



Universidad de Sevilla  
Escuela Politécnica Superior



Trabajo de Fin de Grado en Ingeniería de Diseño Industrial y  
Desarrollo del Producto e Ingeniería Mecánica

---

# Optimización del Sistema de Suspensión trasero de una motocicleta de competición mediante Algoritmos Genéticos

---

Autor: Pablo Díaz Gómez  
Tutores: Dr. Daniel Cagigas Muñiz  
Dra. Rocío García Robles

**Sevilla, 27 de junio de 2022**



## Resumen

El número de variables necesarias para definir el sistema de suspensión trasero de una motocicleta es muy elevado. Además, suele ser bastante complicado predecir cómo una modificación, incluso pequeña, en alguna de ellas puede afectar al comportamiento global del sistema. En el presente trabajo se desarrolla un método para la determinación de dichas variables en función del comportamiento que se desea que tenga un nuevo sistema de suspensión. Este método se fundamenta sobre las bases de la Teoría de Mecanismos, para la medición del comportamiento de los sistemas; y en el uso de Algoritmos Genéticos para el proceso de optimización. El trabajo concluye con la implementación de dicho método en un programa informático que automatiza el procedimiento antes descrito y asiste al ingeniero durante la etapa conceptual del proceso de diseño.

**Palabras clave:** Computación evolutiva, síntesis de mecanismos, diseño mecánico.

## Abstract

The number of variables that are needed to define a motorcycle rear suspension system is high. Moreover, it is usually quite difficult to predict how even a small change in any of these variables may affect the overall behaviour of the system. In this text, a method for defining the value of these variables to achieve a target behaviour is proposed. This method is founded in the Mechanism Theory, for the measure and definition of the behaviour of the systems; and the use of Genetic Algorithms, for the optimization process. The text concludes with the development of a computer application to automate the method described earlier, and assist the engineers during the conceptual phase of the design process.

**Keywords:** Evolutionary computing, mechanism synthesis, mechanical design.





# Índice

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>III</b>
<b>1. Contexto</b>	<b>1</b>
1.1. MotoStudent . . . . .	1
1.2. Sevilla Racing . . . . .	2
<b>2. Introducción y objetivos</b>	<b>3</b>
2.1. Objetivos propuestos . . . . .	4
2.2. Hipótesis de partida . . . . .	5
<b>3. Planificación del proyecto</b>	<b>5</b>
<b>4. Estado del arte</b>	<b>8</b>
4.1. El sistema de suspensión trasero en una motocicleta . . . . .	8
4.1.1. Funciones del sistema de suspensión . . . . .	9
4.1.2. Tipos de sistemas de suspensión traseros . . . . .	9
4.1.3. Comportamiento . . . . .	11
4.2. La teoría de máquinas . . . . .	13
4.2.1. Análisis de mecanismos . . . . .	15
4.2.2. Síntesis de mecanismos . . . . .	15
4.3. Los algoritmos genéticos y otras técnicas de computación evolutiva	16
4.3.1. Componentes de un Algoritmo Evolutivo . . . . .	17
<b>5. Diseño e implementación</b>	<b>20</b>
5.1. Representación . . . . .	20
5.1.1. La superclase <code>SuspensionSystem</code> . . . . .	21
5.1.2. Las subclases de <code>SuspensionSystem</code> . . . . .	23
5.2. Función de aptitud . . . . .	32
5.3. Población . . . . .	35

5.4. Mecanismo de selección de progenitores . . . . .	36
5.5. Mecanismo de generación de la descendencia . . . . .	36
5.6. Mecanismo de selección de supervivientes . . . . .	38
5.7. Inicialización . . . . .	38
5.8. Condición de finalización . . . . .	38
<b>6. Análisis de resultados</b>	<b>39</b>
6.1. Enunciado del problema . . . . .	39
6.2. Proceso de optimización . . . . .	40
6.3. Resultados obtenidos . . . . .	41
<b>7. Discusión de los resultados</b>	<b>43</b>
7.1. Experiencia previa . . . . .	45
7.2. Proceso largo y tedioso . . . . .	45
7.3. Toma de decisiones con escasa información . . . . .	45
<b>8. Conclusiones</b>	<b>46</b>
<b>Referencias</b>	<b>47</b>
<b>Apéndice A. Manual de funcionamiento de la aplicación informática</b>	<b>49</b>
A.1. Análisis . . . . .	49
A.2. Optimización . . . . .	52
A.2.1. Definición del comportamiento objetivo y del espacio de búsqueda . . . . .	52
A.2.2. Proceso de optimización . . . . .	53
<b>Apéndice B. Notación empleada en el programa</b>	<b>57</b>
B.1. Sistema de suspensión tipo Clásico . . . . .	58
B.2. Sistema de suspensión tipo Cantilever . . . . .	59
B.3. Sistema de suspensión tipo Pro-Link . . . . .	60
B.4. Sistema de suspensión tipo Unit Pro-Link . . . . .	61
B.5. Sistema de suspensión tipo Uni-Track . . . . .	62
B.6. Sistema de suspensión tipo Full-Floater . . . . .	63



**Figura 1.** Fotografía grupal de la V Edición de MotoStudent (2017-2018), en la que participaron 47 equipos Petrol y 27 Electric.

## 1 Contexto

El presente proyecto nace en el seno de la asociación de estudiantes Sevilla Racing, uno de los equipos de motociclismo de competición de la Universidad de Sevilla que participan en MotoStudent.

### 1.1 MotoStudent

MotoStudent es una competición internacional en la que estudiantes de universidades de todo el mundo se enfrentan al reto de conceptualizar, diseñar y fabricar sus propias motocicletas de carreras, de características similares a las motocicletas de categoría PreMoto3, en un periodo de 18 meses. Éste culmina con un evento en el circuito de Motorland en el que los prototipos fabricados se enfrentan a una serie de duras pruebas, incluyendo una carrera final, que determinará cuál es el equipo vencedor de la competición.

Para participar en la competición, cada equipo puede optar a inscribirse en dos categorías: *MotoStudent Petrol*, cuyas motocicletas deben utilizar como método de propulsión un motor de combustión interna; y *MotoStudent Electric*, en la que las motocicletas deben estar propulsadas por un sistema completamente eléctrico.

Lo que hace que MotoStudent sea una competición de motociclismo única es el hecho de que no se trate de una competición "de pilotos", sino de ingenieros. No se gana teniendo la motocicleta y el piloto más rápidos, sino que se valora

también el desarrollo del proyecto de ingeniería y las decisiones tomadas por cada equipo a lo largo del mismo. Para conseguirlo, los proyectos y prototipos presentados se evalúan en dos fases con la misma puntuación.

- **Fase MS1.** Para cada edición, el Reglamento Técnico de la competición plantea una situación de diseño en base a la cual los equipos han de tomar todas sus decisiones. Éstas, además, deberán de justificarse en la memoria del proyecto a presentar durante el desarrollo del mismo, y serán defendidas en el evento final mediante una exposición ante un jurado experto.
- **Fase MS2.** La segunda fase de la competición consiste en una serie de pruebas dinámicas que se realizan a los prototipos, ya fabricados, durante el evento final en el circuito de Motorland.

## 1.2 Sevilla Racing

Como se ha comentado antes, Sevilla Racing es una asociación compuesta por estudiantes, principalmente de la Escuela Politécnica Superior, que se creó en 2016 con el propósito de participar en MotoStudent. Desde entonces, han participado en la categoría Petrol de las ediciones IV y V. A día de hoy participa en la actual VI Edición, pero esta vez en la categoría Electric.

Uno de los objetivos principales de la asociación ha sido siempre el tratar de adquirir el mínimo de componentes comerciales para sus prototipos. En vez de ello, prefieren ser los propios integrantes del equipo los que los diseñen y fabriquen. De esta forma, y corriendo el riesgo de que los resultados no sean de calidad comercial, se da cabida al aprendizaje y hay posibilidades para la innovación.

El mejor ejemplo de esta filosofía podemos apreciarlo si pensamos en el carenado de sus motocicletas. El carenado es el conjunto de piezas que resguardan al resto de componentes, protegiéndolos al mismo tiempo que permitiendo aumentar las prestaciones aerodinámicas del vehículo. Además, es el componente que aporta mayor valor estético al conjunto. Muchos de los equipos (los que pueden permitírselo) adquieren carenados comerciales, realizados generalmente en fibra de vidrio. Los carenados de Sevilla Racing hasta ahora han sido específicamente diseñados y fabricados por los estudiantes mediante laminado manual en fibra de carbono. El resultado ha sido carenados mucho más ligeros que los comerciales en fibra de vidrio, y motocicletas con una estética única y rompedora frente al resto de equipos de la competición.

Hay componentes, sin embargo, en los que la fabricación propia no es una alternativa. Aunque existen otras configuraciones, el sistema de suspensión de-



(a) IV Edición de MotoStudent (2016).



(b) V Edición de MotoStudent (2018).

**Figura 2.** Prototipos realizados por Sevilla Racing para ediciones pasadas de MotoStudent.

lantero en motocicletas deportivas que ha demostrado año tras año ser el más eficaz es, sin lugar a dudas, el de horquillas telescópicas. Tratar de fabricar un sistema tan complejo y compacto como éste no tendría ningún sentido considerando los limitados medios de fabricación de los que dispone el equipo.

El trabajo que se desarrolla en estas páginas surge con la intención de ayudar a los estudiantes de la asociación en el proceso de diseño de otro de los componentes más fundamentales para cualquier motocicleta de competición, como es su sistema de suspensión trasero. Se busca con ello obtener resultados personalizados que se adapten perfectamente a las necesidades del equipo y que, como pasaba con el carenado, supongan una ventaja frente al uso de componentes comerciales (pensados para cubrir necesidades más generales).

Como ejemplo del trabajo realizado por la asociación Sevilla Racing desde su creación, en la Figura 2 se muestran unas fotografías de los dos prototipos de motocicleta de combustión que ésta ha realizado hasta la fecha.

## 2 Introducción y objetivos

La suspensión suele ser el sistema más caro de todos los que integran una motocicleta, después del motor (Castillo y col., 2013). Además, el sistema de



suspensión trasero más concretamente, es uno de los subsistemas más complejos durante las etapas de diseño, por presentarse durante el proceso tantas incertidumbres y existir muchas dependencias con el resto de subsistemas. En la práctica, esto hace que los ingenieros, a la hora de realizar un nuevo diseño, tiendan a ser conservadores y se basen en diseños anteriores, realizando únicamente pequeñas modificaciones. Esto se traduce en una falta de exploración de entre todas las posibles alternativas, dando lugar en muchos casos a diseños muy poco diversos (Corbera Caraballo y col., 2020).

Existen una serie de parámetros con los que puede predecirse cómo va a comportarse un determinado sistema de suspensión trasero en pista antes de tener un modelo detallado del mismo (Corbera Caraballo y col., 2020). El proceso de diseño consiste por lo general en: (i) establecer el valor ideal de estos parámetros – aquellos que hacen que la motocicleta se comporte en pista como se desee; (ii) elegir una tipología de sistema de suspensión a emplear y (iii) determinar la geometría del sistema de suspensión que hace que los parámetros sean iguales, o lo más parecidos posibles, a los establecidos como objetivo en el primer paso. Este proceso, a parte de los descritos anteriormente, presenta varios problemas.

- Aunque en la bibliografía puedan encontrarse rangos de valores adecuados en función del tipo de motocicleta, saber interpretar correctamente los parámetros antes mencionados requiere de mucha experiencia en el diseño de sistemas de suspensión.
- Si el proceso de optimización se hace manualmente, este puede hacerse muy largo y tedioso.
- La decisión de la tipología del sistema de suspensión se realiza previo a la optimización. Por ello, si no se cuenta con suficiente experiencia, pueden estar descartándose tipologías que se adecúen mejor a los objetivos establecidos.

## 2.1 Objetivos propuestos

El objetivo de este trabajo es desarrollar una aplicación informática que sirva de ayuda para el proceso diseño conceptual del sistema de suspensión trasero de una motocicleta, buscando resolver los problemas anteriormente mencionados. Para automatizar el proceso de optimización y garantizar, al mismo tiempo, que se evalúa ampliamente todo el espacio de posibilidades, el programa empleará Algoritmos Evolutivos. Esta técnica ya ha sido empleada por algunos autores con el mismo propósito (Castillo y col., 2013) (Corbera Caraballo y col., 2020).

Sin embargo, es también un objetivo clave en este proyecto que el programa informático permita utilizar estas técnicas por un usuario no experto, y que pueda aplicarlas a los tipos de sistemas de suspensión de motocicletas más habituales en la actualidad.

## 2.2 Hipótesis de partida

Como se ha comentado anteriormente, el ámbito de este trabajo es el del diseño conceptual. El análisis realizado parte de las siguientes hipótesis:

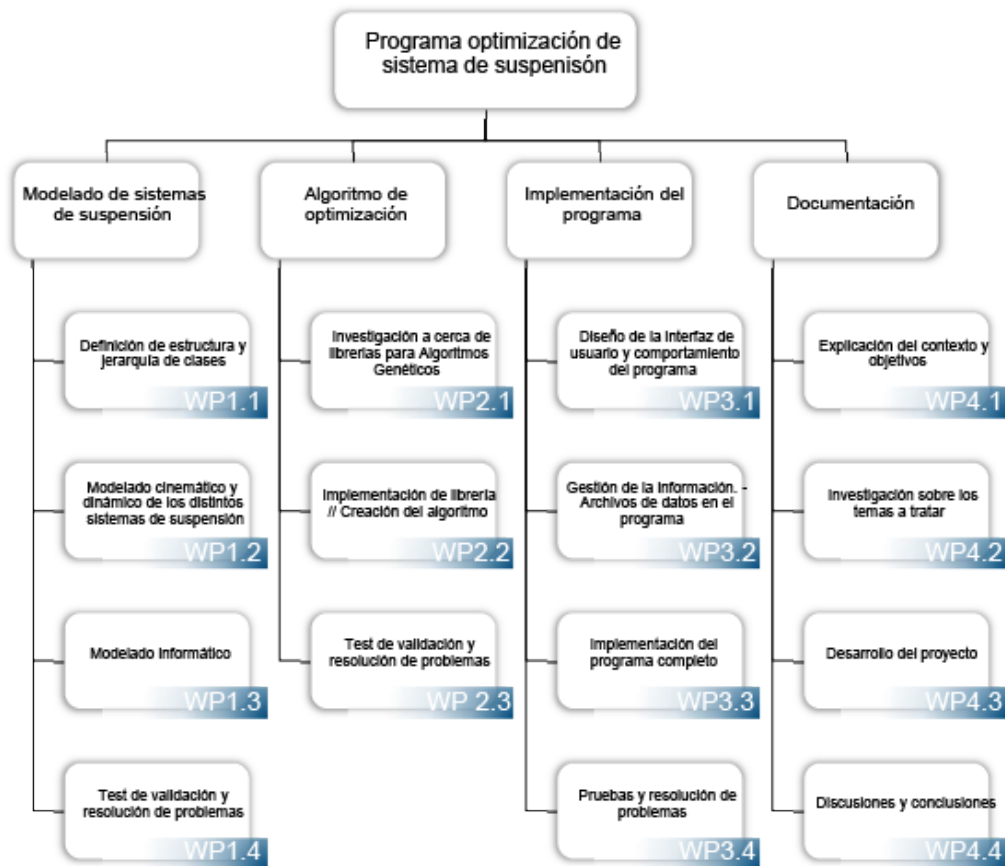
- No se tienen en cuenta la masa ni centros de gravedad de las distintas piezas que componen un sistema de suspensión; solo su geometría básica.
- Se consideran los pares entre las barras perfectos, sin tener en cuenta el rozamiento ni otros fenómenos físicos que separan la realidad del modelo.
- El comportamiento de los sistemas se evalúa únicamente en base a las curvas de comportamiento. No se atiende a otros conceptos como el *squat* y el *dive*, que tienen en cuenta la influencia del elemento de transmisión secundaria sobre la suspensión.
- No se tienen en cuenta los efectos dinámicos, como las inercias o las fuerzas viscosas en el amortiguador. Todos los cálculos se realizan considerando el equilibrio estático.

Los resultados obtenidos por el programa pretenden ser únicamente un punto de partida para el diseñador, que deberá tener en cuenta muchos otros factores para definir por completo el diseño del sistema de suspensión.

Respecto del desarrollo de la aplicación de usuario, ésta se realizará íntegramente usando el lenguaje de programación Java. Para dotar al programa de una interfaz de usuario intuitiva, se empleará la plataforma JavaFX. No se emplearán librerías externas para el uso de Algoritmos Genéticos, sino que se desarrollará todo el código de manera propia para que se ajuste a la perfección con el problema que se pretende resolver.

## 3 Planificación del proyecto

Una vez se ha dejado claro cuáles son los objetivos del presente proyecto, y antes de entrar de lleno en materia, se empleará esta sección para explicar la planificación del mismo y el tiempo estimado que ha de dedicarse a cada actividad.



**Figura 3.** Estructura de desglose de trabajos para el proyecto que nos ocupa.

El objetivo principal se divide en cuatro fases. Cada una de ellas se divide, a su vez, en pequeños entregables o *Work Packs*, lo que constituye la Estructura de Desglose de Trabajos del proyecto, que puede verse en la Figura 3.

Todos los WPs llevan asociado un documento en el que se describe su objetivo de forma clara, así como la duración estimada del mismo. Al finalizar dicho objetivo, se escribe en el documento la fecha de finalización y se adjunta toda la documentación generada para llevar a cabo el WP. Una fase se termina cuando todos sus WP han sido cerrados. En la Figura 4 puede apreciarse la documentación correspondiente al WP3.1.

Considerando un promedio de trabajo de 3 horas al día, 4 días a la semana durante 9 meses, el tiempo estimado que se ha dedicado a cada uno de los entregables se muestra en la Tabla 1.



3 IMPLEMENTACIÓN DEL PROGRAMA

### WP3.1 Diseño de la interfaz de usuario y comportamiento del programa

**Descripción**

Definir por completo cómo será la interfaz de usuario del programa. Incluyendo todas sus opciones y herramientas. También, deberá indicarse cómo funciona el programa y los pasos a seguir para utilizarlo.

En primer lugar se realizarán bocetos de la interfaz del programa, explicando para qué sirve cada elemento de la misma y qué pasa si se interactúa con él. Posteriormente, se crearán todas las clases necesarias para definir por completo la interfaz de us...

**Duración**

El programa principal podrá ser el análisis de sistemas de suspensión de (parámetros) conocidos. En la interfaz deberá mostrarse:

- Representación gráfica del sistema de suspensión que se está analizando en el momento, qué o han graficas
- Datos de sistema de suspensión con el que se está trabajando - nombre del archivo, descripción...
- Lista de todos los parámetros del sistema y la opciones para modificarlos.
- Espacio para la representación de gráficos obtenidos por el usuario.

Esta información importante relativa a la acción del programa. En ejemplo, los ajustes de visualización. También guarda los  $\pi$  que están actualmente abiertos.

File → Almacena datos de un archivo con el sistema de suspensión. El nombre, el sistema de suspensión se si, y herramientas manejo del programa, como el (para poder usar Ctrl+Z, Ctrl+Y...).

Botón para cambiar entre varios sistemas de suspensión que el programa tiene abiertos.

Manejo de archivos por el objeto Session.

Este menú permite visualizar, cambiar de menú o salir de una pantalla, etc.

Así como se puede representar un sistema de suspensión de todos los variables.

→ Como elegir qué graficar se va a mostrar?

Hay que elegir:

- 1) Variable o graficas (pueden ser varias)
- 2) Variable independiente (alternando un variable)

Este menú podrá ser un panel de parámetros, como los que se puede elegir entre ver "grafías" o "representación".

Figura 4. Cierre del entregable WP3.1, con toda la documentación adjunta.

