realizar un sólido muy ligero y resistente, ya que en estos dos aspectos es muy superior al hierro. El cierto carácter elástico del Walk Car hace que los neumáticos se agarren de manera óptima a la superficie de la carretera y permitan una conducción flexible como lo haría un skateboard o patín. El bastidor que soporta las cuatro ruedas, tanto las motoras delanteras como las traseras, utiliza productos de superduraluminio que son empleados en las alas de los aviones para poder conseguir esa ligereza y resistencia que tanto se busca en este tipo de vehículos. Todo esto aporta un concepto de diseño minimalista, eliminando el exceso de materiales y llevando la ligereza al límite máximo. Además, está diseñado para que no se puedan ver tornillos que perjudiquen el diseño, y el aspecto simple del que goza hace que sea incluso un artículo de moda que resulta agradable usar y llevar.

Las ruedas delanteras motoras, de unos 6 cm de diámetro, constan de un motor ultracompac-

to de corriente continua que va dispuesto en el interior de cada una. Dichos motores aportan conjuntamente una potencia nominal de salida de nuestro dispositivo de 260 W en total.[1]. El Walk Car tiene suspensión independiente en las 4 ruedas, además de la suspensión vertical, las ruedas delanteras están equipadas con una suspensión total que suaviza el impacto de caminos con cualquier superficie irregular y logra un alto rendimiento de carrera incluso para los neumáticos de pequeño diámetro que tiene nuestro dispositivo.



Figura 2.2 Suspensión del Walk Car.

Al usar un sensor de 4 puntos integrado en la superficie de contacto, se puede acelerar, frenar y girar con una simple operación de los pies orientada mover el centro de gravedad. Con un simple movimiento de rodillas podremos virar a la derecha o a la izquierda, gozando de un confort y una suavidad en el trazado óptimos y con nada más que levantar ligeramente los dedos de los pies el dispositivo realizará un frenado progresivo.

La carga máxima que soporta es de unos 80 kg, y la mínima de unos 30 kg, con lo que un niño de menos de 10 años no debería utilizar el dispositivo. Dicha carga máxima se reduce hasta los 70 kg si se quiere conseguir sobrepasar pendientes de 10 °, lo que tendremos en cuenta en nuestro estudio. Además, el rango de temperaturas de uso oscila entre -10 °C y 40 °C, lo que es importante tener en cuenta si nos encontramos en verano, cuando las temperaturas puedan sobrepasar dicho límite.

En cuanto a su autonomía, consta de una batería de ion de litio que se carga completamente en 60 min. Sin embargo, con 30 min de carga la batería quedará cargada casi al total de su capacidad. El Walk Car permite realizar la carga de batería de pie, lo que permite hacerlo en cualquier lugar, la oficina o una cafetería, sin estorbar u ocupar más espacio de la cuenta.

Con el fin de que el individuo pueda tener una conducción agradable y sin sobresaltos, el Walk Car incorpora una serie de medidas de seguridad que enunciamos a continuación. Ya se ha mencionado la baja altura del dispositivo de tan solo 74 mm, dicha altura permite que la persona pueda bajarse del dispositivo en cualquier momento ante cualquier situación de peligro, lo que resulta una medida de seguridad muy interesante. Pero sin duda el aspecto más atractivo en cuanto a funcionalidades es el frenado automático, que permite que el Walk Car no se acelere en exceso en bajadas con una pendiente considerable y además emplea la energía regenerativa para cargar baterías y aumentar el rango de cruce.

2.1 Modos de Funcionamiento

Pasamos ahora a describir los 3 modos de funcionamiento de los que consta el Walk Car, describiendo posibles usos para cada uno de ellos y sus características más relevantes.

Tabla 2.1 Parámetros básicos del Walk Car.

Parámetro	Valor
Dimensiones	346x215x74 mm
Masa	2.9 kg
Potencia de salida	260 W
Rango de carga	30-80 kg

2.1.1 Modo de Asistencia para Caminar

Este modo permite conducir con confianza a la velocidad de lo que bien podría ser un paseo rápido. Es adecuado también para ir en paralelo con otros peatones o incluso para su uso en interior, como ahora veremos. Cocoa Motors insta a utilizar este modo no solo para ir por la ciudad, sino para utilizarlo para necesidades básicas o en la industria. Por ejemplo, si una persona que está llevando a otra en una silla de ruedas no quiere realizar esfuerzo tan solo tiene que montarse en el Walk Car y este aplicará la fuerza necesaria para realizar el movimiento. En la industria también podría ser utilizado para que el operario correspondiente pueda transportar la carga de una manera más sencilla y rápida. Este modo tiene una autonomía de 7 km y puede llegar a alcanzar una velocidad máxima de 6 km/h.

**Figura 2.3** El Walk Car en la industria.

2.1.2 Modo Normal

Este modo es posiblemente el más adecuado para viajar lejos y dar un paseo disfrutando del paisaje a la velocidad que uno normalmente tendría si decide correr. Tiene una velocidad máxima de 10 km/h y una autonomía de 7 km, al igual que en el modo de asistencia para caminar. Quizá sería este el modo más óptimo para ir en paralelo con una persona que vaya corriendo o haciendo footing.

2.1.3 Modo Deportivo

Es adecuado cuando la preferencia es moverse rápido o disfrutar de una conducción algo más activa a la velocidad similar a la que pueda tener una bicicleta. Quizá sea este el modo más empleado para esos trayectos conocidos del trabajo a casa, y emplear el menor tiempo posible en ellos. Tiene

una velocidad máxima de 16 km/h y una autonomía con distancia de cruce de 5 km, algo más reducida que en los dos modos anteriores. Esto último puede llegar a ser un inconveniente si buscamos conseguir cubrir un largo trayecto a una velocidad alta, para lo que quizá sería más conveniente el modo normal.



Figura 2.4 El Walk Car virando en modo normal.

3 Modelos y Simulaciones

En este capítulo describiremos en primer lugar de manera breve cómo opera Pyomo y para ello realizaremos un modelado de un motor de corriente continua, que bien podría ser el que lleve el Walk Car en sus ruedas motoras delanteras, de forma que podamos ir introduciendo el modelo completo del Walk Car. Iremos describiendo paso a paso cada bloque de la estructura del código, una vez descrito el modelo y sus ecuaciones.

3.1 Motor de Corriente Continua en Pyomo

Para comenzar a familiarizarnos con el entorno de Pyomo y simplificar nuestro sistema, llevaremos a cabo el modelado y optimización de un motor de corriente continua. Debemos tener en cuenta que nuestro Walk Car lleva instalados dos motores de corriente continua en cada una de sus ruedas delanteras, que ejercen de motoras. Para esta primera simulación hemos escogido un motor que simula a la perfección la máquina que llevarán nuestras ruedas del Walk Car. Teniendo en cuenta los datos facilitados por Cocoa Motors y atendiendo al tamaño ultracompacto de nuestro motor, se ha escogido un modelo que tiene una potencia de salida nominal de 260 W y un par máximo ejercido de 1000 mNm. Además, la tensión a la que está sometido nuestro motor será de 24 V y será capaz de soportar una intensidad máxima de 17.1 A. De esta forma, para esta simulación inicial se tomará un motor de 260 W para ver la respuesta del sistema, y en las siguientes dividiremos esa potencia se repartirá entre los dos motores.



Figura 3.1 Motor ultracompacto del Walk Car.

Para desarrollar el modelo que describe nuestro motor, debemos tener en cuenta tanto el modelo

eléctrico como el mecánico, y combinando ambos podremos realizar nuestra simulación y optimizar el sistema completo. Empezaremos por tanto a describir en primer lugar de forma teórica ambos modelos, describiendo las ecuaciones, variables y parámetros de cada uno de ellos.[6].

3.1.1 Modelo Eléctrico y Mecánico

Modelo Eléctrico

El circuito de un motor de corriente continua se puede dividir en dos, de forma que dividiremos nuestro estudio entre un circuito que llamaremos de campo y otro circuito que denominaremos de armadura. Ambos circuitos constarán por supuesto de su propia tensión e intensidad, además de una resistencia e inductancia propias, y una inductancia mutua que los relaciona.

A continuación se muestran las dos ecuaciones diferenciales que modelan ambas partes, donde los valores constantes se indican en mayúsculas y las variables dependientes en minúsculas. Se muestran también los circuitos de campo y armadura y la tabla de parámetros con sus respectivos valores que modelan el comportamiento de nuestro motor de corriente continua [2].

$$\frac{di_a}{dt} = \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.1)$$

$$\frac{di_f}{dt} = \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.2)$$

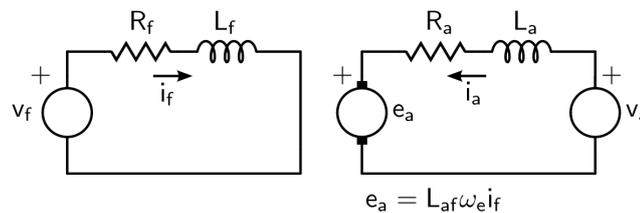


Figura 3.2 Circuitos de campo y armadura.

Tabla 3.1 Valores constantes del motor.

Parámetro	Valor
L_a	0.001882 H
R_a	0.06727 Ω
L_{af}	0.06641 H
L_f	0.2166 H
V_f	24 V
R_f	1 Ω

Es importante denotar que el valor de la frecuencia eléctrica ω_e es igual al número de pares de polos de nuestro sistema por la frecuencia del rotor. En nuestro caso, tendremos un par de polos, por lo que ambos valores son iguales, luego ω_e y ω_r tendrán el mismo valor.

Modelo mecánico

Una vez desarrollada la parte eléctrica del motor con sus respectivos circuitos, pasamos ahora a detallar los elementos mecánicos que tendremos en cuenta para el modelado de nuestro motor. Nos centraremos en desarrollar la 2ª Ley de Newton, que relaciona un sumatorio de pares de fuerza con la variación de velocidad, de esta manera, obtenemos la siguiente ecuación.

$$\frac{d\omega_r}{dt} = \frac{1}{J_r} \cdot (\tau_r - K_f \cdot \omega_r - \tau_l) \quad (3.3)$$

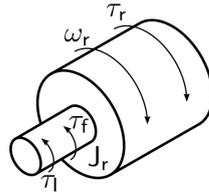


Figura 3.3 Esquema representativo del motor.

Donde J_r es nuestra constante de inercia que adoptará un valor constante por el momento, pero más tarde en otras simulaciones será sustituida por otro conjunto de valores que forman un valor más realista. El par de fricción, τ_f , está compuesto por el producto de la constante de fricción por la velocidad angular del rotor. La frecuencia del rotor como ya se ha mencionado en el apartado anterior, será igual a la eléctrica ya que nuestro número de pares de polos es igual a la unidad. El par de carga τ_l será un conjunto de fuerzas mecánicas que actuarán sobre nuestro sistema, pero que se explicarán más adelante en el modelo mecánico completo. Por el momento tomaremos este valor como nulo para observar el comportamiento del motor sin que esté sometido a ninguna carga externa. Finalmente, el par electromagnético del motor τ_r , es igual al producto de inductancia mutua, intensidad de campo e intensidad de armadura, quedando de la siguiente manera.

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.4)$$

3.1.2 Descripción del Modelo e Implementación en Pyomo

En este apartado describimos las condiciones, variables y restricciones de nuestro primer modelo, el motor de corriente continua. Para ello, y por ser este el primer modelo que optimizamos, explicaremos de manera detallada la implantación del modelo en el código de Pyomo paso a paso, para su mejor comprensión. Como se ha mencionado en apartados anteriores, procedemos al modelado, optimización y simulación de un motor de corriente continua. En este primer estudio nuestro objetivo será alcanzar la máxima potencia, es decir 260 W, en el menor tiempo posible. Eso sí, para ello habrá que fijar límites en algunas otras variables. La estructura de implementación del código seguirá siempre la misma estructura [4]. Para nuestro modelo, enunciado a continuación, trataremos de minimizar el valor del tiempo final t_f . Para ello reescalamos el problema de manera que el horizonte temporal forme parte de nuestro cálculo de optimización. Recordemos que nuestra variable de control es la tensión de armadura y que debemos imponer el límite de 260 W para la potencia de salida. Queda también definida la potencia empleada como el producto del par electromagnético aplicado por el motor y la velocidad angular del mismo.

$$\min : Z = t_f \quad (3.5)$$

s.a

$$\frac{d\omega_r}{dt} = t_f \cdot \frac{1}{J_r} \cdot (\tau_r - K_f \cdot \omega_r) \quad (3.6)$$

$$\frac{di_a}{dt} = t_f \cdot \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.7)$$

$$\frac{di_f}{dt} = t_f \cdot \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.8)$$

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.9)$$

$$p = \tau_r \cdot \omega_r \quad (3.10)$$

$$p \leq 260 \quad (3.11)$$

$$p \geq 0 \quad (3.12)$$

$$\tau_r \leq 1 \quad (3.13)$$

Librerías

En esta parte del código invocamos las librerías que vamos a utilizar. Las más conocidas son la librería Matplotlib, esencial para representar en forma de gráficas los resultados de nuestras simulaciones, y la librería Numpy que nos permite trabajar con vectores y matrices. También importamos otras como Shutil, Sys y DAE, herramienta perteneciente al bloque Pyomo gracias a la cual podemos resolver nuestras ecuaciones diferenciales, y otras librerías para manejar archivos de alto nivel.

```
#imports
import matplotlib.pyplot as plt
import numpy as np
import shutil
import sys
import os.path
from pyomo.environ import*
from pyomo.dae import*
```

Datos y Modelo

Introducimos aquí los datos descritos en los apartados de Modelo Eléctrico y Mecánico, y con una línea de código indicamos a Pyomo el nombre del modelo que vamos a llevar a cabo.

```
#Engine data

J = 0.01
K_f = 0.0001
```

```

R_a = 0.06727
L_a = 0.001882
R_f = 1
L_f = 0.2166
L_af = 0.06641
V_f = 24
R_w = 0.03

#creating the model object
m = ConcreteModel()

```

Variables

Aquí introducimos todas las variables que vamos a utilizar y sobre todo, la de control, que en nuestro caso será la tensión de armadura v_a , aplicada al ejercer el movimiento de pies por parte del individuo. Dicha tensión influye en la intensidad de armadura y esta última en el par aplicado. Así que podríamos decir que realmente lo que el individuo aplica al mover sus pies es una subida o bajada de tensión al motor, que hace que aumente la intensidad y a la postre, el par aplicado.

