



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño de una suspensión
"doble brazo de A" con
ayuda de un algoritmo
basado en COGA-II**

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Sarahi Xanat Trejo Sánchez

DIRECTOR DE TESIS

M. en I. Osiris Ricardo Torres



Ciudad Universitaria, Cd. Mx., 2019

DEDICATORIAS

A mis padres que además de darme la vida me han acompañado por ella apoyando cada una de mis locuras, a mi hermana por enseñarme a temprana edad que si la vida me tira me tengo que levantar, al resto de mi familia que al igual que ellos han sido mi soporte incondicional, a mis profesores que me dieron el mejor regalo que uno puede recibir, el conocimiento. A todos los miembros de UNAM Motorsports que me precedieron porque sin ellos no hubiera podido participar en el proyecto, a los que me sucedieron por continuarlo y a los que participaron conmigo por vivir esa asombrosa experiencia junto a mí. A mis niños, David y Orli, y a mi niña Irene por permitirme ser parte de sus vidas y transmitirles lo poquito que se. A mis amigos con los que he vivido maravillosas experiencias y a Francisco por inspirarme e impulsarme a cumplir mis metas y ser mejor cada día.

DISEÑO DE UNA SUSPENSIÓN “DOBLE BRAZO DE A” CON AYUDA DE UN ALGORITMO BASADO EN COGA-II.

Nomenclatura.....	ii
Acrónimos.....	iv
Ilustraciones de referencia.....	v
Índice de ilustraciones.....	vi
Introducción.....	viii
Capítulo 1: Estado del arte.....	1
1.1: Suspensiones.....	1
1.2: Optimización.....	15
Capítulo 2: Caso de estudio práctico.....	28
2.1: Proceso de diseño de una suspensión para FSAE.....	28
2.2: Proceso propuesto.....	33
Capítulo 3: Optimización.....	37
3.1: Funciones objetivo.....	37
3.2: Restricciones de desigualdad.....	43
3.3: Restricciones de igualdad.....	53
3.4: Programación del algoritmo genético para determinar parámetros primitivos.....	53
3.5: Disminución del error de la razón de instalación.....	54
Capítulo 4: Resultados.....	56
Capítulo 5: Discusión y Conclusiones.....	59
Anexo 1: Fórmula SAE.....	63
Anexo 2: Suspensión 2017.....	64
Anexo 3: Diseño geométrico de una suspensión doble brazo de A.....	68
Anexo 4: Programa para optimización de parámetros primitivos de suspensión.....	70
Anexo 5: Programa para la disminución del error del IR deseado.....	80
Anexo 6: Suspensión 2019.....	83
Bibliografía.....	85

NOMENCLATURA.

Sigla	Nombre	Descripción	Unidades
a		Distancia longitudinal entre el eje delantero y el centro de gravedad del vehículo.	m
a'		Distancia longitudinal entre el centro de gravedad del vehículo y el punto más lejano frontal en el eje x.	m
a_x	Aceleración longitudinal.	Aceleración longitudinal del centro de gravedad del vehículo.	m/s^2
a_y	Aceleración lateral.	Aceleración lateral del centro de gravedad del vehículo.	m/s^2
b		Distancia entre el centro de gravedad del vehículo y el eje trasero.	m
b'		Distancia longitudinal entre el centro de gravedad del vehículo y el punto más lejano trasero en el eje x.	m
c_ϕ	Rigidez al <i>roll</i> .	Puede ser del eje delantero o trasero o de la barra de torsión.	Nm/rad
CG	Centro de gravedad		1
D_{XZ}		Distancia entre el CG del vehículo y el de la figura correspondiente en el plano XZ	m
D_{XY}		Distancia entre el CG del vehículo y el de la figura correspondiente en el plano XY	m
E	Módulo de elasticidad	Relación entre el esfuerzo y la deformación del material de la barra de torsión.	Pa
f_{vsa}	El brazo de oscilación en la vista frontal	Distancia entre el centro instantáneo de rotación y el centro de contacto de la llanta, son siglas del inglés, <i>front view swing arm</i> .	m
g	Gravedad	Aceleración de la gravedad.	m/s^2
h	Altura del prisma equivalente.	Altura del prisma auxiliar para el cálculo de momentos de inercia.	m
h'		Distancia entre el CG del vehículo y el eje de <i>roll</i> .	m
h_{CG}		Altura del CG	m
h_{RC}		Altura del centro de <i>roll</i>	m
I_y	Momento de inercia sobre y.	Momento de inercia en el CG del vehículo sobre su eje y.	$kg\ m^2$
I_z	Momento de inercia sobre z.	Momento de inercia en el CG del vehículo sobre su eje z.	$kg\ m^2$
IR	Razón de instalación	Razón entre el desplazamiento del resorte y el de la llanta.	1
K_R	Razón de viaje vertical (<i>ride rate</i>)	Es la fuerza por unidad de desplazamiento de la llanta en el centro de contacto de esta con el pavimento.	N/m
k_s	Constante del resorte.	Fuerza aplicada al resorte sobre su compresión o extensión.	N/m

k_t	Constante de resorte de la llanta.	Fuerza aplicada a la llanta sobre la compresión de esta.	N/m
l	<i>Wheelbase</i>	Distancia entre el eje delantero y el eje trasero a la altura del centro del centro de contacto de las llantas con el pavimento (<i>contact patch</i> en inglés).	m
L_{ARB}		Longitud de la barra de torsión o ARB, por las siglas de <i>anti-roll bar</i> sus en inglés.	m
R	Radio medio	Radio de una curva de la pista medido al centro de esta.	m
r	Velocidad de <i>guiñada</i> (<i>yaw</i>).	Velocidad de giro sobre el eje y .	rad/s
R_e	Radio exterior	Radio exterior de una curva de la pista.	m
t	Tiempo.		s
tr	<i>Distancia entre llanta derecho e izquierda</i> (<i>track</i>)	Distancia lateral entre el centro del <i>contact patch</i> de las llantas de cada eje.	m
tr'	Ancho del prisma	Ancho del prismas auxiliar utilizado para el cálculo del momento de inercia.	m
t_w	Ancho del neumático.	Ancho total del neumático, no solamente de la huella.	m
t_\emptyset	Diámetro del neumático.	Diámetro de neumático tomado de su catálogo.	m
u	Velocidad longitudinal	Velocidad en el eje "x" del vehículo.	m/s^2
m	Masa del vehículo.	Masa estática del vehículo incluyendo al piloto, medida con este detenido en una superficie nivelada.	kg
m_p	Masa del piloto.		kg
m_s	Masa suspendida.	Masa suspendida del vehículo incluyendo al piloto. Se puede considerar como la masa total menos 4 veces la masa no suspendida.	kg
m_u	Masa no suspendida.	Masa no suspendida de una esquina del vehículo. Si son diferentes considerar un promedio.	kg
V	Velocidad del vehículo	Cambio de posición en el tiempo.	m/s
V_n	Volumen n	Volumen de la figura n, donde n es 1, 2, 3, 4 o 5	
v	Velocidad lateral		m/s
v_p	Módulo de Poisson.		m/s
x_n	Coordenada x	Coordenada x del centro de gravedad del cuerpo n.	m
y_n	Coordenada y	Coordenada y del CG de la figura n.	m
z_n	Coordenada z	Coordenada z del CG de la figura n.	m
ΔF_z	Transferencia de carga	Cambio de fuerza normal en el eje z.	N
$\rho_{p,s}$	Densidad	Densidad de un ser humano o el acero respectivamente.	kg/m^3
φ	Ángulo de <i>roll</i>		rad
∇_{roll}	Gradiente de <i>roll</i>	Rigidez de <i>roll</i> normalizada en grados o radianes de <i>roll</i> sobre aceleración lateral en g's.	$rad/g's$

Subíndices

AC	Punto interior del brazo inferior de la suspensión (vista frontal).
B	Punto exterior del brazo inferior de la suspensión
CG	Centro de gravedad
F	Eje delantero
IC	Centro instantáneo de rotación (<i>Instant Center</i>)
Máx	Máximo
Mín	Mínimo
R	Eje trasero
RC	Centro de <i>roll</i>
1,2,3,4,5	Cuerpo 1,2,3,4 o 5 auxiliar para el cálculo de I_z

ACRÓNIMOS

Acónimo	Significado
ARB	Barra de torsión, por sus siglas en inglés de anti-roll bar
SAE	Siglas en inglés de sociedad de ingenieros automotrices (<i>Society of Automotive Engineers</i>)
FSAE	Competencia de vehículos de tipo formula organizada por la SAE
IR	Razón de instalación (installation ratio)
TTC®	<i>Tire Test Consortium</i> , organismo dedicado a obtener información de calidad de las llantas dirigidas a las competencias de FSAE y FS,
GA	Siglas en inglés de algoritmos genéticos (<i>Genetic Alcorithm</i>)
MUGA	<i>Multi-objective Uniform-diversity Genetic Algorithm</i>
COGA	Siglas en inglés de algoritmo genético de objetivos comprimidos (<i>Compressed-objective Genetic Algorithm</i>)
WS	Puntuación ganadora, siglas del inglés de winning score

ILUSTRACIONES DE REFERENCIA

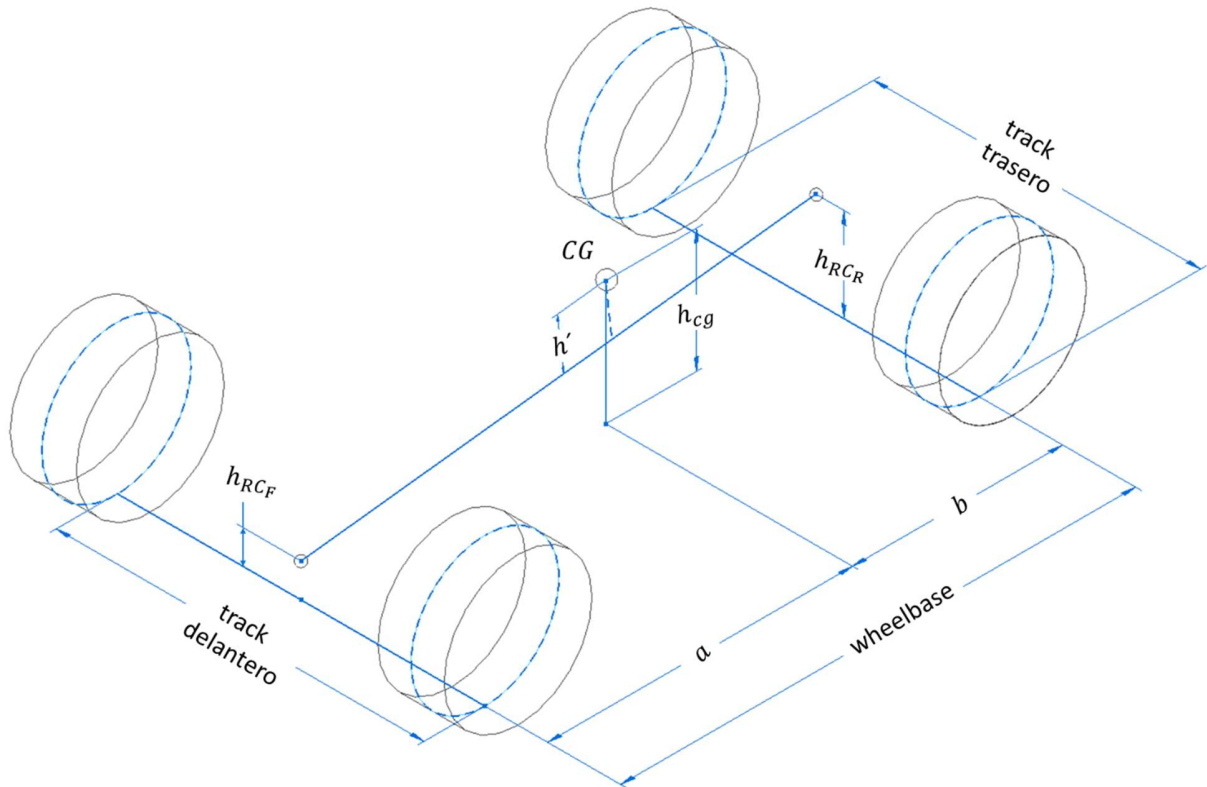


Ilustración 1: Referencia para el *wheelbase*, *tracks*, alturas de centros de roll, altura del cg y altura del eje de roll al centro de gravedad.

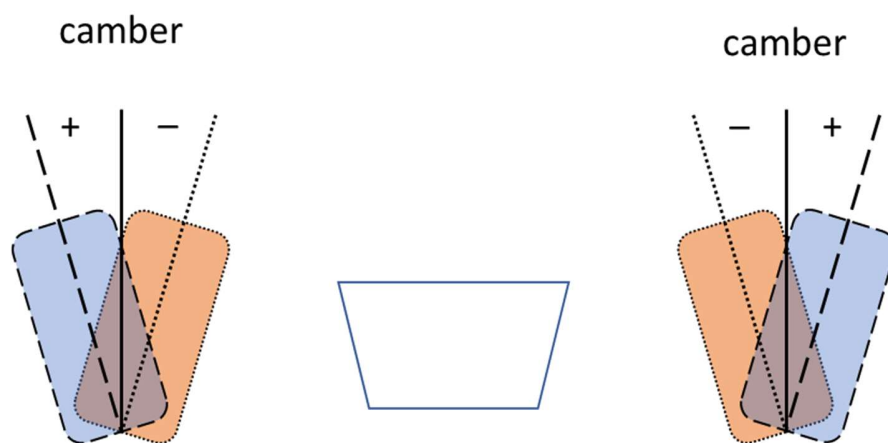


Ilustración 2: Referencia para el ángulo de *camber* de los neumáticos.

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: REFERENCIA PARA EL <i>WHEELBASE</i> , <i>TRACKS</i> , ALTURAS DE CENTROS DE <i>ROLL</i> , ALTURA DEL CG Y ALTURA DEL EJE DE <i>ROLL</i> AL CENTRO DE GRAVEDAD.....	V
ILUSTRACIÓN 2: REFERENCIA PARA EL ÁNGULO DE <i>CAMBER</i> DE LOS NEUMÁTICOS.....	V
ILUSTRACIÓN 1-1: SUSPENSIÓN DE DOBLE BRAZO DE A (<i>DOUBLE WISHBONE</i>) [5].....	3
ILUSTRACIÓN 1-2 SUSPENSIÓN DOBLE BRAZO DE A: A) <i>PUSHROD</i> B) <i>PULLROD</i>	3
ILUSTRACIÓN 1-3: SUSPENSIÓN PASIVA. [1]	4
ILUSTRACIÓN 1-4: SUSPENSIÓN SEMI-ACTIVA CON AMORTIGUADOR VARIABLE. [1]	4
ILUSTRACIÓN 1-5: SUSPENSIÓN ACTIVA [1].....	5
ILUSTRACIÓN 1-6: MODELO DE 1/4 DE VEHÍCULO.....	6
ILUSTRACIÓN 1-7: DIAGRAMA DE CUERPO LIBRE DE LAS MASAS DE UN 1/4 DE VEHÍCULO	6
ILUSTRACIÓN 1-8 DIAGRAMA DE CUERPO LIBRE SIN EXCITACIÓN	8
ILUSTRACIÓN 1-9: MODELO DE ½ VEHÍCULO. [8]	9
ILUSTRACIÓN 1-10: MODELADO DE VEHÍCULO COMPLETO. [8]	11
ILUSTRACIÓN 1-11: MODELO DE COCHE COMPLETO DE PACEJKA.....	11
ILUSTRACIÓN 1-12: MODELADO POR MULTI CUERPOS DE UNA SUSPENSIÓN DOBLE BRAZO DE A. [10]	12
ILUSTRACIÓN 1-13: DIAGRAMA DE FLUJO GENERAL DE UN ALGORITMO GENÉTICO	17
ILUSTRACIÓN 1-14: EJEMPLO DE UNA MUESTRA UNIVERSAL ESTOCÁSTICA	20
ILUSTRACIÓN 1-15: EJEMPLO DE 1 PUNTO DE CRUZA CON $K = 2$	21
ILUSTRACIÓN 1-16: EJEMPLO DE DOS PUNTOS DE CRUZA EN LA POSICIÓN 2 Y 6.....	21
ILUSTRACIÓN 1-17: EJEMPLO DE CRUZA UNIFORME	22
ILUSTRACIÓN 1-18: EJEMPLO DE CRUZA FIJA	22
ILUSTRACIÓN 1-19: CRUZA DE ORDEN.....	23
ILUSTRACIÓN 1-20: DIAGRAMA DE FLUJO DE COGA-II [19]	25
ILUSTRACIÓN 2-1: PROCESO DE DISEÑO DEL 2017	28
ILUSTRACIÓN 2-2: MODELO DE SUSPENSIÓN A) DELANTERA, B) TRASERA.....	32
ILUSTRACIÓN 2-3: BOCETO EN GEOGEBRA PARA DEFINIR LOS PUNTOS DE LOS BRAZOS DE SUSPENSIÓN.....	35
ILUSTRACIÓN 2-4: DIAGRAMA DE PROCESO DE DISEÑO PROPUESTO PARA UNA SUSPENSIÓN DOBLE BRAZO DE A.....	36
ILUSTRACIÓN 3-1 DIAGRAMA PARA EL CÁLCULO DE MOMENTOS DE INERCIA.....	37
ILUSTRACIÓN 3-3 SUMA DE VECTORES PARA EL CÁLCULO DE <i>TRACK</i> MÁXIMO	44
ILUSTRACIÓN 3-2 DIAGRAMA DEL AUTOMÓVIL EN LA PISTA PARA EL CÁLCULO DEL <i>TRACK</i> MÁXIMO.....	44
ILUSTRACIÓN 3-4 DIBUJO SIMPLIFICADO DE UN COCHE TIPO FÓRMULA PARA EL CÁLCULO DEL <i>TRACK</i> MÁXIMO.....	45
ILUSTRACIÓN 3-5 DIAGRAMA PARA EL CÁLCULO DEL <i>TRACK</i> MÍNIMO DELANTERO	47
ILUSTRACIÓN 3-6 CÁLCULO DEL <i>TRACK</i> MÍNIMO DELANTERO.....	47
ILUSTRACIÓN 3-7 CÁLCULO DEL <i>TRACK</i> MÍNIMO TRASERO.....	48
ILUSTRACIÓN 3-8: VISTA FRONTAL PARA EL DISEÑO DE UNA SUSPENSIÓN DE DOBLE BRAZO DE A.....	48
ILUSTRACIÓN 3-9 CÁLCULO DE LA ALTURA MÍNIMA DEL PUNTO AC	49
ILUSTRACIÓN 3-10 DIAGRAMA SIMPLIFICADO DEL ESPACIO INTERIOR DEL RIN.....	50
ILUSTRACIÓN 3-11: DIBUJO AUXILIAR PARA LA OPTIMIZACIÓN DEL IR.....	54
ILUSTRACIÓN 3-12: A) MECANISMO EN 2D DE LA SUSPENSIÓN, B) GRÁFICA PARA EL IR.....	55
ILUSTRACIÓN 4-1: TABLA DE RESULTADOS DEL ALGORITMO GENÉTICO.....	56
ILUSTRACIÓN 4-2: PROPUESTA PARA SUSPENSIÓN DELANTERA	57
ILUSTRACIÓN 4-3: PROPUESTA DE SUSPENSIÓN TRASERA	57
ILUSTRACIÓN 5-1: HISTOGRAMAS DEL MOMENTO DE INERCIA DE ITERACIONES CON 1, 39 Y 2000 GENERACIONES.....	59
ILUSTRACIÓN 5-2: HISTOGRAMAS DEL ERROR DE GRADIENTE DE <i>ROLL</i> DE ITERACIONES CON 1, 39 Y 2000 GENERACIONES.....	59

ILUSTRACIÓN 5-3: HISTOGRAMAS DEL ERROR DE GRADIENTE DE ROLL DE ITERACIONES CON 1, 39 Y 2000 GENERACIONES.....	60
ILUSTRACIÓN 5-4: HISTOGRAMAS DEL ERROR DE GRADIENTE DE ROLL DE ITERACIONES CON 1, 39 Y 2000 GENERACIONES.....	60
ILUSTRACIÓN 5-5: HISTOGRAMAS DE ERROR DE TRANSFERENCIA DE ITERACIONES CON 1, 39 Y 2000 GENERACIONES	61

INTRODUCCIÓN

El primer vehículo que incorporó un motor de combustión interna se remonta a 1886, en los años subsecuentes se logró un aumento considerable en la velocidad de los vehículos por la mejora de los motores [1]. Sin embargo, el análisis de la dinámica vehicular quedó rezagado hasta 1931, cuando se construyó un dinamómetro de llanta, permitiendo así el inicio del estudio de la dinámica de los neumáticos [1] y en consecuencia el entendimiento de la física involucrada cuando el vehículo toma una curva. Desde esa época, no se han dejado de estudiar los componentes que proveen al vehículo de una manejabilidad y confort adecuados. Actualmente existe un gran campo dedicado al estudio de los vehículos de uso cotidiano que se enfoca a los rubros de confort y seguridad de los conductores; asimismo, existe otro campo que enfoca sus estudios en el desarrollo de vehículos de competencia para el mejoramiento de la maniobrabilidad y reducción de tiempo de pista. Este último campo ha impregnado a nivel licenciatura con competencias como las creadas por SAE, las cuales son diseñadas para el involucramiento temprano de los futuros profesionistas en el mundo de las competencias vehiculares.