Work Pack	Tiempo (días)
<b>Modelado del sistema de suspensión</b>	
WP1.1	12
WP1.2	18
WP1.3	18
WP1.4	8
<b>Algoritmo de optimización</b>	
WP2.1	8
WP2.2	18
WP2.3	8
<b>Implementación del programa</b>	
WP3.1	12
WP3.2	2
WP3.3	24
WP3.4	12
<b>Documentación</b>	
WP4.1	6
WP4.2	6
WP4.3	20
WP4.4	8

**Tabla 1.** Tiempo estimado empleado en cada uno de los entregables del proyecto.

## 4 Estado del arte

Este trabajo es inherentemente multidisciplinar. Varias ramas del conocimiento – como son la Síntesis de Mecanismos, los Algoritmos Evolutivos o los Motores Físicos – se relacionan aquí con el objetivo de desarrollar una técnica para la optimización del sistema de suspensión trasero de una motocicleta. A continuación, se explican en qué consiste por separado cada una de estas materias y cómo han evolucionado hasta la actualidad.

### 4.1 El sistema de suspensión trasero en una motocicleta

Todavía a principios del siglo pasado en la mayoría de motocicletas lo más parecido a un sistema de suspensión que podía encontrarse eran unos muelles situados en el asiento del piloto. Sin embargo, muy pronto se hicieron necesarias las suspensiones delanteras, que permitían incrementar considerablemente la direccionabilidad y el control de la motocicleta, sobretudo en terrenos irregulares. Algo más tarde – a partir del primer cuarto de siglo – comenzaron a utilizarse sistemas de suspensión también para el tren trasero. Su función no era tan crítica, en un principio, como la del delantero, pero aumentaba considerablemente la comodidad del piloto (Arias-Paz, 2002).

### 4.1.1 Funciones del sistema de suspensión

Hoy en día existen motocicletas de todo tipo<sup>1</sup>, y cada uno de ellos responde a una serie de necesidades distintas. Sin embargo, en la práctica totalidad de ellos, las motocicletas incorporan sistema de suspensión, tanto delantero como trasero. Las necesidades más habituales a las que han de dar respuesta los sistemas de suspensión de una motocicleta son las siguientes.

- Garantizar la adherencia de las ruedas con el asfalto todo lo posible. Si se pierde la adherencia el piloto perderá el control del vehículo; no podrá girar, frenar ni acelerar.
- Aislar al piloto de las vibraciones de la carretera, haciendo la conducción más cómoda y reduciendo la fatiga.
- Reducir esfuerzos elevados debidos a impactos en la estructura de la motocicleta.
- Controlar la geometría que adquiere la motocicleta en ciertas condiciones críticas. Por ejemplo, durante una frenada brusca el peso se transferirá hacia la rueda de delante, comprimiendo la suspensión. Debe de garantizarse que, aunque esto ocurra, la suspensión siga teniendo capacidad de absorber las irregularidades del terreno. Algo parecido ocurre con la rueda trasera al acelerar.

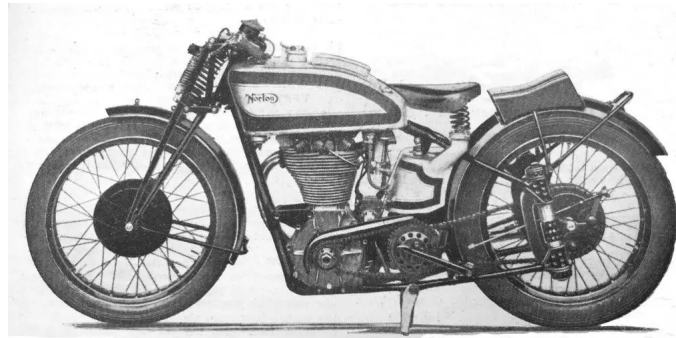
A menudo, algunas de estas necesidades entran en conflicto. El caso más habitual es el de mantener cierto confort durante la conducción con respecto a garantizar el suficiente agarre de los neumáticos con el asfalto. En estos casos, dependiendo del tipo de motocicleta y el uso que vaya a dársele, deberemos llegar a un compromiso en favor de un criterio u otro (Foale, 2002). Por ejemplo, en motocicletas de competición se prioriza mantener el agarre con el asfalto a que la motocicleta sea cómoda (Cossalter, 2006).

### 4.1.2 Tipos de sistemas de suspensión traseros

Entre los primeros intentos por otorgar a la rueda trasera de sistema de suspensión, el que se extendió más ampliamente fue el sistema de suspensión por émbolo. Éste consistía en anclar la rueda a un soporte deslizante, que permitía su movimiento vertical respecto del chasis, y entre el que se intercalaba un muelle y/o amortiguador. El rango de movimiento de la rueda que admitían estos sistemas

---

<sup>1</sup>Deportivas, turismo, trail, enduro, scooters... El lector puede consultar el Capítulo I de *Motocicletas* (Arias-Paz, 2002) para ver una descripción detallada de los principales tipos que existen.



(a) Norton Manx de 1936.



(b) Velocette Mark VIII de 1937.

**Figura 5.** Comparativa entre suspensión trasera de tipo émbolo (a) y de tipo basculante clásica (b).

era limitado, por no hablar de las grandes variaciones de tensión que se producían en el sistema de transmisión secundario (habitualmente mediante correa o cadena). Esto hizo que a poco que se desarrollasen los sistemas basados en brazo basculante, desplazasen rápidamente y casi por completo al sistema por émbolo (Arias-Paz, 2002). Aún hoy en día son estos últimos los sistemas más utilizados con diferencia para la suspensión trasera. Por ello, a continuación se explicará con más detalle en qué consisten y las distintas variaciones que existen.

El basculante es una estructura que se ancla en un extremo al chasis, mediante un eje de rotación denominado *pivot point*. En su otro extremo se ancla el eje de la rueda trasera. El *pivot point* se sitúa lo más cerca posible de piñón del motor, de forma que la variación en la tensión del sistema de transmisión es muy pequeña incluso ante grandes recorridos verticales de la misma. En sus primeras versiones, lo que podríamos denominar **el tipo clásico**, el basculante consiste esencialmente en un tubo en el que se alojan los rodamientos del *pivot point*, soldado a dos brazos, uno a cada lado de la rueda, en cuyos extremos se ancla el eje de la misma. En cada uno de los brazos se ancla en un punto muy cercano al eje de la rueda la parte inferior de un amortiguador. Su extremo superior se une directamente al chasis. Un ejemplo de este tipo de suspensión puede apreciarse en la Figura 5b.

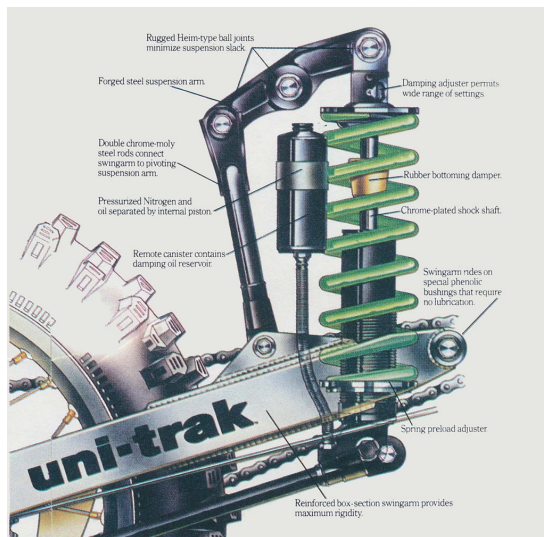
A partir de los años 80 comenzaron a implantarse nuevos sistemas que utilizaban un solo amortiguador en aquellas motocicletas que requerían de mayores prestaciones (Arias-Paz, 2002). El sistema de suspensión tipo **cantiléver**, muy usado por Yamaha, consistía en un basculante más rígido que permitiese utilizar un solo amortiguador centrado y situado delante de la rueda trasera. La posición del amortiguador ahora es mucho más libre – no es necesario disponer los amortiguadores de manera vertical y al final del basculante. Por consiguiente, entran en juego más variables de diseño que permiten ajustar el comportamiento de cada sistema como se requiera.

Estos avances seguían siendo insuficientes, sobretudo en algunos sectores del motociclismo de competición como es el enduro o moto-X. Estas motocicletas requieren que la suspensión admita enormes rangos de movimiento para adaptarse a terrenos altamente irregulares, así como amortiguar los saltos. Surgieron entonces los **sistemas de bieletas** que, a costa de añadir mayor complejidad, permiten ajustar muy bien el comportamiento de un sistema de suspensión a un objetivo especificado. Tanto es así que se extendieron rápidamente a otros ámbitos del mundo del motociclismo, y hoy en día es el tipo de sistema utilizado en la mayoría de motocicletas de altas prestaciones.

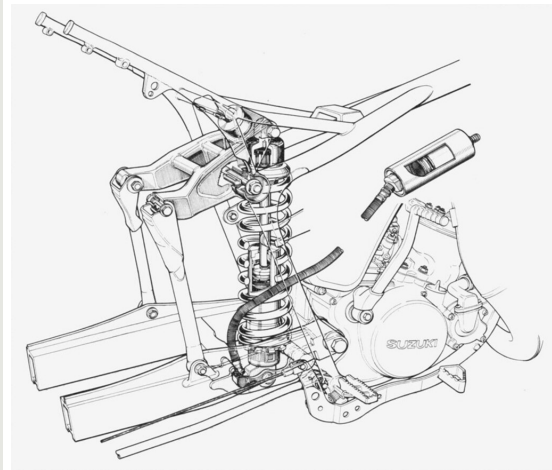
En seguida surgieron de las principales firmas de motocicletas múltiples variantes de este tipo de sistemas. Sin embargo, ninguno de ellos presenta características muy diferentes del resto – la variedad se debe más bien a cuestiones legales de propiedad intelectual, y de marketing (Arias-Paz, 2002). En la Figura 6 pueden apreciarse algunas de estas variantes y sus nombres comerciales.

### 4.1.3 Comportamiento

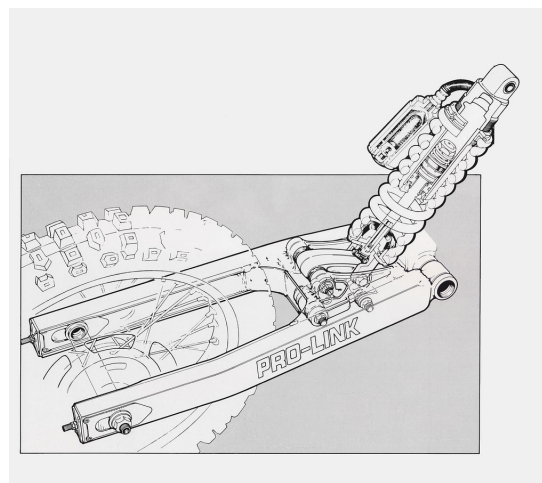
Imaginemos un sistema de suspensión lo más sencillo posible: la unidad de suspensión (conjunto muelle más amortiguador) se coloca verticalmente uniendo el chasis con la rueda trasera. La rueda, a su vez, no puede desplazarse horizontalmente con respecto al chasis. Imaginemos también que el chasis permanece fijo, y que aplicamos una fuerza vertical a la rueda trasera de forma que ésta se desplaza hacia arriba. Dado que el muelle de la unidad de suspensión suele tener un coeficiente de rigidez constante, para desplazar la rueda verticalmente una cierta cantidad debemos sumar a la carga aplicada siempre un mismo incremento, sea cual sea la posición de la rueda. Diremos, en este caso, que **la suspensión es lineal**. En general la topología y geometría del sistema de suspensión son tales que, a lo largo de todo el rango de movimiento del sistema, los desplazamientos verticales que se producen en la rueda trasera no son proporcionales a los que se producen entre los dos extremos de la unidad de suspensión.



(a) Yamaha Uni-Track.



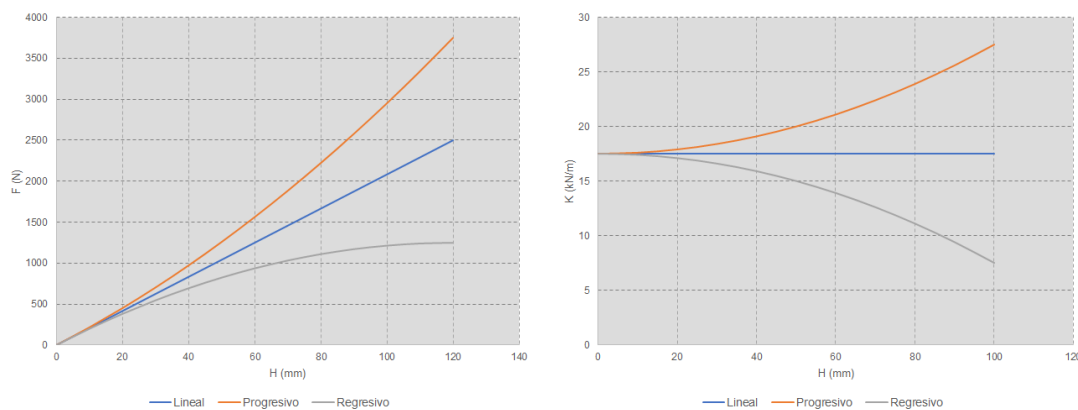
(b) Suzuki Full-Floater.



(c) Honda Pro-Link.

**Figura 6.** Algunos ejemplos de suspensiones con sistemas de bieletas de marcas conocidas.





(a) Fuerza aplicada vs. desplazamiento. (b) Rigidez equivalente vs. desplazamiento.

**Figura 7.** Diferencias de comportamiento entre las suspensiones lineal, progresiva y regresiva.

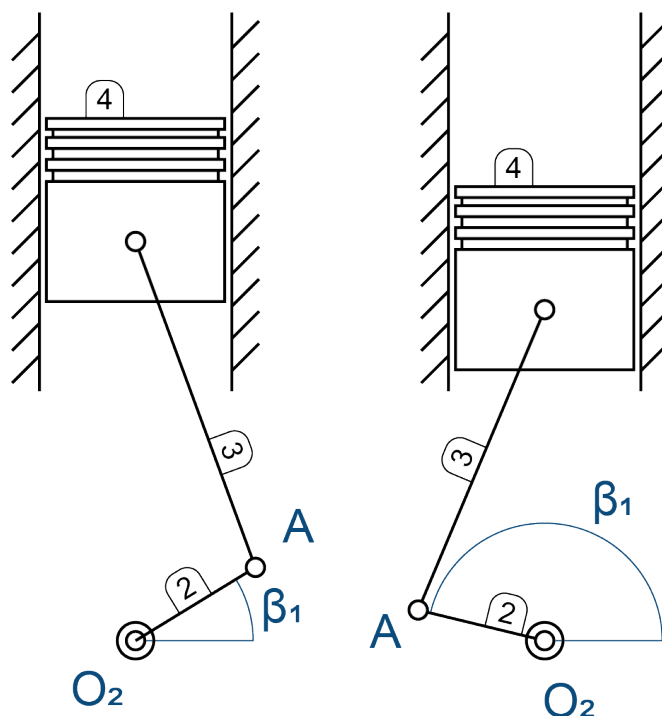
En estos casos, la fuerza necesaria para desplazar la rueda verticalmente una misma cantidad será diferente dependiendo de la posición en la que se encuentre la rueda respecto de su rango de movimiento. Diremos que **la suspensión es progresiva** cuando se requieran incrementos mayores en la fuerza aplicada para desplazar verticalmente la rueda una misma cantidad conforme nos acercamos al límite superior. En el caso opuesto, diremos que **la suspensión es regresiva**. En la Figura 7 puede apreciarse bien las diferencias entre los tres comportamientos.

Podemos asimilar cualquier sistema de suspensión a uno como el caso sencillo que se introdujo anteriormente, siempre que consideremos que la rigidez del muelle de este sistema ficticio es una función del desplazamiento de la rueda, y no una constante. Graficando esta función tendremos la *curva de rigidez equivalente o reducida* del sistema de suspensión (ver Figura 7b), que es uno de los parámetros más empleados para evaluar su comportamiento. Habitualmente, las suspensiones progresivas son más ventajosas respecto de las regresivas o lineales (Cossalter, 2006). Las distintas tipologías para los sistemas de suspensión traseros, que vemos a continuación, han surgido en parte como un intento por mejorar la progresividad y controlar mejor la curva de rigidez equivalente del sistema.

## 4.2 La teoría de máquinas

Un mecanismo es un conjunto de sólidos interconectados entre sí de tal forma que existe movimiento relativo entre ellos, considerándose uno de ellos fijos. A estos sólidos se les conoce como *barras* o *eslabones*, y suelen denotarse con un número entero, siendo la barra 1 el eslabón fijo o *bancada*.

En el espacio un cuerpo libre puede moverse y rotar en los tres ejes carte-



**Figura 8.** Distintas posiciones del mecanismo biela-manivela-corredera.

sianos (3 desplazamientos y 3 rotaciones). Sin embargo, en un mecanismo las barras se unen mediante los denominados *pares*, que restringen los posibles movimientos entre las barras que relaciona, de forma que el movimiento de todas las barras del mecanismo queda completamente condicionado a solo unas pocas variables. Al número de variables necesarias para determinar la posición de todas las barras del mecanismo se le denomina *grados de libertad*.

Existen diferentes tipos de pares, según la forma en que restrinjan el movimiento de los dos sólidos que enlaza. Los pares de rotación, por ejemplo, imponen la condición de que un punto (P) sea coincidente en todo momento entre dos sólidos. O, lo que es lo mismo, que la velocidad respecto de la barra fija en ambos sólidos ( $i$  y  $j$ ) sea la misma ( $\vec{v}_{i1}^P = \vec{v}_{j1}^P$ ). Un par prismático, por el contrario, permite el desplazamiento entre dos barras pero solo en una dirección preestablecida.

El mecanismo de biela-manivela-corredera, muy conocido por su uso en los motores de vehículos de combustión, consta de 4 barras, como puede apreciarse en la Figura 8. Éstas están unidas entre sí mediante tres pares de rotación (entre las barras 1 y 2, las barras 2 y 3, y las barras 3 y 4), y uno prismático (entre las barras 1 y 4). En el ejemplo, el ángulo  $\beta_1$  determina la posición de todas las barras del mecanismo. Por tanto, el mecanismo tiene un solo grado de libertad. Diremos además que la barra 2 es *la barra de entrada*.



### 4.2.1 Análisis de mecanismos

La palabra análisis hace referencia al estudio de algo que es conocido – en este caso un mecanismo – para saber cómo funciona y cuáles son sus características. Podemos diferenciar, en este contexto, entre análisis cinemático y análisis dinámico.

El análisis cinemático de un mecanismo busca obtener el valor de las variables cinemáticas en cualquier punto de todas las barras del mismo. Para ello han de resolverse, de forma secuencial, tres problemas diferentes:

- El problema de posición: Se busca obtener la posición de todas las barras en función de un número de variables igual al número de grados de libertad del mismo.
- El problema de velocidades: El objetivo es obtener el valor de la velocidad de cualquier punto de todas las barras del mecanismo a partir de la velocidad de las barras de entrada.
- El problema de aceleraciones: Análogamente, resolverlo implica poder calcular las aceleraciones de todas las barras dadas las aceleraciones de las barras de entrada.

Por otro lado, el análisis dinámico se centra en la interacción de las barras entre sí y con el entorno. Esta interacción se materializa en términos de fuerzas y momentos. Así, por ejemplo, en cuanto a la interacción de las barras entre sí, interesa en muchas ocasiones conocer qué fuerzas se desarrollan entre los pares que unen las mismas, para así poder dimensionar los elementos mecánicos que lo compondrán (rodamientos, ejes...). Como ejemplo de interacciones con el entorno, podríamos imaginar el momento que hace un motor sobre la barra de entrada de un mecanismo para que ésta se mueva. El análisis dinámico requiere de la realización, en primer lugar, del análisis cinemático.

### 4.2.2 Síntesis de mecanismos

En la síntesis, al contrario que en el análisis, dado un comportamiento deseado y/o una serie de requerimientos, se busca dar con el mecanismo que mejor los cumple. De acuerdo con (Domínguez Abascal, 2016), existen dos categorías principales de síntesis de mecanismos.