También cabe destacar que tendremos que declarar el 'tiempo final' como variable, ya que en esta primera simulación el optimizado del sistema consiste en llevar a cabo las condiciones finales, que más tarde veremos, en el menor tiempo posible.

```

#creating the model object
m = ConcreteModel()

#defining the independent variable

m.tf = Var(domain=NonNegativeReals)
m.t = ContinuousSet(bounds=(0, 1))

#defining control inputs

m.v_a = Var(m.t)

#define the dependent variables

m.omega_r = Var(m.t)
m.i_a = Var(m.t)
m.i_f = Var(m.t)
m.tau = Var(m.t)

```

Ecuaciones Diferenciales

Pasamos ahora a la declaración de las tres ecuaciones diferenciales que modelan nuestro sistema, las variaciones de intensidades del modelo eléctrico y la variación de velocidad del modelo mecánico. Para poder resaltar el modelo, ya que el tiempo final es lo que intentamos minimizar, debemos multiplicar cada ecuación diferencial por el tiempo final.

```

#define derivatives

m.omega_r_dot = DerivativeVar(m.omega_r)

```

```

m.i_a_dot = DerivativeVar(m.i_a)
m.i_f_dot = DerivativeVar(m.i_f)

#define the differential equation as constraints

m.ode_omega_r = Constraint(m.t, rule=lambda m, t:m.omega_r_dot[t] == m.
    tf*(1/J)*(m.tau[t] - K_f*m.omega_r[t]))
m.ode_i_a = Constraint(m.t, rule=lambda m, t:m.i_a_dot[t] == m.tf*(1/L_a
    )*(m.v_a[t] - R_a*m.i_a[t] - L_af*m.omega_r[t]*m.i_f[t]))
m.ode_i_f = Constraint(m.t, rule=lambda m, t:m.i_f_dot[t] == m.tf*(1/L_f
    )*(V_f - R_f*m.i_f[t]))

```

Constraints

Las restricciones de nuestro modelo Pyomo las interpreta como Constraints, en ellas fijaremos tanto algunas fórmulas como los valores máximos y mínimos de algunos parámetros que resultan ser esenciales para nuestro modelo. En la siguiente línea de código se muestra un ejemplo de la sintaxis y en la tabla algunos de los límites más significativos para las variables en esta simulación, en la que nuestro objetivo será alcanzar la máxima potencia en el menor tiempo posible.

```

m.power1 = Constraint(m.t, rule=lambda m, t:m.power[t]>=0)
m.power2 = Constraint(m.t, rule=lambda m, t:m.power[t]<=260)
m.tau1 = Constraint(m.t, rule=lambda m, t:m.tau[t]>=0)
m.tau2 = Constraint(m.t, rule=lambda m, t:m.tau[t]<=1)

```

Tabla 3.2 Valores límite.

Valor	Rango
Potencia	0-260 W
Par	0-1000 mNm
Intensidad	0-17.1 A

Condiciones Iniciales y Finales, Objetivo y Solver

Vemos ahora una de las partes más importantes del modelo, que resulta esencial para resolver cualquier modelo que implique una serie de ecuaciones diferenciales como el nuestro: las condiciones iniciales y finales. Para ello, se creará una lista con todos los valores por cada variable, definiendo de forma óptima el modelo y permitiendo optimizarlo en el sentido que nos convenga, en nuestro caso en el de alcanzar la máxima potencia en el menor tiempo posible.

```

#initial conditions

m.pc = ConstraintList()

m.pc.add(m.omega_r[0]==0)
m.pc.add(m.i_a[0]==0)
m.pc.add(m.i_f[0]==0)
m.pc.add(m.v_a[0]==0)

```

```
#final conditions

m.pc.add(m.omega_r[1]>=0)
m.pc.add(m.i_a[1]>=0)
m.pc.add(m.i_f[1]>=0)

#final conditions on the control inputs

m.pc.add(m.v_a[1]>=0)
m.pc.add(m.power[1]==260)
```

Tabla 3.3 Condiciones iniciales y finales.

Variable	Condición inicial	Condición final
Velocidad del motor	0 rad/s	≥ 0 rad/s
Intensidades de armadura y campo	0 A	≥ 0 A
Tensión de armadura	0 V	≥ 0 V
Potencia	0 W	260 W

Será necesario declarar también el objetivo del modelo, que como hemos mencionado será minimizar la variable del tiempo final `m.tf`, y a su vez debemos indicarle a Pyomo cuál es el solver deseado para resolver nuestro problema de optimización. En nuestro caso utilizaremos el solver `ipopt` y el método de diferencias finitas, que permite discretizar ecuaciones diferenciales y un tratamiento más simple del problema diferencial parcial.

```
#define the optimization objective

m.obj = Objective(expr=m.tf)

#transform and solve

TransformationFactory('dae.finite_difference').apply_to(m, wrt=m.t, nfe
=30)
SolverFactory('ipopt').solve(m).write()
```

3.1.3 Simulaciones

Gráficas y Resultados

Una vez implementado el código resolvemos en apenas unos 0.034 s, según el solver, y obtenemos como resultado que el modelo alcanza el objetivo de llevar el motor a una potencia de 260 W en unos 2.6 segundos, aproximadamente. De esta forma, representamos los siguientes valores, que son los más representativos para entender el comportamiento del motor.

Tanto la potencia como la velocidad del motor suben de manera lineal y uniforme. La velocidad del motor sube hasta las 2500 rpm para potencia nominal, lo que resulta un valor bastante lógico dado

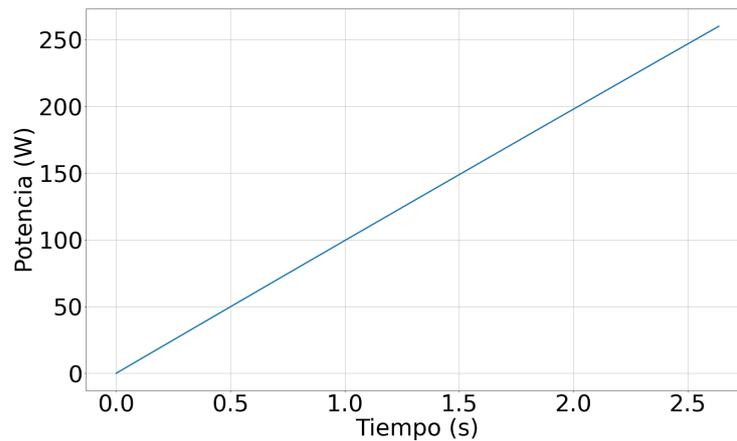


Figura 3.4 Potencia de un motor de 260 W.

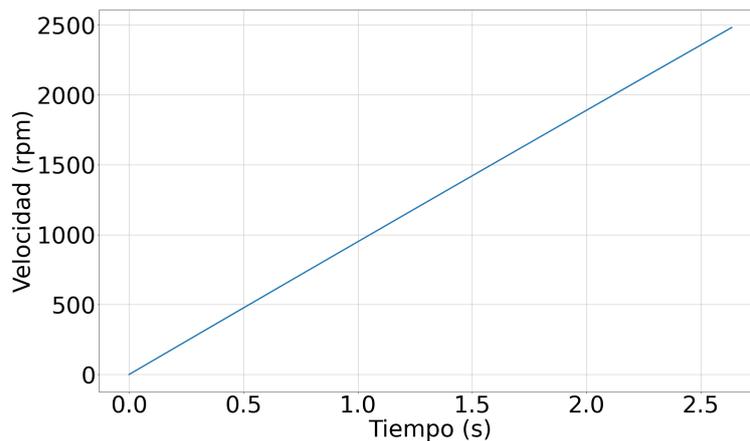


Figura 3.5 Velocidad del motor en rpm.

el reducido tamaño de nuestro motor.

Nuestra variable de control, la tensión de armadura, sube hasta llegar a aplicar un valor cercano a los 400 V si se quiere alcanzar la potencia nominal perseguida de 260 W.

Ambas intensidades (armadura y campo) tienen un pico inicial de 2 A pero se estabilizan en régimen permanente hasta llegar a 0.6 A, aproximadamente.

El par aplicado, ya que solo tiene como restricción un valor máximo de 1000 mNm, sube rápidamente hacia ese valor, con el fin de poder alcanzar la consigna de potencia en el menor tiempo posible.

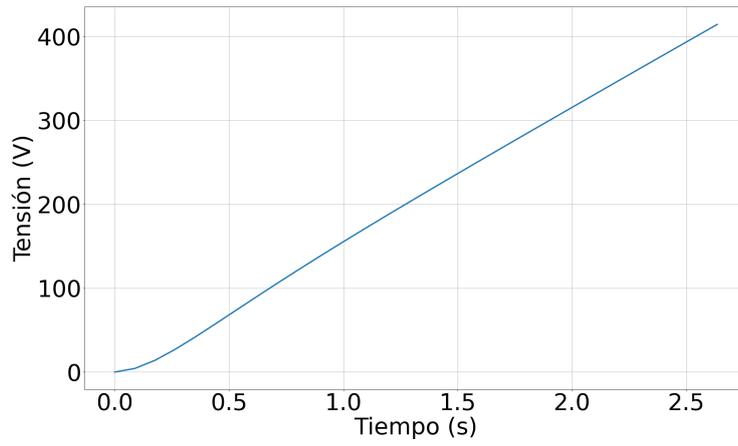


Figura 3.6 Tensión de armadura para motor de 260 W.

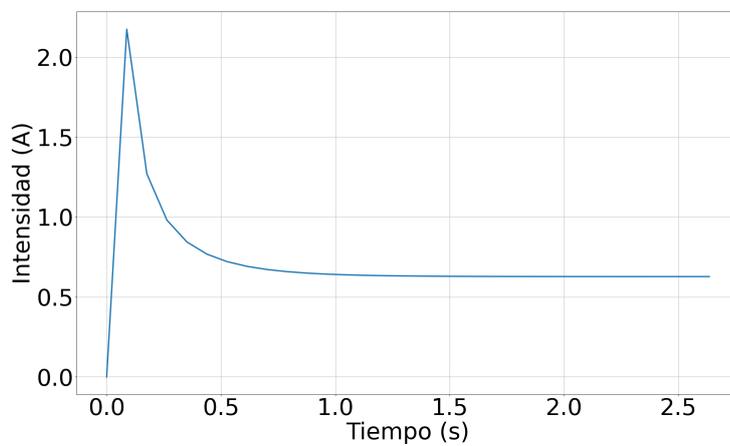
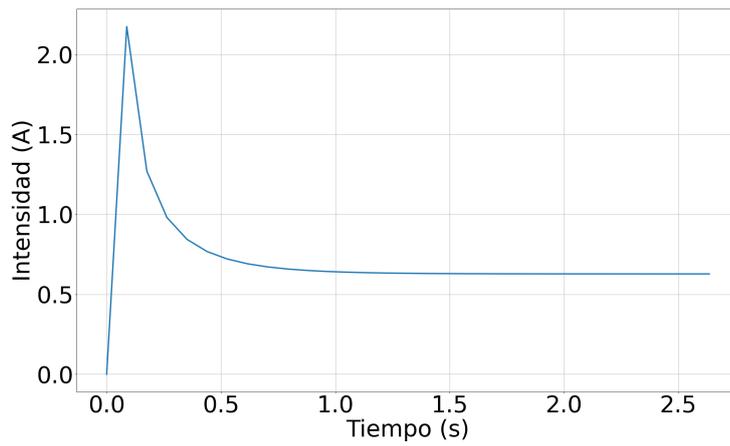


Figura 3.7 Intensidades de campo y armadura para un motor de 260 W.

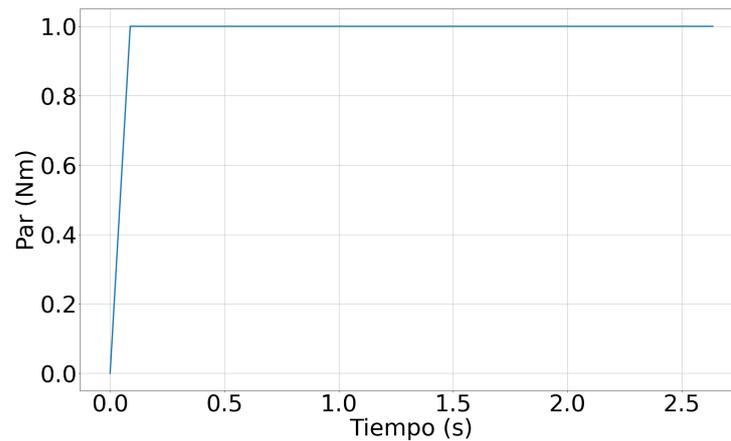


Figura 3.8 Par electromagnético aplicado para un motor de 260 W.

3.2 Modelo Físico Completo

Una vez demostrado el optimizado de un modelo simple como es el motor de corriente continua, en este apartado se realizará la explicación de todas las variables que se han tenido en cuenta para llevar a cabo el sistema global físico, así para conseguir una simulación lo más realista posible dentro de lo que pueda ser un entorno urbano. Junto con las ecuaciones explicadas en apartados anteriores del motor podremos llegar, gracias a la relación de transmisión y otros parámetros, al movimiento de la rueda por el asfalto y una velocidad global del sistema en km/h. Para ello emplearemos distintas fuerzas como la de la gravedad, rozamiento o drag, en la que hemos tenido que emplear parámetros de gran importancia en cuanto a aerodinámica y rozamiento de cualquier vehículo.

3.2.1 Modelo y Ecuaciones

Como se ha introducido, para analizar el comportamiento del sistema completo debemos considerar una serie de aspectos que tienen una influencia mayúscula en todos los sistemas de movilidad mecánica. Recordando la ilustración de la sección anterior que mostraba el diseño del motor de corriente continua, ahora debemos incluir a ese motor la rueda del dispositivo y el sistema de transmisión, que serán los encargados de que nuestro sistema se mueva a través de la superficie. Además de todas las variables y parámetros antes mencionados en los apartados del modelo eléctrico y mecánico del motor, ahora veremos nuevas variables que formarán parte de nuestro modelo.

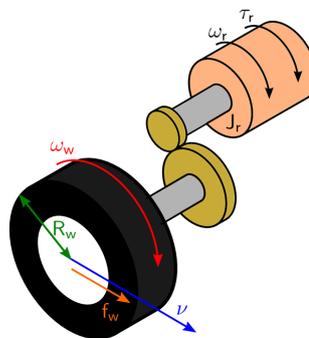


Figura 3.9 Esquema del sistema de transmisión.

Ahora, debemos duplicar tanto el par electromagnético como el par de fricción, ya que nuestro modelo consta de dos motores. Recordando la ecuación que modelaba la velocidad del rotor quedaba definida una constante J_r y un par de carga τ_l que considerábamos nulo para modelar el motor de corriente continua. Pues bien, estas dos variables ahora tomarán un valor totalmente diferente, quedando la velocidad del rotor del motor de la siguiente manera:

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + f_\beta + 4 \cdot f_r)) \quad (3.14)$$

Como vemos se han introducido distintas variables, entre ellas destaca la relación de transmisión K_w y el radio de la rueda R_w que serán los que determinarán la velocidad de la rueda de nuestro Walk Car. El radio de la rueda se tomará de 0.03 m tal y como especifica Cocoa Motors y la relación de transmisión se ha tomado con un valor de 0.9, ya que el motor va dentro de la propia rueda y girará a una velocidad similar, eso sí, teniendo en cuenta algunas pérdidas. De esta forma, relacionamos el valor de la velocidad angular del motor con la de la rueda, y dicha velocidad angular de la rueda con la velocidad lineal de nuestro sistema.

$$\omega_w = K_w \cdot \omega_r \quad (3.15)$$

$$v = R_w \cdot \omega_w \quad (3.16)$$

Para el sistema mecánico, como se puede ver en la ecuación por la que se rige la velocidad del motor, también deberemos tener en cuenta ciertas variables de fuerza que se oponen o que apoyan el movimiento del dispositivo y que resultan indispensables para modelar nuestro sistema de movilidad. Dichas fuerzas son las contempladas en cualquier sistema mecánico y se detallan a continuación.