Este trabajo se enfocará en el diseño de la suspensión para un vehículo de competencia, pasando a un segundo plano la comodidad del piloto y resaltando la manejabilidad y reducción de tiempo de vuelta. Estos parámetros son grandemente influenciados tanto por características geométricas como de los valores de los componentes de la suspensión.

Durante mi participación en el equipo de UNAM Motorsports como integrante del equipo y responsable del desarrollo del sistema de suspensión, tuve la oportunidad de desarrollar propuestas de suspensión de tipo doble brazo de A, tomado como base los valores propuestos en el trabajo “Rediseño de un sistema de suspensión para un auto de competencia mediante Adams/Car” [2]. A través del desarrollo e implementación del sistema de suspensión en el vehículo de FSAE, se identificaron las ventajas y desventajas que se presentan durante la etapa de diseño; uno de los principales retos a los que nos enfrentamos, fue el de determinar y seleccionar los valores de diseño que impactan directamente en el comportamiento dinámico final del prototipo como son: distancia entre la llanta derecha e izquierda (*track*) de ambos ejes, distancia entre el eje delantero y trasero (*wheelbase*), rigidez de alabeo (*roll*), etc. Algunos de estos deben ser aproximados dado que para un mejor cálculo se requieren datos que son determinados durante las etapas posteriores, mientras que para otros se deben realizar una gran cantidad de iteraciones en softwares especializados para el diseño de vehículos. Esto genera que el proceso de diseño sea lento, lo que resulta contraproducente para los tiempos de todo el equipo.

Objetivos

- Proponer una metodología para el diseño de una suspensión del prototipo de FSAE.
- Desarrollar una herramienta mediante la implementación de la teoría de algoritmos genéticos (COGA-II) para la reducción de los tiempos de diseño de una suspensión para un prototipo FSAE.

Capítulo 1: ESTADO DEL ARTE

1.1: SUSPENSIONES

Durante décadas el ser humano ha inventado diferentes formas de desplazarse y transportar mercancía a través de diversos terrenos y distancias, proporcionando cierto grado de confort y con el mejor desempeño posible. Para el desplazamiento terrestre, se han creado diversos automóviles, compuestos por diferentes sistemas mecánicos y electrónicos. Uno de los sistemas más indispensables para proporcionar confort y maniobrabilidad durante la conducción es el sistema de suspensión, ésta tiene las siguientes tres funciones básicas:

- Mantener contacto continuo entre el vehículo y la superficie de avance.
- Reducir la variación de fuerza normal en las llantas
- Absorber las irregularidades e inclinación del camino.

La incorporación de un sistema de suspensión genera *per se* nuevas necesidades en temas de maniobrabilidad debido a la dinámica de las fuerzas generadas por ésta, por lo cual, han sido estudiadas ampliamente a lo largo de los años. Su complejidad en diseño estructural y control de parámetros dinámicos han aumentado como consecuencia de la experiencia y comprensión de los fenómenos que se presentan en este tipo de sistemas.

Al mencionar el término suspensión en un contexto automotriz automáticamente salta a la imaginación el sistema incluido en los vehículos de calle o de competencia, aunque esta expresión no es exclusiva de los vehículos de cuatro ruedas debido a la existencia de diferentes vehículos que también utilizan este tipo de sistemas modificados, por ejemplo, las bicicletas, motocicletas y los vehículos exploradores todo terreno. Sin embargo, como dentro de los objetivos de este trabajo se encuentra el diseño de una suspensión de un automóvil, a continuación, se expresan únicamente los tipos de suspensiones para estos.

1.1.1: TIPOS DE SUSPENSIONES POR SU GEOMETRÍA

Es posible la clasificación de las suspensiones tipo automotriz en dos grandes grupos: de eje sólido e independiente. A continuación, se enlistarán los rasgos principales y tipos de cada grupo.

Suspensiones de eje sólido. Este tipo de suspensiones tiene las siguientes características generales:

- Dependencia entre las llantas.
- Requieren espacio para el movimiento vertical del resorte.
- Poca flexibilidad en el tuneo cinemático y dinámico.

Entre las configuraciones más utilizadas de eje sólido se encuentran:

- Tipo Hotchkiss: El eje está sostenido por los resortes de ballesta, es común en vehículos de carga por su simplicidad y bajo costo [3], sin embargo, para la mejora de la calidad del *ride* es necesario hacer las ballestas más largas, pero se empeora la estabilidad longitudinal.

- Suspensión de 4 barras: El eje sólido es sostenido por 4 eslabones, dos superiores y dos inferiores, provee de un mejor control de la posición del centro de *roll*, comportamiento anti-agacharse (*anti-squat*) y anti-sumergirse (*anti-dive*), aunque tiene un mayor precio que la suspensión Hotchkiss [1].
- Suspensión de Dion: Paso intermedio entre una suspensión de eje sólido y una independiente, el diferencial se encuentra montado en el chasis, por lo que se reduce la masa no suspendida.

Suspensiones independientes. Al generalizar se puede decir que tienen la siguiente tendencia:

- Requieren poco espacio
- Un cambio cinemático y/o elastocinemático de *toe-in*
- Posible tendencia hacia el subviraje
- Masa baja
- No hay influencia entre las llantas [4]

Durante el desarrollo de las suspensiones, se han generado varias configuraciones. Entre las más comunes se encuentran las siguientes:

- Suspensión McPherson: Posee un brazo inferior de control y el amortiguador funciona como eslabón superior el cual puede o no poseer un resorte axial, lo que le permite tener una menor cantidad de partes y una mayor distribución de esfuerzo, pero sacrifica el control cinemático. El chasis debe ser reforzado en la sujeción superior y exige mucho espacio para su instalación [3].
- Brazo de remolque (*trailing arm*): Se sujeta a un eslabón longitudinal con eje de giro perpendicular al eje longitudinal del vehículo, este tipo de suspensión no presenta cambio de *camber* ni de *toe* con las fuerzas longitudinales y laterales [4], además de ahorrar espacio [1], pero puede causar un pequeño sobreviro por la deformación del eslabón [4] y pequeño cambio de *wheelbase* [1].
- Brazo de semi-remolque (*semi trailing arm*): Parecida al brazo de remolque, pero con un ángulo cercano a los 45° [5]. Con la longitud adecuada del brazo de control, se puede afectar positivamente, la altura del centro de *roll*, cambio de *camber* y *toe-in*, pero se tiene una tendencia a sobrevirar [4].
- Suspensión de múltiples eslabones (*multilink*): Es un mecanismo de 6 barras donde los dos eslabones de un brazo de control son unidos por medio de una junta, lo que permite mayor flexibilidad para la manipulación de parámetros de suspensión como el centro de *roll* pero provoca mayor precio y menor confiabilidad que el siguiente tipo de suspensión [5].
- Doble brazo de A (*short long arm* o *double wishbone*): Dos eslabones transversales en forma de A, brazos de control, a cada lado del vehículo, los cuales rotan respecto al chasis, estos brazos son unidos a la mangueta (*wheel knuckle*) por medio de rótulas, ver Ilustración 1-1, entre mayor sea la distancia efectiva entre las rótulas de la mangueta menores serán las fuerzas en los brazos de suspensión [4]. La ventaja de una suspensión doble brazo de A, es el alto número de posibilidades cinemáticas: la posición relativa entre ambos brazos permite la ubicación del centro de *roll* y el polo de *pitch* a criterio del diseñador, mientras que las longitudes de los brazos influyen el ángulo de movimiento de la llanta, es decir el cambio de *camber* [4]. Con un brazo superior de menor longitud se provoca que la suspensión a compresión tenga un *camber* negativo y la llanta a extensión a positivo, lo que se compensa el efecto de *roll pitch* del cuerpo. [4]. Si el polo de *pitch* se localiza arriba del centro de la

llanta produce un mejor comportamiento de anti-dive y reduce el *squat*, pero tiene un alto costo de producción debido a que son necesarios muchos puntos de sujeción [1].

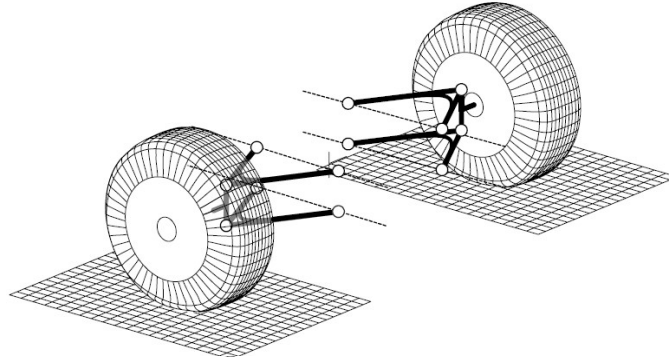


Ilustración 1-1: Suspensión de doble brazo de A (*double wishbone*) [5]

Las suspensiones de doble brazo y *multilink* pueden tener un balancín entre un brazo de control y el amortiguador, lo cual conlleva dos posibles configuraciones para el eslabón que conecte uno de los brazos de control con el balancín, empujar o jalar (*push* o *pull rod*). En la primera este se conecta con el brazo inferior mientras que en la segunda es desde el superior, ver Ilustración 1-2.

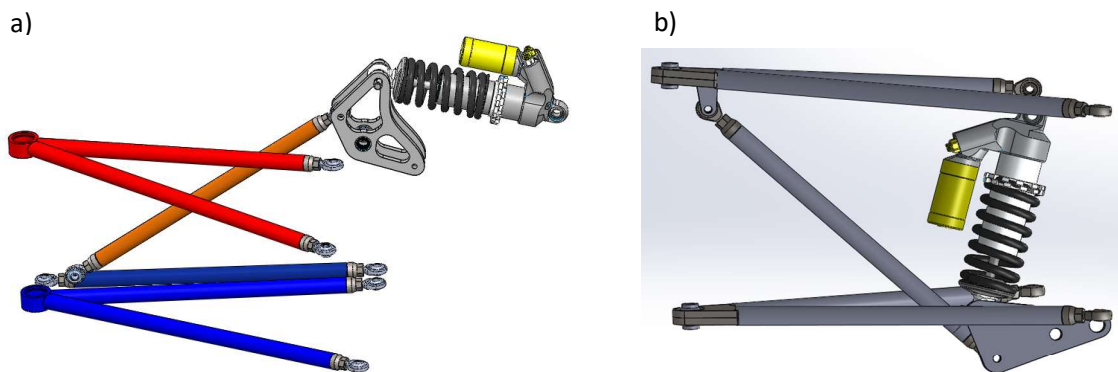


Ilustración 1-2 Suspensión doble brazo de A: a) *pushrod* b) *pullrod*

1.1.2: TIPO DE SUSPENSIÓN POR SU CONTROL

Como se ha visto en la sección anterior existen muchas configuraciones mecánicas para la suspensión de un vehículo, cada una ha surgido por una necesidad diferente y para un tipo de vehículo, debido a las múltiples condiciones a las que se encuentran sometidos. Por ejemplo, un vehículo de carga a comparación de uno de competencia estará bajo aceleraciones menores, pero con cargas normales mucho mayores, sin embargo, en ambos se promueve la maniobrabilidad y confort.

Para que la suspensión sea capaz de proveer una alta maniobrabilidad y confort (a pesar de las condiciones del camino), hoy en día se han desarrollado métodos de control robusto que permiten estimar la respuesta del sistema ante excitaciones y que son capaces de modificar dinámicamente los parámetros involucrados

en esta. En la aplicación de estos métodos los sistemas de suspensión se pueden clasificar de la siguiente forma:

Suspensión pasiva: Contiene componentes con elementos de resorte y amortiguador con constantes invariables en el tiempo (Ilustración 1-3). Los elementos sólo pueden guardar o disipar energía por una porción del ciclo de la suspensión y tampoco proveen directamente de energía [1]. La ventaja de esta suspensión es su simplicidad, al no incorporar elementos de control, su precio es menor comparado con otros sistemas robotizados. Para obtener un comportamiento diferente a una misma entrada es necesario cambiar los elementos de la suspensión, los resortes se sustituyen y algunos amortiguadores tiene la opción de modificar su constante con algunas válvulas. Visto que los sus parámetros son fijos estos son el resultado de un compromiso entre las características del terreno, la carga y el confort de los pasajeros. [6]

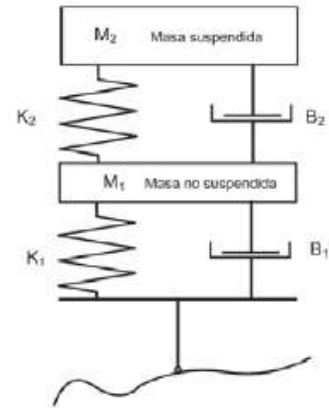


Ilustración 1-3: Suspensión pasiva. [1]

Suspensión semi-activa: Contiene resortes y/o amortiguadores cuyas propiedades pueden ser modificadas por una señal de una fuente externa [1]. El precio de estos es considerablemente mayor a sus similares constantes, al igual que al ser necesaria la incorporación de sensores el precio aumenta y con el costo de mantenimiento. En la Ilustración 1-4 se observa el diagrama de una suspensión semi-activa con únicamente el amortiguador variable. La suspensión semi activa se puede clasificar de la siguiente manera

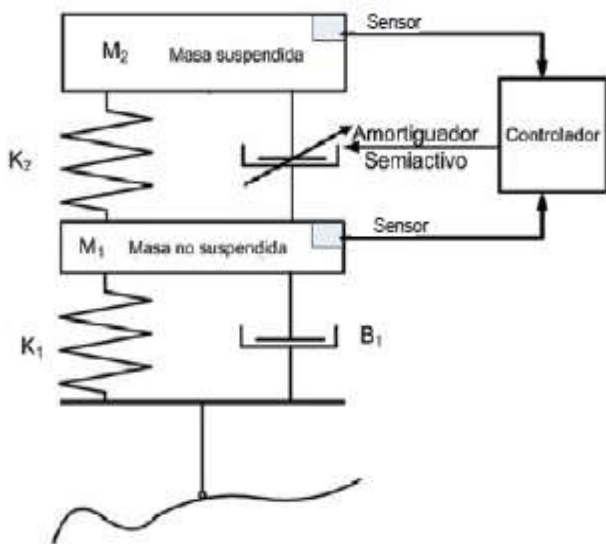


Ilustración 1-4: Suspensión semi-activa con amortiguador variable. [1]

- Lento-activo: La constante del resorte y/o amortiguador puede ser cambiada, en un rango discreto. La presión de la línea de frenos, el ángulo de dirección y el movimiento de la suspensión provocan los cambios de estas, los cambios a constantes mayores ocurren en una fracción de segundo, no obstante, la modificación a constantes menores ocurre con un retraso, este tipo de sistema también puede ser llamado, suspensión adaptativa. Este sistema es capaz de controlar los movimientos de *roll*, *pitch* y *bounce* [1].
- Bajo ancho de banda: La constante del resorte y/o amortiguador son moduladas de manera constante en respuesta a movimientos de baja frecuencia (1-3 [Hz]) de la masa suspendida [1].
- Alto ancho de banda: La constante del resorte y/o amortiguador además de ser modulada a bajas frecuencias también lo son a altas frecuencias (10-15 [Hz]) del eje [1].

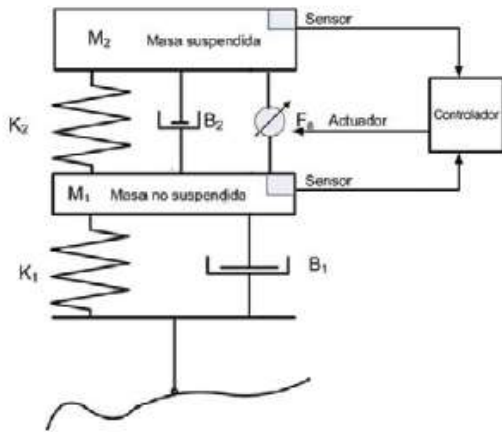


Ilustración 1-5: Suspensión activa [1]

Suspensión activa: Este tipo de suspensión posee una mayor complejidad porque incorpora actuadores que suministran energía adicional (Ilustración 1-5). Estos actuadores normalmente son de tipo hidráulicos, pero también pueden ser neumáticos o electromagnéticos [7]. Al igual que la suspensión semi activa son clasificadas en alto y bajo ancho de frecuencia [1]. Además del controlador es necesario una serie de sensores, en general son acelerómetros colocados en las esquinas del chasis y en las ruedas. En los últimos años gracias al desarrollo de la microelectrónica este tipo de suspensión se ha visto incluida en los vehículos de alta gama [7], como el Mercedes clase S.

1.1.3: MODELADO DE SUSPENSIONES.

Como se ha mencionado, el diseño y desarrollo de suspensiones y sus algoritmos de control, exige un entendimiento profundo de su dinámica (movimientos, fuerzas y pares). De ahí que sea necesario la implementación de modelos y simulaciones durante las primeras etapas del diseño y en fases posteriores se emplean bancos de pruebas para corroborar los resultados de estos.

Existen dos técnicas principales para el modelado de la suspensión, la primera es la técnica de *Bond Graph*, donde se relaciona las componentes mediante su intercambio de energía. La segunda técnica son los sistemas multi cuerpos en donde se relacionan los componentes por medio de juntas cinemáticas. A continuación, se tratarán ambas técnicas.

1.1.3.1: TÉCNICA BOND GRAPH

En *Bond Graph* cualquier componente se puede clasificar en dos grupos: parámetros concentrados y parámetros dispersos. Los primeros se suelen ocupar para el estudio de dinámica vehicular (maniobrabilidad y *ride*), mientras que los segundos se utilizan para análisis térmicos y aerodinámicos [8]. Como el objetivo de este trabajo es el diseño para obtener una mejora en el comportamiento dinámico, se enfocará únicamente en el primer tipo.

Se puede considerar a un vehículo compuesto por dos componentes principales, la masa suspendida y la no suspendida, unidas por elementos elásticos y que disipan energía, expuestos a alteraciones externas [8], la interacción entre estos puede ser descrita de distintas maneras debido a la simetría que generalmente presenta, por lo anterior es viable modelar solo una esquina o hasta todo el vehículo obteniendo modelos con diferentes grados de complejidad y que proporcionan diferente información, a continuación, se presentan los más comunes.

Modelo de un $\frac{1}{4}$ de vehículo: Un modelo altamente recomendable para los inicios del diseño, tiene 2 grados de libertad (GDL), puede modelar el movimiento vertical puro tomando en cuenta la masa suspendida y no suspendida, el resorte, el amortiguador y la llanta, la cual puede estar representada por un resorte y/o amortiguador, ver Ilustración 1-6.

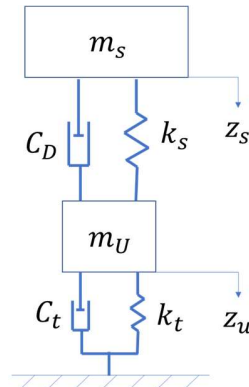


Ilustración 1-6: Modelo de 1/4 de vehículo.