- La síntesis estructural: Se centra en la elección del tipo de mecanismo – el número de barras y la forma en la que éstas se interconectan.

- La síntesis dimensional: Dado un mecanismo definido de manera conceptual, busca determinar las dimensiones básicas del mismo para lograr los objetivos preestablecidos.

El presente trabajo se relaciona directamente con la síntesis de mecanismos. Sin embargo, cabe destacar que la síntesis de tipo estructural no es de aplicación. Partimos de una serie de sistemas de suspensión – mecanismos, en definitiva – ya preestablecidos. El número de barras y cómo éstas se conectan entre sí es conocido. El objetivo será el de ajustar las dimensiones básicas, los parámetros del mismo, para que éste tenga un comportamiento lo más parecido posible al buscado.

### **4.3 Los algoritmos genéticos y otras técnicas de computación evolutiva**

Como en muchas otras áreas del conocimiento, en el ámbito de la informática la naturaleza también ha sido una gran fuente de inspiración. Su papel más relevante reside en la búsqueda de algoritmos genéricos para la solución de problemas complejos. La inspiración se ha tomado principalmente de dos fuentes diferentes: (i) el cerebro humano, dando lugar a la neurocomputación; y (ii) la teoría de la evolución, dando lugar a la Computación Evolutiva (Smith y col., 2015).

Existen multitud de problemas en los que no es posible obtener una solución óptima de forma analítica. Sin embargo, en muchas ocasiones, ante dos soluciones diferentes al problema sí es posible determinar cual de ellas es mejor – podemos calcular cuál es el desempeño de una solución cualquiera. La Computación Evolutiva supone una manera diferente de enfocar los problemas, que parte de esta premisa y aplica los conceptos de la teoría de la evolución darwiniana.

En muy resumidas cuentas, ante un problema en primer lugar se plantea un conjunto de soluciones (que en principio suelen ser aleatorias). Éstas se evalúan y se determina cuál es su desempeño. Las soluciones planteadas en primer lugar se utilizarán para generar nuevas soluciones al problema, modificándose parcialmente o combinándose varias entre sí, en función de su nivel de desempeño. Las soluciones que han obtenido un desempeño mayor tendrán mayor probabilidad de utilizarse como referencia para generar soluciones nuevas, con el objetivo de que las bondades de cada solución se combinen entre sí dando lugar a soluciones mejores. Estas nuevas soluciones obtenidas se evalúan y se emplean en generar otro conjunto de soluciones. El proceso se repite iterativamente hasta que se cumpla alguno de los criterios de terminación preestablecidos que, en el mejor de los casos, consistirá en la convergencia del algoritmo hacia una o varias

soluciones que maximizan el desempeño.

### 4.3.1 Componentes de un Algoritmo Evolutivo

Utilizando una terminología más acorde a la analogía evolutiva, cada una de las soluciones que se plantean son conocidas como *individuos*. El problema planteado es el *entorno* en el que estos individuos, cuando pertenecen a una misma *población*, compiten. Los individuos más aptos tendrán mayor probabilidad de ser seleccionados para la creación de los nuevos individuos de la siguiente *generación*.

Para definir por completo un Algoritmo Genético es necesario establecer previamente 8 conceptos, procedimientos y operadores (Smith y col., 2015). A continuación, se explican con algo más de detalle.

#### Representación

El proceso de representación consiste en la creación de un modelo genérico para las posibles soluciones al problema que se pretende resolver. Este modelo permitirá expresar cualquier solución a partir de datos manejables por un ordenador. El objetivo es que, a partir del valor una serie de variables o parámetros predefinidos, obtengamos una y solo una equivalencia con una solución real al problema.

Distinguiremos, por tanto, entre dos ámbitos dentro de una misma solución. Por un lado tendremos la solución real, que puede ser evaluada y calculado un valor de adecuación. A este ámbito lo denominaremos como el *fenotipo*. Por otro, tenemos la representación de la solución según el modelo establecido, que consiste en el valor de un conjunto de parámetros. Sobre ellos serán los que actúen los operadores del algoritmo para la búsqueda de la solución óptima. Haremos referencia a este ámbito como el *genotipo*.

Usualmente, la representación de las soluciones tiene forma de cadenas de datos – ya sean de bits, de números enteros o decimales. Por ello, a estas cadenas se las conoce también como *cromosoma*, y a cada elemento de la misma como *gen*.

#### Función de aptitud

La función de aptitud o *fitness* asocia a toda solución planteada un valor numérico, que se utilizará como una medida de la bondad de la misma. Es esta función, por tanto, la que define el objetivo de la búsqueda y permite evaluar cuánto mejor o peor es una solución respecto de otra.

## Población

Se conoce como población al conjunto de individuos que se está evaluando en un momento determinado. Los individuos son estáticos – no se modifican durante el transcurso del proceso. Es la población la que va modificándose, albergando cada vez a individuos más aptos. Por ello, la población es la unidad de evolución (Smith y col., 2015). Aunque puede haber otros factores a tener en cuenta, lo más común es que simplemente sea necesario escoger el número de individuos que va a tener la población en todo momento.

## Mecanismo de selección de progenitores

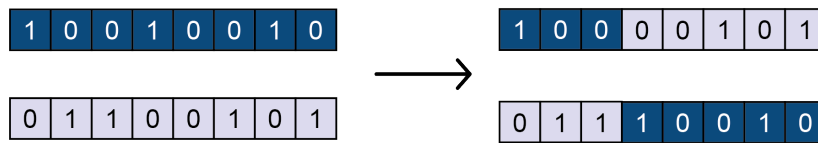
El mecanismo de selección de progenitores define las reglas que van a emplearse para escoger, en cada iteración del proceso, cuáles van a ser los individuos de la generación actual que van a utilizarse para crear nuevos individuos en la siguiente generación. El objetivo es sesgar la selección de los progenitores de forma que los individuos más aptos tengan mayor probabilidad de ser escogidos para dar lugar a la siguiente generación. Así, se trata de conseguir que generación tras generación se vaya aumentando la calidad de los individuos.

## Mecanismo de generación de la descendencia

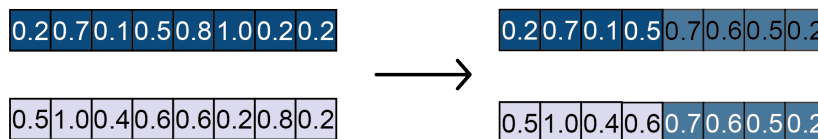
Definen la forma mediante la cual se crean nuevos individuos para la siguiente generación a partir de los individuos seleccionados de la generación anterior. Existen dos operadores típicos para ello:

**Mutación:** La mutación es un operador que actúa sobre un único individuo, cambiando de forma aleatoria parte de su material genético. Se emplea con una escasa probabilidad sobre los individuos que van a formar parte de una nueva generación. Se fomentan así la creación de soluciones innovadoras que permitan abarcar mejor todo el espacio de búsqueda. Además, ayudan a evitar que la población en su totalidad recaiga en máximos locales. Este procedimiento suele conocerse como *exploración*.

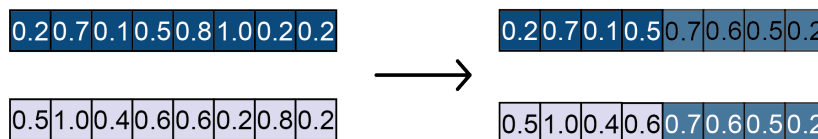
**Reproducción o *crossover*:** Esta operación combina la información genética contenida en varios individuos (usualmente dos), denominados padres, para dar lugar a uno o varios individuos, denominados hijos. Algunos ejemplos se muestran en la Figura 9 para varios tipos de representación. El objetivo de esta operación es tratar de conseguir un nuevo individuo que combine las bondades de los padres, de forma que su adecuación sea mayor. A este procedimiento se le conoce también como *explotación*.



(a) *One point crossover* para una representación de tipo binaria.



(b) *Simple arithmetic recombination* para una representación de tipo coma flotante.



(c) *Uniform crossover* para una representación de tipo entero.

**Figura 9.** Algunos operadores para la reproducción de varios tipos de representaciones, según (Smith y col., 2015).

### Mecanismo de selección de supervivientes

En ocasiones es conveniente que en el proceso de generación de la descendencia se creen más individuos de los que luego van a incluirse en la población. Por ello, deben existir unas reglas que descarten algunos de ellos y que favorezcan que las sucesivas generaciones vayan obteniendo cada vez un mejor desempeño.

### Inicialización

La inicialización consiste en establecer todos los parámetros de la primera generación de individuos. Para ello, lo más común es crear individuos cuyos parámetros se hayan establecido de forma aleatoria.

### Condición de finalización

Establece unos requisitos tales que si se cumple alguno de ellos se da por concluido el proceso de optimización. En el mejor de los casos, esto ocurre cuando se ha alcanzado una solución al problema lo suficientemente buena. Sin embargo, hay que tener en cuenta que el algoritmo puede que nunca llegue a dicha solución, por lo que habrá que especificar otros requisitos para que éste no se convierta en un bucle infinito.

## 5 Diseño e implementación

Una vez hemos comprendido las principales áreas de conocimiento en las que se basa este trabajo de manera independiente, vamos a explicar cómo se combinan todas ellas en busca del objetivo propuesto. Tomaremos como base los componentes, procedimientos y operadores necesarios para definir un Algoritmo Evolutivo según (Smith y col., 2015), y que se resumieron en la sección 4.3.1. En su desarrollo utilizaremos herramientas propias de la teoría de máquinas y la síntesis de mecanismos aplicadas a los conceptos vistos sobre sistemas de suspensión traseros de motocicletas.

### 5.1 Representación

Para definir por completo, de manera conceptual, un sistema de suspensión trasero de motocicleta se necesita conocer: (i) el número de barras que lo componen y la forma en que éstas se conectan entre sí; (ii) la posición de todos los puntos de interés como los pares entre cada barra y los puntos en los que se aplican fuerzas; y (iii) algunos valores específicos como la constante elástica del amortiguador o el diámetro de la rueda.

El número de barras y la forma en que éstas se conectan entre sí viene ya especificado según el tipo de sistema de suspensión que se quiera optimizar. Por este motivo, esta información no debe estar reflejada en el material genético de cada individuo, sino que forma parte de la representación de cada tipo de sistema. El resto de variables – la posición de los puntos de interés y las características del amortiguador – son magnitudes físicas. Por ello, para representar un sistema de suspensión se opta por una cadena de variables tipo `double`, que formará el genotipo de cada individuo. El número de parámetros, o genes, dependerá también del tipo de sistema de suspensión que se pretenda optimizar. La aplicación permitirá al usuario escoger entre uno de los siguientes:

- Sistema de suspensión clásico.

- Sistema de suspensión tipo cantilever.
- Sistema de suspensión tipo Pro-Link.
- Sistema de suspensión tipo Unit-Pro-Link.
- Sistema de suspensión tipo Uni-Track.
- Sistema de suspensión tipo Full-Floater.

Para cada tipo de sistema de suspensión se crea una clase que lo represente. Un objeto de cualquiera de estas clases albergará toda la información y funcionalidad necesaria para poder evaluar la solución, por lo que será considerado el fenotipo de la misma. También, incluirá funcionalidad relacionada con la decodificación del genotipo al fenotipo.

### 5.1.1 La superclase `SuspensionSystem`

Cada tipo de sistema de suspensión requiere de su propia representación. Sin embargo, el resto de componentes del algoritmo genético es igual en todos ellos. Por tanto, se crea una clase abstracta llamada `SuspensionSystem` que define la forma en la que las clases deben interactuar con el resto del programa. Todas las clases antes mencionadas extenderán a ésta, e implementarán su propia funcionalidad sobrescribiendo los métodos de la superclase. Esta superclase establece, por tanto, la interfaz común entre todas las clases relacionadas con un sistema de suspensión y el resto del programa. A continuación, analizamos con algo más de detalle esta interfaz para entender mejor el comportamiento del programa.

En primer lugar, la clase `SuspensionSystem` define una serie de constantes que son utilizadas a lo largo de todo el programa. Por un lado, se establece un identificador numérico para cada uno de los tipos de sistema de suspensión considerados, y se crea un vector con sus nombres. Lo mismo se hace con las variables de cálculo del programa y, además de su nombre, se especifica su abreviatura y sus unidades.

```

1 public static final int CLASSIC_TYPE = 0;
2 public static final int CANTILEVER_TYPE = 1;
3 public static final int PROLINK_TYPE = 2;
4 public static final int UNIT_PROLINK_TYPE = 3;
5 public static final int UNITRACK_TYPE = 4;
6 public static final int FULLFLOATER_TYPE = 5;
7 public static final String[] typeNames = {
8     "Classic suspension system",
9     "Cantilever swingarm suspension system",
10    "Unit Pro-Link suspension system",
11    "Pro-Link suspension system",

```

```

12     "Uni-Track suspension system",
13     "Full floater suspension system"};
14
15 public static final int SWINGARM_ANGLE_VARIABLE = 0;
16 public static final int VERTICAL_WHEEL_DISPLACEMENT_VARIABLE = 1;
17 public static final int SUSPENSION_UNIT_COMPRESSION_VARIABLE = 2;
18 public static final int VERTICAL_FORCE_ON_WHEEL_VARIABLE = 3;
19 public static final int EQUIVALENT_STIFFNESS_VARIABLE = 4;
20 public static final String[] variables = {
21     "Swingarm angle",
22     "Vertical wheel displacement",
23     "Suspension Unit compression",
24     "Vertical force on rear wheel",
25     "Equivalent Stiffness"};
26 public static final String[] symbols = {"B", "H", "SUC", "F", "K"};
27
28 public static final String[] units = { "deg", "mm", "mm", "N", "N/
    mm"};

```

**Bloque de código 1.** Declaración de constantes de la clase SuspensionSystem.

Las variables de cálculo son las que van a permitir definir el comportamiento de un sistema de suspensión determinado. El objetivo de cada subclase es calcular el valor de cada una de estas variables para cada posición dentro del rango de movimiento del sistema de suspensión, y en función de los parámetros de entrada. Cada subclase deberá implementar, por tanto, los métodos que se muestran en el Bloque de código 1. La variable  $b$ , denominada *variable básica*, se utiliza internamente en cada clase para calcular el valor de todas las demás. Un valor concreto de  $b$  se corresponde con una posición determinada del sistema de suspensión y permite establecer el valor del resto de variables justo en dicha posición. Su significado geométrico depende de la implementación de cada subclase. Deben especificarse siempre valores dentro del rango de movimiento del sistema, para lo cual los métodos `getLowerLimit()` y `getUpperLimit()` han de implementarse de forma que devuelvan los límites del mismo.

```

1 public abstract double getSwingarmAngleAt(double b);
2 public abstract double getVerticalWheelDisplacementAt(double b);
3 public abstract double getSuspensionUnitCompressionAt(double b);
4 public abstract double getVerticalForceAt(double b);
5 public abstract double getEquivalentStiffnessAt(double b);
6 public double getVariableValueAt(int variable, double b) {
7     switch (variable) {
8         case SuspensionSystem.SWINGARM_ANGLE_VARIABLE:
9             return getSwingarmAngleAt(b);
10        case SuspensionSystem.VERTICAL_WHEEL_DISPLACEMENT_VARIABLE:
11            return getVerticalWheelDisplacementAt(b);
12        case SuspensionSystem.SUSPENSION_UNIT_COMPRESSION_VARIABLE:
13            return getSuspensionUnitCompressionAt(b);
14        case SuspensionSystem.VERTICAL_FORCE_ON_WHEEL_VARIABLE:
15            return getVerticalForceAt(b);
16        case SuspensionSystem.EQUIVALENT_STIFFNESS_VARIABLE:
17            return getEquivalentStiffnessAt(b);
18        default:

```



```

19         System.out.println("Unexistent variable.");
20     }
21     return 0;
22 }
23
24 public abstract double getLowerLimit();
25 public abstract double getUpperLimit();
26
27 public abstract void recalc();
28

```

**Bloque de código 2.** Métodos que debe implementar cada subclase de `SuspensionSystem`.

Por último, el método `recalc()` se utiliza cuando se ha modificado alguno de los parámetros del sistema, para actualizar las variables internas del mismo y que las funciones anteriormente especificadas devuelvan los datos acorde a los cambios realizados. Básicamente, es la función que se encarga de la decodificación. Los parámetros de cada sistema (el material genético) se almacena también en la clase, en la variable `ArrayList<ParameterGroups> parameterGroups` agrupados de forma que sea más sencilla su identificación por el usuario del programa.

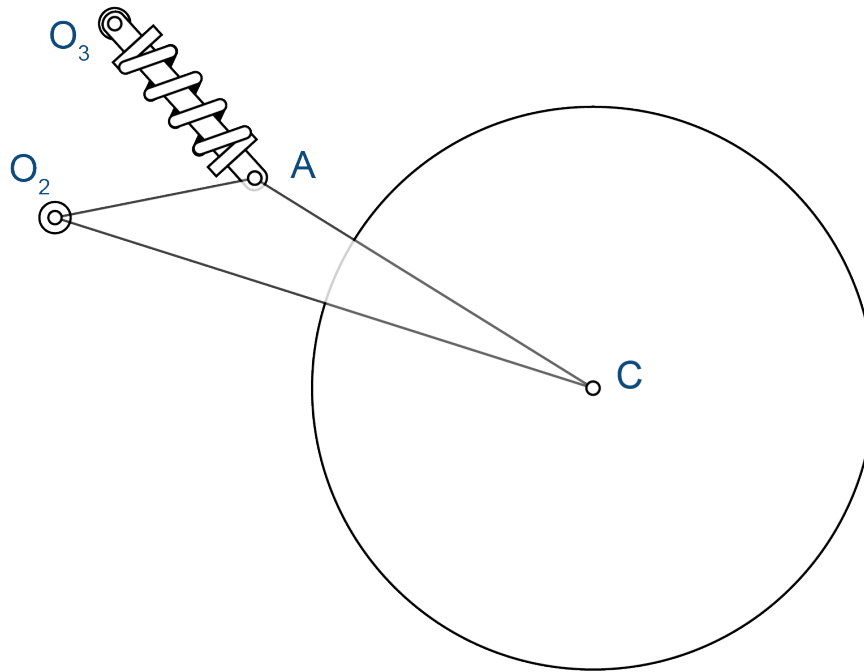
### 5.1.2 Las subclases de `SuspensionSystem`

Definido el esquema general de cualquiera de las clases que representan a un sistema de suspensión, en este apartado se particularizará cada uno de ellos. En primer lugar, se expondrán los parámetros que son necesarios para su definición. Posteriormente se estudiará cómo, a partir de ellos, pueden calcularse las variables que definirán el comportamiento del sistema.

#### Sistemas de suspensión tipo clásico y cantiléver

El sistema de suspensión tipo Cantilever es muy sencillo. Como puede apreciarse en la Figura 10, solo consta de una barra – el basculante – y el amortiguador. Se ha tomado como variable interna  $b$  el ángulo del basculante respecto de la horizontal. En base a ésta, vamos a ver cómo puede obtenerse el valor de las variables de comportamiento. Los parámetros serán considerados como datos de entrada. Todos ellos se muestran en la Tabla 2.

En cuanto al sistema de suspensión tipo clásico, el modelo matemático es exactamente igual que el del tipo Cantilever. La única diferencia es que el tipo clásico utiliza dos amortiguadores en vez de uno, por lo que la fuerza del amortiguador debería ser multiplicada por dos a la hora de realizar los cálculos. Para no presentar información redundante, los cálculos se realizarán considerando únicamente el sistema tipo cantilever.