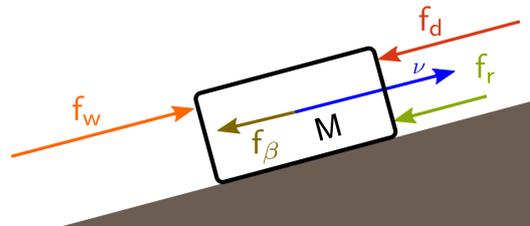


Figura 3.10 Esquema de fuerzas.

Fuerza de las Ruedas

Primeramente, deberemos considerar la fuerza que ejercen las ruedas, en nuestro caso las dos delanteras, ya que son las que llevan los motores de corriente continua y son las que ejercerán el par. En nuestras ecuaciones, deberemos considerar esta fuerza como doble por el hecho de contar con dos ruedas motoras, cuestión que ya se ha tenido en cuenta al duplicar el par aplicado. Así, nuestra fuerza motora total queda como:

$$f_w = \frac{\tau_w}{R_w} \quad (3.17)$$

$$\tau_w = \frac{\tau_r}{K_w} \quad (3.18)$$

Fuerza Gravitatoria

En segundo lugar, y como se muestra en la ilustración más abajo, nuestro sistema es probable que se enfrente normalmente a una inclinación, así que deberemos considerar esa fuerza a vencer, que dependerá de dicho ángulo de inclinación. Recordemos que nuestro Walk Car es capaz de soportar una inclinación máxima de 10° , eso sí, con una carga bastante reducida. M , que representa la masa que engloba tanto al Walk Car como al individuo, se ha tomado de 75 kg. El ángulo de inclinación β se tomará como nulo para nuestras primeras simulaciones, y se estudiará su impacto en un apartado independiente.

$$f_\beta = g \cdot M \cdot \sin \beta \quad (3.19)$$

Fuerza de Arrastre Aerodinámico (Drag)

Además, todo sistema mecánico se enfrenta siempre a la oposición del viento, lo que conocemos como fuerza del drag. Nuestro caso resulta peculiar ya que la sección de nuestro sistema se compone de la base del Walk Car además de la silueta frontal del individuo que va sobre él. Teniendo esto en cuenta, se ha tomado una sección frontal de 0.6 m^2 . La densidad del aire de valor 1.225 kg/m^3 , también tiene una gran influencia a la hora de considerar la acción del viento. El coeficiente aerodinámico C_x se ha tomado de valor 0.9, teniendo en cuenta coeficientes similares para casos como el de un ciclista.

$$f_d = \frac{1}{2} \cdot \rho_{\text{aire}} \cdot S_f \cdot C_x \cdot v^2 \quad (3.20)$$

Fuerza de Rozamiento

Al circular por las aceras de la ciudad, también tendremos un cierto coeficiente de rozamiento por el contacto de las ruedas con la superficie de contacto. Dicha fuerza de rozamiento a vencer será multiplicada por 4 ya que son 4 ruedas las que deben vencer esa fuerza. Tomaremos un C_{rr} de valor 0.005.

$$f_r = C_{rr} \cdot M \cdot g \quad (3.21)$$

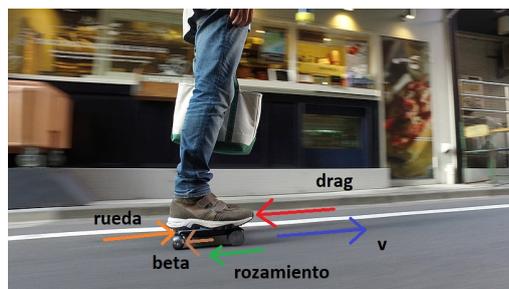


Figura 3.11 Esquema de fuerzas en el Walk Car.

3.2.2 Primera Simulación. Tres modos en movimiento rectilíneo

Ahora, teniendo en cuenta todas las variables y fuerzas antes explicadas llevaremos a cabo las primeras simulaciones del modelo global. Todas las simulaciones que se muestran en este apartado consisten en alcanzar el máximo valor de velocidad permitido para cada modo de uso: asistencia para caminar, normal y rápido. En todas ellas, para este primer ensayo, supondremos un movimiento rectilíneo y con un ángulo de inclinación 0° . Además, para limitar la aceleración, informamos a Pyomo que la máxima aceleración permitida es de 0.8 m/s^2 , favoreciendo así un arrancado suave que permita al individuo permanecer de manera estable sobre el Walk Car. Es importante para entender los resultados, recordar que estamos ante lo que bien podría ser un experimento en el que medimos con un cronómetro, en una carretera aislada, cuánto tiempo tarda el Walk Car en alcanzar la velocidad máxima de cada modo.

$$\min : Z = t_f \quad (3.22)$$

s.a

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + 4 \cdot f_r)) \quad (3.23)$$

$$\frac{di_a}{dt} = t_f \cdot \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.24)$$

$$\frac{di_f}{dt} = t_f \cdot \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.25)$$

$$\frac{dv}{dt} = t_f \cdot a \quad (3.26)$$

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.27)$$

$$p = \tau_r \cdot \omega_r \quad (3.28)$$

$$\omega_w = \omega_r \cdot K_w \quad (3.29)$$

$$v = \omega_w \cdot R_w \quad (3.30)$$

$$p \leq 260 \quad (3.31)$$

$$p \geq 0 \quad (3.32)$$

$$\tau_r \leq 1 \quad (3.33)$$

$$a \leq 0.8 \quad (3.34)$$

```
#define the optimization objective
```

```
m.obj = Objective(expr=m.tf)
```

Esta vez no describiremos el código de forma detallada. Declararemos las restricciones más importantes en cada caso (modo asistencia para caminar, normal y deportivo) y definiremos las condiciones iniciales y finales del sistema, además del objetivo de optimización que como ya se ha comentado, sigue siendo el de minimizar el tiempo.

Modo de Asistencia para Caminar

Recordemos que para este modo el Walk Car alcanzaba una velocidad máxima de 6 km/h. Esto unido al objetivo de minimizar el tiempo en el que se alcanza la velocidad objetivo y a las siguientes condiciones y restricciones arrojará unos resultados óptimos. Las nuevas restricciones y condiciones iniciales y finales quedan enunciadas de la siguiente manera:

- Potencia inferior a 260 W durante todo el intervalo.
- Aceleración inferior al límite impuesto de 0.8 m/s^2

Tabla 3.4 Modo de Asistencia para Caminar.

Valor	Condición inicial	Condición final
Velocidad del motor	0 rad/s	$\geq 0 \text{ rad/s}$
Intensidades de armadura y campo	0 A	$\geq 0 \text{ A}$
Tensión de armadura	0 V	$\geq 0 \text{ V}$
Velocidad	0 m/s	1.7 m/s (6 km/h)

El Walk Car alcanza el objetivo de llegar a una velocidad de 6 km/h en 2.1 s, lo que resulta razonable teniendo en cuenta el límite de aceleración que favorece un arrancado suave y progresivo. Además, por ser esta la primera simulación en la que imponemos dicho límite, representamos la aceleración. Se puede comprobar que alcanza rápidamente el valor máximo, ya que gracias a ello alcanzaremos la velocidad objetivo lo más rápido posible. Para esta primera simulación del modo asistencia para caminar mostraremos una gran cantidad de valores, que iremos omitiendo en las siguientes por obviedad.

En cuanto a la potencia, como cabía esperar la potencia empleada en el modo asistencia para caminar es reducida, llegando a los 60 W para la velocidad objetivo. La velocidad del motor por tanto, también estará muy lejos de su valor nominal, llegando a las 600 rpm. La variable de control, tensión de armadura, que debe ser aplicada por el individuo con los pies será de 100 V. En cuanto a las dos intensidades, se observa que se alcanza el régimen permanente rápidamente tras un pequeño pico de intensidad en el régimen transitorio.

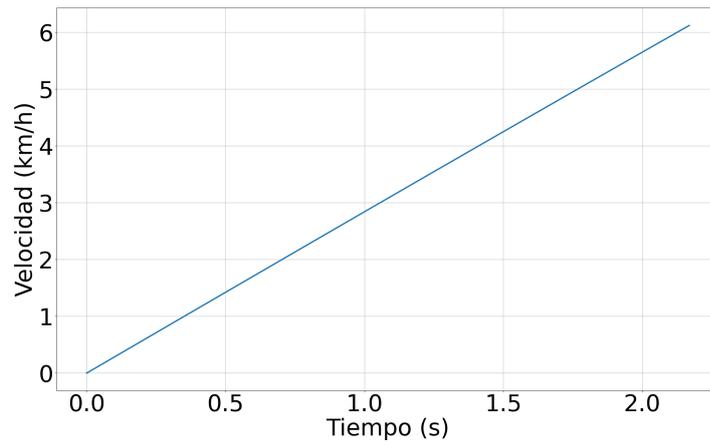


Figura 3.12 Velocidad en Modo Asistencia para Caminar.

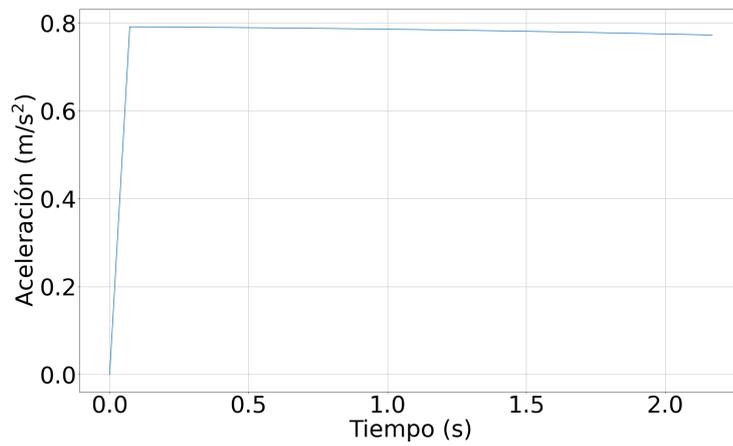


Figura 3.13 Aceleración en Modo Asistencia para Caminar.

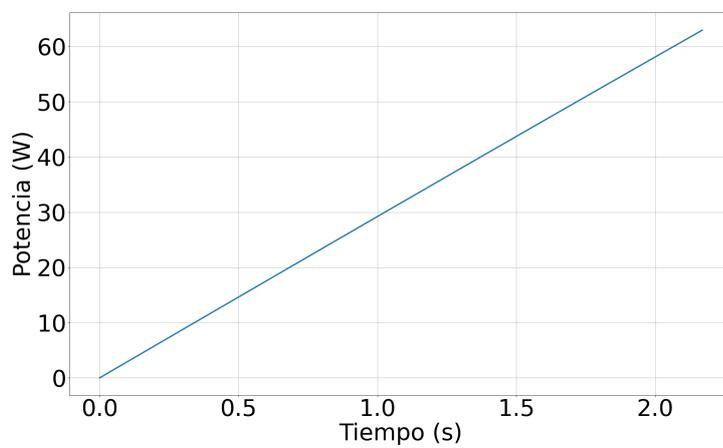


Figura 3.14 Potencia empleada en Modo Asistencia para Caminar.

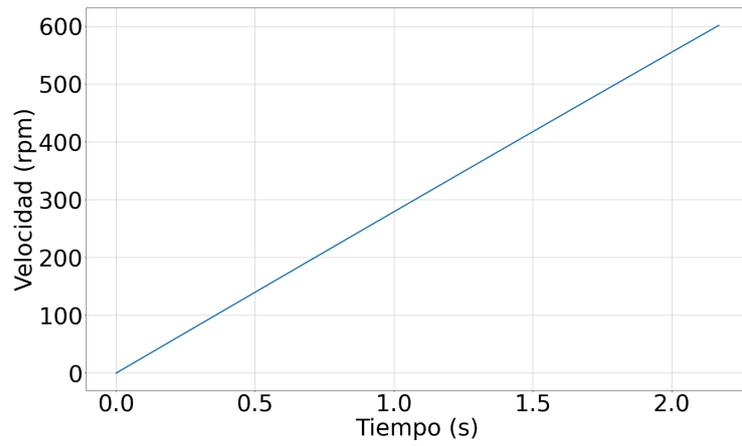


Figura 3.15 Velocidad del motor en Modo Asistencia para Caminar.

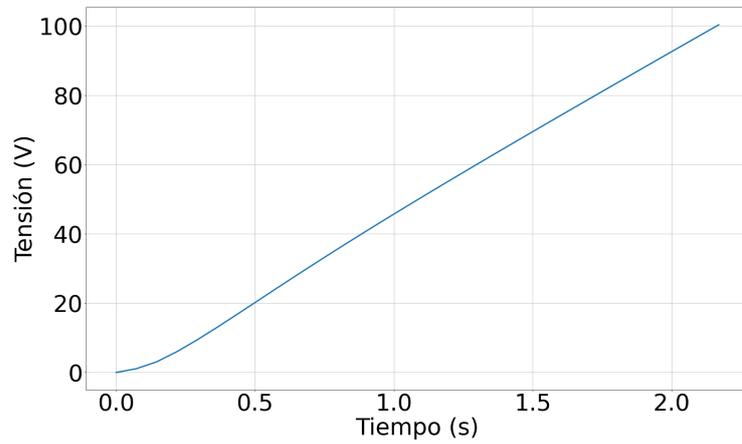


Figura 3.16 Tensión de armadura en Modo Asistencia para Caminar.

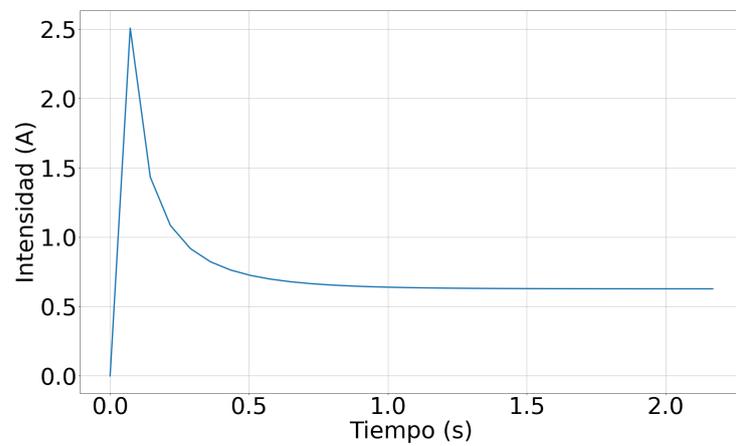


Figura 3.17 Intensidad de armadura en Modo Asistencia para Caminar.