Se comienza con el diagrama de cuerpo libre de las dos masas (Ilustración 1-7):

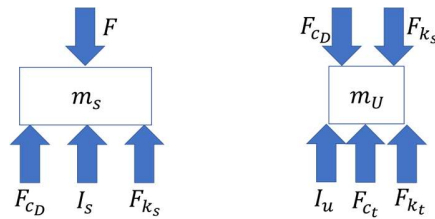


Ilustración 1-7: Diagrama de cuerpo libre de las masas de un

Las ecuaciones de equilibrio:

$$F = F_{C_D} + I_s + F_{k_s} \quad \text{Ec. 1.1}$$

$$F_{C_D} + F_{k_s} = I_u + F_{k_t} + F_{C_t} \quad \text{Ec. 1.2}$$

Las ecuaciones constitutivas de los elementos son las siguientes:

$$\begin{aligned} F_{C_D} &= C(\dot{z}_s - \dot{z}_u) & I_s &= m_s \ddot{z}_s \\ F_{k_s} &= k_s(z_s - z_u) & I_u &= m_u \ddot{z}_u \\ F_{k_t} &= k_t z_u \end{aligned}$$

Al sustituir las ecuaciones constitutivas de los elementos en las ecuaciones de equilibrio (ecuaciones **Ec. 1.1** y **Ec. 1.2**), se obtienen las siguientes expresiones:

$$\begin{aligned} F &= C_D(\dot{z}_s - \dot{z}_u) + m_s \ddot{z}_s + k_s(z_s - z_u) \\ C_D(\dot{z}_s - \dot{z}_u) + k_s(z_s - z_u) &= m_u \ddot{z}_u + k_t z_u + C_t \dot{z}_u \end{aligned}$$

Se resuelve el sistema por variables de estado, definiendo las siguientes sustituciones:

$$x_1 = z_u \quad x_2 = \dot{z}_u \quad x_3 = z_s - z_u \quad x_4 = \dot{z}_s$$

Las salidas deseadas serán las siguientes:

$$s_1 = z_u = x_1 \quad s_2 = z_s = x_3 \quad s_3 = z_s - z_u = -x_1 + x_3 \quad s_4 = \dot{z}_s - \dot{z}_u = -x_2 + x_4$$

Entonces las matrices de variables de estado, de entradas y de salidas son las siguientes respectivamente:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} z_u \\ \dot{z}_u \\ z_s - z_u \\ \dot{z}_s \end{bmatrix} \quad \mathbf{r} = [r_1] = [F] \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} z_u \\ z_s \\ z_s - z_u \\ \dot{z}_s - \dot{z}_u \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ -x_1 + x_3 \\ -x_2 + x_4 \end{bmatrix}$$

Las derivadas de las variables de estado son las siguientes:

$$\dot{x}_1 = \dot{z}_u = x_2$$

$$\begin{aligned} \dot{x}_2 = \ddot{z}_u &= [C_D(\dot{z}_s - \dot{z}_u) + k_s(z_s - z_u) - k_t z_u - C_t \dot{z}_u] \frac{1}{m_u} \\ &= [C_D(x_4 - x_2) + k_s x_3 - k_t x_1 - C_t x_2] \frac{1}{m_u} \end{aligned}$$

$$\dot{x}_3 = \dot{z}_s - \dot{z}_u = x_4 - x_2$$

$$\dot{x}_4 = \ddot{z}_s = [F - C_D(\dot{z}_s - \dot{z}_u) - k_s(z_s - z_u)] \frac{1}{m_s} = [F - C_D(x_4 - x_2) - k_s x_3] \frac{1}{m_s}$$

Se reordenan para obtener la ecuación de estados.

$$\dot{x}_1 = \dot{z}_u = 0x_1 + 1x_2 + 0x_3 + 0x_4 + 0r_1$$

$$\dot{x}_2 = \frac{-k_t}{m_u} x_1 + \frac{-(C_D + C_t)}{m_u} x_2 + \frac{k_s}{m_u} x_3 + \frac{C_D}{m_u} x_4 + 0r_1$$

$$\dot{x}_3 = 0x_1 - 1x_2 + 0x_3 + 1x_4 + 0r_1$$

$$\dot{x}_4 = 0x_1 + \frac{C_D}{m_s} x_2 + \frac{-k_s}{m_s} x_3 + \frac{-C_D}{m_s} x_4 + \frac{1}{m_s} r_1$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_t}{m_u} & -\frac{(C_D + C_t)}{m_u} & \frac{k_s}{m_u} & \frac{C_D}{m_u} \\ 0 & -1 & 0 & 1 \\ 0 & \frac{C_D}{m_s} & \frac{-k_s}{m_s} & \frac{-C_D}{m_s} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_s} \end{bmatrix} [r_1]$$

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Br}$$

Se reescriben las salidas de forma matricial para obtener la ecuación de salidas.

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [r_1]$$

$$s = \mathbf{C}x + \mathbf{D}r$$

Se aplica la transformada de Laplace a la ecuación de estados y despeja el vector de variables de estado.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}r \xrightarrow{L} s\mathbf{x}(s) - \mathbf{x}_0 = \mathbf{A}\mathbf{x}(s) + \mathbf{B}r(s)$$

$$s\mathbf{x}(s) - \mathbf{A}\mathbf{x}(s) = \mathbf{x}(0) + \mathbf{B}r(s)$$

$$\mathbf{x}(s)[s\mathbf{I} - \mathbf{A}] = \mathbf{x}(0) + \mathbf{B}r(s)$$

$$\mathbf{x}(s) = [s\mathbf{I} - \mathbf{A}]^{-1}[\mathbf{x}(0) + \mathbf{B}r(s)]$$

El vector $\mathbf{x}(0)$ representa las condiciones iniciales de las variables de estado, para esto se supondrá que al inició no existe fuerza de excitación y el sistema se encuentra estático, por lo que el diagrama de cuerpo libre se simplifica como se observa en la Ilustración 1-8:

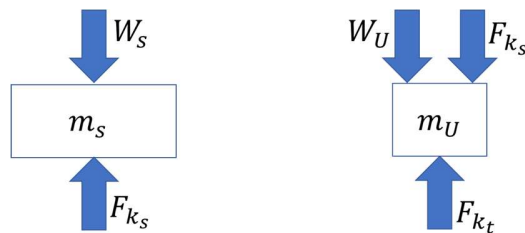


Ilustración 1-8 Diagrama de cuerpo libre sin excitación

Las ecuaciones de equilibrio son las siguientes

$$F_{k_s0} = W_s$$

$$k_s(z_{s0} - z_{u0}) = m_s g$$

$$(z_{s0} - z_{u0}) = \frac{m_s g}{k_s} = x_{30}$$

$$W_u + F_{k_s0} = F_{k_t0}$$

$$W_u + W_s = F_{k_t0}$$

$$m_u g + m_s g = k_t z_{u0}$$

$$z_{u0} = \frac{m_u g + m_s g}{k_t} = x_{10}$$

x_{20} y x_{40} son cero porque al estar el sistema estático $\dot{\mathbf{z}}_u = \mathbf{0}$ y $\dot{\mathbf{z}}_s = \mathbf{0}$.

$$\mathbf{x}(0) = \begin{bmatrix} \frac{m_u g + m_s g}{k_t} \\ 0 \\ \frac{m_s g}{k_s} \\ 0 \end{bmatrix}$$

Como se puede observar en la Ilustración 1-6, únicamente se modela un movimiento vertical, sin embargo, durante el desplazamiento del vehículo en una curva se presenta un cambio de orientación (giro); típicamente conocido como alabeo (o *roll*). Éste involucra al menos dos esquinas del vehículo de manera que se necesita el modelo de medio vehículo para incorporar este efecto al sistema.

Modelo de ½ vehículo o modelo de bicicleta: En términos generales son dos modelos de ¼ unidos por la masa suspendida con 4 GDL, la ventaja que este modelo es la representación del fenómeno de *pitch* o de alabeo (*roll*), dependiendo si se ve el vehículo de forma frontal o lateral, ver Ilustración 1-9.

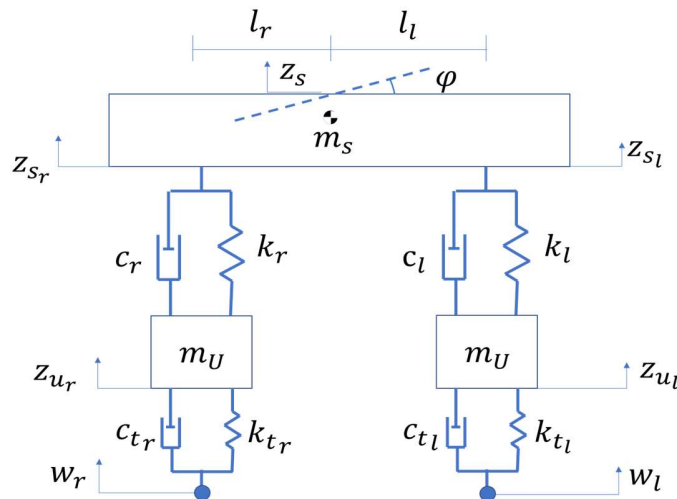


Ilustración 1-9: Modelo de ½ vehículo. [8]

Las ecuaciones de equilibrio son las siguientes:

$$F_{m_s} = F_{k_r} + F_{c_r} + F_{k_l} + F_{c_l}$$

$$F_{m_{ur}} = F_{k_{t_r}} + F_{c_{t_r}} - F_{k_r} - F_{c_r} - F_{ARB}$$

$$F_{m_{ul}} = F_{k_{t_l}} + F_{c_{t_l}} - F_{k_l} - F_{c_l} + F_{ARB}$$

$$M_{m_s} = l_r(F_{k_r} + F_{c_r}) + l_l(F_{k_l} + F_{c_l})$$

Las ecuaciones de elemento son las siguientes:

$$F_{k_r} = k_r(z_{u_r} - z_{s_r})$$

$$F_{c_r} = c_r(\dot{z}_{u_r} - \dot{z}_{s_r})$$

$$F_{k_{t_r}} = k_{t_r}(w_r - z_{u_r})$$

$$F_{c_{t_r}} = c_{t_r}(\dot{w}_r - \dot{z}_{u_r})$$

$$F_{m_{ur}} = m_{u_r} \ddot{z}_{u_r}$$

$$F_{m_s} = m_s \ddot{z}_s$$

$$F_{k_l} = k_l(z_{u_l} - z_{s_l})$$

$$F_{c_l} = c_l(\dot{z}_{u_l} - \dot{z}_{s_l})$$

$$F_{k_{t_l}} = k_{t_l}(w_l - z_{u_l})$$

$$F_{c_{t_l}} = c_{t_l}(\dot{w}_l - \dot{z}_{u_l})$$

$$F_{m_{ul}} = m_{u_l} \ddot{z}_{u_l}$$

$$M_{m_s} = I_x \ddot{\phi}$$

Las siguientes expresiones se deducen del diagrama considerando un ángulo de roll pequeño:

$$l z_s = l_r z_{s_l} + l_l z_{s_r}$$

$$\varphi = \frac{z_{s_l} - z_{s_r}}{l}$$

Se resuelve por variables de estado definiendo las siguientes sustituciones:

$$\mathbf{x} = \begin{bmatrix} z_{s_r} \\ z_{s_l} \\ z_{u_r} \\ z_{u_l} \\ \dot{z}_{s_r} \\ \dot{z}_{s_l} \\ \dot{z}_{u_r} \\ \dot{z}_{u_l} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} w_r \\ w_l \\ \dot{w}_r \\ \dot{w}_l \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} \quad \mathbf{s} = \begin{bmatrix} z_{u_r} - z_{s_r} \\ w_r - z_{u_r} \\ z_{u_l} - z_{s_l} \\ w_l - z_{u_l} \\ \varphi \end{bmatrix} = \begin{bmatrix} x_3 - x_1 \\ r_1 - x_3 \\ x_4 - x_2 \\ r_2 - x_4 \\ \frac{x_2 - x_1}{l} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix}$$

Se derivan las variables de estado y se expresa en función de las entradas y las variables de estado:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{r}$$

Donde:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ k_r \left(\frac{l_r^2}{I_x} - \frac{1}{m_s} \right) & k_l \left(\frac{l_r l_l}{I_x} - \frac{1}{m_s} \right) & k_r \left(\frac{-l_r^2}{I_x} + \frac{1}{m_s} \right) & k_l \left(\frac{-l_r l_l}{I_x} + \frac{1}{m_s} \right) & c_r \left(\frac{l_r^2}{I_x} - \frac{1}{m_s} \right) & c_l \left(\frac{l_r l_l}{I_x} - \frac{1}{m_s} \right) & c_r \left(\frac{-l_r^2}{I_x} + \frac{1}{m_s} \right) & c_l \left(\frac{-l_r l_l}{I_x} + \frac{1}{m_s} \right) \\ k_r \left(\frac{-l_r l_l}{I_x} - \frac{1}{m_s} \right) & k_l \left(\frac{-l_l^2}{I_x} - \frac{1}{m_s} \right) & k_r \left(\frac{l_r l_l}{I_x} + \frac{1}{m_s} \right) & k_l \left(\frac{-l_l^2}{I_x} + \frac{1}{m_s} \right) & c_r \left(\frac{-l_r l_l}{I_x} - \frac{1}{m_s} \right) & c_l \left(\frac{-l_l^2}{I_x} - \frac{1}{m_s} \right) & c_r \left(\frac{l_r l_l}{I_x} + \frac{1}{m_s} \right) & c_l \left(\frac{l_l^2}{I_x} + \frac{1}{m_s} \right) \\ \frac{k_r l + k_{ARB}}{l m_{U_r}} & -\frac{k_{ARB}}{m_{U_r} l} & -\frac{k_{tr} + k_r}{m_{U_r}} & 0 & \frac{c_r}{m_{U_r}} & 0 & -\frac{c_{tr} + c_r}{m_{U_r}} & 0 \\ -\frac{k_{ARB}}{m_{U_l} l} & \frac{k_l l + k_{ARB}}{l m_{U_l}} & 0 & -\frac{k_{tl} + k_l}{m_{U_r}} & 0 & \frac{c_l}{m_{U_l}} & 0 & -\frac{c_{tl} + c_l}{m_{U_l}} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{k_{tr}}{m_{U_r}} & 0 & \frac{c_{tr}}{m_{U_r}} & 0 \\ 0 & \frac{k_{tl}}{m_{U_l}} & 0 & \frac{c_{tl}}{m_{U_l}} \end{bmatrix}$$

Para la ecuación de salida se describe de forma matricial:

$$s = Cx + Dr$$

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -\frac{1}{l} & \frac{1}{l} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

A pesar de que el modelo anterior puede representar el *roll* o *pitch*, no lo puede hacer simultáneamente para esto se utiliza el siguiente modelo.

Modelo de vehículo completo Este modelo tiene 7 GDL, permite la representación de *roll*, *pitch* y movimiento vertical, así como características antivuelco y antiderrape. Como se ve en la Ilustración 1-10 son 4 modelos de ¼ unidos por la masa suspendida, en este modelo se puede incluir los efectos de las barras de torsión delantera y trasera, al igual que fuerzas de entrada diferentes en cada llanta.

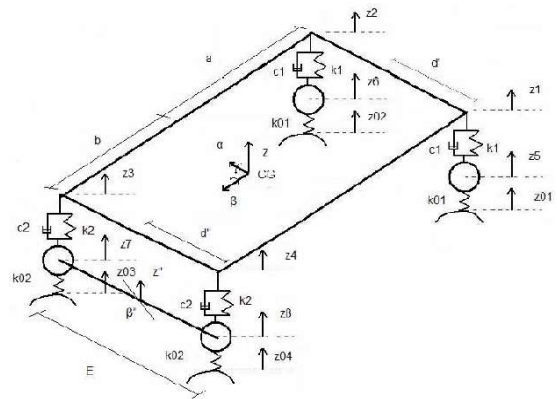


Ilustración 1-10: Modelado de vehículo completo. [8]

El modelo anterior es capaz de incorporar el movimiento de *roll* y *pitch* al mismo tiempo, sin embargo al inicio del diseño no se poseen muchos de estos valores con precisión, en consecuencia, el modelo de vehículo completo no es aplicable, de igual manera tampoco se ve involucrada la velocidad de *yaw* en el modelo, característica primordial para el análisis de un vehículo girando, cualidad que si logra el modelo siguiente, este fue presentado por Hans B. Pacejka [9], este a diferencia de los anteriores no se enfoca en las vibraciones de las masas. Posee 3 grados de libertad, como variables de movimiento define: el ángulo de *roll*, la velocidad lateral y la de *yaw*, mientras que la velocidad longitudinal se considera constante. El coche gira respecto al eje virtual de *roll*, el cual es creado a partir de las alturas de los centros de *roll* de ambos ejes. En representación de la rigidez a *roll* delantera y trasera se coloca un resorte torsional en cada eje, el cual resulta de

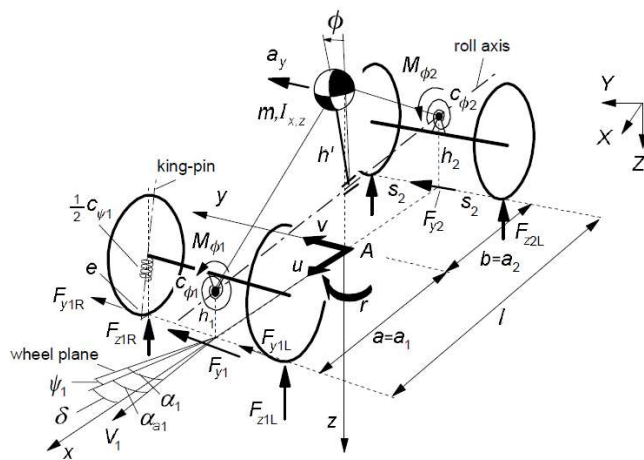


Ilustración 1-11: Modelo de coche completo de Pacejka

las constantes del resorte de la suspensión y la barra de torsión. Se considera una masa puntual en el centro de gravedad.

1.1.3.2: SISTEMAS DE MULTI CUERPOS

Son sistemas que consisten en eslabones rígidos o flexibles unidos entre ellos por juntas cinemáticas, las cuales restringen el movimiento relativo entre los cuerpos. Siguiendo el concepto anterior una suspensión de doble brazo de A se puede modelar como se observa en la Ilustración 1-12.

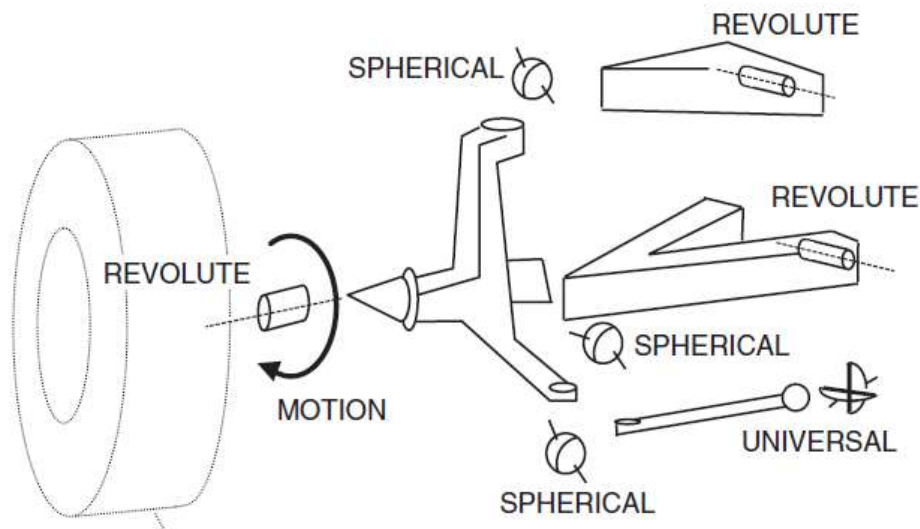


Ilustración 1-12: Modelado por multi cuerpos de una suspensión doble brazo de A. [10]

En la imagen se observa que entre los brazos de suspensión y el chasis existe una sola junta de revolución, reemplazando los dos cojinetes que en realidad son utilizados en la mayoría de los casos. Esto se realiza para evitar sobre restringir el modelo, mediante la introducción de 5 restricciones redundantes [10].

Las matrices de rotación y translación son combinadas para describir el movimiento de cada uno de los eslabones y con una ecuación de lazo, el cual puede ser abierto o cerrado, se puede describir el movimiento del sistema completo.

Como se ha visto con todos los modelos anteriores el aumento de variables en estos no solo introduce efectos no lineales, sino que también incrementan los valores cuantitativos que se deben de poseer al igual que el conocimiento de dinámica vehicular.

1.1.4: TRABAJOS DE OPTIMIZACIÓN DE SUSPENSIONES

Debido a que los objetivos de comodidad y maniobrabilidad se contraponen se ha buscado la forma de obtener los valores de los elementos de las suspensiones que permitan llegar a un justo medio entre ambos, por lo cual se ha estudiado la suspensión por medio de la técnica de Bond Graph y algoritmos

genéticos, los cuales son una metodología de optimización que a diferencia de la mayoría trabaja con un conjunto de soluciones, en lugar de una sola, para luego aplicar operadores que las modifican con estrategias que permiten su mejora progresiva. A continuación, se exponen algunos trabajos que utilizan esta técnica para la optimización de la suspensión.

En el trabajo de R. Alkhatib et. al. [11] busca optimizar por medio de algoritmos genéticos la relación entre la media cuadrática de la aceleración absoluta y la media cuadrática del desplazamiento relativo modelado la suspensión de la forma anterior, pero omitiendo el amortiguamiento de la llanta. Para la selección de los padres se ocupó una probabilidad proporcional a su aptitud mientras que para la cruce se usó una máscara creada aleatoriamente con tasa de mutación del 10%.

Un trabajo posterior realizado por G. Verros et.al. [12] realiza una optimización de varios modelos de $\frac{1}{4}$ de vehículo, en estos se suponen amortiguadores lineales y bi-lineales, también es considerado el salto de la llanta, lo cual provee al modelo de un alto grado de no linealidad. Se busca minimizar la aceleración absoluta de la masa suspendida, la distancia relativa entre la masa suspendida y la no suspendida, y la fuerza entre el neumático y el pavimento. Para la optimización aplica un método pseudo multi objetivo lo que quiere decir que realiza una sumatoria de los objetivos multiplicados por ciertos valores para escalar la importancia del parámetro.