**Figura 10.** Esquema de un sistema de suspensión tipo Cantiléver.

**Ángulo del basculante:** Es el ángulo que forma con la horizontal la línea imaginaria que une el Pivot Point con el eje de la rueda trasera. Corresponde directamente con la variable  $b$  en este caso.

$$B(b) = b \quad (1)$$

**Desplazamiento vertical de la rueda trasera:** Es la distancia a la que la rueda se encuentra por encima del suelo. Su cálculo es inmediato usando trigonometría básica.

$$H(b) = O_{2y} + s \cos(b) - R \quad (2)$$

**Compresión de la unidad de suspensión:** Es la diferencia entre la longitud natural del amortiguador y su longitud en una posición determinada.

$$SUC(b) = l_n - l_{SU}(b) \quad (3)$$

Siendo  $l_{SU}(b) = \sqrt{[O_{3x} - A_x(b)]^2 + [O_{3y} - A_y(b)]^2}$ , la longitud del amortiguador en la posición  $b$ . Las coordenadas del punto  $A$  también dependen de  $b$ . Sus componentes pueden calcularse según  $A_x(b) = O_{2x} + s_x \cos b - s_y \sin b$  y  $A_y(b) = O_{2y} + s_x \sin b + s_y \cos b$ .

**Fuerza vertical en la rueda trasera:** Es la fuerza que hay que aplicar verticalmente y hacia arriba sobre el eje de la rueda trasera para que el sistema esté en equilibrio cuando se encuentra en una posición  $b$ . Tomando momentos

Parámetro	Descripción
$O_{2x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{3x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{3y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
$R$	Radio de la rueda trasera.

**Tabla 2.** Nombre y descripción de los parámetros necesarios para definir un sistema de suspensión tipo clásico y Cantilever.

respecto del punto  $O_2$  nos queda la expresión  $s \cos b F - (s_x \sin b + s_y \cos b) F_{ex} + (s_x \cos b - s_y \sin b) F_{ey}$  que, igualando a 0 debido al equilibrio estático nos quedaría:

$$F = \frac{(s_x \sin b + s_y \cos b) F_{ex}(b) - (s_x \cos b - s_y \sin b) F_{ey}(b)}{s \cos b} \quad (4)$$

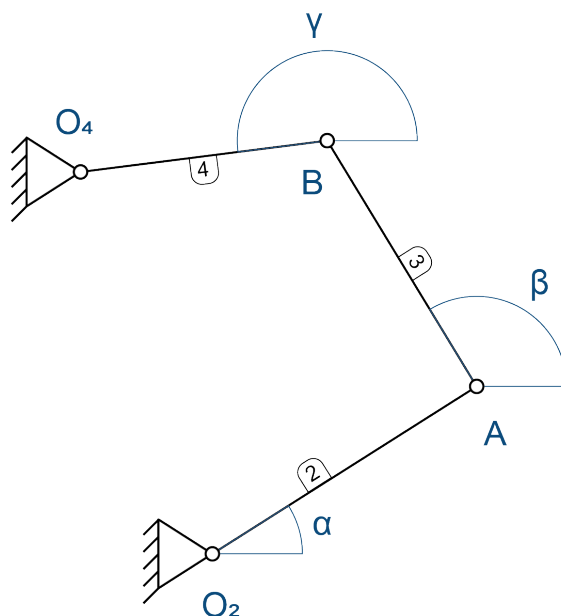
$F_{ex}(b)$  y  $F_{ey}(b)$  son las componentes de la fuerza elástica del muelle sobre el punto  $A$  del basculante, que también dependen de la posición  $b$  del sistema de suspensión. El valor de la fuerza elástica total viene dada por  $F_e(b) = k \times SUC(b)$ . Mediante semejanza de triángulos, podemos establecer el valor de ambas componentes como  $F_{ex}(b) = -\frac{O_{3x}-A_x(b)}{l_{SU}} F_e(b)$  y  $F_{ey}(b) = -\frac{O_{3y}-A_y(b)}{l_{SU}} F_e(b)$ .

**Rigidez equivalente:** Es la derivada de la fuerza  $F(b)$  respecto del desplazamiento vertical de la rueda  $H(b)$ . Su cálculo se hace de forma numérica y es igual para todos los sistemas de suspensión.

## Sistemas de bieletas

Los sistemas de bieletas se basan en un mecanismo de cuatro barras para añadir mayor progresividad a la suspensión. Por ello, en primer lugar, se analizará en detalle este mecanismo y sus características.

Como vimos en la sección la Sección 4.2, el primer paso para analizar un mecanismo es resolver el **problema de posición**. Dado que el mecanismo de cuatro barras tiene un solo grado de libertad, la posición de todas las barras del mecanismo quedará determinada en función de una única variable. Se considerará como barra de entrada la 2, por lo que la variable independiente será el ángulo  $\alpha$  (ver



**Figura 11.** Esquema de un mecanismo de cuatro barras.

la Figura 11. Los datos de partida son las longitudes de las barras 2, 3 y 4 ( $l_2$ ,  $l_3$  y  $l_4$ , respectivamente), y los puntos  $O_2$  y  $O_4$ .

De la ecuación vectorial  $O_2\vec{A} + \vec{AB} + B\vec{O}_4 = O_2\vec{O}_4$ , ordenando los términos adecuadamente, se llega a las dos ecuaciones siguientes:

$$\begin{cases} l_4 \cos \gamma = O_{4x} - O_{2x} - l_2 \cos \alpha - l_3 \cos \beta \\ l_4 \sin \gamma = O_{4y} - O_{2y} - l_2 \sin \alpha - l_3 \sin \beta \end{cases}$$

Elevando al cuadrado ambas ecuaciones y sumándolas, obtenemos:

$$l_4^2 = [(O_{4x} - O_{2x} - l_2 \cos \alpha) - l_3 \cos \beta]^2 + [(O_{4y} - O_{2y} - l_2 \sin \alpha) - l_3 \sin \beta]^2$$

Que, si continuamos operando y simplificando:

$$\begin{aligned} l_4^2 &= [(O_{4x} - O_{2x} - l_2 \cos \alpha)^2 + l_3^2 \cos^2 \beta - 2(O_{4x} - O_{2x} - l_2 \cos \alpha) l_3 \cos \beta] \\ &+ [(O_{4y} - O_{2y} - l_2 \sin \alpha)^2 + l_3^2 \sin^2 \beta - 2(O_{4y} - O_{2y} - l_2 \sin \alpha) l_3 \sin \beta] = \\ &= (O_{4x} - O_{2x})^2 + (O_{4y} - O_{2y})^2 + l_2^2 + l_3^2 \\ &\quad - 2(O_{4x} - O_{2x})l_2 \cos \alpha - 2(O_{4y} - O_{2y})l_2 \sin \alpha \\ &\quad - 2(O_{4x} - O_{2x} - l_2 \cos \alpha) l_3 \cos \beta - 2(O_{4y} - O_{2y} - l_2 \sin \alpha) l_3 \sin \beta \end{aligned}$$

Tomando  $A = (O_{4x} - O_{2x})^2 + (O_{4y} - O_{2y})^2 + l_2^2 + l_3^2 - l_4^2$ ,  $B = -2(O_{4x} - O_{2x} - l_2 \cos \alpha)$  y  $C = -2(O_{4y} - O_{2y} - l_2 \sin \alpha) l_3$ , que dependen únicamente del ángulo  $\alpha$ , la

ecuación puede simplificarse como sigue.

$$\begin{aligned}
 A + B \cos \beta + C \sin \beta &= 0 \\
 A + B \cos \beta + C \sqrt{1 - \cos^2 \beta} &= 0 \\
 A + B \cos \beta &= C \sqrt{1 - \cos^2 \beta} \\
 A^2 + B^2 \cos^2 \beta + 2AB \cos \beta &= C^2 (1 - \cos^2 \beta) \\
 \underbrace{(B^2 + C^2)}_{=a} \cos^2 \beta + \underbrace{(2AB)}_{=b} \cos \beta + \underbrace{A^2 - C^2}_{=c} &= 0
 \end{aligned}$$

Tomando ahora las constantes  $a = B^2 + C^2$ ,  $b = 2AB$  y  $c = A^2 - C^2$ , y haciendo  $\cos \beta = x$ , se obtiene una ecuación de segundo grado, de la cual pueden calcularse dos posibles soluciones para  $x$ .

$$\begin{aligned}
 x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} &\rightarrow \begin{cases} \beta_{11} = \arccos(x_1) \\ \beta_{12} = -\arccos(x_1) \end{cases} \\
 x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} &\rightarrow \begin{cases} \beta_{21} = \arccos(x_2) \\ \beta_{22} = -\arccos(x_2) \end{cases}
 \end{aligned}$$

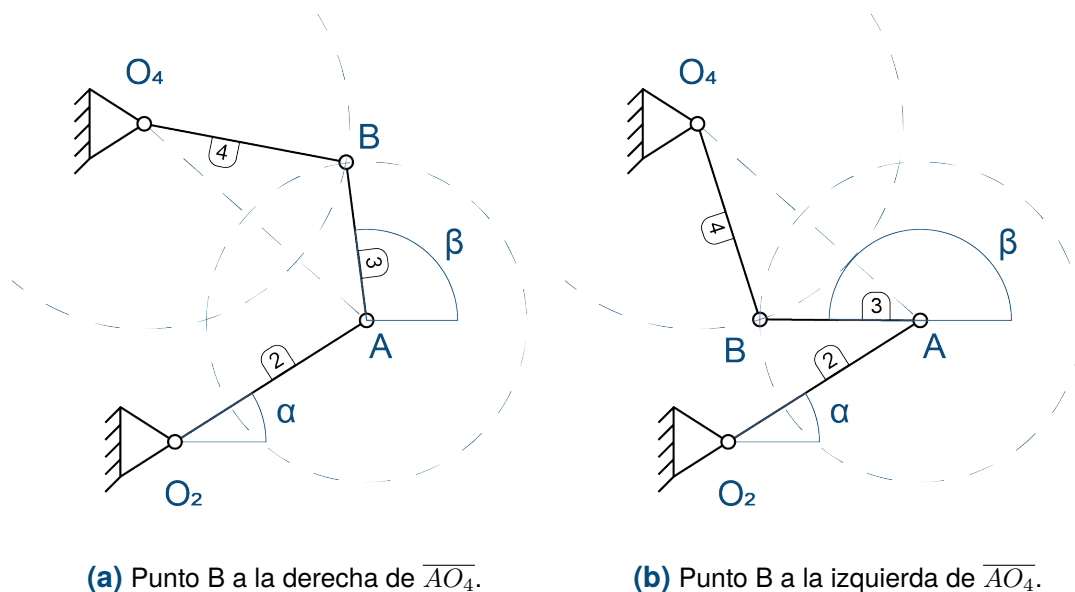
De esta forma obtenemos dos pares de posibles soluciones para el ángulo  $\beta$ . De cada par, solo una de las dos soluciones es válida. Para saber cuál, se calcula la distancia  $\overline{O_4B}$  que resulta de cada uno de los dos posibles valores, como se muestra a continuación:

$$\overline{O_4B}(\beta) = \sqrt{[O_{4x} - (O_{2x} + l_2 \cos \alpha + l_3 \cos \beta)]^2 + [O_{4y} - (O_{2x} + l_2 \cos \alpha + l_3 \cos \beta)]^2}$$

El valor de  $\beta$  válido en cada par de soluciones será aquel que cumpla que  $\overline{O_4B}(\beta) = l_4$ . Esto podemos expresarlo como:

$$\begin{aligned}
 \beta_1 &= \begin{cases} \beta_{11}, & \text{si } \overline{O_4B}(\beta_{11}) = l_4 \\ \beta_{12}, & \text{si } \overline{O_4B}(\beta_{12}) = l_4 \end{cases} \\
 \beta_2 &= \begin{cases} \beta_{21}, & \text{si } \overline{O_4B}(\beta_{21}) = l_4 \\ \beta_{22}, & \text{si } \overline{O_4B}(\beta_{22}) = l_4 \end{cases}
 \end{aligned}$$

Una vez obtenido el ángulo  $\beta$  en función del ángulo de la barra de entrada ( $\alpha$ ), ya puede calcularse la posición de cualquier punto del mecanismo. Sin embargo, se han obtenido dos posibles valores de  $\beta$ . Esto es debido a que el problema de posición de un mecanismo de cuatro barras tiene dos soluciones, como puede apreciarse en la Figura 13. El mayor inconveniente es que los valores  $\beta_1$  y  $\beta_2$  no se corresponden con las Soluciones A y B – para un valor de  $\alpha$  el ángulo  $\beta_1$  pue-



**Figura 12.** Diferentes soluciones al problema de posición en el mecanismo de cuatro barras.

de ser el correspondiente a la Solución B, pero para otro  $\alpha$  cambiar y serlo de la Solución A. Para que el programa funcione de manera consistente es necesario, por consiguiente, especificar en un principio cuál es la solución que queremos obtener, y devolver el valor  $\beta_1$  o  $\beta_2$  según corresponda en cada caso, para cualquier valor de  $\alpha$ .

Para diferenciar las dos soluciones se empleará el producto vectorial  $\vec{AB} \times \vec{BO_4}$ , ya que éste será positivo cuando la solución sea tipo A y negativo en el caso contrario. Si se especifica que la solución al problema de posiciones debe ser de tipo A, entonces se tomará como valor de  $\beta$  de entre  $\beta_1$  y  $\beta_2$  aquel con el cual el producto vectorial  $\vec{AB} \times \vec{BO_4}$  sea positivo. En caso de que se busque una solución de tipo B, lo contrario.

Con esto quedaría resuelto el problema de posición. El siguiente paso es resolver el **problema de velocidades**. Ya que la barra 2 es la de entrada, se tomará su velocidad angular  $\omega_2$  como dato de partida. El objetivo es obtener los valores de  $\omega_3$  y  $\omega_4$  en función de la misma. Recorriendo el lazo  $O_2$ -A-B- $O_4$ , y aplicando el método de las velocidades relativas, obtenemos las ecuaciones:

$$\begin{aligned}\vec{v}_{21}^A &= \underbrace{\vec{v}_{21}^{O_2}}_{=\vec{0}} + \vec{\omega}_{21} \times \overline{O_2A} \\ \vec{v}_{31}^A &= \vec{v}_{21}^A \\ \vec{v}_{31}^B &= \vec{v}_{31}^A + \vec{\omega}_{31} \times \overline{AB} \\ \vec{v}_{41}^B &= \vec{v}_{31}^B\end{aligned}$$

$$\underbrace{\vec{v}_{41}^{O_4}}_{=\vec{0}} = \vec{v}_{41}^B + \vec{\omega}_{41} \times \overrightarrow{BO_4}$$

Tomando la última de las expresiones y sustituyendo con las anteriores, obtenemos la ecuación vectorial:

$$\vec{\omega}_{21} \times \overrightarrow{O_2A} + \vec{\omega}_{31} \times \overrightarrow{AB} + \vec{\omega}_{41} \times \overrightarrow{BO_4} = \vec{0}$$

$$\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & \omega_{21} \\ O_2A_x & O_2A_y & 0 \end{vmatrix} + \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & \omega_{31} \\ AB_x & AB_y & 0 \end{vmatrix} + \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & \omega_{41} \\ BO_{4x} & BO_{4y} & 0 \end{vmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} -l_2 \sin \alpha \\ l_2 \cos \alpha \end{pmatrix} \omega_{21} + \begin{pmatrix} -l_3 \sin \beta \\ l_3 \cos \beta \end{pmatrix} \omega_{31} + \begin{pmatrix} -l_4 \sin \gamma \\ l_4 \cos \gamma \end{pmatrix} \omega_{41} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Multiplicando la segunda ecuación por  $\tan \gamma$  y restándosela a la primera:

$$\begin{cases} l_4 \sin \gamma \omega_{41} = -l_2 \sin \alpha \omega_{21} - l_3 \sin \beta \omega_{31} \\ l_4 \sin \gamma \omega_{41} = -l_2 \cos \alpha \tan \gamma \omega_{21} - l_3 \cos \beta \tan \gamma \omega_{31} \end{cases}$$

$$l_2 (\cos \alpha \tan \gamma - \sin \alpha) \omega_{21} + l_3 (\cos \beta \tan \gamma - \sin \beta) \omega_{31} = 0$$

De la que es posible despejar  $\omega_{31}$  en función de  $\omega_{21}$ .

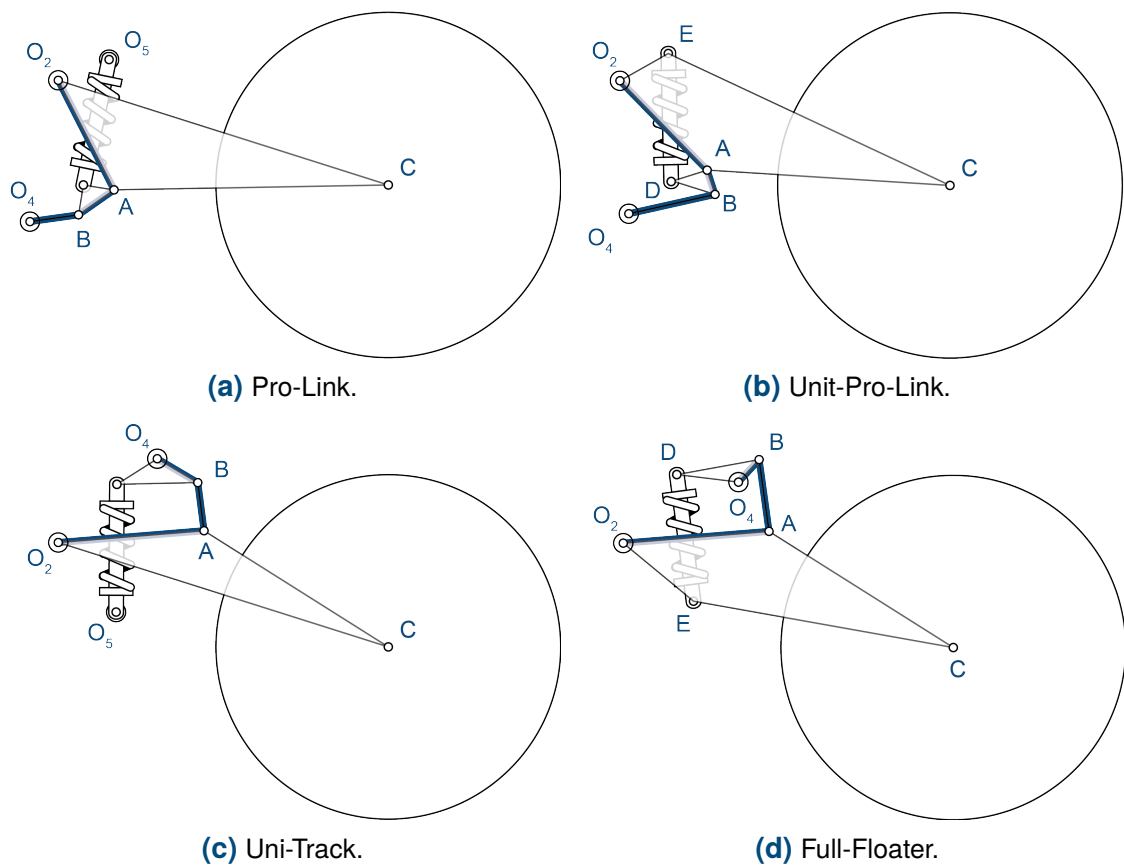
$$\omega_{31} = \frac{l_2 (\sin \alpha - \cos \alpha \tan \gamma)}{l_3 (\cos \beta \tan \gamma - \sin \beta)} \omega_{21} = C_{31} \omega_{21} \quad (5)$$

Realizando un procedimiento similar se obtiene también el valor de  $\omega_{41}$  en función de  $\omega_{21}$ , quedando resuelto el problema de velocidades del mecanismo.

$$\omega_{41} = \frac{l_2 (\sin \alpha - \cos \alpha \tan \beta)}{l_4 (\cos \gamma \tan \beta - \gamma)} \omega_{21} = C_{41} \omega_{21} \quad (6)$$

La relación que existe entre el mecanismo de cuatro barras y los distintos tipos de sistemas de suspensión de bieletas se muestra en la Figura 13. A parte del chasis y el basculante, estos sistemas de suspensión cuentan con dos piezas principales más, que son las que completan el mecanismo de cuatro barras. Suele conocerse como *balancín* a la pieza que tiene tres pares de rotación en el mecanismo. La *bieleta* es la que solamente tiene dos.