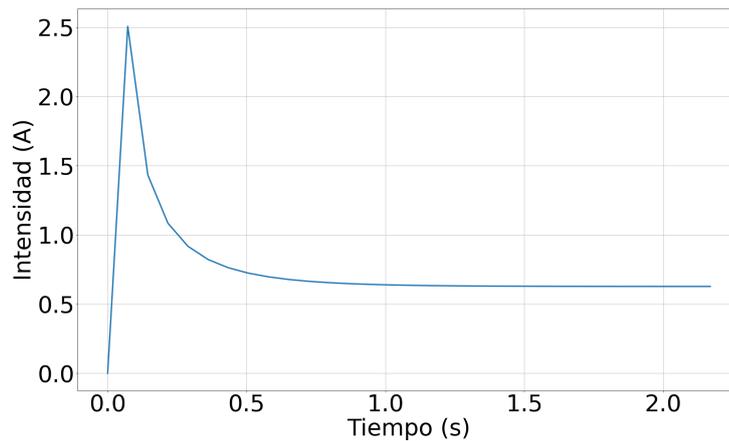


Figura 3.18 Intensidad de campo en Modo Asistencia para Caminar.

Modo Normal

Para este modo normal el Walk Car alcanzaba una velocidad máxima de 10 km/h. Esto unido al objetivo de minimizar el tiempo en el que se alcanza la velocidad objetivo y a las siguientes condiciones y restricciones arrojará unos resultados que interpretamos a continuación. Ahora obviaremos algunas gráficas que se han comentado para el modo normal, que se consideran repetitivas. Las restricciones y condiciones iniciales y finales quedan enunciadas de la siguiente manera:

- Potencia inferior a 260 W durante todo el intervalo.
- Aceleración inferior al límite impuesto de 0.8 m/s^2

Tabla 3.5 Modo Normal.

Valor	Condición inicial	Condición final
Velocidad del motor	0 rad/s	$\geq 0 \text{ rad/s}$
Intensidades de armadura y campo	0 A	$\geq 0 \text{ A}$
Tensión de armadura	0 V	$\geq 0 \text{ V}$
Velocidad	0 m/s	2.8 m/s (10 km/h)

El Walk Car alcanza el objetivo de llegar a una velocidad de 10 km/h en 3.6 s, un segundo y medio más tarde, (aproximadamente) lo que resulta razonable teniendo en cuenta la subida de velocidad con respecto al modo anterior, manteniendo ese arrancado suave y progresivo. En cuanto a la potencia, como cabía esperar la potencia empleada en el modo normal sube considerablemente con respecto al modo anterior y alcanza el valor de 100 W cuando el Walk Car circula a 10 km/h. La velocidad del motor sigue muy lejos de su valor nominal, llegando a las 1000 rpm. La variable de control, tensión de armadura, que debe ser aplicada por el individuo con los pies sube hasta llegar a un valor de 155 V. En cuanto a las dos intensidades, se observa que se alcanza el régimen permanente rápidamente tras un pequeño pico de intensidad en el régimen transitorio, de nuevo como en el caso anterior.

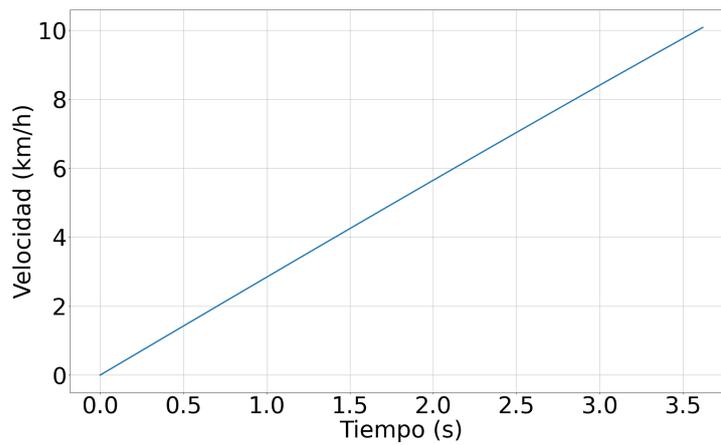


Figura 3.19 Velocidad del Walk Car en Modo Normal.

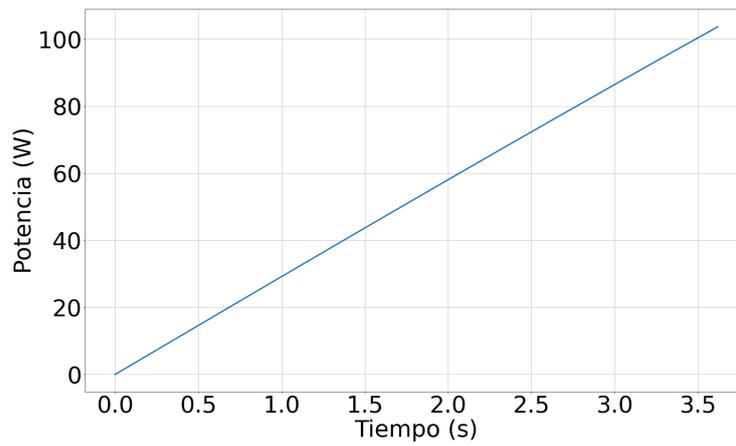


Figura 3.20 Potencia empleada en Modo Normal.

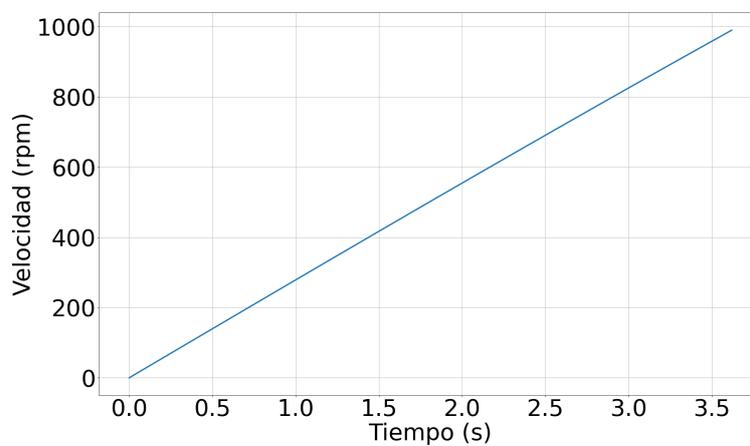


Figura 3.21 Velocidad del motor en Modo Normal.

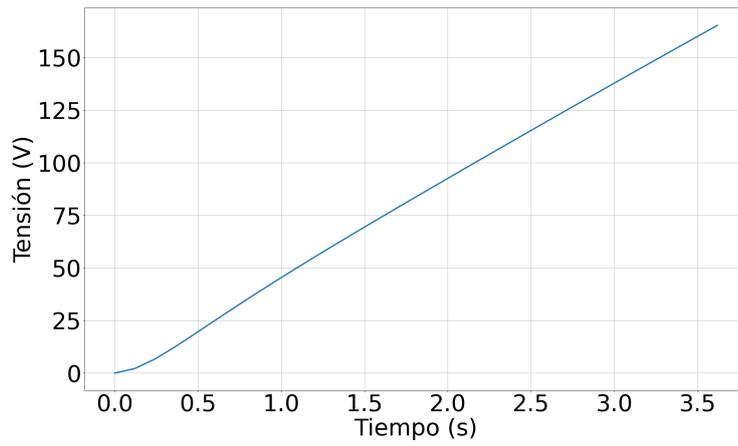


Figura 3.22 Tensión de armadura en Modo Normal.

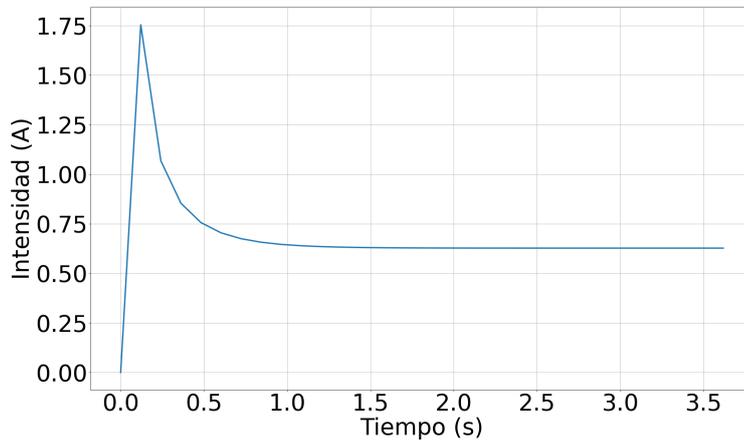


Figura 3.23 Intensidad de armadura en Modo Normal.

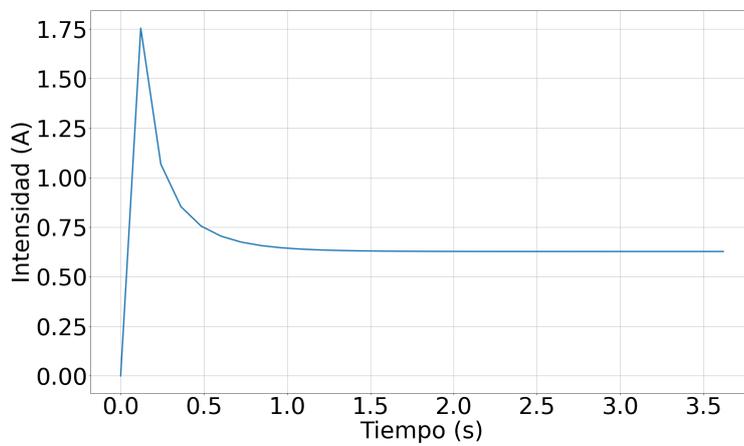


Figura 3.24 Intensidad de campo en Modo Normal.

Modo Deportivo

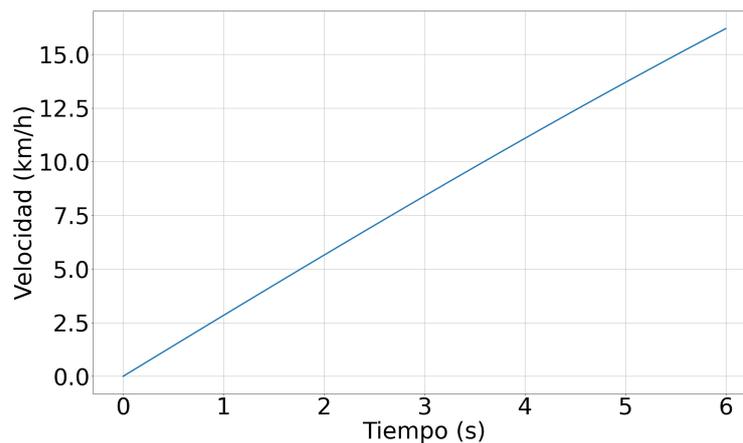
Vamos ahora al último de los tres modos, el que permite alcanzar la velocidad más alta, y por tanto el que requerirá la mayor aplicación de potencia. El objetivo de la simulación de alcanzar los 16 km/h se alcanza en un tiempo de 6 s, lo que resulta razonable teniendo en cuenta el arrancado suave y progresivo.

- Potencia inferior a 260 W durante todo el intervalo.
- Aceleración inferior al límite impuesto de 0.8 m/s^2

Tabla 3.6 Modo Deportivo.

Valor	Condición inicial	Condición final
Velocidad del motor	0 rad/s	$\geq 0 \text{ rad/s}$
Intensidades de armadura y campo	0 A	$\geq 0 \text{ A}$
Tensión de armadura	0 V	$\geq 0 \text{ V}$
Velocidad	0 m/s	4.5 m/s (16 km/h)

La velocidad de 16 km/h se alcanza satisfactoriamente en esos 6 s. Las intensidades de armadura y campo, no mostradas en esta simulación, tienen un valor y forma de onda muy similar a los dos modos anteriores, teniendo un pequeño pico en el transitorio pero estabilizándose rápidamente en el régimen permanente. Vemos ahora como la velocidad del motor se va acercando a ese valor nominal, y la potencia sube ya hasta casi los 175 W.

**Figura 3.25** Velocidad del Walk Car en Modo Deportivo.

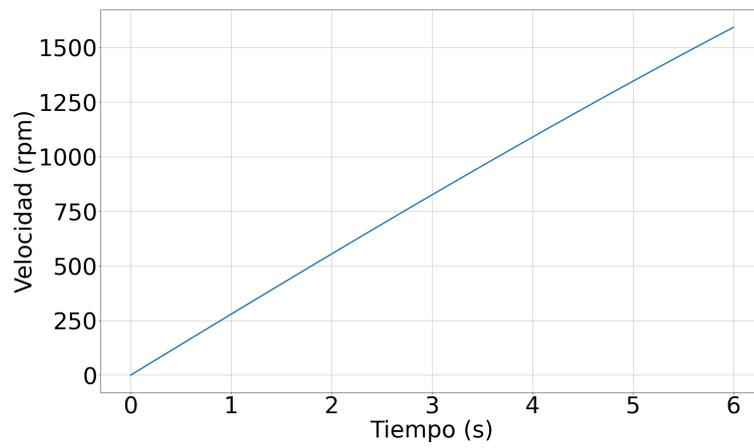


Figura 3.26 Velocidad del motor en Modo Deportivo.

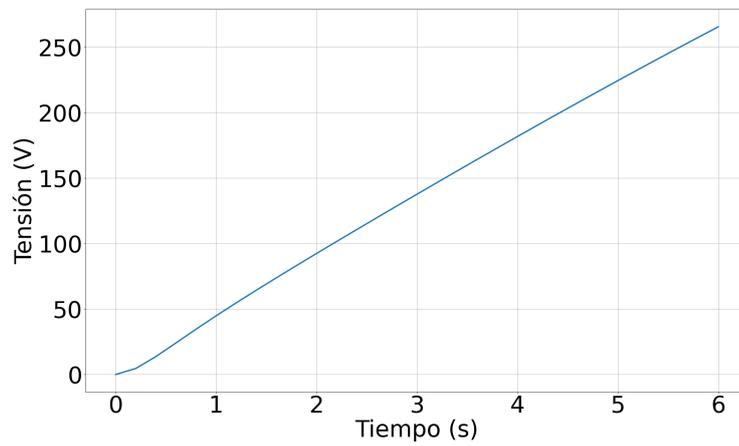


Figura 3.27 Tensión de armadura en Modo Deportivo.