En el trabajo publicado en el 2007 por Ö.Gündoğdu [13] utiliza el modelo de $\frac{1}{4}$ de vehículo con dos grados de libertad en serie de otros dos sistemas de masa-resorte-amortiguador, uno representando el asiento y el otro el tórax y pelvis, los elementos de resorte y amortiguamiento utilizados son lineales. Se utilizan algoritmos genéticos para optimizar: la constante de amortiguamiento y de resorte de la suspensión y del asiento, sus objetivos son el factor de cresta (*crest factor*) y el valor de dosis de vibración VDV (*vibration dose value*). Respecto al algoritmo genético fue utilizado el incluido en MATLAB® con una longitud del genoma de 30, una probabilidad de cruce y de mutación de 0.001 y 0.002 respectivamente.

En el artículo de Ming-Yuan Shing et. al. [14] propone optimizar una suspensión activa representada con un $\frac{1}{4}$ de vehículo con un método de optimización híbrido que une algoritmos de sistema inmune artificial con reglas difusas con algoritmos genéticos, dicha unión se realiza para disminuir los altos tiempos de cómputo necesarios para un algoritmo genético y evitar la tendencia de los algoritmos inmunes de ser atrapados en un óptimo local. Dentro de este trabajo se busca optimizar el control de una suspensión activa al mejorar la comodidad durante el viaje sin tener que sacrificar la maniobrabilidad. El algoritmo es capaz de calcular la concentración de afinidad entre cada grupo de cromosomas para luego ser guardados en la memoria del sistema o en el sistema inhibidor. Los cromosomas de los anticuerpos dentro del sistema inhibidor son eliminados mientras que los que se encuentran en la memoria del sistema son replicados. Para evitar caer en algún óptimo local se utiliza la concentración de cromosomas en el pozo genético, en caso de ser muy similar indica que el algoritmo está atrapado en una solución óptima local, para salir se aumenta el porcentaje de mutación. Dentro de este trabajo el algoritmo propuesto es comparado contra: un algoritmo genético clásico, un algoritmo genético difuso y un algoritmo de optimización evolutiva por enjambre de partículas, dentro de dicho trabajo se concluye que el algoritmo genético inmunológico artificial y reglas difusas es capaz de obtener soluciones óptimas con un menor número de generaciones.

En el trabajo de Nagarkar et. al. [15] se estudia un modelo no lineal con modelo de llantas cuadrática y constante de resorte cúbica, el modelo de suspensión utilizado fue de $\frac{1}{4}$ de vehículo con el marco y cojín del asiento, además al modelo fue agregada la representación de un pasajero con 4 GDL. En dicho trabajo se realiza un análisis comparativo entre un algoritmo multiobjetivo por enjambre de partículas con distancia de apiñamiento o MOPSO-CD por sus siglas en inglés contra un algoritmo genético de ordenamiento no dominado II o NSGA-II por sus siglas en inglés. Este trabajo concluye que MOPSO-CD converge más rápido que NSGA-II.

Continuando con el estudio anterior de suspensiones no lineales Nagarkat et. al. [16] agregan las estrategias de control: PID (siglas en inglés de proporcional, integral y derivativo) y LQR (siglas en inglés de regulador cuadrático lineal). Este trabajo y el realizado por Ming-Yuan Shing et. al. [14] son una muestra de la tendencia hacia la suspensión activa.

Debido a que la posición del conductor respecto a los ejes también influye en la aceleración experimentada por este, en el trabajo de N. Nariman-Zadeh et. al. [17] se utiliza un modelo de suspensión pasiva de $\frac{1}{2}$ vehículo con 5 grados de libertad, 4 provistos por la suspensión y uno por un modelo de masa resorte amortiguador sobre la masa suspendida representando el asiento. Sus parámetros por optimizar son las constantes de resorte y amortiguamiento del asiento y suspensión, así como la posición relativa entre el centro de gravedad y el asiento.

Las funciones por minimizar son: la aceleración del asiento, la velocidad de las llantas y su desplazamiento relativo en ambos ejes. Fue explorada la optimización ocupando diferentes pares de objetivos y uno con los 5 objetivos, utilizando el concepto de frontera de Pareto para distinguir las soluciones óptimas. En este trabajo se introduce la función de eliminación ϵ al algoritmo genético de multiobjetivo con diversidad uniforme, conocido como MUGA por sus siglas en inglés (*Multi-objective Uniform-diversity Genetic Algorithm*), la función propuesta realiza la eliminación de los individuos con gran similitud en el espacio de objetivos y de variables asociadas simultáneamente.

Después de la eliminación de los individuos con dicha característica son generados individuos de manera aleatoria para conservar un tamaño de población constante, además en caso de que el individuo con el ranking 1 se mantenga constante n generaciones continuas también son introducidos individuos aleatorios a la población. Los parámetros ocupados en el algoritmo genético fueron: una población de 80 individuos con una probabilidad de cruce de 0.9 y de mutación de 0.1 y la condición de término del algoritmo fue 240 generaciones. Al utilizar un modelo de $\frac{1}{2}$ vehículo se puede calcular el movimiento angular, el cual también influye en la comodidad, de ahí que el artículo publicado por Vladimir Goga et. al. [18] además de minimizar la aceleración vertical de la masa suspendida y los desplazamientos relativos recurre a la aceleración angular. Este artículo no considera el asiento, por lo que solo cuenta con 4 GDL, con un resorte en representación de la llanta. Los parámetros por optimizar son: las constantes de amortiguamiento y de resorte de ambos ejes, mientras que la función objetivo se genera a partir de la sumatoria de la multiplicación de constantes por las siguientes funciones: la aceleración vertical y angular de la masa suspendida y desplazamiento relativo entre esta y la no suspendida. Para el algoritmo genético fueron ocupados 100 individuos, los 7 mejores de cada generación fueron seleccionados para la cruce,

sobre la mutación esta fue adaptativa y aleatoria. La condición de término del algoritmo fue 250 generaciones.

En todos los trabajos anteriores existen dos formas de que el algoritmo genético termine su proceso, la primera es alcanzar el número de generaciones previamente establecido, la otra es parar manualmente el algoritmo, esto puede causar utilizar más tiempo de cómputo que el estrictamente necesario o en un caso peor puede terminar el proceso prematuramente lo cual significa que no se obtiene una opción óptima, de ahí que Kittipong Boonlong propone en su artículo [19] un operador para determinar si el algoritmo ha convergido y así poder asegurar una solución óptima reduciendo el tiempo necesario, este solo comparando diferentes set de soluciones con una función propuesta en el trabajo. Utiliza una modificación del algoritmo genético con objetivos comprimidos, COGA (*Compressed-Objective Genetic Algorithm*) por sus siglas em inglés. El algoritmo original de COGA calcula directamente la clasificación de los individuos no dominante por medio de la cantidad de los objetivos en los que el individuo es superior o inferior causando así el agrupamiento de los individuos alrededor de los mejores valores individuales en lugar de espesarse por la frontera de Pareto. Para solucionar esto COGA-II utiliza diferentes pesos para la clasificación dependiendo de cada objetivo. Para comprobar la eficiencia de este método fueron utilizados el mismo modelo, parámetros y objetivos que los utilizados por Nariman-Zadeh [17], concluyendo que las generaciones necesarias para encontrar una solución óptima dependen de las características de los objetivos y del problema, esto se deduce que de los 5 diferentes conjuntos de objetivos con los que se trata para 4 de ellos es necesarios menos generaciones que las propuestas por Nariman-Zadeh pero en uno es necesario más de 3 veces las cantidad de generaciones propuestas.

Con el avanzar de los años los modelos de suspensión y/o algoritmos utilizados en los trabajos de investigación son más complejos provocando mayor cercanía con el fenómeno real y disminución en los tiempos de cómputo respectivamente, sin embargo, se han ido aumentando las funciones por optimizar sin tener una métrica de que tan necesario es este proceso por lo cual Papaioannou et. al. en el 2018 realizaron un estudio [20] comparativo para una suspensión pasiva utilizando un modelo de $\frac{1}{2}$ de vehículo más un modelo de asiento-pasajero con más de 6 GDL. Fueron seleccionadas 6 funciones a optimizar que afectan la sensación de confort y de manejabilidad del vehículo, después fue aplicado un algoritmo genético para la optimización del modelo, por los resultados de dicho trabajo se concluye que la optimización de menos funciones es más eficiente que si se trata de optimizar las 6 funciones al mismo tiempo. Como se puede observar en los trabajos anteriores es muy común la optimización de modelos de parámetros concentrados, enfatizando el confort de los pasajeros, sin embargo, en ninguno de los artículos anteriores se tratan las dimensiones del vehículo o geometría de la suspensión como parámetros para modificar el comportamiento dinámico del coche, el cual se ve ampliamente afectado por estos. El trabajo de Smith et. al. del 2004 presenta un acercamiento a este enfoque realizando la optimización de la geometría de una suspensión de doble brazo de A, para la disminución del cambio de *camber* en *bump* y *roll*. Se utiliza un algoritmo genético a 50 generaciones con una población de 1000 individuos.

1.2: OPTIMIZACIÓN

La definición clásica de optimización es la búsqueda de uno o varios puntos óptimos, es decir, inmejorables. No obstante, esta definición no toma en cuenta factores como los recursos disponibles o el tiempo

necesario para encontrar la solución, por consiguiente, se ha propuesto que el fin más importante de la optimización sea el mejoramiento con un adecuado uso de los recursos [21].

Los problemas para ser optimizados deben tener la siguiente forma [17]:

Vector de variables de diseño	$X^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in \mathcal{R}^n$
Vector de funciones objetivo	$F(X) = [f_1(X), f_2(X), \dots, f_k(X)]^T \in \mathcal{R}^k$
Sujeto a m restricciones de desigualdad	$g_i(X) \leq 0, \quad i = 1 \text{ hasta } m$
Y p restricciones de igualdad	$h_j(X) = 0, \quad j = 1 \text{ hasta } p$

Buscar la mejor solución a un problema siempre ha sido una inquietud humana por lo que se han desarrollado diferentes técnicas para lograrlo, a continuación, se profundiza en lo anterior.

1.2.1: MÉTODOS DE OPTIMIZACIÓN

Los métodos clásicos para optimizar se pueden separar en 2 grandes grupos: los que se basan en el cálculo enumerativo y los heurísticos.

El primer grupo, se divide en dos clases: indirectos y directos. Los métodos indirectos están basados en encontrar el punto donde el gradiente de la o las funciones objetivo sean cero, mientras que los métodos directos buscan el óptimo local saltando en la función con una dirección relacionada con el gradiente local [21]. Este método supone la existencia de la derivada de la función de interés lo que implica que debe de ser continua y sus derivadas laterales coincidir. Sin embargo, lo anterior hace que este método no sea apto para muchas funciones. Al ser un método de búsquedas locales resulta poco robusto [21].

En los métodos enumerativos se buscan valores uno a la vez de la función objetivo dentro de un espacio de búsqueda finito o uno infinito discretizado. Muchos espacios de búsqueda son demasiado grandes como para buscar un valor a la vez llevando a este método a carecer de eficiencia, por lo cual podemos decir que tampoco es robusto [21].

El segundo grupo son los heurísticos. En estos tipos de optimización, se propone un valor de forma aleatoria, luego se valúa en la función objetivo y se repite este proceso, se va guardando el valor más alto. El problema con este tipo de métodos es el mismo que con los métodos enumerativos ya que al aumentar el espacio de búsqueda, el tiempo necesario para la optimización aumenta de sobre manera, resultando en un algoritmo poco eficiente llevando así a una falta de robustez [21].

En los últimos años ha florecido la inteligencia artificial proveyendo de herramientas de búsqueda local y global, por otro lado, con los métodos de heurística se puede llegar a generar algoritmos eficientes y robustos. Una las ramas de la inteligencia artificial son los algoritmos evolutivos, siendo los algoritmos genéticos los más populares de éstos, a continuación, se explicarán más ampliamente.

1.2.1.1: ALGORITMOS GENÉTICOS.

Los algoritmos genéticos, o GA por sus siglas en inglés, es una metodología de optimización estocástico, que a diferencia de otros algoritmos manipula una población de soluciones en lugar de una sola [20].

Como muchos avances tecnológicos, los algoritmos genéticos están basados en la naturaleza. Este método basa su estructura en la teoría de la evolución de las especies, tal como es descrita por Charles Darwin.

En términos biológicos, en una población de seres vivos un nuevo individuo es creado por el cruce de información genética de dos padres. La información genética es guardada en el ADN; son los **genes** los encargados de codificar una o más de las características del individuo, el conjunto de estas es llamado **fenotipo**. Un conjunto de genes es llamado **cromosoma**. Un gen puede tomar diferentes valores. Estos son llamados **alelo** y el conjunto de todos los genes con su valor correspondiente es el **genotipo**.

En términos de programación se comienza con una población de individuos generada aleatoriamente. La información genética es guardada en una cadena, normalmente binaria de longitud constante. Ésta es el equivalente del **genotipo**, cada bit es un **gen**. Si es binaria solo puede tener uno de dos **alelos**, cero o uno. La cadena puede ser dividida por **cromosomas**, en cada uno se encuentra cifrada una de las características del resultado, el conjunto de características decodificadas es el equivalente del **fenotipo**.

En términos biológicos un individuo para lograr reproducirse tiene que sobrevivir a diferentes amenazas, entre ellas está el clima, la accesibilidad a la comida y los depredadores. Según indica Darwin, los individuos más aptos logran reproducirse y de esta manera transmitir sus genes a la siguiente generación, además, es por medio de la mutación que el genotipo sufre un cambio aleatorio; esta es la forma de la naturaleza de probar la variación de una característica.

En términos de programación para calcular la aptitud de cada individuo se ocupa la función de evaluación y es así como se encuentran a los más aptos, los cuales son modificados por medio de dos operadores: mutación y cruza. Durante la mutación el valor de los genes puede cambiar o no, este cambio es determinado de forma aleatoria, mientras que en una cruza se toma dos individuos y se intercambia segmentos de su cadena. Ambos operadores serán expuestos de forma más clara en secciones sucesivas. Los nuevos individuos reemplazarán a los individuos con menos aptitud para volver a comenzar el proceso, hasta llegar a la condición de término.

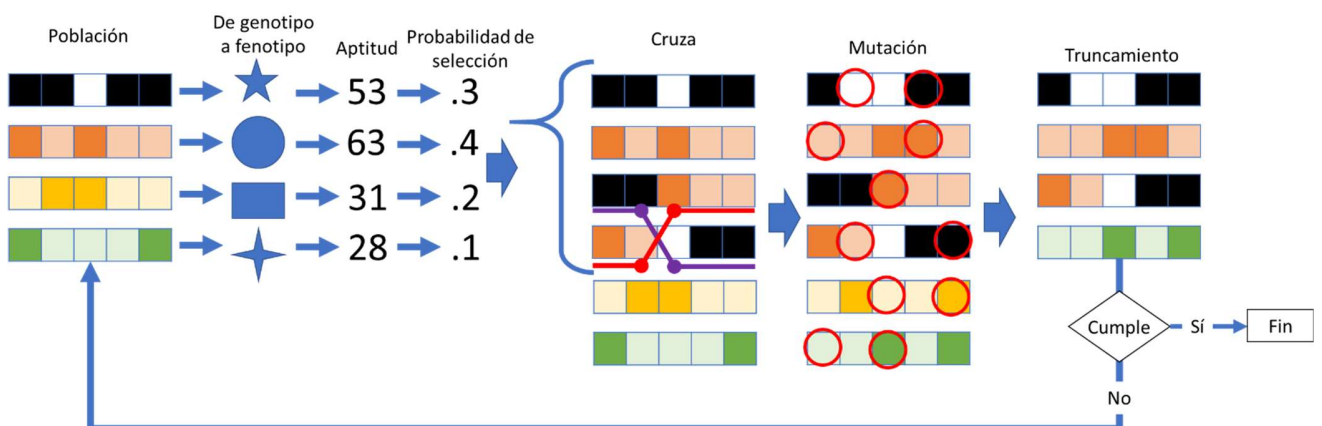


Ilustración 1-13: Diagrama de flujo general de un algoritmo genético

En la Ilustración 1-13 se puede observar el procedimiento general de un algoritmo genético, con el estudio de la inteligencia artificial se han desarrollado cada vez algoritmos más eficientes y complejos que permiten reducir los tiempos de cómputo y evitar caer en mínimos locales.

A continuación, se describirán cada una de las partes que componen a los algoritmos genéticos.

1.2.1.1.1: CODIFICACIÓN

Los algoritmos genéticos han demostrado un buen rendimiento para todo tipo de problemas, por ejemplo: encontrar el camino más rápido entre dos lugares, el orden de procesos para el tiempo mínimo en la fabricación, hallar la cantidad de dientes de una transmisión para que ésta sea del menor tamaño posible, entre muchos otros. La solución de cada uno de estos tiene una forma diferente por lo que deben de ser codificadas de distintas maneras para ser procesados por un programa computacional de forma óptima. Los tipos de codificación más usados son:

Binaria: es la forma más utilizada. Los cromosomas están representados con números binarios. Las ventajas de este método radican en la existencia de muchas posibilidades de cromosomas con sólo dos alelos 0 o 1 [22], pero resulta poco natural esta forma de representar las soluciones. Para este tipo de codificación se puede ocupar la cruce de un punto, n puntos y uniforme, respecto a la mutación se puede ocupar el método de volteo [23].

No binarios: son similares a la codificación binaria, pero basados en otros sistemas de numeración como el octal y el hexadecimal. Estos ocupan los mismos operadores que la codificación binaria.

Permutación: esta forma es la más adecuada para soluciones que implican el ordenamiento [22]. Cada gen representa números de una secuencia. Si se trata de optimizar un recorrido cada gen representaría un sitio y su posición el orden de visita. Para este tipo de codificación es necesario operadores especiales, en el caso de cruce se puede ocupar cruce parcialmente mapeada, cruce cíclica y cruce de orden, respecto a la mutación la más utilizada es la inversión [23].

Real: Es también llamado codificación de valor directo [24] pues cada gen está compuesto de una cadena de valores, los cuales pueden ser números enteros, reales, caracteres o incluso objetos [23]. Esta técnica es usada cuando se requieren valores complicados. Además, este método resulta más práctico a comparación del binario, si es necesario codificar valores con decimales, pero se tienen que ocupar métodos especiales para la cruce y la mutación.

Árbol: es la forma más adecuada para evolucionar programas o expresiones pues cada cromosoma es un árbol de algún objeto, función o comando de un lenguaje de programación [22]. El protocolo de separación localizador/identificador o LISP por sus siglas en inglés es utilizado para esta codificación puesto que ayuda para el análisis. De esta forma, se puede llevar a cabo la mutación y cruce de una manera fácil [23].

Desordenado: Si en un problema una combinación particular de bits para algunos genes separados ampliamente es una construcción de bloques será difícil mantener ese bloque de construcción en los tipos de codificación anteriores, para resolver esto se propone que la posición y valor de cada gen

estén codificados en una cadena, así se reduce la probabilidad de interrupción de bloques de construcción importantes [24].

Existen más tipos de codificaciones pero son utilizados en problemas específicos como la codificación de graficas usando vectores o basadas en gramática, usada para los problemas de coloreado con la mínima cantidad de colores y optimización de redes autómatas respectivamente [24].

1.2.1.1.2: APTITUD

La forma de calcular la aptitud depende de cada problema, pues las funciones objetivo son diferentes, en forma y número. Los problemas en los que existe más de una función objetivo son llamados multiobjetivo, de estos existen dos formas principales de manejarlos. La primera es hacer una sumatoria ponderada de los objetivos, lo cual convierte en problema multiobjetivo en un solo objetivo, a esta clase de problemas se les llama pseudo multiobjetivo. Su ventaja es la reducción de la complejidad del algoritmo necesario, pero su desventaja es la considerable influencia de los valores de ponderación en el resultado final, la selección de estos es hecha por el diseñador/programador, por lo que el resultado óptimo en realidad será subjetivo. La otra forma de manejar un problema multiobjetivo es utilizando el concepto de dominación de Pareto, en esto en lugar de obtener un solo resultado óptimo es encontrado un conjunto de estos, llamados la frontera de Pareto.

La idea atrás de una frontera de Pareto es muy simple: existen dos clases de individuos, los no dominados y los dominados, los algoritmos genéticos se adaptan a este método debido a que utilizan una población. Para que un individuo \bar{x} domine a uno \bar{y} debe de cumplir las dos siguientes condiciones

1. \bar{x} no es peor que \bar{y} en todos los objetivos
2. \bar{x} es mejor que \bar{y} en por lo menos un objetivo

Si \bar{x} las cumple se dice que \bar{x} domina a \bar{y} o $\bar{x} < \bar{y}$. Y si un objetivo no es dominado por ningún otro este es parte de la frontera de Pareto. A partir de la dominancia de Pareto se han desarrollado diferentes formas de ranking y ponderación de la aptitud, para seleccionar a los padres con los que se generará la siguiente generación. A continuación, se exponen algunos de estos métodos.