Para este tipo de sistemas de suspensión, la variable interna de cálculo ( $b$ ) se ha tomado como el ángulo de la barra de entrada del mecanismo de cuatro barras. En base a ella se obtendrá el valor del resto de variables.



**Figura 13.** Esquema de los diferentes sistemas de suspensión de bieletas y su relación con el mecanismo de cuatro barras.



**Ángulo del basculante:** El ángulo del basculante no coincide en este caso con la variable  $b$ .

$$B(b) = b - b_i \quad (7)$$

Siendo  $b_i$  el ángulo interno del basculante, que es el que hay entre los segmentos  $\overline{O_2C}$  y  $\overline{O_2A}$ . Su valor se obtiene de  $\arctan \frac{s_y}{s_x}$  actualizando, eso sí, el resultado al cuadrante que corresponda. Esto se aprecia mejor en el Bloque de código 3.

```

1 public double getAngle (double dx, double dy) {
2     double angle;
3     angle = dx != 0? Math.atan(dy/dx): Math.PI/2;
4     angle += dx < 0? Math.PI: 0;
5     return angle;
6 }
```

**Bloque de código 3.** Función para calcular el ángulo de un triángulo a partir de sus catetos.

**Desplazamiento vertical de la rueda trasera:** Su valor se obtiene de la misma forma que en los sistemas de suspensión tipo clásico y cantilever. Lo único que hay que tener en cuenta que ahora que el ángulo del basculante se obtiene de la expresión anterior, y no coincide directamente con la variable básica  $b$ .

$$H(b) = O_{2y} + s \cos(B) - R \quad (8)$$

**Compresión de la unidad de suspensión:** La ecuación 3 sigue siendo perfectamente válida en este caso. Sin embargo, el valor de  $l_{SU}(b)$  ha de calcularse específicamente para cada sistema de suspensión. En todo caso, es la distancia entre los dos anclajes de la unidad de suspensión. En los sistemas tipo Pro-Link y Uni-Track, un extremo es fijo y el otro se ancla al balancín. En los Unit-Pro-Link y Full-Floater, ambos extremos son móviles. Sin embargo, las posiciones de todos estos puntos pueden obtenerse en función de  $b$  según el análisis que se ha realizado del mecanismo de cuatro barras.

En el sistema Pro-Link, por ejemplo,  $B_x(b) = \sqrt{s_x^2 + s_y^2} \cos b + r \cos(\beta(b))$  y  $B_y(b) = \sqrt{s_x^2 + s_y^2} \sin b + r \sin(\beta(b))$ . La longitud del amortiguador queda, por tanto, como  $l_{SU}(b) = \sqrt{(O_{5x} - B_x(b))^2 + (O_{5y} - B_y(b))^2}$ .

**Fuerza vertical en la rueda trasera:** Dado que solo es necesario obtener el valor de la fuerza  $F$  aplicada sobre la rueda trasera, y no todas las acciones que se generan entre cada barra del mecanismo, se utilizará el Principio de las Potencias Virtuales. Como el análisis es en condiciones estáticas, no actúan momentos, y además no se considera la masa de las barras ni su inercia, la expresión del

Principio de las Potencias Virtuales se simplifica a:

$$\sum_{j=1}^{n_S} \left( \sum_{k=1}^{n_f^j} \vec{F}_k \cdot \vec{v}_{j1}^{P_k^*} \right) = 0$$

Tomando como ejemplo, de nuevo, el sistema tipo Pro-Link, la expresión anterior quedaría como:

$$\vec{F} \cdot \vec{v}_{21}^{C^*} + \vec{F}_e \cdot \vec{v}_{41}^{D^*} = 0$$

$$\vec{F} \cdot (\vec{\omega}_{21}^* \times \overrightarrow{O_2C}) + \vec{F}_e \cdot (\vec{\omega}_{21}^* \times \overrightarrow{O_2A} + \vec{\omega}_{31}^* \times \overrightarrow{AD}) = 0$$

$$\vec{F} \cdot \begin{pmatrix} -O_2C_y \\ O_2C_x \end{pmatrix} \vec{\omega}_{21}^* + \vec{F}_e \left[ \begin{pmatrix} -O_2A_y \\ O_2A_x \end{pmatrix} \vec{\omega}_{21}^* + \begin{pmatrix} -AD_y \\ AD_x \end{pmatrix} \vec{\omega}_{21}^* \right] = 0$$

$$(-O_2C_x \omega_{21}) F + \left( -O_2A_x - AD_y \underbrace{\omega_{31}}_{=C_{31}\omega_{21}} \right) F_{ex} + (O_2A_x \omega_{21} + AD_x \omega_{31}) F_{ey} = 0$$

Donde se ha tenido en cuenta que  $\vec{F}$  es una fuerza en la dirección positiva de  $\vec{j}$  (hacia arriba), y la ecuación 5. El valor de la fuerza  $F$  se obtiene finalmente según:

$$F = \frac{(O_2A_y + AD_y) F_{ex} - (O_2A_x + AD_x C_{31}) F_{ey}}{O_2C_x} \quad (9)$$

## 5.2 Función de aptitud

El objetivo de la optimización es conseguir un sistema de suspensión con un comportamiento determinado. Como ya hemos visto, el comportamiento va a especificarse mediante una serie de curvas. El usuario indicará al programa el tipo de curva objetivo y una serie de puntos de control para establecer su forma. Cualquier curva que relacione dos de las cinco variables de cálculo puede ser tomada para realizar la optimización. Sin embargo, lo normal es establecer el comportamiento deseado tomando como variable independiente el desplazamiento de la rueda trasera, o el ángulo del basculante; y como variable dependiente la fuerza vertical en la rueda trasera o la rigidez equivalente del sistema. Los datos indicados por el usuario se guardan en una instancia de la clase `TargetCurveDefinition`.

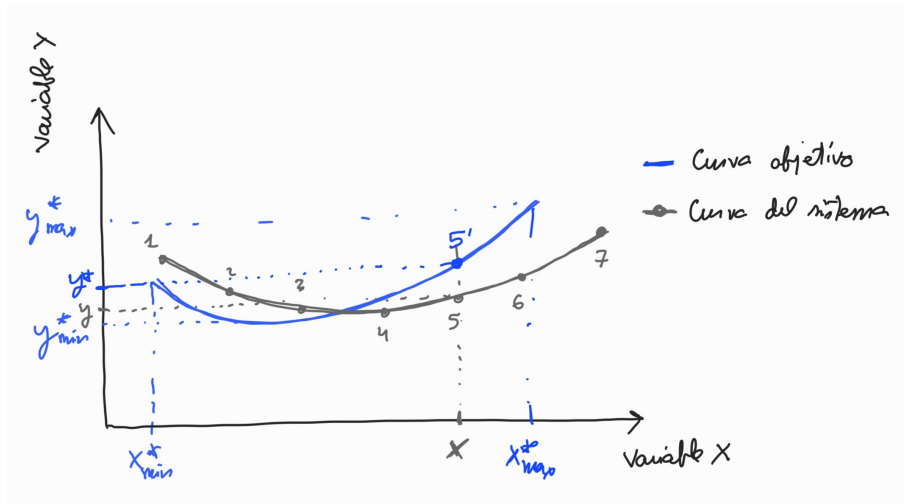
Parámetro	Descripción
$O_{2x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{4x}$	Coordenada x del punto de anclaje la bieleta al chasis.
$O_{4y}$	Coordenada y del punto de anclaje la bieleta al chasis.
$O_{5x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{5y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
$r$	Longitud del balancín.
$r_x$	Coordenada x, en ejes locales del balancín, del punto de anclaje con el basculante.
$r_y$	Coordenada y, en ejes locales del balancín, del punto de anclaje con el basculante.
$l$	Longitud de la bieleta.
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
$R$	Radio de la rueda trasera.

**Tabla 3.** Nombre y descripción de los parámetros necesarios para definir un sistema de suspensión tipo Pro-Link.

Para calcular la adecuación de un sistema de suspensión al comportamiento especificado, valdrá con medir la diferencia que existe entre la curva objetivo y la curva del sistema que está siendo evaluado. Para ello, se tomarán una serie de puntos repartidos por todo el rango de movimiento del sistema. Si el punto considerado está dentro del rango en la variable  $x$  de la curva objetivo, se sumará al valor de adecuación (*fitness*) una cantidad que dependerá de la distancia entre la curva del sistema y la objetivo en la variable  $y$ . Para calcular el valor de adecuación de uno de estos puntos, especificado por un valor  $b$  de su variable básica, se ha tomado la siguiente expresión:

$$f(b) = \begin{cases} 0, & \text{si } x < x_m^* \\ 2^{-\frac{|y-y^*|}{y_x^*-y_m^*}}, & \text{si } x_m^* \geq x \geq x_x^* \\ 0, & \text{si } x > x_x^* \end{cases} \quad (10)$$

Donde  $x$  e  $y$  son las coordenadas del punto en la gráfica del sistema, que vienen dadas por el valor de la variable  $b$ .  $y^*$  es el valor de la coordenada  $y$  de la curva objetivo cuando la variable independiente vale  $x$ . Los subíndices  $m$  y  $x$  denotan el valor mínimo y máximo de las coordenadas  $x^*$  e  $y^*$  de la curva objetivo, respectivamente. Todas estas variables se muestran en la Figura 14 para el



**Figura 14.** Cálculo del valor de adecuación correspondiente al punto 5 de la gráfica del sistema.

cálculo del valor de adecuación correspondiente al punto 5.

El valor de adecuación total del sistema podría calcularse directamente sumando el de todos los puntos considerados. Sin embargo, esto favorecería soluciones en las que la curva del sistema quede dentro de la curva objetivo, ya que los puntos para el cálculo quedaría más cercanos entre sí, como puede apreciarse en la Solución 3 de la Figura 15. Interesa más bien lo contrario, que la curva del sistema abarque por completo el dominio de la curva objetivo. Por ello, la suma del valor de adecuación en cada punto se multiplicará por un coeficiente  $c$  cuyo valor depende únicamente de los límites en  $x$  de las curvas del sistema y objetivo, y se calcula como:

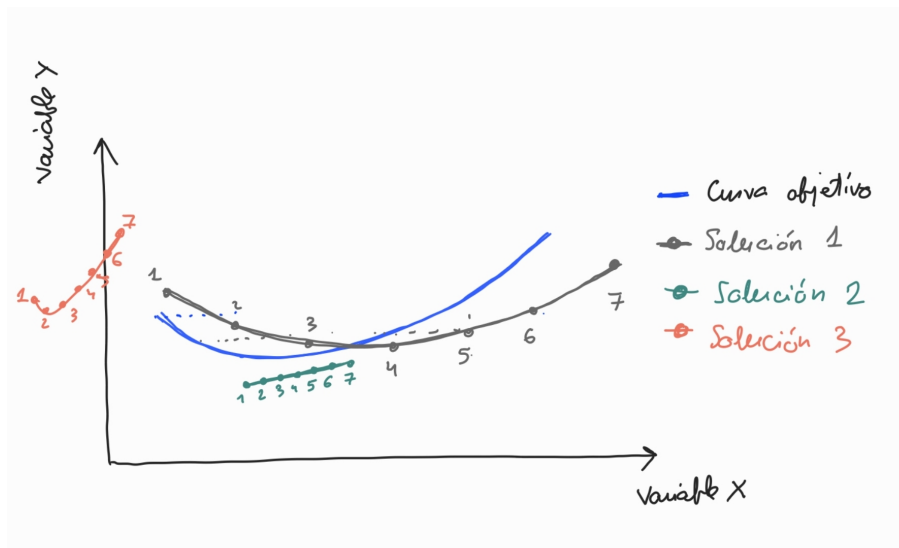
$$c = \frac{\max(x_x, x_x^*) - \min(x_m, x_m^*)}{x_x^* - x_m^*} \quad (11)$$

Este coeficiente valdrá 1 si la curva del sistema abarca por completo el dominio de la curva objetivo. Si no, su valor corresponderá al tanto por 1 en que el dominio de la curva objetivo es cubierto por el de la curva del sistema.

Teniendo en cuenta las ecuaciones 10 y 11, el cálculo del valor de adecuación total para un sistema de suspensión será:

$$\text{fitness} = \left( \sum_{i=0}^n f(b_i) \right) \times c \quad (12)$$

Donde  $n$  es el número de puntos tomados para comparar ambas curvas, y  $b_i$  cada uno de los  $i$  valores de la variable  $b$  tomados dentro del rango de movimiento



**Figura 15.** Algunas casuísticas que podemos encontrarnos a la hora de comparar la gráfica objetivo con las de un sistema de suspensión. Los puntos de la Solución 2 sumarán un valor muy alto, pero la curva es mala por que apenas abarca el dominio de la curva objetivo. La Solución 3 está completamente fuera del dominio de la curva objetivo; su valor de adecuación será 0.

del sistema de suspensión, especificado según las funciones `getLowerLimit()` y `getUpperLimit()`.

### 5.3 Población

Se crea una clase denominada `Solution` que alberga información de cada individuo dentro de la población – el sistema de suspensión al que se refiere y el valor del desempeño que se ha calculado, entre otros.

Para poder consultar datos útiles por el usuario, el programa guarda no solo la información de los individuos de la generación actual, sino también la de algunas generaciones atrás. Todos estos datos se almacenan en la variable `ArrayList<Solution>[] population`, que es un vector de listas tipo `ArrayList<Solution>`. Cada elemento del vector corresponde a una generación. Así, `population[0]` almacena toda la información referente a la generación actual; `population[1]` la correspondiente a la generación anterior; y así sucesivamente hasta acabar con el tamaño del vector.

Se ha optado por dejar que el usuario decida la cantidad de individuos que conforman la población. Podrá establecer dicho tamaño ya sea antes de que comience el proceso de optimización como durante el mismo. En el segundo caso, los cambios afectarán únicamente a las generaciones que se obtienen tras el mismo.

## 5.4 Mecanismo de selección de progenitores

De entre todos los individuos que conforman la población en una generación dada, se escogen solo algunos de ellos para dar lugar a los nuevos individuos de la siguiente generación.

Los individuos se escogerán con una probabilidad proporcional a su valor de adecuación. Para lograrlo, se toma una variable que almacene la suma de los valores de adecuación de todas las soluciones que componen la generación actual. Se escoge un número aleatorio entre 0 y el resultado de la suma anterior. Por último, se recorre la población sumando los valores de adecuación de nuevo. El bucle se detiene cuando se encuentra al individuo que hace que la suma actual del valor de adecuación sobrepase el número aleatorio obtenido. El código correspondiente puede apreciarse en el Bloque de código 4.

```

1 private int selectParent () {
2     int parentIndex = 0;
3     double sumFitness = 0;
4
5     for (Solution s: population[lastGeneration - currentGeneration
6         + 1])
7         sumFitness += s.getFitness();
8
9     double r = sumFitness*Math.random();
10    sumFitness = 0;
11
12    for(parentIndex = 0; parentIndex < population[lastGeneration -
13        currentGeneration + 1].size() - 1; parentIndex++) {
14        sumFitness += population[lastGeneration - currentGeneration
15            + 1].get(parentIndex).getFitness();
16        if (r <= sumFitness)
17            break;
18    }
19    return parentIndex;
20 }

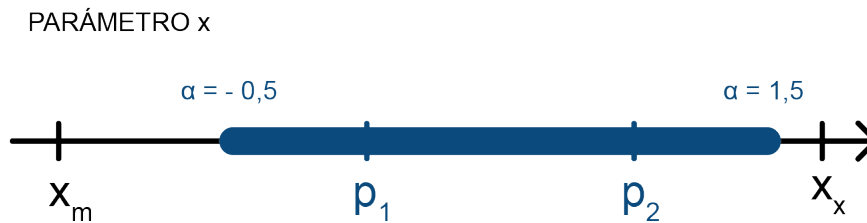
```

**Bloque de código 4.** Función que devuelve la posición del individuo a utilizar como padre en la generación de uno nuevo.

## 5.5 Mecanismo de generación de la descendencia

Para definir cada individuo de la nueva generación, se tomarán dos progenitores de la generación anterior, siguiendo el procedimiento especificado en el apartado anterior. Posteriormente, el valor de todos los parámetros del nuevo individuo se tomarán en base al de los padres, según un coeficiente  $\alpha$  cuyo valor varía aleatoriamente entre  $\alpha = -0,5$  y  $\alpha = 1,5$ , tal y como se muestra en el Bloque de código 5.

Si para un parámetro  $\alpha = 0$ , el valor del parámetro del hijo corresponderá con el del primero de los progenitores. Si  $\alpha = 1$  ocurrirá lo mismo pero para el segundo



**Figura 16.** Esquema de funcionamiento de la reproducción tipo *blend crossover* para un parámetro real cualquiera.  $p_1$  y  $p_2$  es el valor del parámetro de los padres;  $x_m$  y  $x_x$  los límites inferior y superior, respectivamente, del parámetro. La zona azul son los posibles valores del parámetro que pueden heredar el hijo

de los progenitores. Para valores intermedios de  $\alpha$ , el valor de los parámetros se estará tomando entre el de ambos padres de forma aleatoria. Para valores de  $\alpha \geq 1$  y  $\alpha \leq 0$ , el valor del parámetro para la solución hijo quedará fuera del rango que marcan los padres. De esta forma, la capacidad de explorar el espacio de soluciones es mayor. Esto es lo que se conoce como *Blend crossover* (Smith y col., 2015). En la Figura 16 se representa un esquema de cómo funciona este procedimiento para un solo parámetro. Si, para un parámetro, el valor resultante quedase fuera de los límites del mismo, se tomaría el valor del límite más cercano.

Para añadir mayor variabilidad a las soluciones, el valor de todos los parámetros es susceptible de ser modificado a un valor aleatorio dentro de los límites del mismo. Este procedimiento, correspondiente al operador de mutación, se realiza con una probabilidad determinada por el parámetro `mutationRate`, cuyo valor podrá ser modificado por el usuario de la aplicación.

```

1 public Solution (double mutationRate, SuspensionSystem parent1,
2   SuspensionSystem parent2) {
3   this.suspensionSystem = parent1.getCopy();
4   for (int pIndex = 0; pIndex < parent1.getLastLevelParameters().
5     size(); pIndex++) {
6     if (parent1.getLastLevelParameters().get(pIndex).
7       isToOptimize()) {
8       //Blend crossover
9       double r = 2.0*Math.random() - 0.5;
10      double newValue = parent1.getLastLevelParameters().get(
11        pIndex).getValue() + (parent2.getLastLevelParameters().get(
12        pIndex).getValue() - parent1.getLastLevelParameters().get(pIndex)
13        ).getValue()*r;
14
15      this.suspensionSystem.getLastLevelParameters().get(
16        pIndex).setValue(newValue);
17
18      //Mutation
19      r = Math.random();
20      if (r < mutationRate)
21        this.suspensionSystem.getLastLevelParameters().get(
22        pIndex).setValue(this.suspensionSystem.getLastLevelParameters().
23        get(pIndex).getLowerLimit() + Math.random()*(this.

```

```

15     suspensionSystem.getLastLevelParameters().get(pIndex).
16     getUpperLimit() - this.suspensionSystem.getLastLevelParameters()
17     .get(pIndex).getLowerLimit());
    }
}

```

**Bloque de código 5.** Uno de los constructores de la clase `Solution` funciona como mecanismo de generación de la descendencia.

Cabe destacar, por último, que el usuario puede establecer previamente al proceso de optimización si un parámetro debe o no ser optimizado. Si se establece que un parámetro no debe ser optimizado, los operadores de mutación y crossover no se aplicarán sobre el mismo, y todas las soluciones utilizarán siempre el mismo valor para este parámetro.

## 5.6 Mecanismo de selección de supervivientes

En cada generación se generan un número de individuos igual al tamaño de la población establecido. Por ello, no es necesario ningún mecanismo de selección de supervivientes – todos los individuos nuevos creados pasan a formar parte de la población.