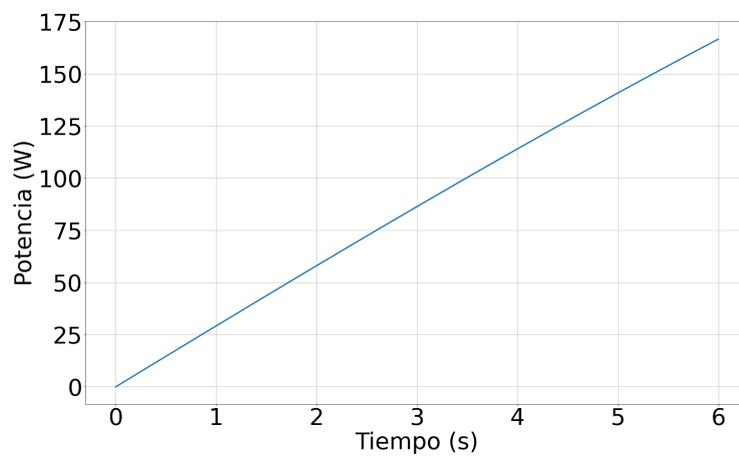


Figura 3.28 Potencia empleada en Modo Deportivo.

3.2.3 Simulación del efecto de la inclinación

Una vez estudiados los tres modos de operación en movimiento rectilíneo y visto su correcto comportamiento, pasamos a analizar la influencia de la pendiente del terreno. Para ello, llevaremos a cabo tres simulaciones distintas. En cada una de estas simulaciones supondremos que estamos en un entorno ideal sin rozamiento y con una masa reducida como indica Cocoa Motors. Nuevamente, limitaremos la aceleración máxima en 0.8 m/s^2 y nuestro movimiento será rectilíneo. Así, llevaremos a cabo simulaciones para un ángulo de inclinación de 0° , 5° y 10° en la que para cada una de ellas el objetivo será alcanzar la potencia nominal de salida de 260 W . Nos bastará por tanto estudiar en cuánto tiempo se alcanza dicha potencia, así que esa será la gráfica que tendremos en cuenta. Nuestro modelo será idéntico pero incluyendo la fuerza gravitacional y eliminando la fuerza de rozamiento.

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + f_\beta)) \quad (3.35)$$

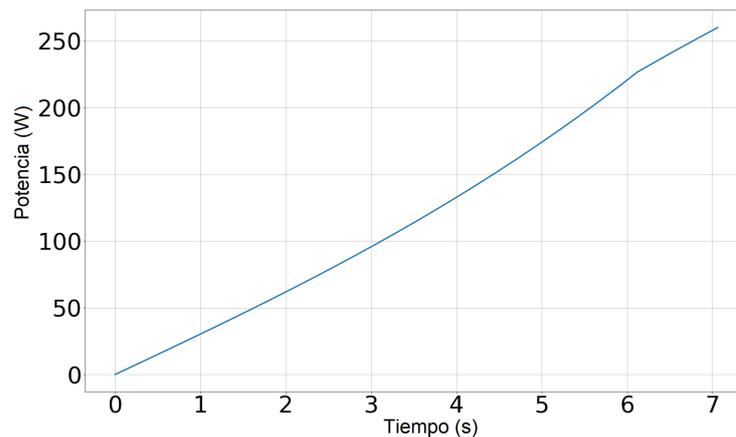


Figura 3.29 Potencia empleada para 0° .

Para la primera simulación de 0° se alcanza dicha potencia en unos 7 segundos. Es conveniente tener en cuenta que en este apartado no estamos considerando la fuerza de rozamiento y no hacer comparaciones temporales con apartados anteriores.

Para el segundo caso en el que nos enfrentamos a una inclinación de 2.5° , todavía reducida, podemos observar como ya se aprecia una subida en el tiempo requerido para alcanzar dicha potencia nominal. Al sistema le cuesta más llegar a ese punto. Finalmente, para una inclinación de 5° , al sistema ya le cuesta bastante más tiempo alcanzar la potencia de 260 W , concretamente unos 30 s.

Tal y como indica Cocoa Motors al mostrar que para inclinaciones considerables la carga máxima debe ser de 60 kg , quizá sea este uno de los puntos débiles del Walk Car, ya que le cuesta alcanzar una potencia que permita una velocidad considerable cuando se enfrente a inclinaciones de 5° .

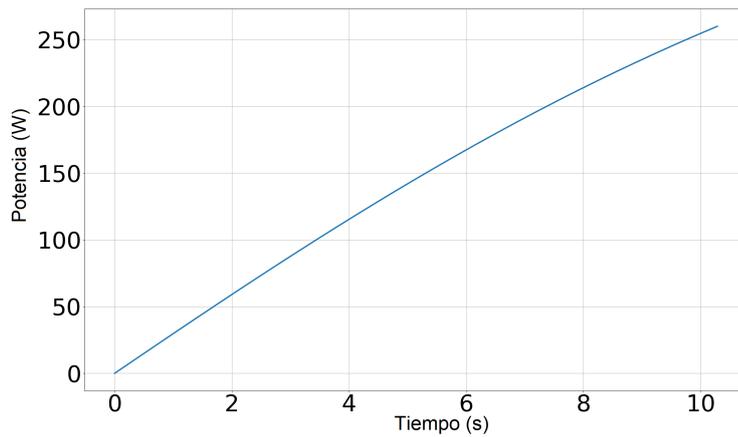


Figura 3.30 Potencia empleada para 2.5°.

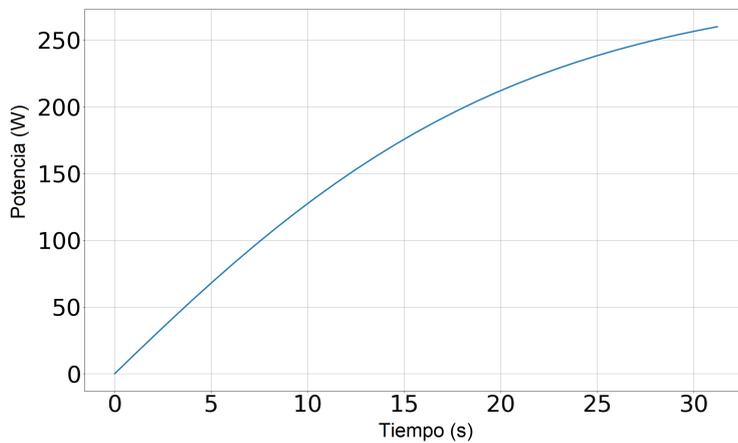


Figura 3.31 Potencia empleada para 5°.

3.3 Modelo Físico con Giros y Aceleración Lateral

En este nuevo apartado cambiaremos nuestra consigna, ya no buscaremos alcanzar el objetivo de nuestra simulación en el menor tiempo posible. Ahora nos centraremos en observar cómo gira el Walk Car ante la introducción de la aceleración lateral y un eje de coordenadas x-y. Para aclarar algo más cómo vamos a proceder en la siguiente simulación, se muestra el modelo matemático a utilizar.

3.3.1 Modelo Matemático y Ejes de Coordenadas

Al ángulo de inclinación de la rueda con respecto a su posición de equilibrio le llamaremos ϕ y al ángulo de inclinación del coche con respecto a la horizontal le denominaremos θ . De esta forma,

pasamos a indicar cómo cambia de posición nuestro sistema, si parte desde el origen de coordenadas ($x=0, y=0$).

$$\frac{dx}{dt} = v \cdot \cos(\theta) \quad (3.36)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta) \quad (3.37)$$

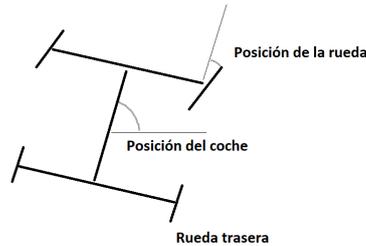


Figura 3.32 Esquema del Walk Car con los ángulos considerados.

Queda claro entonces como cambiarán nuestras posiciones x e y con respecto al ángulo de inclinación del propio coche, pero debemos determinar cómo se modifica el ángulo de inclinación del coche respecto a la horizontal con el ángulo de inclinación de las ruedas ϕ . Para ello, debemos introducir la longitud del Walk Car, que denominaremos L .

$$\frac{d\theta}{dt} = \frac{v}{L} \cdot \tan(\phi) \quad (3.38)$$

Uno de los parámetros más importantes en la conducción es la conocida aceleración lateral o fuerza 'g'. Esta fuerza será igual a la velocidad del sistema al cuadrado partida del radio de curvatura, que no es más que la longitud del vehículo partida del seno de ϕ . De esta forma, definimos así la aceleración lateral.

$$a_r = \frac{v^2 \cdot \sin(\theta)}{L} \quad (3.39)$$

3.3.2 Primera Simulación

Para nuestra primera simulación impondremos un tiempo de 30 segundos, y trataremos de que el sistema consiga, partiendo del reposo, llegar a una cierta posición y se pare en dicha posición. De esta manera, nuestro objetivo será minimizar la aceleración y la ya descrita aceleración lateral, para mejorar así el confort y favorecer ya no solo el arrancado sino también un giro suave para la seguridad del individuo. Volveremos a imponer un valor máximo de aceleración 0.8 m/s^2 , que nos valdrá para ambas aceleraciones. Así todo, nuestro modelo será ahora:

$$\min : Z = \int_0^{t_f} (a^2 + a_r^2) \cdot dt \quad (3.40)$$

s.a

$$\frac{dx}{dt} = v \cdot \cos(\theta) \quad (3.41)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta) \quad (3.42)$$

$$\frac{d\theta}{dt} = \frac{v}{L} \cdot \tan(\phi) \quad (3.43)$$

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + 4 \cdot f_r)) \quad (3.44)$$

$$\frac{di_a}{dt} = t_f \cdot \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.45)$$

$$\frac{di_f}{dt} = t_f \cdot \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.46)$$

$$\frac{dv}{dt} = t_f \cdot a \quad (3.47)$$

$$a_r = \frac{v^2 \cdot \sin(\theta)}{L} \quad (3.48)$$

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.49)$$

$$p = \tau_r \cdot \omega_r \quad (3.50)$$

$$\omega_w = \omega_r \cdot K_w \quad (3.51)$$

$$v = \omega_w \cdot R_w \quad (3.52)$$

$$p \leq 260 \quad (3.53)$$

$$p \geq 0 \quad (3.54)$$

$$\tau_r \leq 1 \quad (3.55)$$

$$a \leq 0.8 \quad (3.56)$$

$$a_r \leq 0.3 \quad (3.57)$$

$$a_r \geq -0.3 \quad (3.58)$$

Nuestra función objetivo será:

```
m.integral = Integral(m.t, wrt=m.t, rule=lambda m, t: m.aceleracion[t]**2 + m.aceleracion_lateral[t]**2)
m.obj = Objective(expr=m.integral)
```

Las condiciones que le impondremos al individuo serán las siguientes: *Partiendo del sistema en reposo y desde el origen de coordenadas, lleve el Walk Car hasta la posición $x=1.5$ m, $y=3$ m y deténgalo en ese punto al cabo de 30 segundos. Además, el ángulo de inclinación del coche debe ser 0° tanto en la posición inicial como en la final.* Una vez asimilada la consigna, este sería el resultado.

```
#initial conditions

m.pc = ConstraintList()

m.pc.add(m.x[0]==0)
m.pc.add(m.y[0]==0)
m.pc.add(m.theta[0]==0)
m.pc.add(m.velocidad[0]==0)
m.pc.add(m.v_a[0]==0)
m.pc.add(m.tau[0]==0)

#final conditions

m.pc.add(m.x[tf]==1.5)
m.pc.add(m.y[tf]==3)
m.pc.add(m.theta[tf]==0)
m.pc.add(m.velocidad[tf]==0)

#final conditions on the control inputs

m.pc.add(m.v_a[tf]==0)
m.pc.add(m.tau[tf]==0)
m.pc.add(m.phi[tf]==0)
```

Como podemos observar, para poder detener el vehículo en la posición pedida y con el ángulo de inclinación requerido, se aprecia la suave curva que debería de realizar el individuo. En la siguiente gráfica se puede apreciar el bajo valor de ambas aceleraciones, ya que el objetivo no requiere una mayor aplicación de estas. Sí que es visible el aumento de la frenada al final del recorrido, ya que entre nuestros objetivos estaba el de frenar el vehículo en la posición pedida.

En cuanto a los ángulos y observando a la vez el recorrido realizado por nuestro Walk Car, para el valor del ángulo de inclinación de la rueda ϕ , es claro que para los 15 primeros segundos el dispositivo gira en una dirección (hacia la derecha) y en el tramo final en la contraria (izquierda). El ángulo de inclinación del dispositivo describe exactamente igual que el recorrido el trazado que realiza el Walk Car, llegando en el instante final a encontrarse en la misma posición que la inicial, cumpliendo nuestro objetivo. De la misma forma que para los ángulos, en la gráfica de la velocidad se puede intuir también el trazado del dispositivo, los valores son bajos ya que la posición objetivo se encuentra relativamente cerca del objetivo, y recordemos que el individuo dispone de 30 segundos para realizar la maniobra.

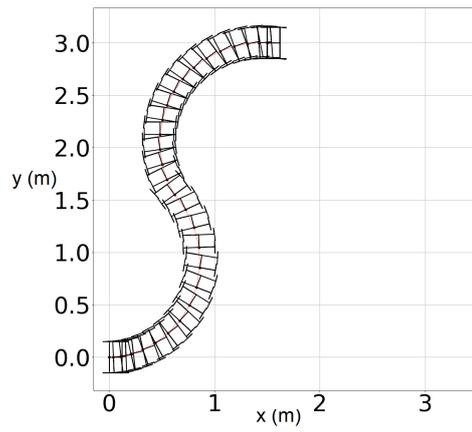


Figura 3.33 Recorrido trazado por el individuo en la primera simulación con giros.

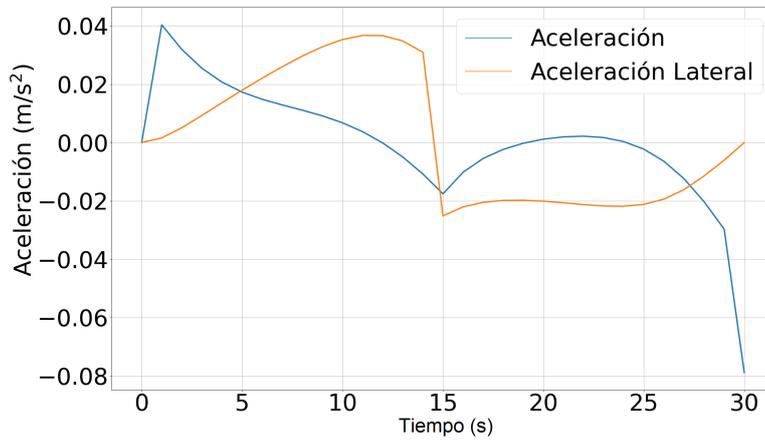


Figura 3.34 Aceleraciones del sistema en la primera simulación con giros.