1.2.1.1.3: SELECCIÓN DE PADRES

Siguiendo la teoría de la evolución de las especies de Darwin, existe una selección de los mejores individuos para pasar sus genes a generaciones posteriores, por lo cual primero se deben de seleccionar los individuos más aptos, entre más alta sea la aptitud de un individuo más alta es su probabilidad de ser seleccionado. Se introduce el concepto de presión de selección como el grado en el que los mejores individuos son favorecidos, una presión alta provoca una velocidad de convergencia alta, si esta es demasiado alta puede llevar a una convergencia temprana y por lo tanto a un subóptimo. Por otro lado, si la presión de selección es demasiado baja, aumenta innecesariamente el tiempo de convergencia. Además de una adecuada presión de selección es necesario asegurar la preservación de la diversidad de la población para evitar caer en mínimos locales [25].

Se distinguen dos tipos de esquemas de selección, selección proporcional y basada en un ranking. El primer tipo selecciona los individuos respecto a la proporción de su aptitud y la de los otros individuos,

mientras en el segundo esquema no se utiliza la aptitud pura, en su lugar se utiliza la posición que ocupa el individuo en el ranking de aptitudes [25].

Ruleta: Es la selección aleatoria de los individuos a través de una ruleta con ranuras proporcionales a la aptitud de los individuos. Este método tiene una desventaja cuando las aptitudes difieren mucho entre ellas, pues un individuo toma gran parte de la ruleta haciéndolo fácilmente seleccionable, en consecuencia, es altamente probable que solo copias de este individuo llenen el conjunto de padres.

Aleatoria: Como su nombre lo indica los individuos son seleccionados de forma aleatoria, su ventaja es su fácil programación, sin embargo, no provee de ningún tipo de presión de selección.

Torneo: Son seleccionados n individuos para el torneo y el que tenga la aptitud o ranking más alto gana y pasa al conjunto para la cruce [25].

Boltzmann: Este método varía continuamente la presión de selección, se comienza con una presión baja para permitir una búsqueda global y se va aumentando gradualmente para acercarse a los individuos más aptos y manteniendo una diversidad adecuada. Se utiliza una función logarítmica para la modificación de la presión porque ha demostrado buena convergencia sin quedar encerrado en mínimos locales [25].

Muestra universal estocástica: Se crea una línea con los segmentos contiguos de todos los individuos, la longitud de cada individuo es proporcional a su aptitud, considerando n como el número de individuos a ser seleccionados, se generan n punteros a una distancia entre ellos igual a un número aleatorio en el rango de $[0, 1/n]$, y finalmente los individuos seleccionados serán los señalados por dichos indicadores. En la Ilustración 1-14 se muestra un ejemplo donde la distancia entre los marcadores aleatoria es igual a " a " y los individuos 1, 2 y 3 son seleccionados.

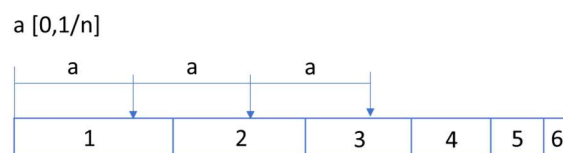


Ilustración 1-14: Ejemplo de una muestra universal estocástica

1.2.1.1.4: CRUZA

La eficiencia y eficacia del algoritmo genético depende en gran parte de la forma en que se realice la cruce, pues si la cruce y/o la mutación es poca el algoritmo puede quedarse en óptimos locales y si es demasiada puede que los individuos salten de una región a otra sin encontrar óptimos, por lo que entender y luego seleccionar la forma de cruce es fundamental. A continuación, se describen los tipos más utilizados de cruce.

1 punto de cruce: Los individuos seleccionados son emparejados aleatoriamente, luego el genotipo es separado en una posición k , este número debe ser entero y puede ser constante o generado aleatoriamente entre 1 y la longitud máxima del cromosoma. El nuevo individuo es creado por la primer parte del primer padre y la segunda del segundo y el otro individuo es creado por las partes restantes.

En la Ilustración 1-15 se presenta un ejemplo de este tipo de cruce con $k = 2$ y codificación binaria simbolizada por las casillas claras y oscuras.

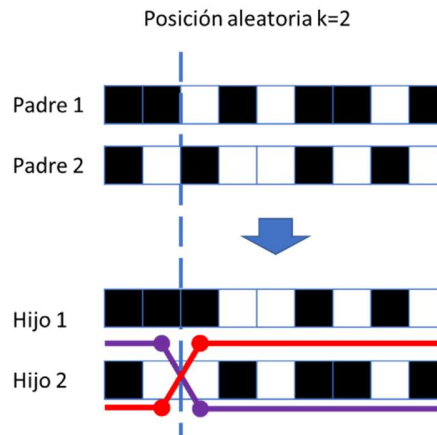


Ilustración 1-15: Ejemplo de 1 punto de cruce con $k = 2$

N puntos de cruce: Este proceso de cruce se realiza de forma similar al de un solo punto de cruce solo que en este caso existe más de un punto de cruce, el cual también puede ser fijo o generado aleatoriamente. Los nuevos individuos son creados uniendo la primera parte del primer padre, y luego, la siguiente parte de del segundo padre y repitiendo este proceso con todas las partes. Se muestra un ejemplo en la Ilustración 1-16, donde al igual que el anterior se utiliza la codificación binaria.

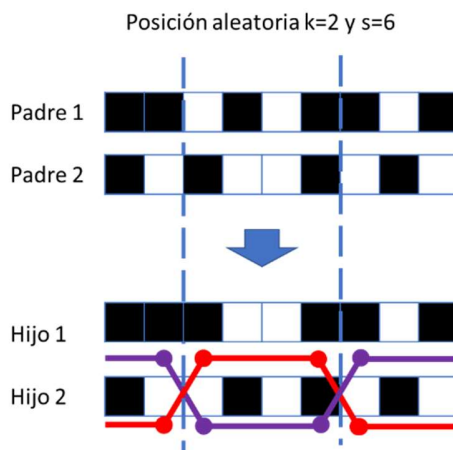


Ilustración 1-16: Ejemplo de dos puntos de cruce en la posición 2 y 6

Cruza uniforme: Se genera un vector de números aleatorios con una longitud igual a la de la cadena que codifica a un individuo, se revisa en cada posición del vector aleatorio si el número generado es mayor a la probabilidad de cruce previamente definida, de tal modo que, en caso positivo se realiza el intercambio de ese gen, de lo contrario, permanece el gen original, se continúa de esta forma en todas las posiciones.

Como se puede observar la Ilustración 1-17 presenta un ejemplo de este tipo de cruce, si el número aleatorio fue mayor o no a la probabilidad de cruce es representado con un recuadro blanco o negro respectivamente, en este ejemplo solo los genes en las posiciones 4, 6,7 y 9 de izquierda a derecha cambiaron.

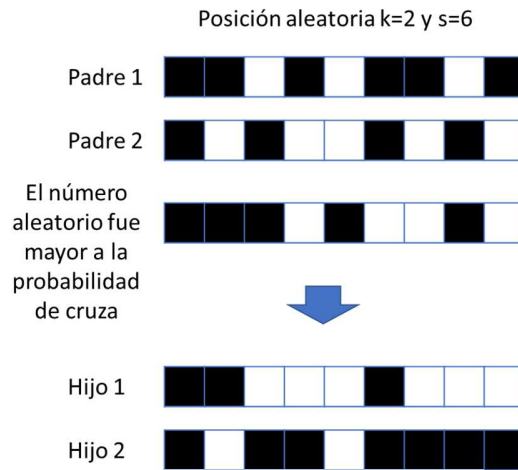


Ilustración 1-17: Ejemplo de cruce uniforme

Cruza fija: Este tipo de cruce es mejor aplicado para la codificación real, pues en lugar de solo intercambiar los genes se realiza una ponderación entre los valores de cada padre con un número aleatorio generado entre cero y uno empleando la siguiente fórmula para calcular el valor del nuevo gen. [22]

$$x_i^1 = r_i x_{1,i} + (1 - r_i) x_{2,i}$$

En la Ilustración 1-18 se muestra un ejemplo de la cruce fija de uno de los hijos.

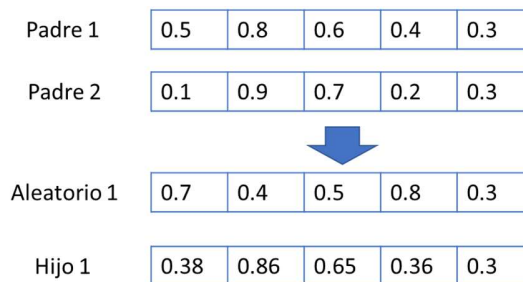


Ilustración 1-18: Ejemplo de cruce fija

Cruza de orden: Esta forma de cruce pretende mantener parcialmente el orden relativo, la posición absoluta y los individuos adyacentes. Este método se utiliza para la codificación de permutación. Primero son generados dos puntos aleatorios de cruce, el centro del primer padre es copiado al primer hijo, luego se copian los elementos no utilizados en el orden del segundo padre comenzando desde el segundo punto de cruce [26].

Un ejemplo de la cruce se puede observar en la Ilustración 1-19, como puede notarse, se deben generar dos puntos de cruce, el centro del primer padre pasa al segundo hijo y viceversa, a continuación se anotan los elementos del padre uno en el hijo uno que aún no se encuentren en dicho hijo y se realiza la acción equivalente en el hijo dos, en el ejemplo para el hijo uno son los elementos **J, K, L, B, C, E, F,** y **G**, mientras que para el hijo dos son **b, j, c, l, k, a, d** e **i**, estos elementos son anotados desde el segundo punto de cruce en el orden en el que aparecen en el padre [26].

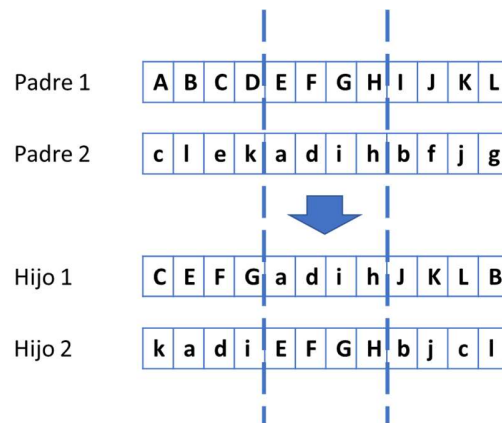


Ilustración 1-19: Cruza de orden

1.2.1.1.5: MUTACIÓN

Mientras que la cruce se utiliza para la exploración local, la mutación sirve para explorar todo el espacio de resultados y así encontrar máximos globales.

Al igual que con el operador de cruce existen muchos métodos, pero la idea general es la selección y modificación de un gen de forma aleatoria, el procedimiento para realizarlo depende del tipo de codificación, a continuación, se exponen algunos de los más comunes.

Inserción: Se ocupa con codificación de permutación. Primero se escogen de forma aleatoria dos alelos, luego coloca el segundo alelo a continuación del primero y se desplaza el resto [27].

Inversión: También es utilizado para la codificación de permutación, pero en este son seleccionados de forma aleatoria dos alelos y a continuación se invierte la cadena que quedó entre ellos, cuando se realiza esto, se mantiene casi toda la información adyacente al solo romper dos uniones [27].

Intercambio: Es para la codificación de permutación. Primero se seleccionan dos alelos al azar y luego se intercambian las posiciones de estos [27].

Cambio de cara: Es generalmente usada con la codificación binaria, un gen o más son aleatoriamente seleccionado y son cambiados de 0 a 1 o de 1 a 0.

Deslizamiento: Para la codificación real, un gen aleatorio es seleccionado y cambiado su valor en un rango predefinido.

No uniforme: Es una mutación para codificación real. Para aplicar este operador es necesario conocer la generación actual y la generación máxima, se ocupan las siguientes ecuaciones [28].

$$c'_i = \begin{cases} c_i + \Delta(t, b_i - c_i) & \text{si } \tau = 0 \\ c_i - \Delta(t, c_i - a_i) & \text{si } \tau = 1 \end{cases} \quad \Delta(t, y) = y \left(1 - r \left(1 - \frac{t}{g_{max}} \right)^b \right)$$

Donde:

- τ es un número con valor de cero o uno
- t generación actual
- g_{max} generación máxima
- r es valor aleatorio entre 0 y 1
- b es una constante escogida por el programador, que determina el grado de dependencia del número de generaciones
- c_i es el valor actual del gen i
- a_i es el valor máximo del gen i
- b_i es el valor mínimo del gen i

La probabilidad de que $\Delta(t, y)$ regrese cero aumenta al avanzar las generaciones, esta propiedad permite proveer de una búsqueda uniforme al inicio y local en etapas posteriores [28].

Los operadores de mutación anteriores son ocupados solos o en las estrategias de mutación para ayudar a evitar la convergencia temprana. A continuación, se explican algunas de estas.

Algoritmo Genético de Mutación Dinámica: También llamado DMGA por sus siglas en inglés, ocupa varios operadores de mutación simultáneamente, cada uno de estos inicia con una tasa de mutación igual, esta es $1/n$ de la tasa total de mutación para el programa, donde n es la cantidad de operadores que se emplean. Cada operador de mutación es aplicado según su probabilidad y luego la aptitud de los hijos es evaluada. La tasa de los operadores que hayan provisto con los hijos de mayor aptitud promedio es aumentada, mientras que se disminuye la de los que suministraron hijos de una menor aptitud promedio. A pesar de proporcionar automáticamente de un operador de mutación apropiado tiene la desventaja de la posibilidad de una convergencia prematura [29].

Algoritmo Genético de Mutación de Esquema: También llamado SMGA por sus siglas en inglés, se revisan los individuos con una mejor aptitud en busca de un esquema es decir, cromosomas con los mismos alelos, los cuales van al esquema quasi-óptimo H , los individuos en este esquema tendrán una probabilidad de mutación baja en comparación de la probabilidad aplicada a los elementos que no estén en el esquema H , por lo que se va a tender a mantener con menos modificaciones a los individuos similares y con mayor aptitud [29].

Algoritmo Genético de hiper-mutación: También llamado HMGA por sus siglas en inglés. En los problemas de optimización dinámica es necesario mantener una alta diversidad en la población para asegurar una adaptabilidad rápida. El objetivo de la hiper-mutación es aumentar la probabilidad de

mutación a un valor predefinido cuando el ambiente cambie e ir bajando este gradualmente o incluso de una generación a otra [29].

Debido a los mecanismos previamente expuestos existe la posibilidad que los mejores individuos no sean seleccionados, o en caso de que lo sean, la información contenida en ellos sea destruida por la cruce o mutación, es por esto que se introduce en algunos algoritmos el mecanismo de **elitismo**, este propone que los mejores individuos sean copiados a la siguiente generación, mejorando así el rendimiento del algoritmo [25].

1.2.1.1.6: CONDICIÓN DE TÉRMINO

Las formas más comunes de detener a los algoritmos genéticos son: que alcance un número de generaciones máximas propuestas por el diseñador o haya corrido por un tiempo predefinido, no obstante, en ambas formas se depende de la ecuanimidad del diseñador a causa de que si las generaciones o tiempo máximo propuesto es demasiado se desperdiciará tiempo de cómputo, al contrario, si es muy poco se llegará a un subóptimo. De ahí que se hayan propuesto nuevas formas de detener el algoritmo una de ellas es, si después de n generaciones la aptitud de la población no ha sufrido cambios se detiene este, otra es cuando se llega a una población que domine a más individuos de una n anterior que los individuos que la dominan de esta se para el proceso, entre otras.

Existen algoritmos genéticos que combinan condiciones de término como las dos últimas, con métodos para asignar un ranking o una aptitud ponderada que también dependa de la cercanía del individuo con el resto, esto para promover la diversidad de la población, con técnicas especiales para el truncamiento de la población, una de estas es COGA-II, la cual se explica más ampliamente a continuación.

1.2.1.2: COGA-II

Existe una metodología llamada COGA, siglas en inglés de algoritmo genético de objetivos comprimidos, que permite cambiar un problema con M-objetivos a uno con solo dos: puntuación ganadora (o WS por sus siglas en inglés, *winning score*) e índice de vecindad (*vicinity index*). Este fue mejorado con COGA-II que utiliza solo el primer objetivo, pero con diferentes pesos en cada objetivo k [19]. El procedimiento general para COGA-II se expone en la Ilustración 1-20.

Al igual que para su predecesor COGA-II utiliza la siguiente función para el cálculo de WS.

$$WS_i = \sum_{j=1}^N \omega_{ij}$$

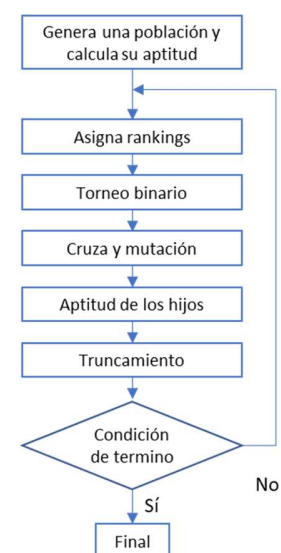


Ilustración 1-20: Diagrama de flujo de COGA-II [19]

Mientras que en COGA ω_{ij} se calcula como $\omega_{ij} = \mathbf{sup}_{ij} + \mathbf{inf}_{ij}$ en COGA-II se utilizan las siguientes funciones [19]:

$$\omega_{ij} = \sum_{k=1}^M W F_k q_{ijk}$$

$$q_{ijk} = \begin{cases} 1, & \text{si en el objetivo } k \text{ } i \text{ es mejor que } j \\ -1, & \text{si en el objetivo } k \text{ } i \text{ es peor que } j \\ 0, & \text{si son iguales en el objetivo } k \end{cases}$$

$$W F_k = \frac{V_k}{\sum_{l=1}^M V_l}$$

$$V_k = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \rho_{ijk}$$

$$\rho_{ijk} = \begin{cases} \frac{\mathbf{sup}_{ij} + \mathbf{inf}_{ij}}{2\mathbf{sup}_{ij}}, & \text{si } i \text{ es mejor que } j \\ \frac{\mathbf{sup}_{ij} + \mathbf{inf}_{ij}}{2\mathbf{inf}_{ij}}, & \text{si } i \text{ es peor que } j \\ 1, & \text{si } i \text{ es igual a } j \end{cases}$$

Donde \mathbf{sup}_{ij} es el número de objetivos que son superiores en el individuo i a los correspondientes del objetivo j , \mathbf{inf}_{ij} es la cantidad de objetivos en los que el individuo i es inferior al individuo j , M es el número de individuos no dominados y N el número de objetivos [19].

COGA-II utiliza rankings para ordenar los individuos de acuerdo con su aptitud, este proceso es realizado de la siguiente forma [19]:

1. Separar individuos dominados y no dominados.
2. Calcular la WS de los individuos no dominados.
3. Buscar individuos extremos no dominados y ordenarlos por WS. Asignarles ranking comenzando desde 1 hasta E , donde E es el número de individuos extremos.
4. Ordenar el resto de los individuos no dominados de forma descendente por WS y asignarles ranking comenzando de $E+1$.
5. El ranking de los elementos dominados es M más la cantidad de individuos que lo dominan.

Debido a que la asignación de WS no asegura la diversidad de la población se utiliza el siguiente operador para seleccionar Q individuos entre la población de padres e hijos y conformar la siguiente generación, este proceso es llamado truncamiento y se realiza con la siguiente serie de pasos [19]:

1. Encontrar los individuos extremos de los no-dominados, pasarlos a la siguiente generación y eliminarlos de la actual.

2. Calcular la distancia Euclidiana, d_i^{RL} , entre los individuos de la generación actual y su vecino más cercano en la siguiente generación.
3. Seleccionar $Q-L$ individuos con las mayores d_i^{RL} , donde Q es la cantidad de individuos deseados en la siguiente generación y L la cantidad que se encuentran en esta. Seleccionar el candidato con mayor WS de dicho subconjunto de individuos.
4. Regresar al paso 3 hasta que L se igual a Q .

Después de realizar este procedimiento el algoritmo de COGA-II revisa la convergencia.

Se dice que el algoritmo ha convergido si para la generación actual, A , y una anterior aleatoria, B [19].

$$C(A, B) \leq C(B, A)$$

Donde:

$$C(X', X'') = \frac{|\{x'' \in X''; \exists x' \in X': x' < x''\}|}{|X''|}$$

Si $C(X', X'') = 1$ significa que todas las soluciones de X'' son dominadas o iguales a las de X'

Capítulo 2: CASO DE ESTUDIO PRÁCTICO

El equipo de UNAM Motorsports participa en la competencia FSAE desde el año 2010, para lo cual diseña un prototipo de vehículo tipo fórmula, el cual debe de cumplir el reglamento de la competencia publicado por SAE. El equipo es dividido en subgrupos, llamados sistemas, uno de estos es el de suspensión, encargado del diseño y manufactura de la suspensión y dirección del prototipo.