## 5.7 Inicialización

La primera generación se crea de forma aleatoria. Para ello, todos los parámetros de cada individuo se establecen como un valor aleatorio de entre los límites inferiores y superiores establecidos por el usuario.

## 5.8 Condición de finalización

Se ha querido que el usuario tenga total control respecto del proceso de optimización a través de la aplicación. Por ello, se puede detener la optimización tantas veces como se desee; analizar los resultados y continuar posteriormente con la misma. Incluso es posible volver a generaciones anteriores. Por ello, la condición de finalización la establece fundamentalmente el usuario del programa.

Sin embargo, cabe destacar que tal y como se definió la función de adecuación en la Sección 5.2, existe un valor máximo que puede alcanzar la misma. Este valor máximo corresponde con el número de puntos de comparación tomados entre la curva del sistema y la curva objetivo. Esto se debe a que, en el caso ideal en que todos los puntos de comparación de la curva del sistema coincidiesen con los de la curva objetivo, cada uno sumaría una unidad al valor de adecuación, ya que  $2 \frac{0}{y_x^* - y_m^*} = 1$ . Esto permite conocer cuán buena es una solución de manera absoluta – no solo comparando entre varias soluciones. Así, el usuario puede



tener esto en cuenta para saber cuándo una solución es suficientemente buena para él y así terminar el proceso de optimización.

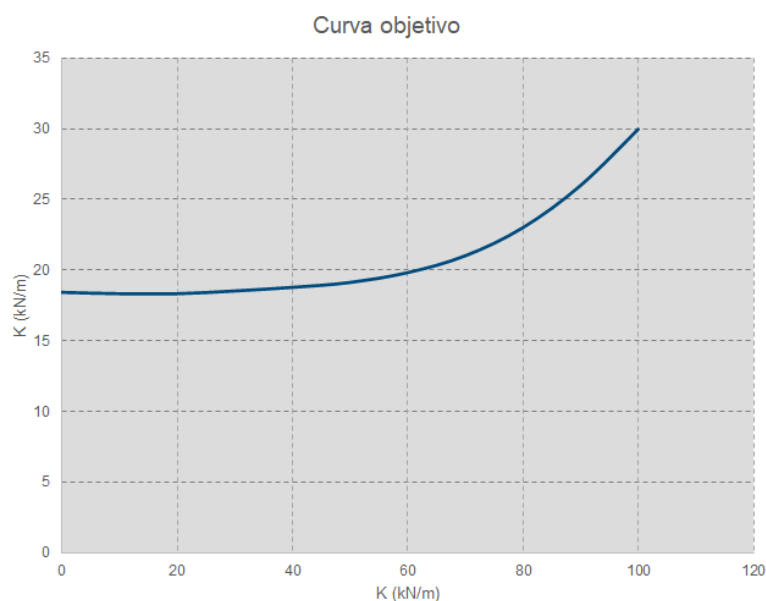
## 6 Análisis de resultados

Las bases sentadas en los apartados anteriores se materializan en el desarrollo de una aplicación informática que asista a los ingenieros durante la fase conceptual del proceso de diseño. En este apartado se describe un breve caso práctico del uso de la aplicación dentro en este contexto. Sin embargo, muchas de las herramientas del programa no se explican aquí. El lector puede consultar el Anexo A, que constituye un pequeño manual para conocer todos los detalles de la aplicación.

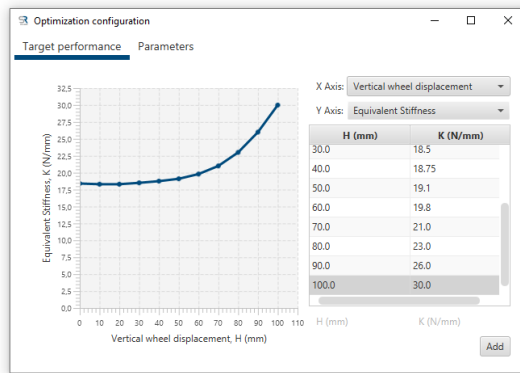
### 6.1 Enunciado del problema

Se está trabajando en el nuevo diseño de una motocicleta de competición. La rueda que va a utilizarse es una OZ Racing que tiene un radio de 300mm. Para el sistema de suspensión trasero, se ha optado por adquirir una unidad de suspensión de características:

- Constante de rigidez: 90 KN/m.
- Longitud extendida: 267 mm.
- Longitud mínima: 210 mm.



**Figura 17.** Curva objetivo de la optimización.



**Figura 18.** Definición de la curva objetivo en el programa.

H (mm)	K (N/mm)
0	18,40
10	18,30
20	18,30
30	18,50
40	18,75
50	19,10
60	19,80
70	21,00
80	23,00
90	26,00
100	30,00

**Tabla 4.** Puntos empleados para definir la curva objetivo en el programa.

Los ingenieros del equipo han establecido, en base a su experiencia, que para que la motocicleta se comporte como ellos desean la curva de Rigidez equivalente de la suspensión vs. Desplazamiento de la rueda trasera debe ser tal y como se muestra en la Figura 17. Se utilizará un sistema de tipo Pro-Link.

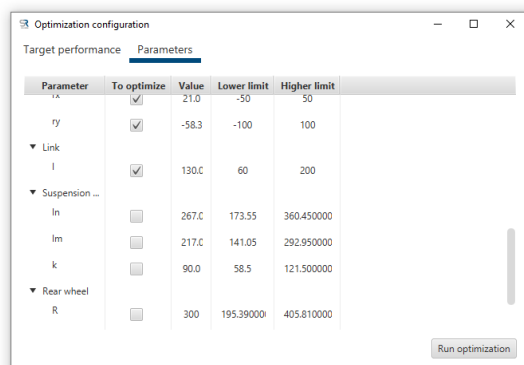
## 6.2 Proceso de optimización

En primer lugar, se abre el programa y se crea un nuevo Sistema de Suspensión de tipo Pro-Link. El programa asigna un valor por defecto para todos los parámetros. Si se cuenta con un modelo de referencia, podemos modificar el valor de los parámetros a los correspondientes a dicho modelo. Esto es útil cuando se cuenta con un sistema de suspensión más o menos definido, a falta de ajustar mínimamente el valor de algunos de los parámetros, ya que los límites superior e inferior de los parámetros a la hora de hacer el análisis se establecerán de acuerdo a al valor de los mismos, de manera porcentual.

En este caso, dado que no contamos con ninguna información preliminar, ignoraremos los parámetros que viene definidos por defecto y ejecutaremos directamente la herramienta de optimización.

Se considerará que 11 puntos es suficiente para definir la curva objetivo del sistema. Estos se obtienen de la gráfica mostrada en la Figura 17, y se muestran en la Tabla 4. En la pestaña de definición de la curva objetivo, se eliminan los puntos que viniesen por defecto y se añaden los correspondientes a la curva objetivo que nos ocupa.

Posteriormente, en la pestaña *Parameters*, establecemos cuáles son los parámetros que quedan fuera de la optimización y cuales sí se tendrán en cuenta.



**Figura 19.** Definición de los parámetros de la optimización.

Parámetro	L. inferior	L. superior
O2y	450,00	650,00
O4x	-150,00	150,00
O4y	200,00	400,00
O5x	-150,00	150,00
O5y	350,00	550,00
s	450,00	600,00
sx	-150	200,00
sy	-150,00	150,00
r	20,00	60,00
rx	-30,00	40,00
ry	-30,00	40,00
l	-60,00	200,00

**Tabla 5.** Límites superior e inferior de los parámetros a optimizar.

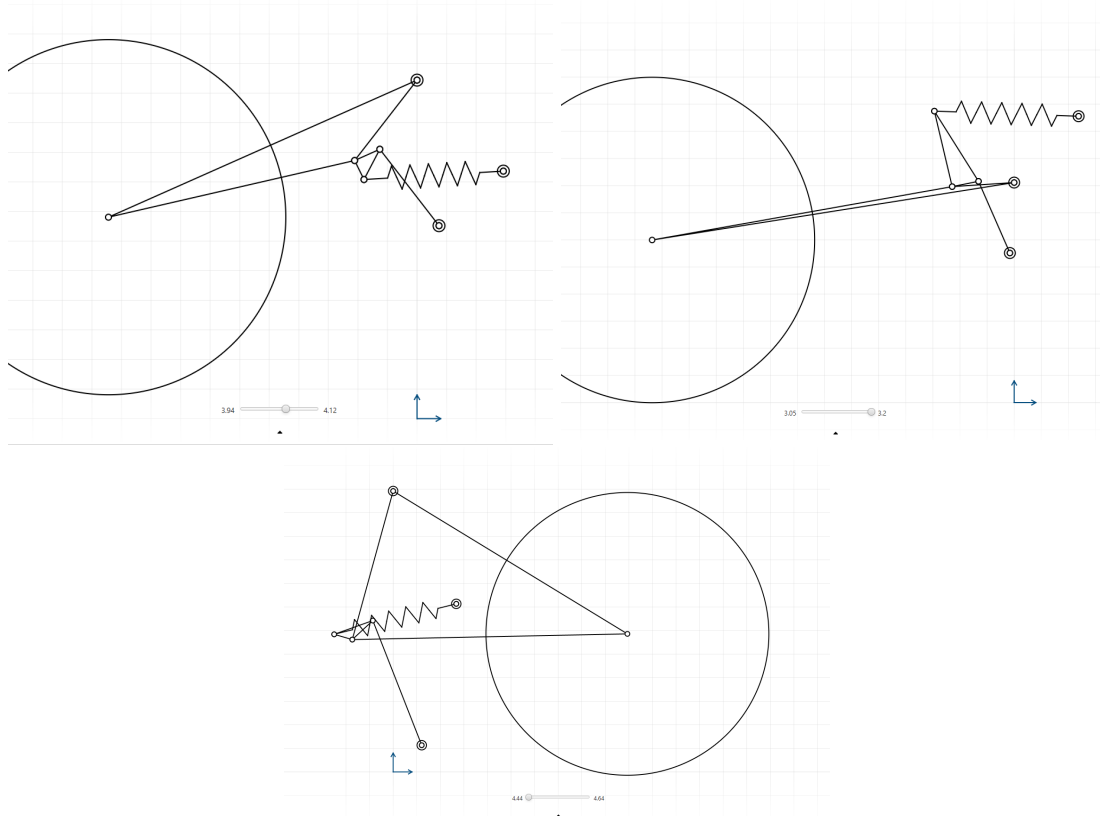
Para los parámetros que no van a optimizarse, debemos de asegurar que el valor del mismo es el deseado – en este caso, los parámetros de la suspensión y el radio de la rueda. Desmarcaremos la casilla de la columna *To optimize*. Para el resto de parámetros, es necesario establecer los límites del mismo. Podemos dejar los valores que se calculan por defecto, o ampliar/reducir el espacio de búsqueda modificando los límites del sistema. Para este ejemplo, se ha ampliado considerablemente el espacio de búsqueda para abarcar una mayor casuística en las soluciones. Los límites establecidos se muestran en la Tabla 5, y la interfaz del programa con los datos introducidos en la Figura 19.

Por último, se ejecuta la optimización pulando el botón *Run optimization*, y se inicia el proceso considerándose una población de 100 individuos y un ratio de mutación del 2%.

### 6.3 Resultados obtenidos

Los primeros resultados no fueron del todo satisfactorios. La curva del sistema se aproximaba bastante bien a la curva establecida como objetivo. Sin embargo, los resultados obtenidos eran poco prácticos por varios motivos:

- Algunas de los procesos de optimización llevados a cabo convergieron en soluciones en las que el basculante quedaba hacia el lado izquierdo. Esto es perfectamente válido en cuanto a la formulación y los cálculos realizados. Sin embargo, hace que los límites establecidos para los parámetros no concuerden con los de los resultados, ya que el sistema de referencia sobre el que se basan asume que el basculante queda en el lado positivo del Eje X.



**Figura 20.** Algunos resultados de la optimización, considerados como no válidos.

- Otras soluciones obtenidas tenían el problema de que, aunque la curva de rigidez equivalente se aproximase bastante bien, en la práctica los resultados no tenían sentido. Por ejemplo, en algunos casos la variable de la fuerza  $F$  era en todo el rango de movimiento del sistema negativa; es decir, que para comprimir el amortiguador en un sistema de suspensión de este tipo habría que desplazar la rueda hacia abajo, y no al contrario.
- Por último, destacar un problema generalizado con la convergencia. Después de pocas generaciones, casi todos los individuos de la población adquieren una topología parecida, de forma que la optimización sigue ocurriendo pero en menor medida. Claramente el algoritmo está convergiendo en máximos locales. Esto es evidente también por el hecho de que, después de realizar varias optimizaciones bajo las mismas condiciones los resultados sean siempre tan dispares.

Para hacer frente a estos problemas se realizan los siguientes cambios:

- Se modifica el mecanismo de selección de los progenitores para que no haya tanta diferencia entre la probabilidad de escoger a una solución buena y a una mala. Para hacerlo, durante el cálculo de `sumFitness`, se añade no

solamente el valor de adecuación del individuo, sino el propio valor promedio del mismo. De esta forma, incluso aunque el valor de adecuación de un individuo sea nulo hay cierta probabilidad de que sea escogido.

- Se modifica el proceso de generación de la descendencia. Ahora, según una probabilidad dada por la variable `mutationRate`, el algoritmo ignora el `crossover` y genera una solución completamente nueva, estableciendo sus parámetros de forma aleatoria.
- Se añade un coeficiente al cálculo del valor de adecuación de un individuo, que penaliza el mismo si en el límite inferior del mecanismo la fuerza sobre la rueda trasera se aleja de 0 N. De esta forma estamos buscando soluciones en las que, en su límite inferior del recorrido útil, el amortiguador esté completamente extendido.

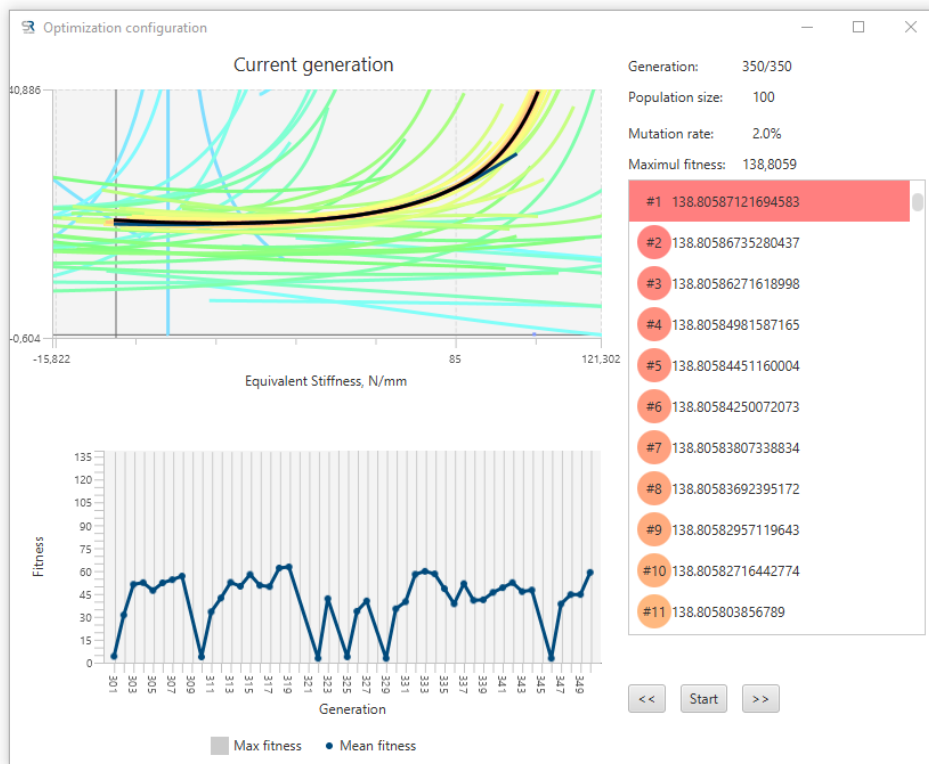
Se lleva a cabo la optimización con los mismos parámetros que en el caso anterior. Tras 350 generaciones, el algoritmo ha convergido en una solución con un valor de adecuación de 138,8, tal y como se muestra en la Figura 21. Dado que se han utilizado 150 puntos de control para la comparación entre la gráfica objetivo y la gráfica del sistema, se observa que la solución obtenida tiene un comportamiento muy cercano al establecido como objetivo (valor máximo de 150,00). Se aprecia que los resultados son mucho más satisfactorios tras los cambios realizados, aunque sigue estando presente el problema de la convergencia prematura.

## 7 Discusión de los resultados

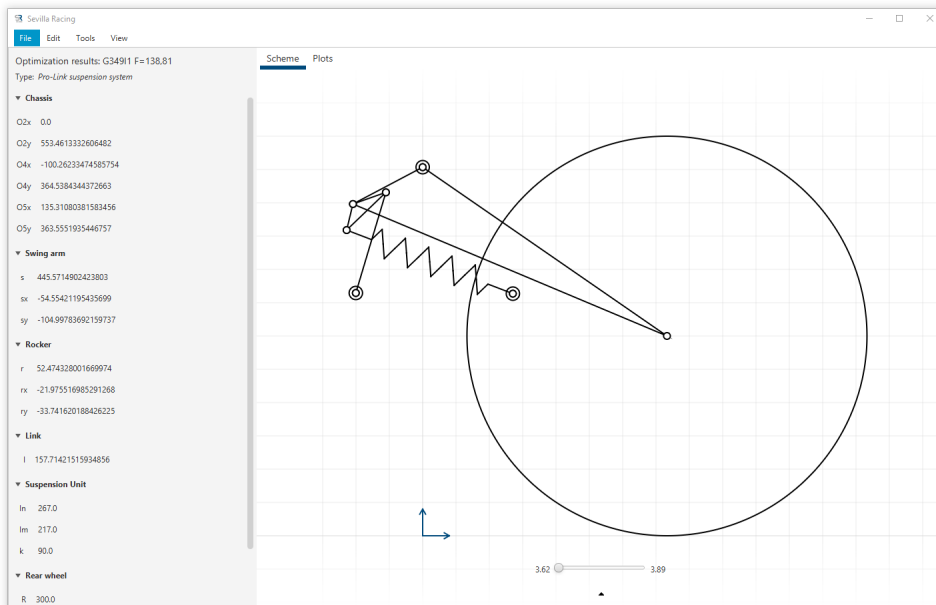
Resumiendo lo que ya se comentó en la Sección 2 de este texto, el proceso de diseño de un sistema de suspensión presenta los siguientes inconvenientes:

- Se requiere experiencia previa.
- Proceso largo y tedioso.
- Necesario tomar decisiones sobre el tipo de sistema de suspensión con poca información.

A continuación se discuten estos inconvenientes en relación al método desarrollado, con el propósito de evaluar en qué medida se han cumplido los objetivos propuestos al inicio de este proyecto.



(a) Proceso de optimización hasta la generación 350.



(b) Esquema de la solución obtenida y valor de sus parámetros.

Figura 21. Resultados del proceso de optimización

## 7.1 Experiencia previa

En el contexto que nos ocupa, la experiencia previa se hace especialmente relevante en dos vertientes diferentes.

1. Determinar las curvas de comportamiento ideales de acuerdo con las características del problema de diseño que se plantea. Esto es, traducir los objetivos de diseño a curvas de comportamiento que los garanticen.
2. Saber de qué manera la modificación de los parámetros del sistema afecta al comportamiento del mismo.

El primero de los casos queda completamente fuera del ámbito de actuación de este trabajo, ya que parte de que las curvas de comportamiento objetivo son un dato. Sin embargo, el método/aplicación desarrollado si ayuda en gran medida en la segunda vertiente. Dado que los parámetros del sistema se ajustan de forma automática, no es determinante tener la experiencia y pericia necesarias para predecir como un cambio determinado va a afectar a las curvas de comportamiento del sistema.