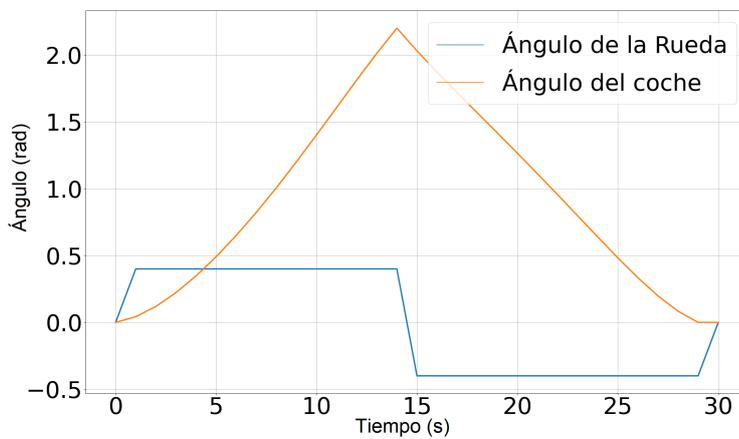


Figura 3.35 Ángulos del sistema en la primera simulación con giros.

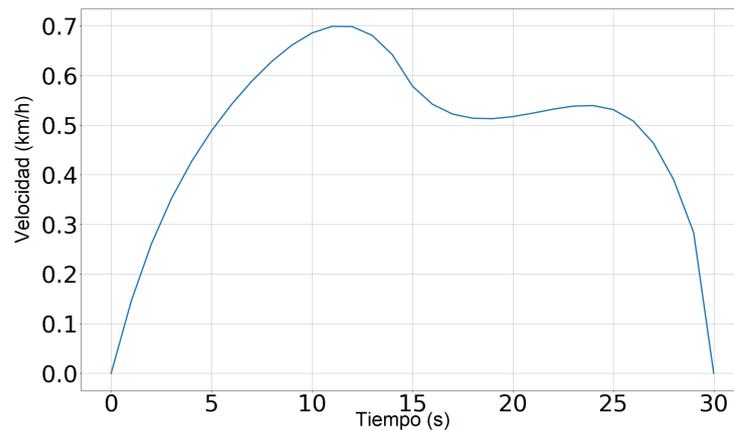


Figura 3.36 Velocidad del Walk Car en la primera simulación con giros.

3.3.3 Segunda Simulación

Realizaremos ahora una nueva simulación, en la que la consigna y el objetivo que le marcaremos al individuo será exactamente el mismo con una salvedad, deberá colocar el dispositivo en la posición objetivo pero colocado al revés, es decir, con un ángulo de inclinación respecto a la horizontal de π rad. Esta vez, la posición final será $x=2$, $y=2$. La curva del recorrido, muestra que ahora para cumplir el objetivo hay que realizar una maniobra algo más compleja, lo que tendrá repercusión en los valores de aceleraciones, ángulos y velocidad.

```
#initial conditions

m.pc = ConstraintList()

m.pc.add(m.x[0]==0)
m.pc.add(m.y[0]==0)
m.pc.add(m.theta[0]==0)
m.pc.add(m.velocidad[0]==0)
m.pc.add(m.v_a[0]==0)
m.pc.add(m.tau[0]==0)

#final conditions

m.pc.add(m.x[tf]==2)
m.pc.add(m.y[tf]==2)
m.pc.add(m.theta[tf]==pi)
m.pc.add(m.velocidad[tf]==0)

#final conditions on the control inputs

m.pc.add(m.v_a[tf]==0)
m.pc.add(m.tau[tf]==0)
m.pc.add(m.phi[tf]==0)
```

Los valores de aceleración han subido considerablemente con respecto a la simulación previa. De nuevo, para alcanzar el objetivo de frenar el Walk Car en la posición objetivo, se puede apreciar una

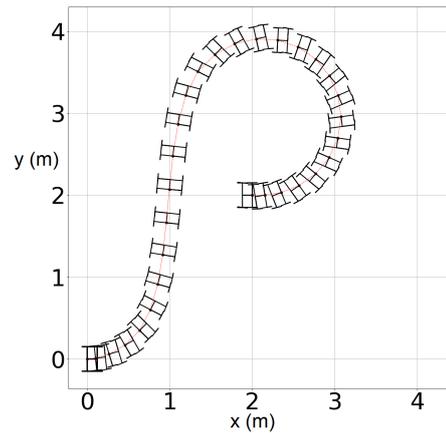


Figura 3.37 Recorrido realizado por el individuo en la segunda simulación con giros.

frenada relativamente fuerte en los instantes finales.

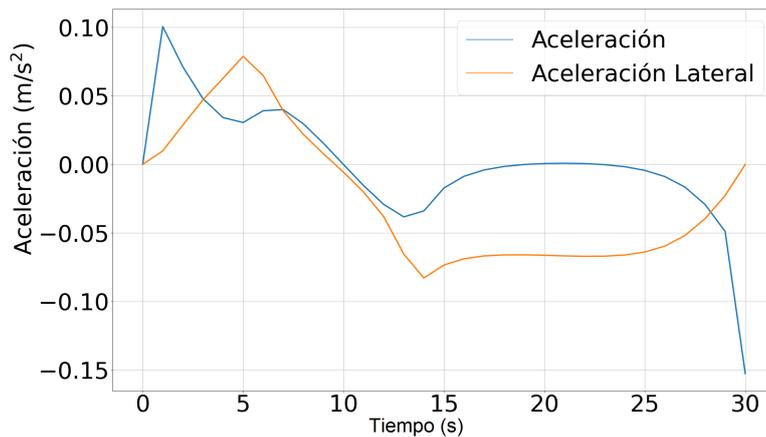


Figura 3.38 Aceleraciones del sistema en la segunda simulación con giros.

Para los ángulos, es obvio que ϕ tiene que pasar del valor inicial 0 rad al valor final π . De esta forma, se produce una bajada en el valor del ángulo de inclinación del vehículo. Para el valor del ángulo de giro, y teniendo en cuenta el recorrido, se puede ver que entre el instante 14 s y 25 s el giro es continuo y por tanto ϕ es constante. La velocidad sube notablemente, debido a las mayores aceleraciones, llegando a alcanzar una velocidad máxima de simulación de unos 1.5 km/h. Recordemos que estas bajas velocidades se deben a que le hemos dado 30 segundos para que el individuo realice la acción, siempre minimizando la aceleración y la aceleración lateral.

De nuevo podemos concluir que las velocidades siguen siendo muy bajas, por lo que en la siguiente simulación trataremos de minimizar el tiempo en el que el individuo realiza la acción. Podríamos decir que estas dos primeras simulaciones han sido realizadas con un individuo novato para el Walk Car, y ahora llevaremos a cabo el mismo tipo de experimento con objetivos de posición pero con una persona que tenga un control más avanzado del dispositivo y pueda cumplir el objetivo en menos tiempo y por tanto, yendo a más velocidad.

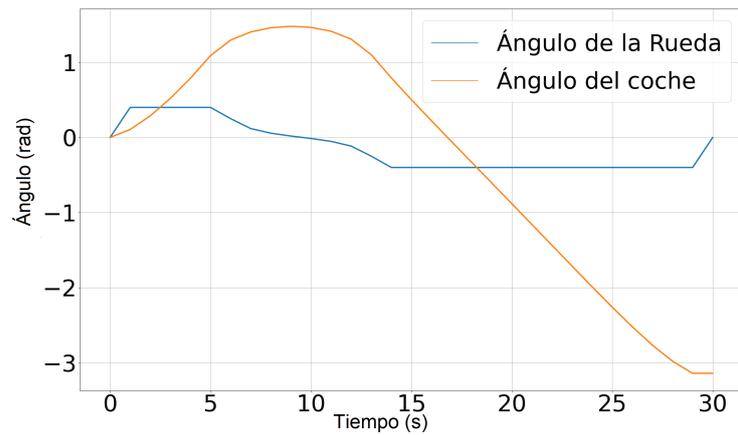


Figura 3.39 Ángulos del sistema en la segunda simulación con giros.

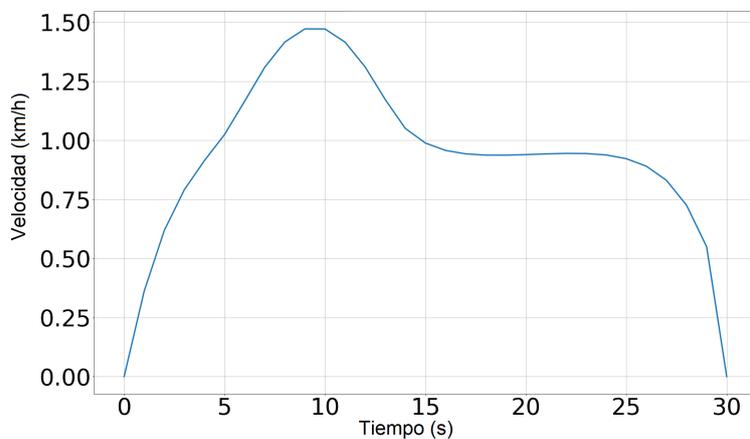


Figura 3.40 Velocidad del Walk Car en la segunda simulación con giros.

3.3.4 Tercera simulación

Una vez contemplados dos casos en los que no hemos tratado de minimizar el tiempo a ejecutar, sino las aceleraciones implicadas en nuestro sistema, pasamos ahora a estudiar un enunciado similar a los anteriores pero con el objetivo de cumplirlo en el menor tiempo posible. De nuevo le indicaremos al individuo una posición objetivo y un ángulo específico de llegada, pero esta vez no le impondremos que se detenga, solo que llegue y pase por dicha posición en el ángulo requerido.

$$\min : Z = t_f \quad (3.59)$$

s.a

$$\frac{dx}{dt} = v \cdot \cos(\theta) \quad (3.60)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta) \quad (3.61)$$

$$\frac{d\theta}{dt} = \frac{v}{L} \cdot \tan(\phi) \quad (3.62)$$

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + 4 \cdot f_r)) \quad (3.63)$$

$$\frac{di_a}{dt} = t_f \cdot \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.64)$$

$$\frac{di_f}{dt} = t_f \cdot \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.65)$$

$$\frac{dv}{dt} = t_f \cdot a \quad (3.66)$$

$$a_r = \frac{v^2 \cdot \sin(\theta)}{L} \quad (3.67)$$

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.68)$$

$$p = \tau_r \cdot \omega_r \quad (3.69)$$

$$\omega_w = \omega_r \cdot K_w \quad (3.70)$$

$$v = \omega_w \cdot R_w \quad (3.71)$$

$$p \leq 260 \quad (3.72)$$

$$p \geq 0 \quad (3.73)$$

$$\tau_r \leq 1 \quad (3.74)$$

$$a \leq 0.8 \quad (3.75)$$

$$a_r \leq 0.3 \quad (3.76)$$

$$a_r \geq -0.3 \quad (3.77)$$

```
#define the optimization objective
```

```
m.obj = Objective(expr=m.tf)
```

Luego el enunciado que le pasaríamos al individuo para realizar la prueba sería: *Partiendo del sistema en reposo y desde el origen de coordenadas, lleve el Walk Car hasta la posición $x=11$ m, $y=11$ m y pase por ese punto en el menor tiempo posible. Además, el ángulo de inclinación del coche debe ser mayor o igual a 1 rad.* No debemos olvidar que pese a que nuestro objetivo ha

pasado a ser el de minimizar el tiempo de ejecución, tenemos establecidos unos valores límite para ambas aceleraciones, fijados en 0.6 m/s^2 y 0.3 m/s^2 para la aceleración lateral.

```
#initial conditions

m.pc = ConstraintList()

m.pc.add(m.x[0]==0)
m.pc.add(m.y[0]==0)
m.pc.add(m.theta[0]==0)
m.pc.add(m.omega_r[0]==0)
m.pc.add(m.i_a[0]==0)
m.pc.add(m.i_f[0]==0)
m.pc.add(m.velocidad[0]==0)
m.pc.add(m.v_a[0]==0)
m.pc.add(m.tau[0]==0)

#final conditions

m.pc.add(m.x[1]>=11)
m.pc.add(m.y[1]>=11)
m.pc.add(m.theta[1]>=1)
m.pc.add(m.omega_r[1]>=0)
m.pc.add(m.i_a[1]>=0)
m.pc.add(m.i_f[1]>=0)
m.pc.add(m.velocidad[1]<=4.4)

#final conditions on the control inputs

m.pc.add(m.v_a[1]>=0)
m.pc.add(m.tau[1]>=0)
```

En el recorrido y en la gráfica de la velocidad en km/h se puede apreciar la optimización de la curva que realiza el individuo para alcanzar la máxima velocidad posible, para llegar así a la posición pedida en un tiempo de 7.15 s.

Para las aceleraciones, ambas alcanzan su pico en algunos instantes de la simulación lo que no supone ningún problema. Resulta lógico pensar que si aumentásemos dichos límites el objetivo de la simulación se alcanzaría en un tiempo inferior a 7.15 s.

Para los ángulos también se puede observar como se alcanza el límite del ϕ máximo en un instante de la simulación, adquiriendo en este caso un valor de 0.4 rad. Esto no supone ningún problema y bien podríamos decir que sería el equivalente de girar todo el volante en un coche. Observamos también que el objetivo de llegar con una inclinación de 1 rad sobre la horizontal se cumple satisfactoriamente.

En cuanto a la potencia empleada y a la tensión aplicada por nuestra variable de control v_a , que recordemos que lo es en todas nuestras simulaciones, podemos ver como la potencia empleada es alta, y la velocidad roza el límite del modo deportivo de 16 km/h. La potencia llega a alcanzar un

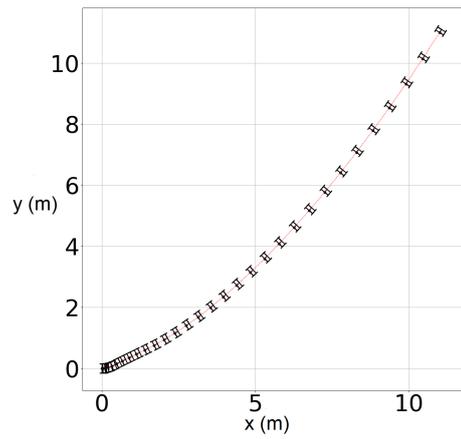


Figura 3.41 Recorrido realizado por el individuo en la tercera simulación con giros.