En este trabajo se propone encontrar el valor de los parámetros necesarios para el diseño de una suspensión doble brazo de A mediante el desarrollo de una herramienta aplicando la teoría de algoritmos genéticos. Para dicho fin, se toma como caso de estudio el sistema de suspensión del prototipo de FSAE de la UNAM del año 2017

2.1: PROCESO DE DISEÑO DE UNA SUSPENSIÓN PARA FSAE

La Ilustración 2-1 muestra el proceso de diseño general utilizado en el 2017:

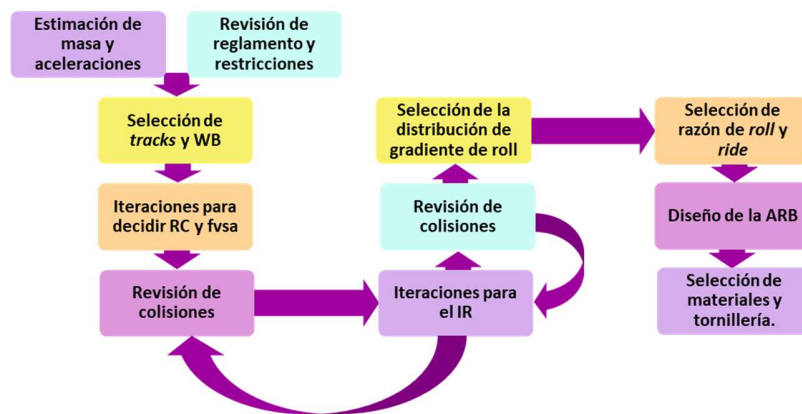


Ilustración 2-1: Proceso de diseño del 2017

La primera etapa en el desarrollo de la especificación del vehículo consiste en la aproximación del peso de este. Esto permite a los equipos responsables de los subsistemas del vehículo iniciar su trabajo de diseño. Uno de los requisitos que la escudería se ha planteado de forma anual, consiste en la disminución en el peso del vehículo. Para nuestro caso, el peso del vehículo sin incluir al piloto es de 194 [kg].

Posteriormente, se toman como referencia los mejores tiempos de las competencias de FSAE de un año anterior, en nuestro caso se tomaron los tiempos (datos) de la competencia del año 2016¹ para fijar los tiempos objetivo: 4.7 [s] para el skidpad y 4.0 [s] para la aceleración. Con estos datos, se procede a realizar la aproximación de las aceleraciones que permitirán alcanzar los tiempos objetivo.

Se realiza el cálculo de la aceleración lateral durante la prueba de *skidpad* (para más detalles ver Anexo 1). En esta prueba el vehículo debe desplazarse en una curva (pista) con un radio medio de 9.125 [m]:

$$a_y = \left(\frac{2\pi}{t}\right)^2 R = \left(\frac{2\pi}{4.7}\right)^2 9.125 = 16.308 \left[\frac{m}{s^2}\right] = 1.66 [g's]$$

¹ En *skidpad* 4.714 [s] por Universität Stuttgart y en aceleración, 4.065 [s] por Michigan State University

Igualmente, para el cálculo de la aceleración longitudinal en la prueba de aceleración (para más detalle ver Anexo 1), se utilizó una recta de 75 [m]:

$$a_x = \frac{2 \times \text{distancia recorrida}}{t^2} = \frac{2 \times 75}{4^2} = 9.375 \left[\frac{m}{s^2} \right] = 0.956 [g's]$$

De forma simultánea, se revisa el reglamento completo del año en curso debido a que puede existir una regla que comprometa el desarrollo de algún sistema que no necesariamente se incluya en las especificaciones que estén incluidas en los apartados de cada uno. De igual manera, es necesario revisar el reglamento de todas las competencias en las que se pretenda participar debido a que es posible que se publiquen modificaciones al reglamento original de FSAE. A continuación, se enlistan las reglas que afectan el diseño de la suspensión:

- Llantas descubiertas. (T1.1.2)
- *Wheelbase* mínimo de 1525 [mm]. (T1.7.1)
- La proporción entre el *track* de menor tamaño y *track* más grande sea mayor o igual a 0.75. (T1.8.1)
- Viaje mínimo de llanta de 2 [in], 1 [in] de extensión y 1 [in] de compresión. (T1.3.1)
- Un intento serio de una suspensión delantera y trasera operacionales. (T1.3.1)

Al iniciar la temporada se decidieron los siguientes componentes del vehículo:

- Rin 10: Durante el final de la temporada 2016 se tomó la decisión de utilizar rin 10 para el prototipo del año 2017, mientras que los rines 13 serían utilizados para el primer prototipo eléctrico del equipo.
- Suspensión doble brazo de A: debido a que era la que se había utilizado previamente y es la más común durante la competencia.
- Construcción de chasis tubular: Se mantuvo este tipo de chasis, debido a que un monocasco representaba un extenso desafío en diferentes áreas.
- Öhlins® TTX25 MkII: Se utilizaron estos amortiguadores debido a que ya se poseían y son diseñados para la competencia.
- Resortes: Las constantes de resorte que el equipo posee son 150, 250, 350, 450 y 600 [lb/in].

Durante la etapa inicial de diseño del sistema de suspensión se deben de **seleccionar *tracks* y *wheelbase***. Dichas distancias son delimitadas con la transferencia de carga que ocasionarían y la experiencia de los miembros del sistema.

Para el cálculo de transferencia de carga se ocuparon las siguientes fórmulas, suponiendo una altura del centro de gravedad, CG, de 26 [cm]:

$$\Delta F_{zLateral} = \frac{a_y \cdot m \cdot h_{cg}}{tr}$$

$$\Delta F_{zLongitudinal} = \frac{a_x \cdot m \cdot h_{cg}}{l}$$

Después de un día de ponderación entre dimensiones más cortas y largas se decidió 64 [in] de *wheelbase* 48 y 49 [in] de *track* delantero y trasero correspondientemente.

Debido a que durante el año 2017 se utilizó por primera vez un rin de tamaño 10, fue necesario seleccionar el tipo de llantas debido a que éstas tienen una gran influencia en el diseño del vehículo. Éstas son la conexión entre la suspensión y el pavimento, por lo tanto, su selección es determinante para el comportamiento y desempeño del vehículo. Debido a lo anterior se seleccionó los modelos para rin 10 que están disponibles para envío a México analizados por el TTC[®] (*Tire Test Consortium*[®]), los cuales son los siguientes:

- Hoosier 18.0 x 6.0 10 R25B (Item 43101).
- Hoosier 6.0 / 18.0 10 LCO C2000 (Item 41100).
- Hoosier 18.0 x 7.5 10 R25B (Item 43105).

Para cada llanta se modeló su comportamiento de fuerza lateral contra *slip angle* a cada fuerza normal con ayuda del programa desarrollado por el equipo, éste utiliza un modelo semi-empírico comúnmente llamado fórmula mágica (*magic formula*), desarrollado por la universidad de TU-Delf y la empresa automotriz Volvo [2]. De acuerdo con los datos obtenidos la llanta Hoosier 18.0 x 6.0 10 R25B es capaz de proveer más fuerza lateral que las demás, por lo que esta fue la seleccionada.

El siguiente paso fue la selección de los puntos de los brazos de suspensión, para esto fueron realizadas varias iteraciones con ayuda de Win-Geo[®], en este proceso **se busca alterar principalmente la altura del centro de roll y el fvsa**, se trató de lograr el centro del *roll* más alto con el fvsa más largo posible. Se proporcionaron los puntos encontrados al sistema de chasis y USM pues de estos parten para la geometría de su diseño. Luego se revisaron colisiones con los componentes cercanos a los brazos de suspensión, para esto se realiza un borrador del CAD.

Con los puntos de los brazos definidos se prosiguió con los puntos en del balancín y amortiguador, para esto es necesario primero decidir el tipo de suspensión doble brazo de A, esta puede ser *push* o *pull*. El prototipo del año 2016 era tipo *pull*, permitiendo un centro de gravedad de la suspensión bajo, sin embargo, debido al empaquetamiento por la reducción del tamaño del rin se tuvo que cambiar a *push*, para poder colocar los amortiguadores en la parte superior del vehículo, para evitar que causarían drag los amortiguadores delanteros se alinearon con el “eje x” y así poder colocar el cofre encima de estos. Mientras que los amortiguadores traseros se colocaron alineados con el “eje y”, pues esto facilita la manufactura. Para la decisión de dichos puntos se realizaron iteraciones en Adams Car[®], se revisaba cada conjunto de puntos propuestos con un análisis de viaje de llantas opuestas (*opposite wheel travel*) con el objetivo de encontrar un IR similar a la unidad, por lo que principalmente se modificaron los puntos en el balancín. Al encontrar un conjunto de puntos que satisfacía a los integrantes del sistema se realizaba un borrador del CAD de la suspensión y este era enviado a los demás sistemas para revisar colisiones, este tuvo que ser repetir en varias ocasiones. Las gráficas de los análisis de los puntos finales se encuentran en el Anexo 2, al igual que los puntos de ambas suspensiones.

La suspensión delantera cuenta con un IR de 1.057 y una ganancia de *camber* en todo el recorrido de la suspensión menor a medio grado, mientras que la suspensión trasera tiene 0.995 y 0.04 respectivamente. Transcurrieron aproximadamente 3 meses entre el inicio del diseño y la selección final de los puntos.

Para la selección de razones de *roll* y *ride* los integrantes del sistema se apoyaron con el programa hecho por Aldo Espejel [2] que realiza los cálculos que se describen a continuación.

Con la altura de los centros de *roll* definidas se puede aproximar mejor la transferencia de carga y encontrar la distribución de gradiente de *roll* que permita una distribución de carga uniforme, este es definido como:

$$q = \frac{c_{\varphi F}}{c_{\varphi F} + c_{\varphi R}}$$

Se seleccionó con ayuda de las gráficas de transferencia de carga generadas por el programa el valor de q que hiciera la distribución de fuerza normal lo más uniforme, este valor fue 0.5124.

El primer paso para la obtención de las razones de *roll* y *ride* es calcular el **gradiente de roll**, la definición de este es los grados de *roll* por aceleración lateral en g's. Para vehículo del 2017 se utilizó 1.55 grados de *roll*, por lo que:

$$\nabla roll = \frac{-m g h'}{c_{\varphi F} + c_{\varphi R}} = \frac{\varphi}{\frac{a_y}{g}} = \frac{1.55}{1.66} = 0.937 \left[\frac{^\circ}{g} \right]$$

Si se despeja $c_{\varphi F} + c_{\varphi R}$ de la ecuación anterior es posible calcular la rigidez total que necesita el vehículo para lograr el gradiente deseado.

$$c_{\varphi F} + c_{\varphi R} = \frac{-m g h'}{\nabla roll}$$

Con el valor de la distribución de gradiente de *roll* previamente obtenido se puede calcular la rigidez necesaria en cada eje.

Para cada resorte se calcula la razón de *ride*, frecuencia natural de masa suspendida y no suspendida con las siguientes fórmulas:

$$K_R = \frac{K_W k_t}{K_W + k_t}$$

$$\omega_S = \frac{1}{2\pi} \sqrt{\frac{K_R}{m_s}}, \quad [Hz]$$

$$\omega_U = \frac{1}{2\pi} \sqrt{\frac{K_W + k_t}{m_u}}, \quad [Hz]$$

Donde: $K_W = IR^2 k_s$

Se escoge los resortes que causen frecuencias naturales dentro de los rangos recomendados presentados por Aldo Espejel [2].

Con los resortes seleccionados se puede calcular la rigidez necesaria de la barra de torsión para lograr el gradiente de *roll* deseado con las siguientes fórmulas:

$$c_{\varphi F_{ARB}} = \frac{c_{\varphi F} k_t \frac{t_F^2}{2}}{k_t \frac{t_F^2}{2} - c_{\varphi F}} - K_{W_F} \frac{t_F^2}{2}$$

$$c_{\varphi R_{ARB}} = \frac{c_{\varphi R} k_t \frac{t_R^2}{2}}{k_t \frac{t_R^2}{2} - c_{\varphi R}} - K_{W_R} \frac{t_R^2}{2}$$

Después se decide su material y diámetro, proceso realizado con el mismo programa, los resultados se muestran en el Anexo 2.

Con todo lo anterior se puede realizar el CAD y se revisan colisiones con los otros sistemas, en caso de existir se modifican esos componentes, el resultado final se muestra en la Ilustración 2-2.

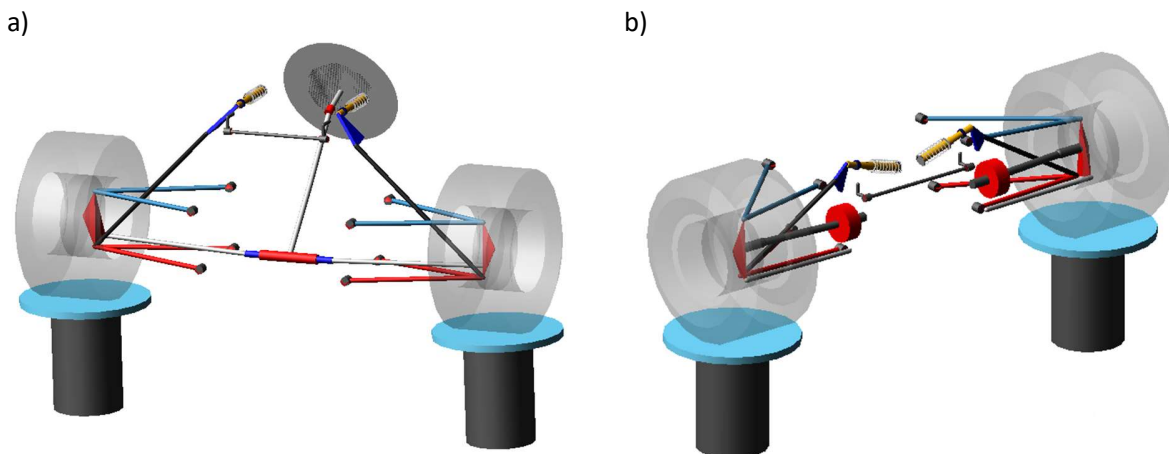


Ilustración 2-2: modelo de suspensión a) delantera, b) trasera

Como se vio en el proceso anterior existen parámetros que son propuestos con base en los conocimientos de los diseñadores y reglamentos a los cuales el vehículo estará adscrito, sin embargo esta forma resulta subjetiva, además basarse en los resultados de años anteriores para la conservación o modificación de estos sin una métrica exacta del impacto de cada uno en el comportamiento final es ambiguo, pues una mejora en los tiempos de pista puede ser o no debido a la modificación de la suspensión. Lo ideal para medir el impacto de estos sería mantener los demás sistemas del vehículo constantes y solo variar estos y por ende los consecuentes, como la proposición anterior resulta irreal el estudio a través de modelos matemáticos es ideal para mejorar los parámetros. Por lo que se propondrá un proceso que buscará su obtención de forma objetiva y rápida por medio de algoritmos genéticos.

Para disminuir la subjetividad del proceso de diseño primero deben ser identificados los parámetros propuestos por los diseñadores, a partir de los cuales parte el resto del diseño, estos serán llamados parámetros primitivos, para esto se tomará como base el proceso previamente expuesto. Visto que durante dicho proceso la selección de *wheelbase* y *tracks* fue realizada basándose en la experiencia, estos formarán parte de los parámetros primitivos.

En lugar de iterar los puntos de los brazos de la suspensión, estos pueden ser definidos casi en su totalidad con ayuda de los centros de *roll* y la distancia *fvsa* siguiendo el proceso expuesto en el Anexo 3, los centros de *roll* serán parte de los parámetros primitivos mientras que el *fvsa* será calculado con la ganancia de *camber* permitida obtenida por los datos de las llantas.

El último de los parámetros propuesto son los grados de *roll* que gira el vehículo a una aceleración lateral, esto para el cálculo del gradiente de *roll*, por lo que en realidad se está proponiendo este indirectamente, el cual también puede ser calculado con el peso del vehículo, la distancia del centro de gravedad al eje de *roll* y la rigidez torsional de cada eje, el primero de estos parámetros estará fijo, el segundo es fácilmente calculable si la altura de los centros de *roll* y la del centro de gravedad es conocida, el último tendría que ser propuesto por lo que será un parámetro primitivo, lo cual no solo permite el cálculo del gradiente de *roll*, sino también su distribución. En resumen, los parámetros primitivos son el *wheelbase*, *tracks*, alturas de centro de *roll* y rigideces a *roll*.

2.2: PROCESO PROPUESTO

Para comenzar el diseño primero es necesario identificar las necesidades, a continuación, se muestran con su respectiva interpretación en orden de importancia:

#	Necesidad del equipo	Interpretación
1	Seguro	Los componentes deberían ser lo suficientemente robustos para resistir los esfuerzos presentes durante la competencia y las pruebas previas, sin arriesgar la integridad de algún observador o del piloto, además de proveer a éste de la maniobrabilidad necesaria para recorrer la pista sin percance alguno.
2	Pasar inspección técnica (<i>tech inspection</i>).	Cumplir con el reglamento en curso de la o las competencias a participar: dimensiones y comportamiento.
3	Realizar todas las pruebas dinámicas, ver Anexo 1.	Pueda recorrer todas las pistas descritas en el reglamento sin cometer alguna penalización.
4	Fácil instalación y servicio	Los espacios alrededor de la tornillería deberían permitir un fácil acceso a las herramientas, al igual de proveer una visibilidad suficiente para la instalación y desinstalación del sistema.
5	Fácil manufactura	Debería ser posible la manufactura por los integrantes del equipo y dentro de las instalaciones de la UNAM.

Los puntos 2 y 3 se encuentran fuertemente ligados al reglamento de la competencia, por lo que el siguiente paso es la revisión de este, entre el reglamento de FSAE 2017 y 2019 no ocurrieron grandes modificaciones en las reglas respectivas a suspensión, a continuación, se enlista la pequeña modificación a las reglas presentadas anteriormente:

- Se solicita casi el mismo viaje vertical de las llantas, 50 [mm], pero ya no especifica el viaje de compresión o extensión. (V.3.1.1)

Además de las reglas anteriores también es necesario resaltar todas las restricciones de dimensiones en las que el diseño de la suspensión pueda tener algún efecto debido a que posteriormente se obtendrán los rangos de las dimensiones del vehículo.

- El espacio de la cabina debe permitir el paso longitudinalmente de una plantilla desde la columna de dirección hasta 10 [cm] antes de los pedales. El ancho y alto de la plantilla es de 350[mm] (T.3.2.1).
- Debe haber máximo 700 [mm] entre la parte más frontal de las llantas delanteras y la parte más lejana de cualquier elemento aerodinámico delantero (T.9.3.1).
- Debe haber máximo 250 [mm] entre la parte más posterior de las llantas traseras y la parte más lejana de cualquier elemento aerodinámico trasero (T.9.4.1).
- El ancho mínimo de la pista en las pruebas dinámicas es de 3 [m], es en *skipad* (D.9.1.1).
- El diámetro mínimo exterior de una curva en las pruebas dinámicas es en *autocross* y es de 9 [m] (D.10.1.1).

A la par de la revisión del reglamento también pueden ser identificadas las restricciones por los componentes que se utilizarán en el diseño de la suspensión, para fin de este trabajo se utilizarán las mismas que en el 2017.

Respecto a la masa del vehículo su reducción resulta beneficiosa, pero es un proceso complejo que puede llevar varias temporadas, en consecuencia, no se puede fijar un valor deseado arbitrariamente, por lo que a consideración personal la mejor forma de obtener esta es a partir de la estimación de reducción de peso de cada sistema, sin embargo, esta debe de ser realizada con gran reserva pues afecta más una suposición menor que una mayor de este valor.

A diferencia del proceso anterior una estimación de las aceleraciones no es necesaria, pero se recomienda realizarla como referencia para todo el equipo

Hasta ahora el proceso descrito es muy similar al realizado durante la temporada del 2017 pero en lugar de comenzar a tantear el valor de los parámetros se comienza la preparación de la optimización, por lo que se debe obtener el rango en el que los parámetros primitivos permiten un automóvil factible y de acuerdo con el reglamento, dicho proceso se desarrolla en la sección de restricciones de desigualdad en el capítulo 3.

Los últimos valores de entrada para el algoritmo genético es el valor de gradiente de *roll* deseado, las razones de instalación y la altura de centro de gravedad, el primero puede ser tomado de las recomendaciones de Milliken [30], para un vehículo de competencia este valor es de 1.5 [°/g], los segundos son la relación entre el cambio de longitud del resorte y el viaje vertical de la llanta. Para FSAE es

recomendable que su valor sea cercano a 1 [31]. La altura del centro de gravedad puede ser aproximada con la del vehículo anterior.