## 7.2 Proceso largo y tedioso

El procedimiento habitual que los ingenieros de Sevilla Racing llevaban a cabo consistía en modificar manualmente los parámetros del sistema, tratando de predecir cómo estos cambios iban a afectar, hasta dar con unas curvas de comportamiento aceptables. Normalmente este procedimiento involucraba a 3 o 4 ingenieros durante al menos 4 o 5 días. Más aún si, por motivos ajenos al sistema de suspensión, era necesario modificar la geometría y volver a optimizar el mismo

El procedimiento que aquí se plantea solo requiere por parte del ingeniero del establecimiento de la curva objetivo y el tipo de sistema de suspensión. La aplicación realiza de forma automática el proceso de optimización, y tarda apenas unos minutos en converger a una solución que se aproxima bastante a los resultados óptimos. En consecuencia, es justo afirmar que el proceso largo y tedioso se minimiza en gran medida.

## 7.3 Toma de decisiones con escasa información

El método desarrollado sigue teniendo como premisa el escoger un tipo de sistema de suspensión previo a la optimización. Esta decisión previa sigue estando presente, con sus correspondiente inconvenientes. Sin embargo, cabe destacar que, dado que el método planteado es mucho más rápido que el manual, es viable

plantear en un primer momento varios tipos de sistemas de suspensión, realizar la optimización y considerar finalmente el que mejores resultados arroje.

## 8 Conclusiones

Sin duda alguna los resultados del proyecto son muy positivos. El método propuesto y la aplicación informática suponen una ayuda muy interesante para el ingeniero encargado del diseño de un sistema de suspensión trasero de una motocicleta. Definitivamente, el uso de Algoritmos Genéticos está justificado en una aplicación como la que aquí se presenta.

Sin embargo, hay que destacar que el proyecto aún necesita pequeñas mejoras. El principal problema es el de la convergencia prematura de las soluciones planteadas por el algoritmo. Hace falta atinar mejor con el valor de los parámetros de la optimización para garantizar que esto no ocurra. También cabría discutir sobre si la implementación de los componentes de un algoritmo genético que se ha realizado es la más adecuada. Son muchos los métodos y las variantes que entran en juego cuando hablamos sobre algoritmos genéticos.

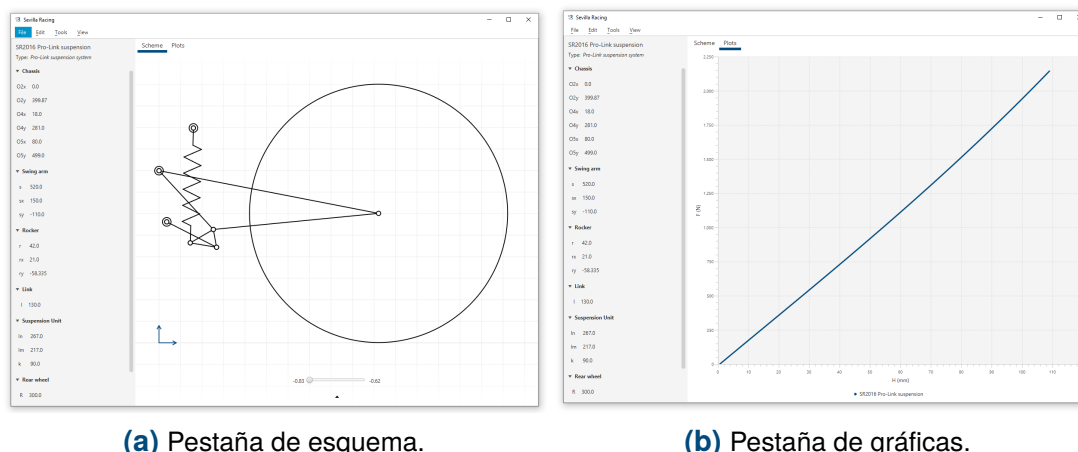
En cualquier caso, sirva este proyecto como ejemplo de lo potentes que pueden llegar a ser estos algoritmos. Una implementación sencilla de cada uno de sus componentes y en muy poco tiempo se obtienen soluciones suficientemente válidas ante el problema planteado.



## Referencias

- ARIAS-PAZ, M, 2002. *Motocicletas*. 32.<sup>a</sup> ed. CIE DOSSAT 2000. ISBN 84-9512-07-7.
- CASTILLO, J. J.; GINER, P.; SIMÓN, A. y CABRERA, J. A., 2013. Optimal Design of Motorcycle Rear Suspension Systems Using Genetic Algorithms. Disp. desde doi: 10.1007/978-94-007-4902-3.
- CORBERA CARABALLO, Sergio y FERNANDEZ ALVAREZ, Roberto, 2020. A performance-based design framework for enhancing decision-making at the conceptual phase of a motorcycle rear suspension development. *Optimization and Engineering*. Vol. 21, págs. 1283-1317. ISBN 0123456789. Disp. desde doi: 10.1007/s11081-019-09475-w.
- COSSALTER, Vittore, 2006. *Motorcycle dynamics*. 2.<sup>a</sup> ed. Lulu. ISBN 978-1-4303-0861-4.
- DOMÍNGUEZ ABASCAL, Jaime, 2016. *Teoría de máquinas y mecanismos*. 1.<sup>a</sup> ed. Sevilla: Universidad de Sevilla. ISBN 978-84-472-1856-1.
- FOALE, Tony, 2002. *Motorcycle Handling and Chassis Design*. Tony Foale Designs. ISBN 84-933286-1-8.
- SMITH, J. E. y EIBEN, A. E., 2015. *Introduction to Evolutionary Computing*. 2.<sup>a</sup> ed. Springer. ISBN 9783662448731. **urlalso**: papers2 : // publication / uuid / F189C4FB-EB51-43FE-B6FC-17BAF1BF36C8.





(a) Pestaña de esquema.

(b) Pestaña de gráficas.

Figura 22. Ventana principal de la aplicación.

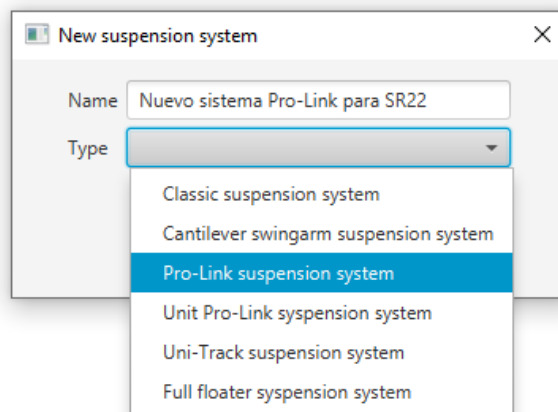
## Apéndice A. Manual de funcionamiento de la aplicación informática

Por un lado, la aplicación permite al usuario crear modelos de cada uno de los tipos de sistemas de suspensión, editar manualmente sus parámetros y estudiar cómo esto afecta a las curvas de funcionamiento del sistema. Por otro, la aplicación cuenta con una herramienta para automatizar el proceso de optimización, creando y evaluando nuevas soluciones de manera automática hasta converger en la solución más adecuada.

### A.1 Análisis

Al iniciar la aplicación, se abre la ventana principal que corresponde al *Análisis* (Figura 22). Lo primero que se aprecia es una barra de menú, típica de cualquier aplicación de escritorio, en la que se incluyen varios comandos organizados en cuatro menús.

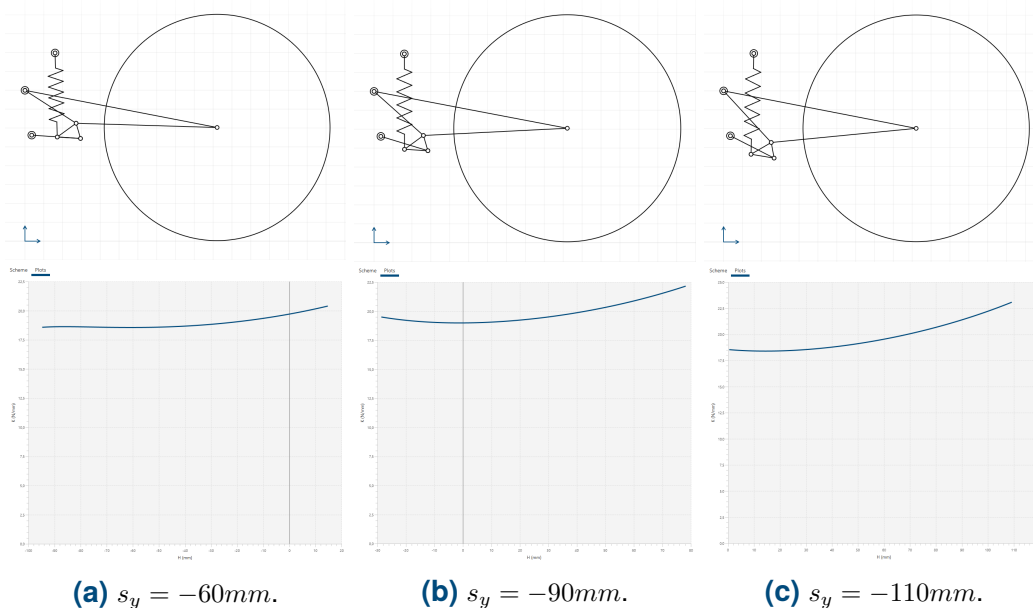
- Menú *File*.
  - Opened Files. Muestra los archivos que el programa tiene abiertos. Permite moverse entre los mismos, cambiando el archivo actual que el programa está analizando.
  - New (Ctrl. + N). Muestra el cuadro de diálogo para crear un archivo nuevo, tal y como puede apreciarse en la Figura 23. Éste pedirá al usuario un nombre y el tipo de sistema de suspensión que pretende modelar.
  - Open (Ctrl. + O). Abre un archivo previamente guardado en un fichero.



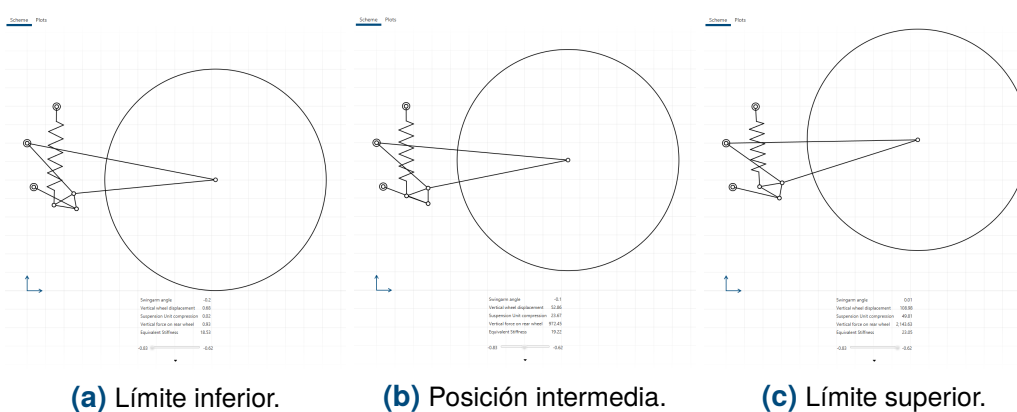
**Figura 23.** Creación de un nuevo archivo. Es necesario indicar un nombre para identificar el archivo, y el tipo de sistema de suspensión que va a modelar.

- Save (Ctrl. + S). Guarda los cambios que se hayan realizado al archivo actual.
- Save As. Guarda el archivo actual en un nuevo fichero.
- Menú *Edit*.
  - Undo (Ctrl. + Z). Permite deshacer el último cambio realizado en el archivo (cualquier parámetro modificado del sistema de suspensión).
  - Redo (Ctrl. + Y). Rehace los cambios deshechos tras el uso del comando Undo.
- Menú *Tools*.
  - Run optimization (Ctrl. + R). Ejecuta la herramienta de optimización.
- Menú *View*.
  - X Axis. Permite seleccionar cuál de las 5 variables de cálculo será la que se utilice para el Eje X de la gráfica.
  - Y Axis. Permite seleccionar cuál de las 5 variables de cálculo será la que se utilice para el Eje Y de la gráfica.

A la izquierda de la ventana de Evaluación se muestra un panel en el que aparecen todos los parámetros del sistema de suspensión que se esté trabajando, organizados por categorías. Desde este panel puede modificarse el valor de cualquiera de estos parámetros. Los cambios realizados se verán reflejados en el



**Figura 24.** Influencia del parámetro  $s_y$  sobre la representación gráfica del sistema y su curva de funcionamiento.



**Figura 25.** Diferentes posiciones dentro del rango de movimiento de un sistema Por-Link.

esquema del sistema de suspensión, en los límites superior e inferior del sistema, y en las gráficas de comportamiento, como se muestra en la la Figura 24.

En el panel central de la interfaz de usuario nos encontramos dos pestañas. La primera corresponde al esquema cinemático del sistema de suspensión. Éste es interactivo, ya que puede manipularse la barra situada en la zona inferior para cambiar la posición del sistema representado entre cualquier punto de sus límites superior e inferior. Además, puede conocerse el valor de cualquiera de las variables de cálculo en el punto seleccionado si se abre el desplegable que hay justo debajo.

Por último, la pestaña de gráficas permite visualizar las curvas de comportamiento del sistema. Utilizando las herramientas del menú *View*, puede graficarse una curva que relacione cualquiera de las variables de comportamiento.

## A.2 Optimización

Modificando los parámetros del sistema y atendiendo a cómo estos afectan a las curvas de comportamiento, el usuario podría buscar, a base de ensayo y error, los parámetros que consiguen que el sistema tenga el comportamiento deseado. Sin embargo, uno de los principales objetivos del programa es que esta búsqueda se realice de forma automática.

Para ello se ha implementado una herramienta de optimización, que puede ejecutarse desde el menú *Tools ->Run optimization*.

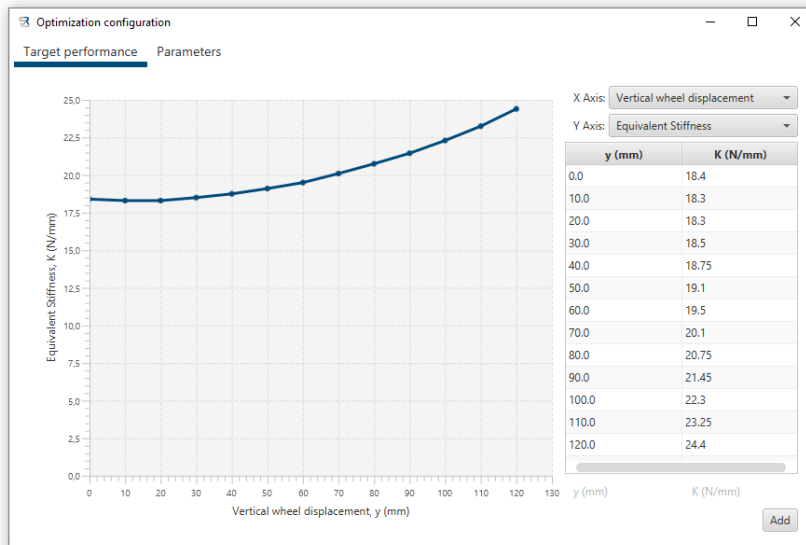
### A.2.1 Definición del comportamiento objetivo y del espacio de búsqueda

Nada más ejecutar la herramienta de optimización, aparece una nueva ventana con dos pestañas, tal y como se muestra en la Figura 26.

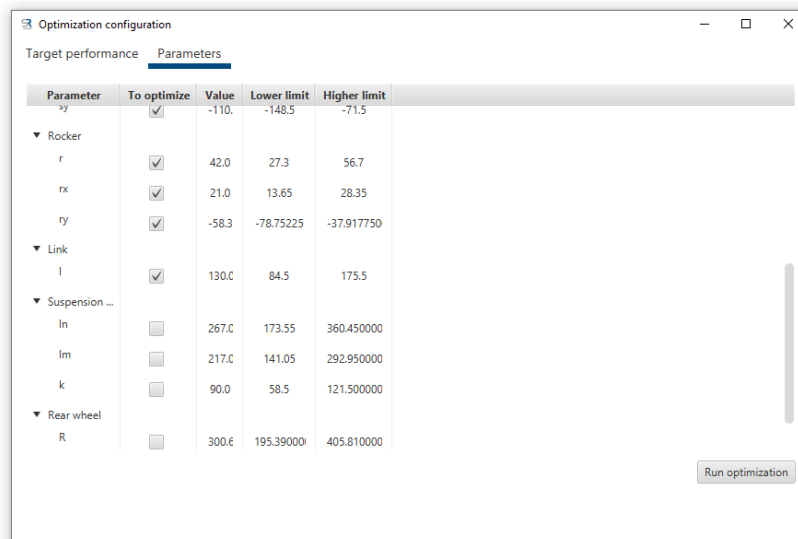
La primera pestaña (Figura 26a) sirve para establecer cuál es el comportamiento objetivo – el que se busca que tenga el sistema de suspensión. Para ello, en primer lugar han de seleccionarse las variables para los Ejes X e Y que definen el tipo de curva de comportamiento. Para hacerlo, se usan los dos desplegados de la derecha. En segundo lugar, se deben introducir suficientes puntos de control para definir la forma de la curva objetivo. Cada punto se introduce rellenando los dos cuadros de texto de debajo de la tabla con las coordenadas x e y del punto, y pulsando el botón *Add*. En este momento, se añadirá el punto a la tabla y se incluirá en la gráfica mostrada a la izquierda. Si se desea eliminar algunos de los puntos ya introducidos en la tabla, basta con seleccionarlo, hacer click derecho y pulsar *Delete* en el menú contextual.

La segunda pestaña (Figura 26) permite definir el espacio de búsqueda sobre el que quedará circunscrita la optimización. Se muestra una tabla en la que se listan de nuevo todos los parámetros del sistema. Para cada uno de ellos puede seleccionarse si va a incluirse en la optimización o no.

Que un parámetro se incluya en la optimización quiere decir que podrá ser modificado por el algoritmo para buscar el valor óptimo – aquel que hace que el sistema de suspensión se comporte según la curva objetivo establecida. Los límites inferior y superior limitan el rango de búsqueda. Se muestra un valor por defecto pero el usuario puede modificarlos según convenga. Por el contrario, si un parámetro no se incluye en la optimización significa que su valor no va a ser modificado. En dicho caso, debemos asegurarnos de que el valor establecido es el adecuado.



(a) Definición de la curva objetivo para la optimización.



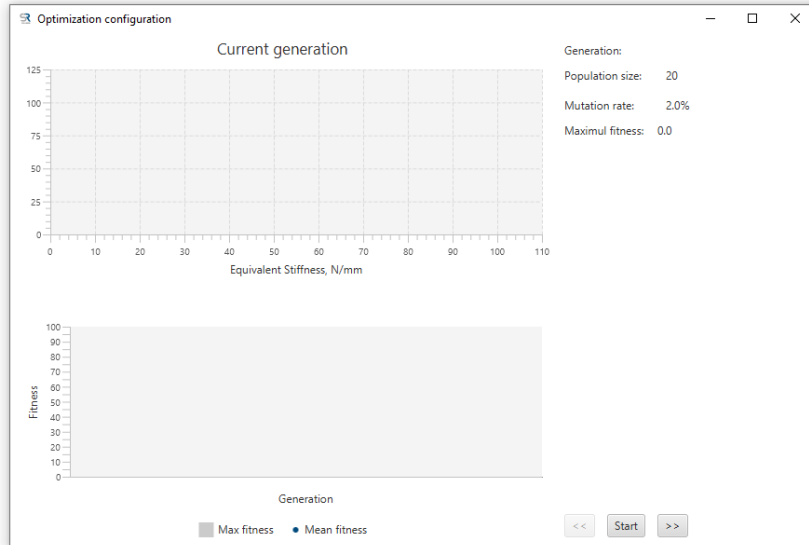
(b) Selección de los parámetros y sus límites superior e inferior.

Figura 26. Ventana para configurar la herramienta de optimización.