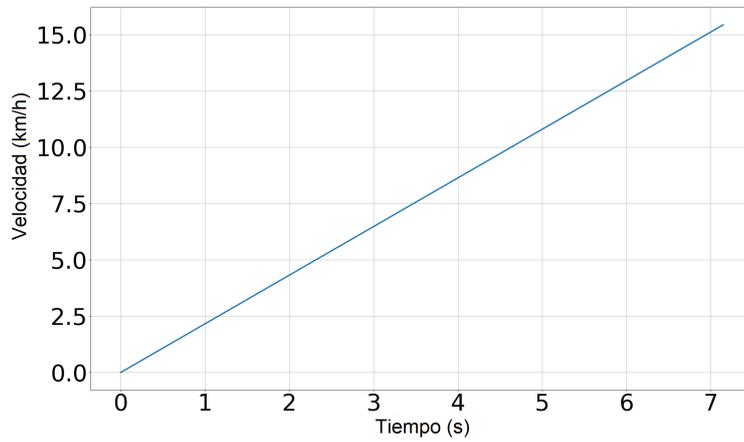


Figura 3.42 Velocidad del Walk Car en la tercera simulación con giros.

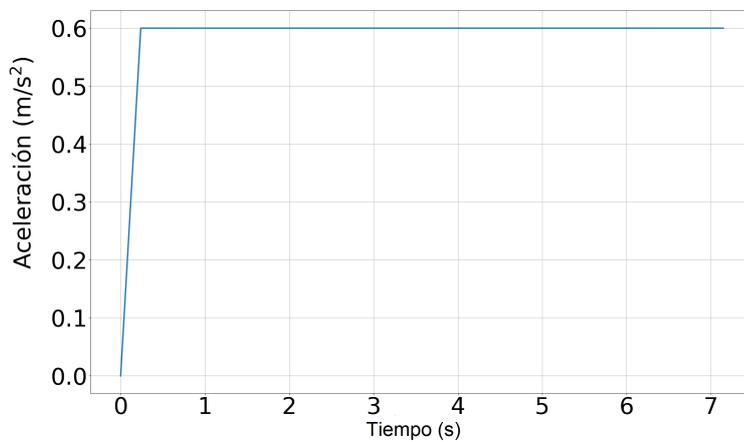


Figura 3.43 Aceleración en la tercera simulación con giros.

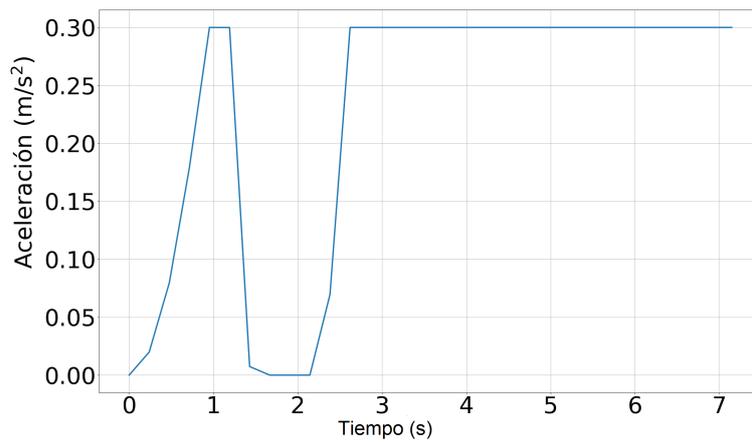


Figura 3.44 Aceleración lateral en la tercera simulación con giros.

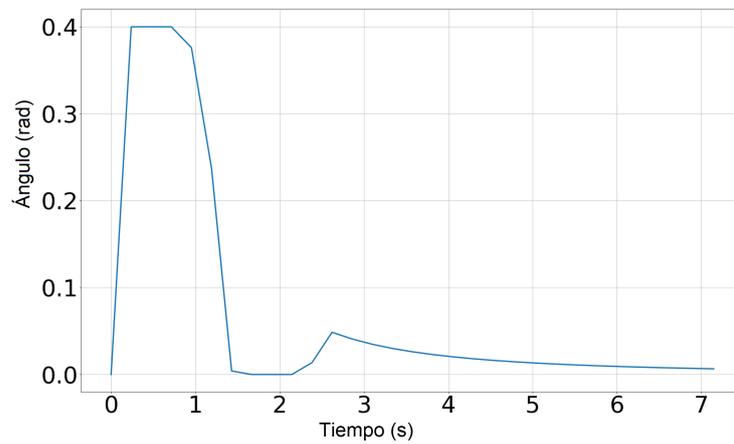


Figura 3.45 Ángulo de las ruedas en la tercera simulación con giros.

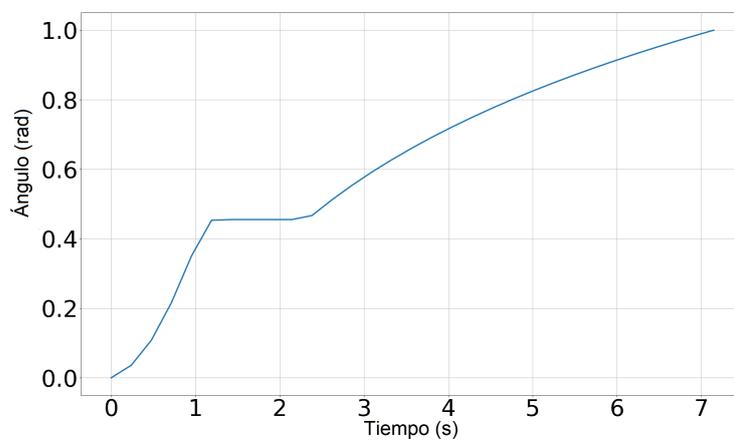


Figura 3.46 Ángulo del Walk Car en la tercera simulación con giros.

valor cercano a los 150 W, lo que todavía se encuentra lejos de la potencia nominal del Walk Car.

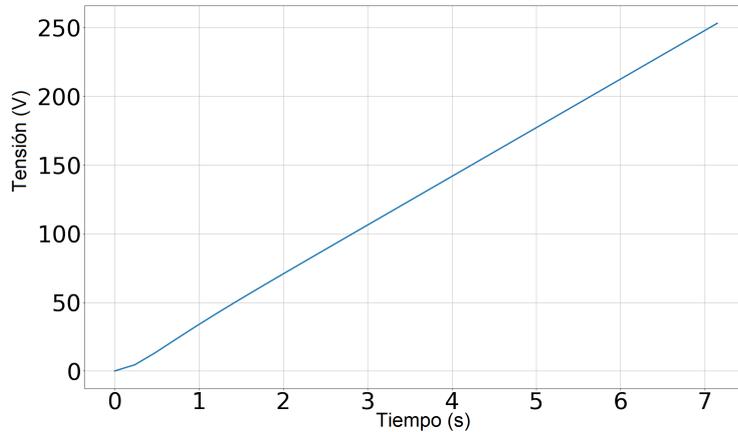


Figura 3.47 Tensión aplicada en la tercera simulación con giros.

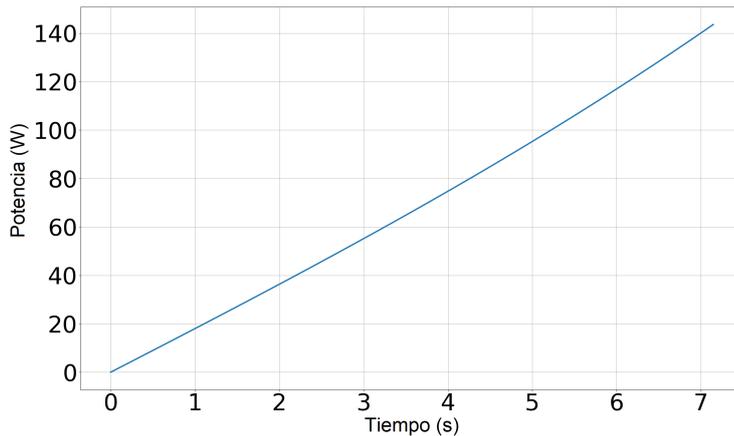


Figura 3.48 Potencia empleada en la tercera simulación con giros.

Como conclusión podemos deducir que hemos conseguido aumentar la velocidad en la que el individuo realiza este tipo de maniobras con posición en los ejes x,y y que además resulta muy interesante observar las trazadas realizadas. Dichas trazadas son consecuencia única y exclusivamente de la aplicación de un límite para la aceleración lateral y el ángulo de giro máximo.

Como novedad y para comenzar a introducir la siguiente simulación, introducimos el valor de energía, definido en Wh, como potencia multiplicada por el tiempo. Vemos como se sitúa en torno al valor de 1000 Wh, y en la siguiente y última simulación de este apartado trataremos de minimizar dicha energía empleada.

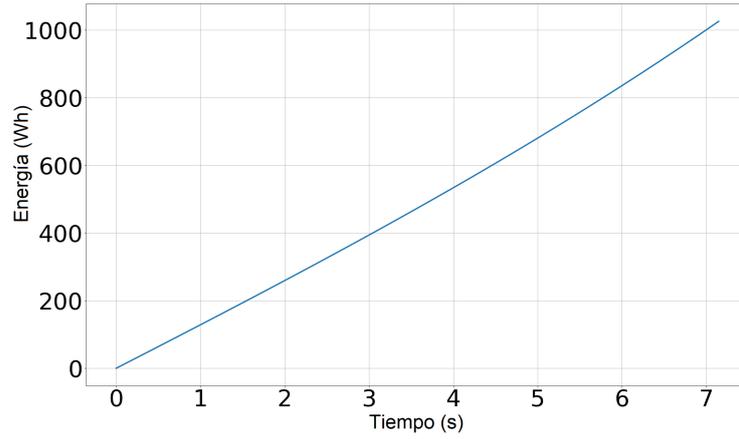


Figura 3.49 Energía empleada en la tercera simulación con giros.

3.3.5 Cuarta Simulación

Pasamos ahora en esta cuarta simulación a optimizar nuestro modelo de una manera novedosa hasta ahora y que no hemos implementado por el momento. Una vez vista la energía empleada en la tercera simulación, con un valor de pico cercano a los 1000 Wh, vamos a tratar de minimizar dicho valor para una simulación similar a la del apartado anterior. De una forma similar, le pasamos al individuo la orden de llevar el Walk Car a la posición $x=10$ m, $y=10$ m. Establecemos de nuevo los límites ya mencionados en anteriores simulaciones de aceleración, aceleración lateral y ángulo de giro de la rueda máximo. El objetivo de minimizar la energía lo llevamos a cabo minimizando la integral de la potencia al cuadrado en el tiempo.

$$\min : Z = \int_0^{t_f} p^2 \cdot dt \quad (3.78)$$

s.a

$$\frac{dx}{dt} = v \cdot \cos(\theta) \quad (3.79)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta) \quad (3.80)$$

$$\frac{d\theta}{dt} = \frac{v}{L} \cdot \tan(\phi) \quad (3.81)$$

$$\frac{d\omega_r}{dt} = \frac{1}{M \cdot R_w^2 \cdot K_w^2} \cdot (2 \cdot \tau_r - 2 \cdot K_f \cdot \omega_r - R_w \cdot K_w \cdot (f_d + 4 \cdot f_r)) \quad (3.82)$$

$$\frac{di_a}{dt} = t_f \cdot \frac{1}{L_a} \cdot (v_a - R_a \cdot i_a - L_{af} \cdot \omega_e \cdot i_f) \quad (3.83)$$

$$\frac{di_f}{dt} = t_f \cdot \frac{1}{L_f} \cdot (V_f - R_f \cdot i_f) \quad (3.84)$$

$$\frac{dv}{dt} = t_f \cdot a \quad (3.85)$$

$$a_r = \frac{v^2 \cdot \sin(\theta)}{L} \quad (3.86)$$

$$\tau_r = L_{af} \cdot i_f \cdot i_a \quad (3.87)$$

$$p = \tau_r \cdot \omega_r \quad (3.88)$$

$$\omega_w = \omega_r \cdot K_w \quad (3.89)$$

$$v = \omega_w \cdot R_w \quad (3.90)$$

$$p \leq 260 \quad (3.91)$$

$$p \geq 0 \quad (3.92)$$

$$\tau_r \leq 1 \quad (3.93)$$

$$a \leq 0.8 \quad (3.94)$$

$$a_r \leq 0.3 \quad (3.95)$$

$$a_r \geq -0.3 \quad (3.96)$$

$$tf = 12 \quad (3.97)$$

```
#define the optimization objective
```

```
m.integral = Integral(m.t, wrt=m.t, rule=lambda m, t: m.power[t]**2)
m.obj = Objective(expr=m.integral)
```

Las condiciones finales e iniciales quedan:

```
#initial conditions
```

```
m.pc = ConstraintList()
```

```
m.pc.add(m.x[0]==0)
m.pc.add(m.y[0]==0)
m.pc.add(m.theta[0]==0)
m.pc.add(m.velocidad[0]==0)
m.pc.add(m.v_a[0]==0)
m.pc.add(m.tau[0]==0)
```

```
#final conditions
```

```
m.pc.add(m.x[tf]>=10)
m.pc.add(m.y[tf]>=10)
```

```

m.pc.add(m.theta[tf]>=0)
m.pc.add(m.velocidad[tf]<=4.4)

#final conditions on the control inputs

m.pc.add(m.v_a[tf]>=0)
m.pc.add(m.tau[tf]>=0)
m.pc.add(m.phi[tf]==0)

```

De esta forma, vemos como minimizando la energía, manteniendo un tiempo de simulación constante de 12 segundos, el pico de energía baja hasta los 220 Wh. La velocidad baja con tal de cumplir la consigna.

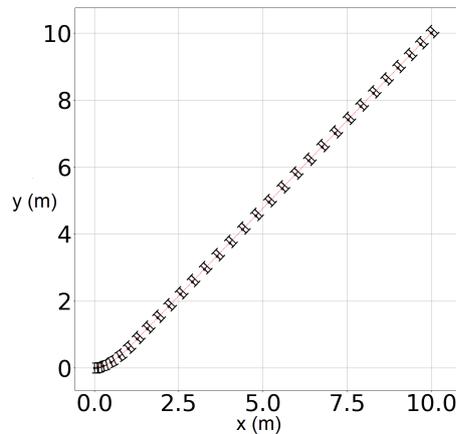


Figura 3.50 Recorrido realizado por el individuo en la cuarta simulación con giros.

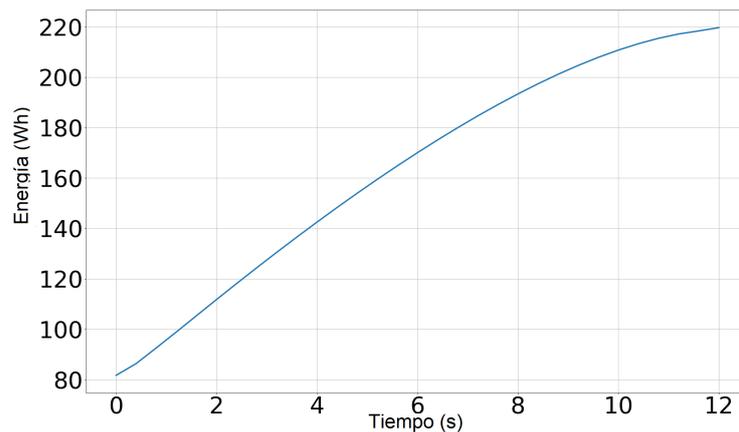


Figura 3.51 Energía empleada en la cuarta simulación con giros.