Con todos los valores anteriores se hace uso del programa propuesto, Anexo 4, su funcionamiento será aclarado en las secciones de funciones objetivo y programación del capítulo 3, este regresa una hoja de cálculo que contiene los individuos de la última generación, la aptitud de cada individuo y para los individuos en la frontera de Pareto la distancia euclidiana promedio en el espacio de variables de diseño. Es deseable que dicha distancia sea mínima debido a que es altamente probable que los valores encontrados sean diferentes a los reales, por las tolerancias de manufactura. Así pues, los valores finales se podrían acercar a otro individuo de la frontera de Pareto

Con los parámetros finales seleccionados se pueden generar los puntos de los brazos de suspensión con un boceto que relacione el RC, fvsa con el IC y puntos de los brazos de suspensión como se muestra en el Anexo 3, para fines de este trabajo se utilizó GeoGebra®, pero se recomienda el uso de un software para dibujo en computadora, el boceto realizado se muestra en la Ilustración 2-3.

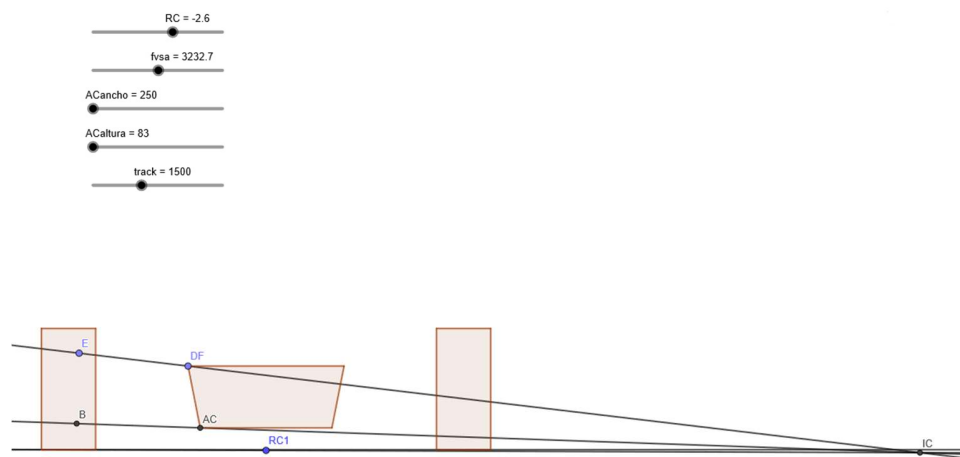


Ilustración 2-3: Boceto en GeoGebra para definir los puntos de los brazos de suspensión.

Con los puntos de los brazos definidos se prosigue a seleccionar la ubicación de los puntos del balancín y el amortiguador. Para evitar las iteraciones manuales se escribió un programa en Mathematica®, que ejecuta un algoritmo genético para encontrar los puntos del balancín que minimicen el error al IR deseado.

Debido a que ya fue definida la rigidez a *roll* de ambos ejes, ya no es necesario el cálculo de la distribución de gradiente de *roll*. Se procede con la selección del resorte y de la razón de *ride* (*ride rate*) con ayuda de las frecuencias. Con la razón del resorte seleccionada se puede calcular K_W y con este valor se calcula la rigidez necesaria de la barra de torsión para alcanzar la rigidez deseada de cada eje.

Finalmente, con la rigidez de la ARB establecida y el diseño de la geometría de la suspensión se procede a realizar el cálculo de las fuerzas para la selección de materiales y optimización topológica. El proceso propuesto es mostrado por el diagrama en la Ilustración 2-4

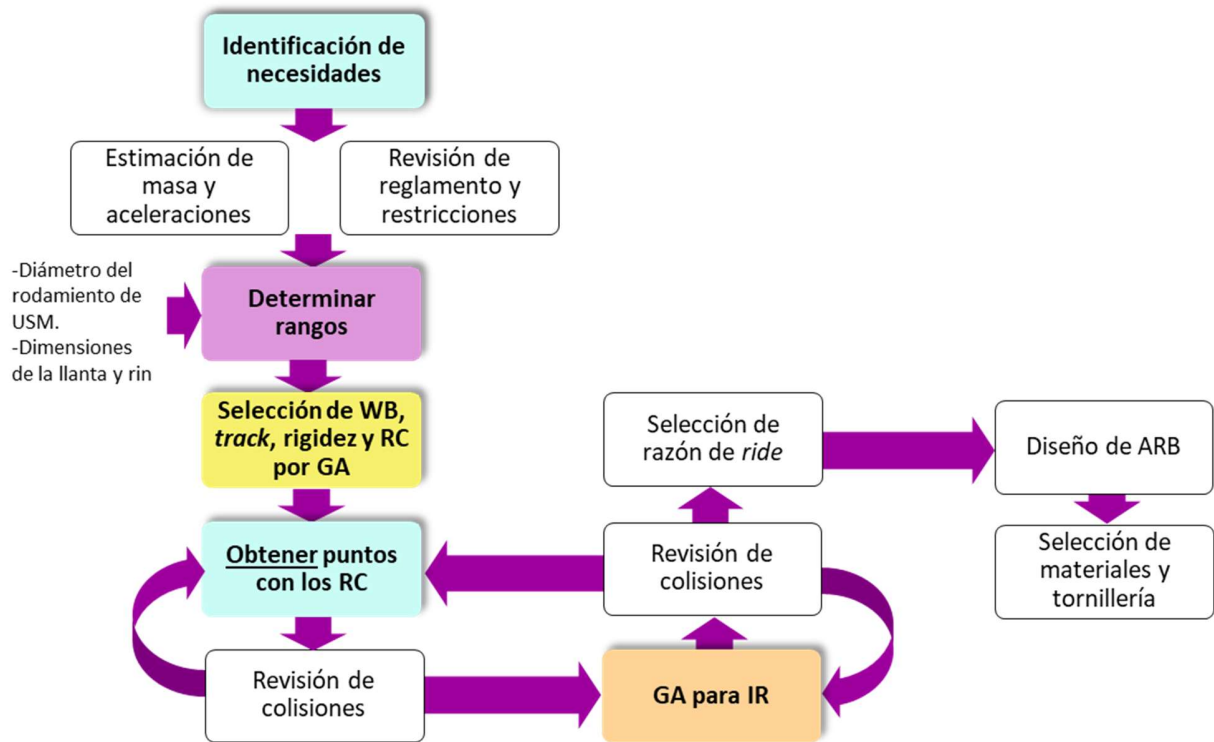


Ilustración 2-4: Diagrama de proceso de diseño propuesto para una suspensión doble brazo de A

Capítulo 3: OPTIMIZACIÓN

Dentro del diseño de la suspensión se deben de seleccionar parámetros que beneficiarán cierto comportamiento dinámico, pero perjudicarán otro, es decir se tienen objetivos en conflicto, el ejemplo más sencillo de estos es la agilidad del vehículo y la fuerza lateral total que pueden proveer los neumáticos, pues para obtener un vehículo ágil se desea que tenga un menor momento de inercia y por lo tanto menores dimensiones. Para obtener una mayor fuerza lateral es necesario disminuir la transferencia de carga, por lo que se buscarán dimensiones mayores.

Con el fin de mejorar las técnicas de diseño y fabricación del sistema de suspensión del equipo de FSAE de la UNAM se propone un algoritmo de optimización para encontrar los justos medios de los valores que tengan objetivos en conflicto como el caso previamente expuesto. Para esto es necesario encontrar las funciones en conflicto que sean representativas del comportamiento dinámico del prototipo FSAE e involucren a los parámetros de los que se parte el diseño de la suspensión, parámetros primitivos, estas funciones serán llamadas funciones objetivo.

Las variables de las funciones objetivo que no sean parte de los parámetros primitivos serán fijadas ya sea por valores esperados o del prototipo anterior, estas serán las restricciones de igualdad. Debido a que pretende diseñar un vehículo que está restringido por el reglamento una competencia, los parámetros primitivos no pueden tomar cualquier valor, por lo que es necesario definir un rango para cada uno de estos, estos serán llamados restricciones de desigualdad.

3.1: FUNCIONES OBJETIVO

Un cuerpo que posee un momento de inercia (I_z) bajo tiene menor resistencia a aceleraciones rotacionales, por lo que provee de una respuesta más rápida al girar [32]. El momento de inercia depende de la geometría, material y ubicación de todos los elementos del vehículo, sin embargo en una etapa temprana del diseño no se poseen dichos datos, por lo que, se aproximará I_z con un modelo rudimentario, este representado con un prisma rectangular con cilindros en las esquinas en imagen de las llantas como se observa en la Ilustración 3-1 .

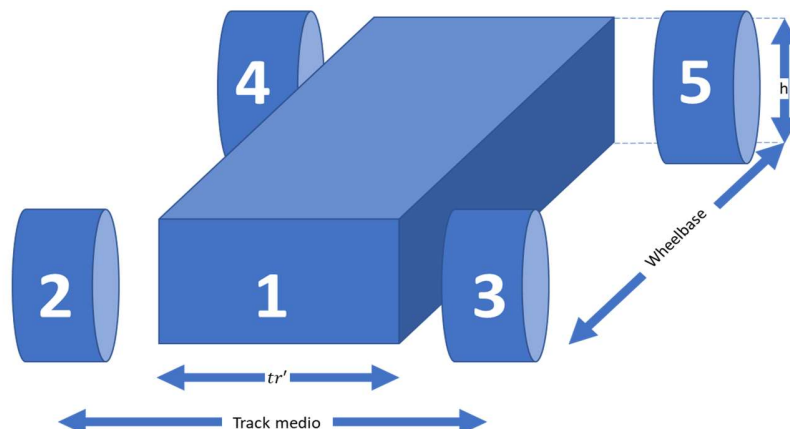


Ilustración 3-1 Diagrama para el cálculo de momentos de inercia.

El momento de inercia respecto al “eje z” en el CG será la suma de los momentos de inercia de cada uno de los elementos, es decir:

$$I_z = \sum_{i=1}^n I_{z_i}$$

Donde “ n ” es la cantidad de elementos e I_{z_i} el momento de inercia del cuerpo i respecto al “eje z” en el “CG” del vehículo. Al haber 5 cuerpos la fórmula anterior queda de la siguiente manera.

$$I_z = I_{z_1} + I_{z_2} + I_{z_3} + I_{z_4} + I_{z_5} \quad \text{Ec. 3.1}$$

Donde $I_{z_{1-5}}$ son el momento de inercia respecto al “eje z” en el centro de gravedad del vehículo del cuerpo 1-5 correspondientemente.

Primero se calcularán los momentos de inercia de cada elemento en su CG, $I_{z_i}^{cg}$, luego para obtener I_{z_i} se aplicará el teorema de Steiner o de ejes paralelos, el cual se expresa de la siguiente forma.

$$I_{z_i} = I_{z_i}^{cg} + M D^2 \quad \text{Ec. 3.2}$$

Donde $I_{z_i}^{cg}$ es el momento de inercia respecto el “eje z” en el CG del cuerpo i , M es la masa de ese elemento y D es la distancia entre los ejes.

Para el cuerpo uno:

Se aplica el teorema de Steiner, ecuación Ec. 3.2, resultando de esto la siguiente ecuación:

$$I_{z_1} = I_{z_1}^{cg} + m_s D_{XY_1}^2 \quad \text{Ec. 3.3}$$

Donde:

- $I_{z_1}^{cg}$ momento de inercia respecto al “eje z” en el CG del prisma.
- m_s masa suspendida incluyendo al piloto.
- D_{XY_1} distancia entre el CG del vehículo y el del prisma en el plano XY.

Para calcular $I_{z_1}^{cg}$ se ocupará la siguiente fórmula de momento de inercia de un prisma, quedando:

$$I_{z_1}^{cg} = \frac{m_s}{12} (tr'^2 + l^2) \quad \text{Ec. 3.4}$$

Donde:

- l wheelbase
- t_ϕ diametro de la llanta
- t_w ancho de la llanta
- tr' ancho del prisma

El ancho del prisma, tr' , se puede considerar como el *track* promedio menos el ancho del neumático y su diámetro, esto expresado de la siguiente forma.

$$tr' = \frac{tr_F + tr_R}{2} - t_\emptyset - t_w \quad \text{Ec. 3.5}$$

Al sustituir la ecuación Ec. 3.5 en la ecuación Ec. 3.4 se encuentra la siguiente expresión

$$I_{z_1}^{cg} = \frac{m_s}{12} \left(\left(\frac{tr_F + tr_R}{2} - t_\emptyset - t_w \right)^2 + l^2 \right) \quad \text{Ec. 3.6}$$

Se considerará que el centro de gravedad del prisma y del vehículo están centrados lateralmente pero no longitudinalmente, es decir su "coordenada y" es 0 pero no así la "coordenada x", por lo que la fórmula para calcular la distancia entre sus "ejes z" se expresa de la siguiente forma:

$$D_{xY_1}^2 = (x_1 - a)^2 \quad \text{Ec. 3.7}$$

Donde: x_1 es la "coordenadas x" del centro de gravedad del prisma y a distancia del centro de gravedad del vehículo al eje delantero.

Debido a que la ubicación del centro de gravedad del prisma no es conocida se hace uso del cálculo del centroide de figuras compuestas para su obtención, las expresiones resultantes se muestran a continuación.

$$\begin{aligned} x_1 &= \frac{aV_1 + 4aV_2 - 2lV_2}{V_1} \\ y_1 &= 0 \\ z_1 &= \frac{h_{cg}V_1 + 4h_{cg}V_2 - 2V_2t_\emptyset}{V_1} \end{aligned} \quad \text{Ec. 3.8}$$

Donde:

- x_1, y_1, z_1 "coordenadas x,y,z" del centro de gravedad del prisma
- $V_{1,2}$ volumen de la figura 1 o 2 respectivamente
- h_{cg} altura del centro de gravedad

Al sustituir la ecuación Ec. 3.8 en la ecuación Ec. 3.7 se obtiene la siguiente expresión

$$D_{XY} = \frac{-aV_1 - 4aV_2 + 2lV_2}{V_1} - a \quad \text{Ec. 3.9}$$

El volumen dos, V_2 , es fácilmente calculable con la fórmula para el volumen de un cilindro la cual se expresa a continuación:

$$V_2 = \frac{1}{4} \pi t_\emptyset^2 t_w \quad \text{Ec. 3.10}$$

Una de las formas para el cálculo de volúmenes es a partir de la densidad y masa, debido a que el vehículo está hecho de muchos componentes diferentes su volumen sería la sumatoria de la división de la masa de cada uno entre su densidad, sin embargo, la realización de este proceso implica que se conoce en su totalidad, pero en una etapa temprana de diseño lo anterior no es posible por consiguiente se calculará *grosso modo* considerando que este está hecho puramente de acero. Lo anterior se aproxima a la realidad

debido a que su elemento más pesado, el motor, está hecho principalmente de dicho material, otro de los elementos que aporta una parte importante del peso es el chasis, también de dicho material. Además de la masa suspendida del vehículo en el volumen 1 también considera al piloto, por lo que el volumen uno puede ser calculado de la siguiente forma:

$$V_1 = \frac{m_p}{\rho_p} + \frac{m_s - m_p}{\rho_s} \quad \text{Ec. 3.11}$$

Donde “ m_p ” y “ m_s ” son la masa del piloto y suspendida respectivamente mientras que “ ρ_p ” y “ ρ_s ” son sus densidades correspondientes.

El desarrollo para los cuerpos 2-5 se hará de forma simultánea pues existe gran similitud entre ellos.

Primero se aplica el teorema de Steiner, ecuación Ec. 3.2, resultando de esto las siguientes ecuaciones:

$$I_{z_2} = I_{z_2}^{cg} + m_u D_{XY_2}^2 \quad \text{Ec. 3.12}$$

$$I_{z_3} = I_{z_3}^{cg} + m_u D_{XY_3}^2$$

$$I_{z_4} = I_{z_4}^{cg} + m_u D_{XY_4}^2 \quad \text{Ec. 3.13}$$

$$I_{z_5} = I_{z_5}^{cg} + m_u D_{XY_5}^2$$

Donde:

- $I_{z_{2,3,4,5}}^{cg}$ momento de inercia respecto al “eje z” en el CG del cuerpo 2,3,4 o 5 correspondientemente.
- $D_{XY_{2,3,4,5}}$ distancia en el plano XY entre el CG del vehículo y el de la figura 1, 2,3,4 o 5 respectivamente.

Primero se obtendrá los momentos de inercia en el CG de cada figura con ayuda de la fórmula de momento de inercia para un cilindro:

$$I_{z_2}^{cg} = \frac{m_u}{12} \left(\frac{3}{4} t_{\phi}^2 + t_w \right) \quad \text{Ec. 3.14}$$

Donde “ m_u ” es la masa no suspendida de una esquina del vehículo.

Debido a que los cuerpos 2-5 son los mismos su I_z^{cg} son iguales, lo que se expresa de la siguiente manera.

$$I_{z_2}^{cg} = I_{z_3}^{cg} = I_{z_4}^{cg} = I_{z_5}^{cg} \quad \text{Ec. 3.15}$$

Ahora se prosigue con el cálculo de la distancia entre los ejes y del CG del vehículo y cada figura:

$$D_{z_2}^2 = D_{z_3}^2 = a^2 + \frac{1}{4} tr_F^2 \quad \text{Ec. 3.16}$$

$$D_{z_4}^2 = D_{z_5}^2 = b^2 + \frac{1}{4} tr_R^2 \quad \text{Ec. 3.17}$$

Donde “ b ” es la distancia entre el eje trasero y el CG. Debido a $D_{z_2}^2 = D_{z_3}^2$ resulta $I_{y_2} = I_{y_3}$, así mismo con $D_{y_4}^2 = D_{y_5}^2$ resulta $I_{y_4} = I_{y_5}$.

En conclusión, el momento de inercia sobre el “eje z” puede ser calculado al sustituir las ecuaciones Ec. 3.12-Ec. 3.17 en la ecuación Ec. 3.1, se obtiene:

$$I_z = I_{z_1} + 4I_{z_2}^{cg} + \frac{1}{2}m_u \left(\frac{4l^2(m_F^2 + m_R^2)}{m^2} + tr_F^2 + tr_R^2 \right) \quad \text{Ec. 3.18}$$

Donde:

$$I_{z_1} = I_{z_1}^{cg} + m_s D_{XY_1}^2 \quad \text{Ec. 3.3}$$

$$I_{z_1}^{cg} = \frac{m_s}{12} \left(\left(\frac{tr_F + tr_R}{2} - t_\phi - t_w \right)^2 + l^2 \right) \quad \text{Ec. 3.6}$$

$$D_{XY_1}^2 = \left(\frac{-aV_1 - 4aV_2 + 2lV_2}{V_1} - a \right)^2 \quad \text{Ec. 3.9}$$

$$V_1 = \frac{m_p}{\rho_p} + \frac{m_s - m_p}{\rho_s} \quad \text{Ec. 3.11}$$

$$V_2 = \frac{\pi}{4} t_\phi^2 t_w \quad \text{Ec. 3.10}$$

$$I_{z_2}^{cg} = \frac{m_u}{12} \left(\frac{3}{4} t_\phi^2 + t_w^2 \right) \quad \text{Ec. 3.15}$$

Para disminuir el tiempo al girar se debe de aumentar la aceleración lateral, para aumentar esta es necesario maximizar la fuerza lateral que suministran los neumáticos. Por regla general se puede decir que la fuerza máxima total que los neumáticos pueden proveer es cuando no hay transferencia de carga, por lo que minimizar esta se vuelve una función objetivo

La transferencia de carga se puede calcular de la siguiente forma siguiendo el modelo de Pacejka [9]

$$\Delta F_z = \sigma m a_y \quad \text{Ec. 3.19}$$

$$\sigma_F = \frac{1}{tr_F} \left(\frac{c_{\phi F}}{c_{\phi F} + c_{\phi R} - m g h'} h' + \frac{b}{l} h_{RCF} \right) \quad \text{Ec. 3.20}$$

$$\sigma_R = \frac{1}{tr_R} \left(\frac{c_{\phi R}}{c_{\phi F} + c_{\phi R} - m g h'} h' + \frac{a}{l} h_{RCR} \right) \quad \text{Ec. 3.21}$$

Donde:

- σ_F, σ_R es el coeficiente de transferencia de carga delantero o trasero respectivamente
- h_{RCF}, h_{RCR} es la altura del centro de roll delantero o trasero respectivamente

La fuerza normal en las llantas cuando el vehículo está únicamente bajo una aceleración lateral, es decir que no hay aceleración longitudinal, son las siguientes.

$$F_{zF}^I = \frac{1}{2} m_F g - \Delta F_z^F \quad \text{Ec. 3.22}$$

$$F_{zF}^O = \frac{1}{2} m_F g + \Delta F_z^F \quad \text{Ec. 3.23}$$

$$F_{zR}^I = \frac{1}{2} m_R g - \Delta F_z^R \quad \text{Ec. 3.24}$$

$$F_{z_R}^O = \frac{1}{2} m_R g + \Delta F_z^R \quad \text{Ec. 3.25}$$

Donde:

- $F_{z_{F,R}}^I$ fuerza normal en la llanta interna delantera o trasera respectivamente.
- $F_{z_{F,R}}^E$ fuerza normal en la llanta externa delantera o trasera respectivamente.