## A.2.2 Proceso de optimización

Definido el comportamiento objetivo y el espacio de búsqueda, puede pulsarse el botón *Run optimization*, que modifica la interfaz a la que se muestra en 27. En esta nueva interfaz, destinada al proceso de optimización, se muestra la siguiente información:

- Gráfica de comportamiento de la generación actual. Muestra las gráficas de comportamiento de todos los individuos de la generación en la que se encuentre el proceso de optimización, junto con la curva objetivo.
- Gráfica historial del valor de adecuación. Representa el valor de adecuación

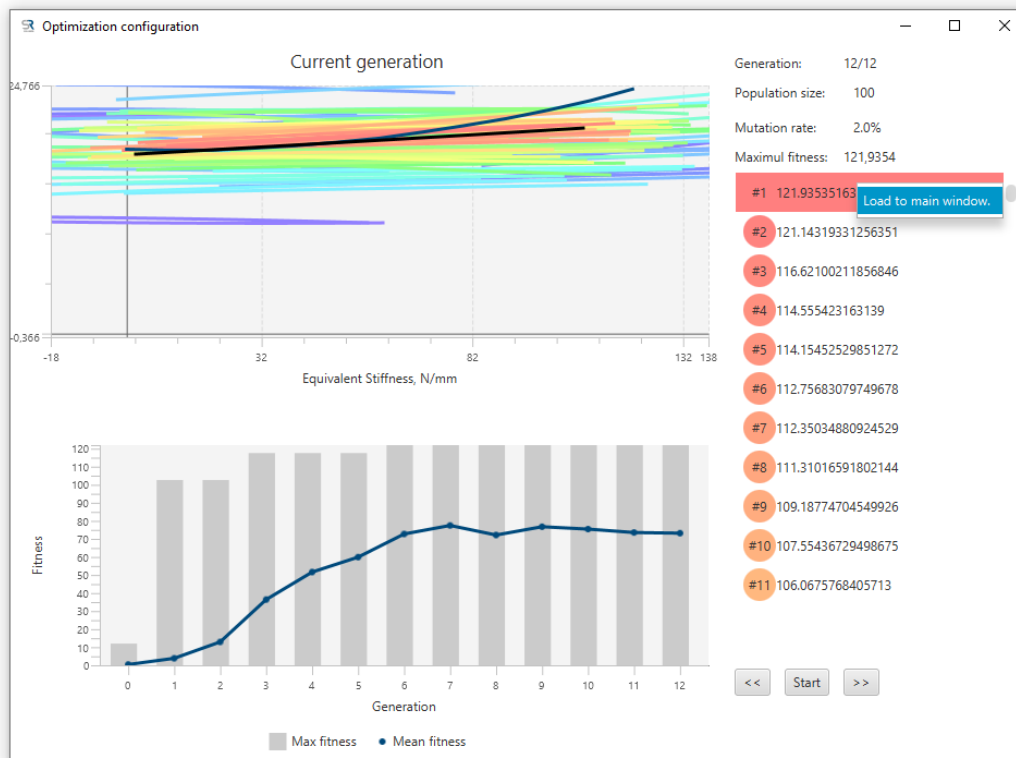


**Figura 27.** Ventana del proceso de optimización justo antes de comenzar.

máximo y el promedio en cada generación. El primero en formato de gráfica de barras, y el segundo de líneas.

- Información relevante durante el proceso. Se muestra:
  - Generación. Informa sobre la generación en la que se encuentra el programa así como la última generación obtenida. Este valor no siempre es el mismo debido a que el programa permite volver a generaciones anteriores para analizar los resultados.
  - Tamaño de la población. Indica el tamaño actual. Se permite modificar dicho valor tanto antes como durante la optimización, aunque en este último caso los cambios serán aplicables únicamente a las generaciones nuevas.
  - Ratio de mutación. Probabilidad con la que ocurre la mutación. También puede modificarse su valor tanto antes como durante la optimización.
  - Valor de adecuación máximo. Muestra el mayor valor obtenido en el transcurso de todo el proceso.
- Lista de individuos de la generación actual. Se muestran por orden descendente según su valor de adecuación. El color es el mismo que el de su correspondiente gráfica de comportamiento, por lo que sirve a modo de leyenda.
- Controles del proceso de optimización. El usuario tiene control total sobre el proceso de optimización usando los botones que están bajo la leyenda.





**Figura 28.** Ventana del proceso de optimización transcurridas 12 generaciones.

- Botón de generación anterior ( $\ll$ ). Permite volver a la generación anterior para consultar los individuos y sus gráficas de comportamiento.
- Botón de siguiente generación ( $\gg$ ). Avanza a una generación más 'actual'. Si el programa ya se encontraba en la última generación calculada, se obtiene una nueva.
- Botón Start. Comienza el proceso automático de optimización. Éste puede detenerse en cuanto se desee pulsando el mismo botón, e incluso reiniciar el proceso de nuevo.

Iniciado el proceso de optimización, los datos que muestra la interfaz se van actualizando acorde a cada nueva generación (ver la Figura 28). En muchos casos, la gráfica de comportamiento de algunos de los individuos abarcan dominios y recorridos muy elevados. Esto hace que el resto de curvas dejen de apreciarse en la gráfica de comportamiento. Para ello, pueden seleccionarse en la leyenda algunos individuos (que tengan especial interés). La gráfica ajustará los rangos del Eje X e Y para abarcar únicamente las curvas de dichos individuos junto con la objetivo. De esta forma, que interpretar los datos mostrados por la gráfica es mucho más sencillo.

Por último, si se desea analizar con más detalle algunas de las soluciones que

ha generado el algoritmo – por ejemplo para ver el esquema cinemático de las mismas – pueden seleccionarse de la lista y hacer click en la opción *Load to main window* que aparece como menú contextual.

## **Apéndice B. Notación empleada en el programa**

Este Anexo se recoge la notación que se ha empleado durante todo el documento y la aplicación informática para representar los parámetros de cada uno de los seis tipos de sistemas de suspensión trasera de motocicleta que se consideran en este trabajo.

## B.1 Sistema de suspensión tipo Clásico

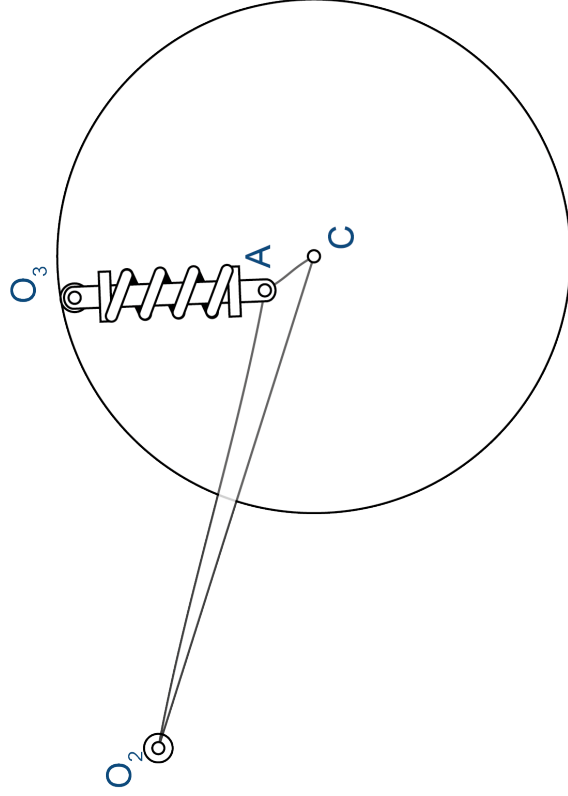


Figura 29. Esquema del sistema de suspensión tipo Clásico.

Parámetro	Descripción
<b>Chasis</b>	
$O_{2,x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2,y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{3,x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{3,y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
<b>Amortiguador</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

Tabla 6. Descripción de los parámetros que definen un sistema de suspensión tipo Clásico.

## B.2 Sistema de suspensión tipo Cantilever

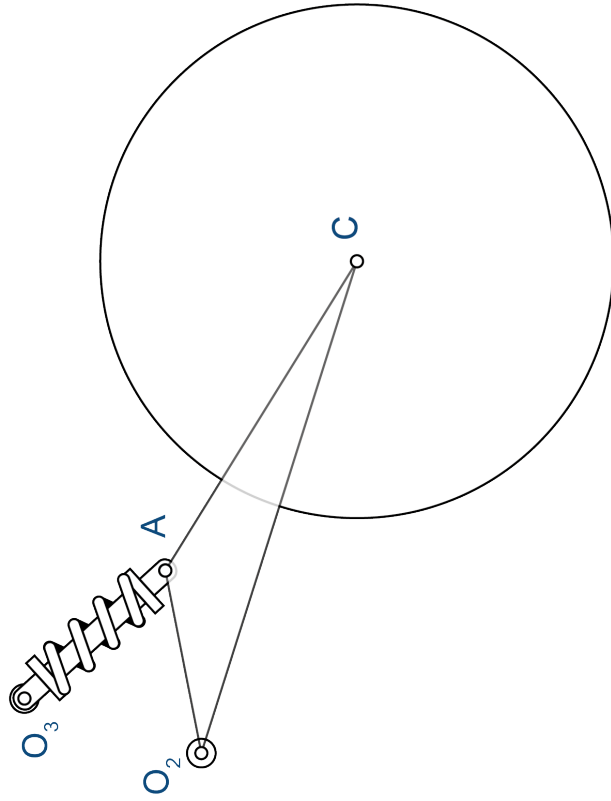


Figura 30. Esquema del sistema de suspensión tipo Cantilever.

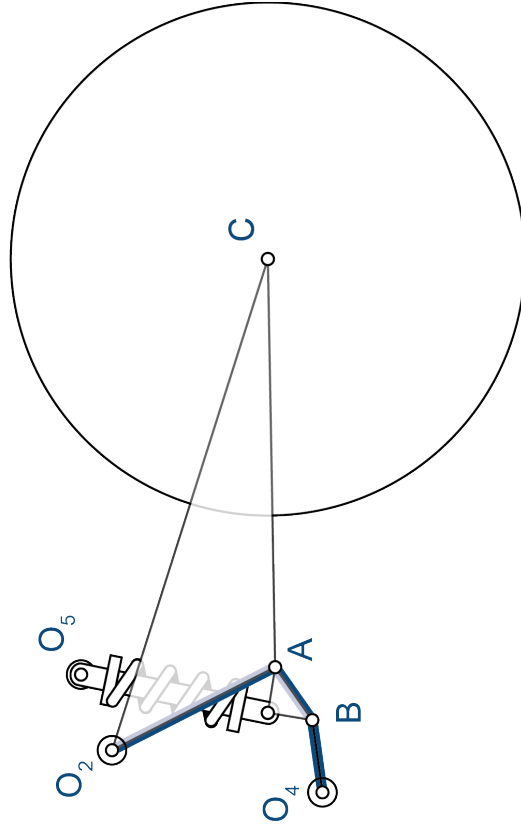
Parámetro	Descripción
<b>Chasis</b>	
$O_{2,x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2,y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{3,x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{3,y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
<b>Amortiguador</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

Tabla 7. Descripción de los parámetros que definen un sistema de suspensión tipo Cantilever.

### B.3 Sistema de suspensión tipo Pro-Link

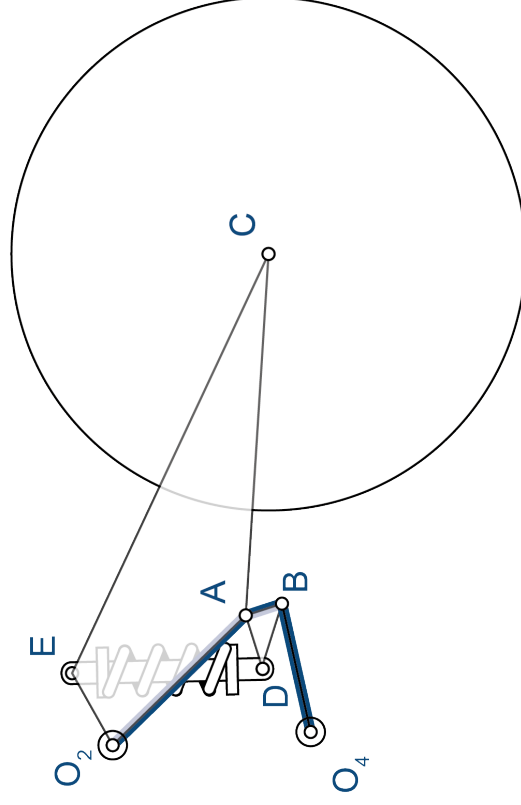
Parámetro	Descripción
<b>Chasis</b>	
$O_{2x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{4x}$	Coordenada x del punto de anclaje la bieleta al chasis.
$O_{4y}$	Coordenada y del punto de anclaje la bieleta al chasis.
$O_{5x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{5y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del balancín.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del balancín.
<b>Balancín</b>	
$r$	Longitud del balancín.
$r_x$	Coordenada x, en ejes locales del balancín, del punto de anclaje con el amortiguador.
$r_y$	Coordenada y, en ejes locales del balancín, del punto de anclaje con el amortiguador.
<b>Bieleta</b>	
$l$	Longitud de la bieleta.
<b>Unidad de Suspensión</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

**Tabla 8.** Descripción de los parámetros que definen un sistema de suspensión tipo Pro-Link.



**Figura 31.** Esquema del sistema de suspensión tipo Pro-Link.

## B.4 Sistema de suspensión tipo Unit Pro-Link



**Figura 32.** Esquema del sistema de suspensión tipo Unit Pro-Link.

Parámetro	Descripción
<b>Chasis</b>	
$O_{2,x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2,y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{4,x}$	Coordenada x del punto de anclaje la bieleta al chasis.
$O_{4,y}$	Coordenada y del punto de anclaje la bieleta al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del balancín.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del balancín.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
<b>Balancín</b>	
$r$	Longitud del balancín.
$r_x$	Coordenada x, en ejes locales del balancín, del punto de anclaje con el amortiguador.
$r_y$	Coordenada y, en ejes locales del balancín, del punto de anclaje con el amortiguador.
<b>Bieleta</b>	
$l$	Longitud de la bieleta.
<b>Unidad de Suspensión</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima del amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

**Tabla 9.** Descripción de los parámetros que definen un sistema de suspensión tipo Unit Pro-Link.

## B.5 Sistema de suspensión tipo Uni-Track

Parámetro	Descripción
<b>Chasis</b>	
$O_{2x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{4,x}$	Coordenada x del punto de anclaje balancín al chasis.
$O_{4,xy}$	Coordenada y del punto de anclaje la balancín al chasis.
$O_{5,x}$	Coordenada x del punto de anclaje del amortiguador al chasis.
$O_{5,y}$	Coordenada y del punto de anclaje del amortiguador al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante ( $O_2\bar{C}$ ).
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje de la bieleta.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje de la bieleta.
<b>Balancín</b>	
$r$	Longitud del balancín.
$r_x$	Coordenada x, en ejes locales del balancín, del punto de anclaje con el amortiguador.
$r_y$	Coordenada y, en ejes locales del balancín, del punto de anclaje con el amortiguador.
<b>Bieleta</b>	
$l$	Longitud de la bieleta.
<b>Unidad de Suspensión</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima el amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

Figura 33. Esquema del sistema de suspensión tipo Uni-Track.

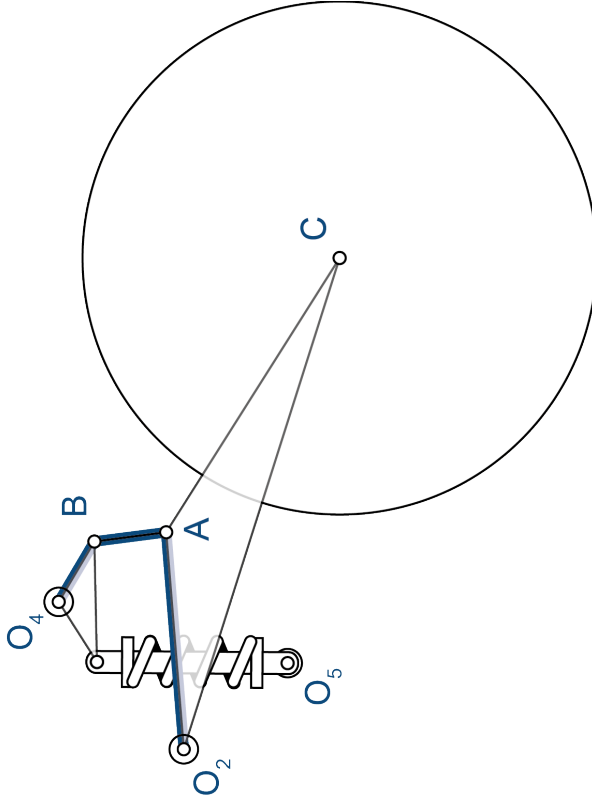


Tabla 10. Descripción de los parámetros que definen un sistema de suspensión tipo Uni-Track.



## B.6 Sistema de suspensión tipo Full-Floater

Parámetro	Descripción
<b>Chasis</b>	
$O_{2x}$	Coordenada x del punto de anclaje del basculante al chasis.
$O_{2y}$	Coordenada y del punto de anclaje del basculante al chasis.
$O_{4x}$	Coordenada x del punto de anclaje del balancín al chasis.
$O_{4y}$	Coordenada y del punto de anclaje del balancín al chasis.
<b>Basculante</b>	
$s$	Longitud del basculante ( $\overline{O_2C}$ ).
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje de la bieleta.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje de la bieleta.
$s_x$	Coordenada x, en ejes locales del basculante, del punto de anclaje del amortiguador.
$s_y$	Coordenada y, en ejes locales del basculante, del punto de anclaje del amortiguador.
<b>Balancín</b>	
$r$	Longitud del balancín ( $\overline{BD}$ ).
$r_x$	Coordenada x, en ejes locales del balancín, del punto de anclaje con el amortiguador.
$r_y$	Coordenada y, en ejes locales del balancín, del punto de anclaje con el amortiguador.
<b>Bieleta</b>	
$l$	Longitud de la bieleta.
<b>Unidad de Suspensión</b>	
$l_n$	Longitud natural (máxima) del amortiguador.
$l_m$	Longitud mínima del amortiguador.
$k$	Constante de rigidez del muelle.
<b>Rueda</b>	
$R$	Radio de la rueda trasera.

Figura 34. Esquema del sistema de suspensión tipo Full-Floater.

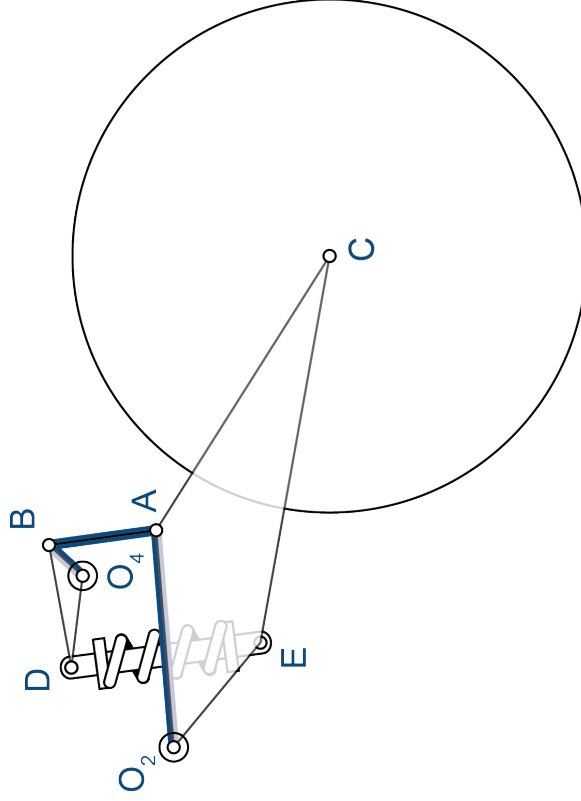


Tabla 11. Descripción de los parámetros que definen un sistema de suspensión tipo Full-Floater.