La velocidad también ha bajado considerablemente, en favor de minimizar ese valor de energía empleada. Vemos como su valor de pico durante los 11.3 s de maniobra se sitúa en unos 5.4 km/h, aproximadamente.

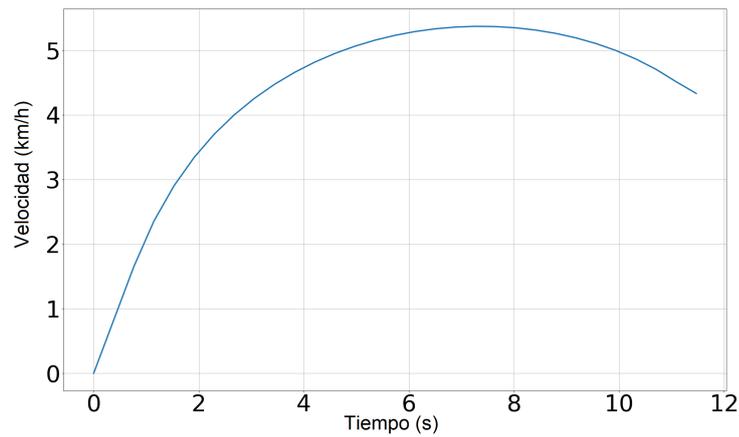


Figura 3.52 Velocidad del Walk Car en la cuarta simulación con giros.

Podemos concluir que resulta muy útil minimizar la energía ya que comparándola con su valor en la tercera simulación la bajada es considerable y el tiempo de maniobra no se ha incrementado demasiado.

4 Conclusiones y Posibles Ampliaciones

En este capítulo describiremos qué clase de conclusiones hemos podido sacar en base al modelo que hemos optimizado. Describiremos cómo se ha comportado el Walk Car en primer lugar, teniendo en cuenta algunos posibles puntos débiles para su desarrollo y alguna que otra mejora. También se comentará cómo ha sido la experiencia de trabajar en Pyomo, sus ventajas e inconvenientes, y algunas cosas a tener en cuenta en un paquete relativamente novedoso y que está en constante desarrollo.

Además, tendremos en cuenta qué más se podría haber hecho en este trabajo, comentando posibles ampliaciones o aproximaciones a lo que podría ser un trabajo más fiel a la realidad y que pueda contemplar todas las posibles variantes de un modelo físico completamente real.

4.1 Resultados

Recorriendo paso a paso todas las simulaciones realizadas hemos podido comprobar distintos resultados, teniendo en cuenta el objetivo de cada simulación y el comportamiento mostrado. De esta forma y a modo de resumen de los resultados podríamos decir:

- El motor de corriente continua simulado al principio alcanzó el objetivo de llegar a 260 W en un tiempo de 2.55 segundos, con una velocidad nominal del motor de 2500 rpm.
- Para las simulaciones de los tres modos de funcionamiento podemos concluir que los 3 modos: Asistencia para caminar (6 km/h), Normal (10 km/h) y Deportivo (16 km/h) alcanzan sus respectivas velocidades objetivo en unos tiempos de 2.2 s, 3.55 s y 6 s, respectivamente.
- Cuando nos enfrentamos a la simulación que tenía en cuenta la inclinación del terreno alcanzábamos el objetivo de llegar a la potencia nominal en 10 segundos para una inclinación de 2.5° y 30 segundos para una inclinación de 5°.
- En la primera simulación del modelo físico completo el tiempo no fue nuestro objetivo, sino el hecho de minimizar las aceleraciones. Conseguimos parar el Walk Car en la posición $x=1.5$ $y=3$ en un tiempo establecido de 30 segundos teniendo un pico de aceleración de -0.08 m/s^2 y un pico de 0.04 m/s^2 para la aceleración lateral.
- En la segunda simulación del modelo físico completo seguimos con nuestro objetivo de minimizar las aceleraciones pero esta vez cambiamos la posición de parada con un ángulo de llegada de 180°. De nuevo llegamos a nuestro objetivo con unos picos de aceleración algo superiores: -0.15 m/s^2 y 0.08 m/s^2 para la aceleración lateral.

- En la tercera simulación del modelo físico volvimos a intentar minimizar el tiempo objetivo, consiguiendo dejar el Walk Car en la posición pedida en unos 7 segundos, empleando una energía de 1000 Wh.
- En el último apartado nuestro objetivo pasó a ser limitar dicha energía, como integral de la potencia, llegando a un pico de 220 Wh consiguiendo dejar el Walk Car en la posición pedida en un tiempo establecido de 12 segundos.

4.2 Walk Car

El Walk Car, como se ha podido ver a lo largo del capítulo descriptivo de sus características y funcionalidades, es un dispositivo relativamente cómodo de utilizar. Tiene una autonomía lo suficientemente amplia y unos rangos de velocidades que se adecuan a la perfección al tráfico urbano. Una de los aspectos más importantes contemplados en la simulación y modelado ha sido el de poder gozar de un arrancado suave y cómodo, ya que el individuo que va sobre el Walk Car no va agarrado a un volante, ni sentado a un asiento. Hemos podido ver que el arrancado, favorecido por los límites y capacidad de minimizar la aceleración, se realiza de forma ergonómica y segura, sin posibilidad de desestabilizar a la persona del dispositivo.

Uno de los aspectos en los que se han encontrado más dificultades ha sido el de definir las características de los motores de corriente continua que van insertados en las ruedas delanteras motoras del Walk Car. Cocoa Motors no ofrece apenas datos sobre los mismos, más que una breve descripción sobre su tamaño ultra compacto y su tecnología innovadora. Es por ello que se han seleccionado valores de motores con un tamaño similar, de entre varios catálogos de motores de corriente continua. Los resultados obtenidos se corresponden con las características de cada modo de uso así que se entiende que la aproximación ha sido correcta y no muy lejana de lo que puedan ser los motores reales del Walk Car.

Uno de los posibles puntos flacos del Walk Car, que se ha mencionado en nuestro modelo y optimizado, es el de afrontar una pendiente de subida considerable. Cocoa Motors ya informa en sus manuales e instrucciones que para llevar a cabo determinadas subidas la carga máxima que puede soportar es de unos 60 kg, lo que no resulta ser exactamente el peso medio del ser humano. Es por eso que consideramos este el principal punto débil del Walk Car. Por lo demás el dispositivo se comporta bastante bien, es lo bastante rápido y lo suficientemente suave.

Es por todo esto que, intentando olvidarnos de su elevado precio, el Walk Car tiene todo (o casi todo) para hacerse dueño de la movilidad urbana. Ya lo es en Asia y seguramente en los próximos años empecemos a verlo por las calles europeas y americanas, ya que tiene un diseño bastante más atractivo que otros dispositivos de movilidad similares.

4.3 Pyomo

Utilizar Pyomo dentro del lenguaje de programación Python y el entorno Spyder ha sido una experiencia muy positiva. Gracias a una cantidad enorme de información disponible por la comunidad de Pyomo, se ha podido aprender una estructura de programación simple, ordenada y funcional que permite resolver ecuaciones diferenciales para definir un modelo de forma cómoda. La manera de optimizar cada sistema con la función objetivo resulta extremadamente útil y sencilla, ya que es ahí donde le decimos a Pyomo qué variable o variables son las que queremos optimizar.

El tiempo de simulación suele ser reducido, salvo en algunas ocasiones que alcanza un límite

máximo de iteraciones. Resulta interesante comentar que la instalación del solver ipopt resultó muy complicada y compleja, y una gran cantidad de personas tienen problemas para su instalación, por lo que muchos usuarios emplean otros solvers. Sin embargo, en nuestro caso se pudo descargar y utilizar correctamente tras muchos intentos.

Creo también que es muy interesante el concepto de restricción o constraint, en el que podemos definir valores límite para nuestras variables, o definir variables que no sean variables en el tiempo. Ambas funciones las hemos ido utilizando a lo largo de todas las simulaciones de forma cómoda y sencilla. Además, la estructura simple que se ha utilizado ha permitido de manera cómoda declarar todo el conjunto de valores iniciales y finales necesarios dentro de una lista de restricciones.

4.4 Modelo

Seguramente faltan muchos detalles o cosas para mejorar nuestro modelo. Además de muchas otras maneras de optimizarlo además de reduciendo energía, aceleraciones o tiempo, podríamos haber incluido distintas particularidades para mejorar nuestro sistema.

En primer lugar, podríamos haber implementado un modelo que contemplase el deslizamiento de la rueda, haciendo que nuestra fuerza ejercida por la rueda, f_w , sea función de dicho deslizamiento. Este deslizamiento dependería del radio de la rueda, la relación de transmisión, la velocidad del rotor y de la velocidad lineal en m/s del Walk Car. No la hemos considerado porque parece que el Walk Car pueda tener muy poco deslizamiento en sus ruedas y prácticamente sea imposible patinar o incluso derrapar en una arrancada. Además, Cocoa Motors no menciona nada en absoluto sobre este aspecto, ya que se entiende que es un aspecto negativo que no tiene mucho efecto.

Otro de los aspectos para poder ampliar este trabajo habría sido el de poder considerar cómo y de qué manera aplica la tensión el individuo con los pies. Es decir, qué efecto se produce realmente en ese proceso y qué consecuencias tiene en nuestro modelo mecánico y eléctrico. En este caso nuestra variable de control sería ese movimiento de pies y no la tensión aplicada directamente. Este sistema viene modelado por lo que se conoce como módulo del péndulo invertido, a partir del cual con unas referencias de posición y unos ejes de coordenadas, cualquier perturbación del mismo afecta sobre la tensión aplicada. Para ello sería necesario la creación de una serie de matrices de posición y ángulos con sus respectivas perturbaciones. Hoy en día vemos muchos sistemas que funcionan con este sistema, por lo que su aplicación nos resulta familiar. A pesar de todo lo descrito hasta ahora sobre el módulo del péndulo invertido, sobre el que versan numerosos trabajos de grado dada su alta dificultad, no se ha visto conveniente tratar de incluirlo en nuestro trabajo dada su elevada dificultad y atendiendo a qué nuestro objetivo era crear, modelar y optimizar un sistema y entender de forma clara como hacerlo a través de Pyomo, objetivo que se ha cumplido con un sistema lo suficientemente complejo.

Índice de Figuras

1.1	Distintos dispositivos de movilidad eléctrica urbana	1
1.2	Modelo de una ciudad actual que integra distintos sistemas de movilidad	1
1.3	Walk Car	2
1.4	Logotipo de Cocoa Motors	2
1.5	Logotipo de Pyomo DAE	3
2.1	Dimensiones del Walk Car	5
2.2	Suspensión del Walk Car	6
2.3	El Walk Car en la industria	7
2.4	El Walk Car virando en modo normal	8
3.1	Motor ultracompacto del Walk Car	9
3.2	Circuitos de campo y armadura	10
3.3	Esquema representativo del motor	11
3.4	Potencia de un motor de 260 W	16
3.5	Velocidad del motor en rpm	16
3.6	Tensión de armadura para motor de 260 W	17
3.7	Intensidades de campo y armadura para un motor de 260 W	17
3.8	Par electromagnético aplicado para un motor de 260 W	18
3.9	Esquema del sistema de transmisión	18
3.10	Esquema de fuerzas	19
3.11	Esquema de fuerzas en el Walk Car	20
3.12	Velocidad en Modo Asistencia para Caminar	23
3.13	Aceleración en Modo Asistencia para Caminar	23
3.14	Potencia empleada en Modo Asistencia para Caminar	23
3.15	Velocidad del motor en Modo Asistencia para Caminar	24
3.16	Tensión de armadura en Modo Asistencia para Caminar	24
3.17	Intensidad de armadura en Modo Asistencia para Caminar	24
3.18	Intensidad de campo en Modo Asistencia para Caminar	25
3.19	Velocidad del Walk Car en Modo Normal	26
3.20	Potencia empleada en Modo Normal	26
3.21	Velocidad del motor en Modo Normal	26
3.22	Tensión de armadura en Modo Normal	27
3.23	Intensidad de armadura en Modo Normal	27
3.24	Intensidad de campo en Modo Normal	27
3.25	Velocidad del Walk Car en Modo Deportivo	28

3.26	Velocidad del motor en Modo Deportivo	29
3.27	Tensión de armadura en Modo Deportivo	29
3.28	Potencia empleada en Modo Deportivo	29
3.29	Potencia empleada para 0º	30
3.30	Potencia empleada para 2.5º	31
3.31	Potencia empleada para 5º	31
3.32	Esquema del Walk Car con los ángulos considerados	32
3.33	Recorrido trazado por el individuo en la primera simulación con giros	35
3.34	Aceleraciones del sistema en la primera simulación con giros	35
3.35	Ángulos del sistema en la primera simulación con giros	35
3.36	Velocidad del Walk Car en la primera simulación con giros	36
3.37	Recorrido realizado por el individuo en la segunda simulación con giros	37
3.38	Aceleraciones del sistema en la segunda simulación con giros	37
3.39	Ángulos del sistema en la segunda simulación con giros	38
3.40	Velocidad del Walk Car en la segunda simulación con giros	38
3.41	Recorrido realizado por el individuo en la tercera simulación con giros	41
3.42	Velocidad del Walk Car en la tercera simulación con giros	41
3.43	Aceleración en la tercera simulación con giros	41
3.44	Aceleración lateral en la tercera simulación con giros	42
3.45	Ángulo de las ruedas en la tercera simulación con giros	42
3.46	Ángulo del Walk Car en la tercera simulación con giros	42
3.47	Tensión aplicada en la tercera simulación con giros	43
3.48	Potencia empleada en la tercera simulación con giros	43
3.49	Energía empleada en la tercera simulación con giros	44
3.50	Recorrido realizado por el individuo en la cuarta simulación con giros	46
3.51	Energía empleada en la cuarta simulación con giros	46
3.52	Velocidad del Walk Car en la cuarta simulación con giros	47

Índice de Tablas

2.1	Parámetros básicos del Walk Car	7
3.1	Valores constantes del motor	10
3.2	Valores límite	14
3.3	Condiciones iniciales y finales	15
3.4	Modo de Asistencia para Caminar	22
3.5	Modo Normal	25
3.6	Modo Deportivo	28

Bibliografía

- [1] <https://es.cocoamotors.com/>.
- [2] <https://es.rs-online.com/web/>.
- [3] <https://es.wikipedia.org/wiki/python>.
- [4] *Pyomo-workshop: Summer 2018*, 2018.
- [5] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola, *Pyomo-optimization modeling in python*, second ed., 2017.
- [6] Juan Manuel Mauricio, *Apuntes de la asignatura de accionamientos y movilidad eléctrica*, Curso 2020/2021.