Sustituimos las ecuaciones Ec. 3.19-Ec. 3.21 en las ecuaciones Ec. 3.22-Ec. 3.25

$$\begin{aligned} F_{z_F}^I &= \frac{1}{2} m_F g - \Delta F_z^F & F_{z_F}^O &= \frac{1}{2} m_F g + \Delta F_z^F \\ F_{z_F}^I &= \frac{1}{2} m_F g - \sigma_F m a_y & F_{z_F}^O &= \frac{1}{2} m_F g + \sigma_F m a_y \\ F_{z_F}^I &= g \left(\frac{1}{2} m_F - \sigma_F m A_y \right) & F_{z_F}^O &= g \left(\frac{1}{2} m_F + \sigma_F m A_y \right) \end{aligned} \quad \begin{array}{l} \text{Ec. 3.26} \\ \text{Ec. 3.27} \end{array}$$

$$\begin{aligned} F_{z_R}^I &= \frac{1}{2} m_R g - \Delta F_z^R & F_{z_R}^O &= \frac{1}{2} m_R g + \Delta F_z^R \\ F_{z_R}^I &= \frac{1}{2} m_R g - \sigma_R m a_y & F_{z_R}^O &= \frac{1}{2} m_R g + \sigma_R m a_y \\ F_{z_R}^I &= g \left(\frac{1}{2} m_R - \sigma_R m A_y \right) & F_{z_R}^O &= g \left(\frac{1}{2} m_R + \sigma_R m A_y \right) \end{aligned} \quad \begin{array}{l} \text{Ec. 3.28} \\ \text{Ec. 3.29} \end{array}$$

Donde $a_y = A_y g$.

Es deseable que, al tener una transferencia de carga lateral, las fuerzas normales de las llantas internas sean similares, al mismo tiempo que las externas mantengan la misma condición, es decir:

$$\begin{aligned} F_{z_F}^I &= F_{z_R}^I \\ F_{z_F}^O &= F_{z_R}^O \end{aligned} \quad \text{Ec. 3.30}$$

A la diferencia del estado anterior le llamaremos **error_I** y **error_O**, para el error entre las cargas normales de las llantas internas y externas respectivamente, es decir $|F_{z_F}^I - F_{z_R}^I| = \text{error}_I$ $|F_{z_F}^O - F_{z_R}^O| = \text{error}_O$. Sustituimos las ecuaciones Ec. 3.26-Ec. 3.29 en las dos anteriores:

$$\begin{aligned} |F_{z_F}^I - F_{z_R}^I| &= \text{error}_I & |F_{z_F}^O - F_{z_R}^O| &= \text{error}_O \\ \left| g \left(\frac{1}{2} m_F - \sigma_F m A_y \right) - g \left(\frac{1}{2} m_R - \sigma_R m A_y \right) \right| & & \left| g \left(\frac{1}{2} m_F + \sigma_F m A_y \right) - g \left(\frac{1}{2} m_R + \sigma_R m A_y \right) \right| & \\ &= \text{error}_I & &= \text{error}_I \\ \left| \frac{1}{2} g (m_F - m_R) - m g A_y (\sigma_F - \sigma_R) \right| &= \text{error}_I & \left| \frac{1}{2} g (m_F - m_R) + m g A_y (\sigma_F - \sigma_R) \right| &= \text{error}_O \end{aligned}$$

Se supondrá que durante la competencia el vehículo gira en un rango de 1 a 2 [g's] por lo que será en este rango en el que se calcule el error y la disminución de este será una función objetivo.

$$error_{I@1g} + error_{O@1g} + error_{I@2g} + error_{O@2g} \quad \text{Ec. 3.31}$$

Para evitar aumentar innecesariamente la cantidad de funciones objetivos, solo será considerada la disminución de transferencia de carga delantera, en lugar de ambas, y se agregará la del error.

Al analizar las ecuaciones Ec. 3.20 y Ec. 3.21 es posible notar que el aumento de $c_{\varphi F}$ y $c_{\varphi R}$ tiene un resultado positivo en ambos casos, por lo que es muy probable que si únicamente estas funciones son utilizadas para medir la aptitud de los individuos, estos tenderían a valores muy altos de rigidez, para limitar este efecto agregaremos otra función objetivo, esta es el error a un gradiente de *roll* deseado, por lo cual primero comenzamos definiendo el gradiente de *roll* como los grados de *roll* que gira el vehículo por aceleración lateral que sufre, normalmente es indicada en [°/g], se puede expresar de la siguiente forma.

$$\nabla roll = \frac{\varphi}{\frac{a_y}{g}} = \frac{m g h'}{c_{\varphi F} + c_{\varphi R} - m g h'} \quad \text{Ec. 3.32}$$

El error del gradiente de *roll* calculado contra el deseado, se expresa de la siguiente forma.

$$\frac{\nabla roll_c - \nabla roll_d}{\nabla roll_d} \quad \text{Ec. 3.33}$$

Donde:

- $\nabla roll_c$ es el gradiente de *roll* calculado
- $\nabla roll_d$ es el gradiente de *roll* deseado

En resumen, las siguientes ecuaciones serán las funciones objetivo:

$$I_z = I_{z_1} + 4I_{z_2}^{cg} + \frac{1}{2}m_u \left(\frac{4l^2(m_F^2 + m_R^2)}{m^2} + tr_F^2 + tr_R^2 \right) \quad \text{Ec. 3.18}$$

$$\sigma_F = \frac{1}{tr_F} \left(\frac{c_{\varphi F}}{c_{\varphi F} + c_{\varphi R} - m g h'} h' + \frac{b}{l} h_{RCF} \right) \quad \text{Ec. 3.20}$$

$$error_{I@1g} + error_{O@1g} + error_{I@2g} + error_{O@2g} \quad \text{Ec. 3.31}$$

$$\frac{\nabla roll_c - \nabla roll_d}{\nabla roll_d} \quad \text{Ec. 3.33}$$

3.2: RESTRICCIONES DE DESIGUALDAD

Para que un valor cuantitativo sea elegible debe ser físicamente posible que el vehículo tome dicho valor y este no infrinja el reglamento, por lo cual en las secciones subsecuentes se delimitará un rango para cada parámetro dentro del cual el programa propondrá valores.

3.2.1: TRACKS MÁXIMOS.

Vamos a suponer que el vehículo viaja por el centro de la pista, el *whelbase* es tangente al centro de la curva, una distribución de carga 50/50 lateral y se propone un sistema coordenado en el centro de la curva (ver Ilustración 3-3), entonces

$$R\mathbf{j} + a'\mathbf{i} + \frac{1}{2}c\mathbf{j} = e R_e \quad \text{Ec. 3.34}$$

Donde:

- R el radio al centro de la pista
- a' distancia del CG al frente del vehículo
- c el ancho frontal del coche
- R_e radio externo de la curva
- e vector hacia la esquina exterior frontal

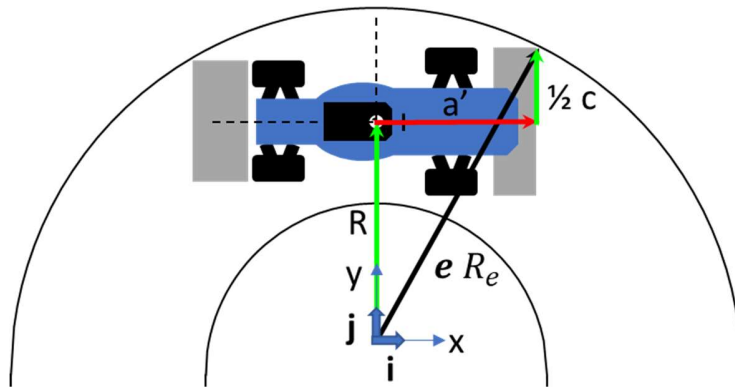


Ilustración 3-3 Diagrama del automóvil en la pista para el cálculo del track máximo.

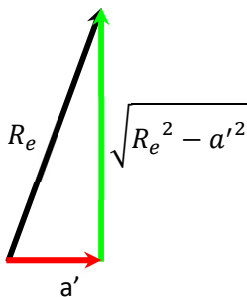


Ilustración 3-2 Suma de vectores para el cálculo de track máximo

El vector $e R_e$ puede ser descompuesto en componentes ij más simples y con ayuda del teorema de Pitágoras (ver Ilustración 3-2) y considerando la máxima distancia del frente del vehículo la ecuación Ec. 3.34 queda de la siguiente forma.

$$R\mathbf{j} + a'\mathbf{i} + \frac{1}{2}c_{Max}\mathbf{j} = a'\mathbf{i} + \left(\sqrt{R_e^2 - a'^2}\right)\mathbf{j}$$

$$R + \frac{1}{2}c_{Max} = \sqrt{R_e^2 - a'^2}$$

$$c_{Max} = 2 \left(\sqrt{R_e^2 - a'^2} - R \right) \quad \text{Ec. 3.35}$$

De acuerdo con el reglamento de FSAE para el 2019, la parte más lejana del alerón delantero debe estar a máximo 0.7 [m] de la parte delantera de las llantas, (Ilustración 3-4), el valor de a' debe ser la suma de esta medida, la distancia del CG al eje delantero y la mitad del diámetro de la llanta, lo anterior se puede escribir como:

$$a' = a + \frac{1}{2}t_\phi + 0.7 \quad \text{Ec. 3.36}$$

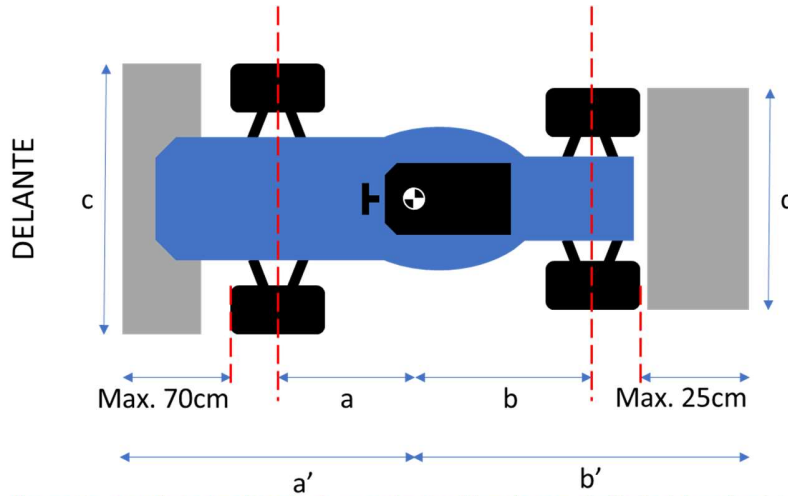


Ilustración 3-4 Dibujo simplificado de un coche tipo fórmula para el cálculo del track máximo.

Suponemos que la distancia c es el track delantero más el ancho del neumático, t_{wF} , ver Ilustración 3-4. Por lo anterior el track máximo delantero, tr_{FMax} , se relaciona con c_{Max} de la siguiente forma:

$$c_{Max} = tr_{FMax} + t_{wF} \quad \text{Ec. 3.37}$$

Para finalizar el caculo se sustituyen las ecuaciones Ec. 3.36 y Ec. 3.37 en la ecuación Ec. 3.35, se considerará una tolerancia de 0.1 [m] de cada lado para evitar el alto riesgo de derribar conos al recorrer la pista y se despeja el track máximo delantero, obteniendo las siguiente expresión

$$tr_{FMax} = 2 \left(\sqrt{R_e^2 - \left(a + \frac{1}{2} t_{\phi} + 0.7 \right)^2} - R \right) - t_{wF} - 0.2 \quad \text{Ec. 3.38}$$

Para los fines de esta tesis se considerará la prueba de *skid-pad* descrita en el reglamento de FSAE 2019 [33] (ver Anexo 1), el cual describe el diámetro mínimo y el ancho de pista máximo como 9.0 [m] y 3.5 [m] respectivamente, por lo tanto, R_e es 4.5 [m] y R de 2.75 [m]. Dentro de este trabajo se supondrá el uso de llantas Hoosier 18.0x6.0-10 R25B con diámetro de 18.1 [in] y ancho de 8.1 [in] [34]. Al sustituir los valores anteriores en la ecuación Ec. 3.38, la expresión para encontrar el track máximo queda de la siguiente forma.

$$tr_{FMax} = 2 \left(\sqrt{4.5^2 - \left(a + \frac{1}{2} 18.1 * 0.0254 + 0.7 \right)^2} - 2.75 \right) - 8.1 * 0.0254 - 0.2$$

$$tr_F = 2 \sqrt{4.5^2 - (a + 0.92987)^2} - 5.906$$

Ec. 3.39

De forma análoga generamos la ecuación para el eje trasero.

$$\begin{aligned} Rj - b'i + \frac{1}{2}dj &= R_e(p) \\ Rj - b'i + \frac{1}{2}dj &= -b'i + \left(\sqrt{R_e^2 - b'^2} \right) j \\ R + \frac{1}{2}d &= \left(\sqrt{R_e^2 - b'^2} \right) \\ d &= 2 \left(\sqrt{R_e^2 - b'^2} - R \right) \end{aligned}$$

Ec. 3.40

Donde " b " es la distancia del eje a la parte trasera más alejada del vehículo y " d " es el ancho de la parte trasera del vehículo.

Suponemos que la distancia d es el *track* trasero más el ancho del neumático, t_{wR} , ver Ilustración 3-4. Por lo anterior el *track* máximo trasero, tr_{RMax} , se relaciona con d_{Max} de la siguiente forma:

$$d_{Max} = tr_{RMax} + t_{wR} \quad \text{Ec. 3.41}$$

De acuerdo con el reglamento de FSAE para el 2019 la parte más lejana de los elementos aerodinámicos en la parte trasera es a máximo 0.25 [m] de la parte trasera de las llantas, ver Ilustración 3-4, entonces el valor de " b " debe ser la suma de esta medida, la dimensión del CG al eje trasero, la mitad del diámetro de la llanta, lo anterior se puede escribir como:

$$b' = b + \frac{1}{2}t_{\phi} + 0.25 \quad \text{Ec. 3.42}$$

Para finalizar el cálculo se sustituyen las ecuaciones Ec. 3.41 y Ec. 3.42 en la ecuación Ec. 3.40, al igual que para eje delantero se considerará una tolerancia de 0.1 [m] y se despeja el *track* máximo trasero, obteniendo la siguiente expresión:

$$\begin{aligned} tr_{RMax} + t_{wR} + 0.2 &= 2 \left(\sqrt{R_e^2 - \left(b + \frac{1}{2}t_{\phi} + 0.25\right)^2} - R \right) \\ tr_{RMax} &= 2 \left(\sqrt{R_e^2 - \left(b + \frac{1}{2}t_{\phi} + 0.25\right)^2} - R \right) - t_{wR} - 0.2 \end{aligned} \quad \text{Ec. 3.43}$$

Al sustituir en la ecuación Ec. 3.43 los valores de las dimensiones de las llantas y los radios previamente mencionados se encuentra la siguiente ecuación:

$$\begin{aligned} tr_{RMax} &= 2 \left(\sqrt{4.5^2 - \left(b + \frac{1}{2}18.1 * 0.0254 + 0.25\right)^2} - 2.75 \right) - 8.1 * 0.0254 - 0.2 \\ tr_{RMax} &= 2\sqrt{4.5^2 - (b + 0.48)^2} - 5.906 \end{aligned}$$

Las ecuaciones Ec. 3.38 y Ec. 3.43 pueden ser reescritas en términos del *wheelbase* si es conocida la distribución de masa, quedando de la siguiente manera:

$$tr_{FMax} = 2 \left(\sqrt{R_e^2 - \left(\frac{m_R}{m}l + \frac{1}{2}t_{\phi} + 0.7\right)^2} - R \right) - t_{wF} - 0.2 \quad \text{Ec. 3.44}$$

$$tr_{RMax} = 2 \left(\sqrt{R_e^2 - \left(\frac{m_F}{m}l + \frac{1}{2}t_{\phi} + 0.25\right)^2} - R \right) - t_{wR} - 0.2 \quad \text{Ec. 3.45}$$

3.2.2: TRACKS MÍNIMOS

Al tratar esta tesis de un vehículo para FSAE el “track” mínimo delantero está delimitado por el posible menor habitáculo permitido por el reglamento de esta competencia. Este indica el diámetro mínimo de los tubos del arco frontal y una plantilla, la cual debe pasar libremente. Además, debe existir suficiente espacio entre el centro de la llanta y la carrocería para el giro de la llanta. Esta distancia se aproximará con el radio del neumático.

Con la información anterior y las suposiciones se generó un diagrama simplificado para una mejor aclaración del cálculo del “track” mínimo, Ilustración 3-5.

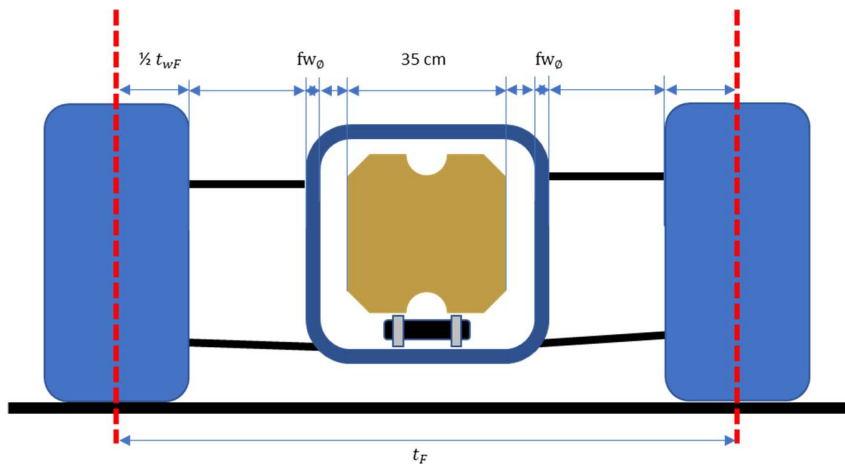


Ilustración 3-5 Diagrama para el cálculo del track mínimo delantero

Para el caso delantero el reglamento exige que una plantilla con un ancho de 0.35 [m] pase libremente en el habitáculo, por lo que a la dimensión de la plantilla se le agrega una pulgada de cada lado para evitar que imprecisiones de la manufactura de chasis lleguen a causar el incumplimiento de esta regla. Como se observa en la Ilustración 3-5, se debe considerar el diámetro del tubo al que se anclan los puntos de suspensión, en general, este tubo es el arco frontal (*front hoop*) del cual el reglamento exige un diámetro de 1.0 [in] [33]. Además, se debe de agregar $\frac{1}{2} * t_{\theta}$ por cada lado para el espacio de la llanta al girar. Finalmente se agrega $\frac{1}{2} t_w$ de cada lado para llegar al centro de la llanta, pues el track es medido entre los centros de los neumáticos. Todo lo anterior se resume en la Ilustración 3-6.

0.35	Ancho de la plantilla
2 x 0.0254	Tolerancia de manufactura del chasis
2 x 0.0254	Diámetro del tubo del arco frontal
18.1 x 0.0254	Diámetro de la llanta
8.1 x 0.0254	Ancho de la llanta
<hr/>	
1.117 [cm]	Total

Ilustración 3-6 Cálculo del track mínimo delantero.

Para el *track* trasero se realiza un proceso similar, pero en lugar de la plantilla se utiliza el tamaño del motor actual del equipo de FSAE de la UNAM, Yamaha® R8, a continuación, se muestra en la Ilustración 3-7.

0.4	Motor
2 x 0.0254	Tolerancia de manufactura del chasis
2 x 0.0254	Diámetro del tubo del arco frontal
18.1 x 0.0254	Diámetro de la llanta
8.1 x 0.0254	Ancho de la llanta
<hr/>	
1.167 [cm]	Total

Ilustración 3-7 Cálculo del *track* mínimo trasero.

La última condición para el *track* mínimo es debido al reglamento pues este limita al menor *track* como al menos el 75% del mayor. [33]

3.2.3: WHEEL BASE

El reglamento [33] limita el *wheelbase* a mínimo 1.525 [m], sin embargo, no especifica uno máximo, para permitir explorar soluciones con un valor alto su valor máximo será fijado en 1.85 [m], 0.15 [m] más que el máximo encontrado en la competencia el cual es de 1.70 [m].

3.2.4: CENTROS DE ROLL

La posición de centro de *roll* está delimitada por las dimensiones interiores del rin, este rango es fácilmente calculable con el diagrama de vista frontal, Ilustración 3-8.

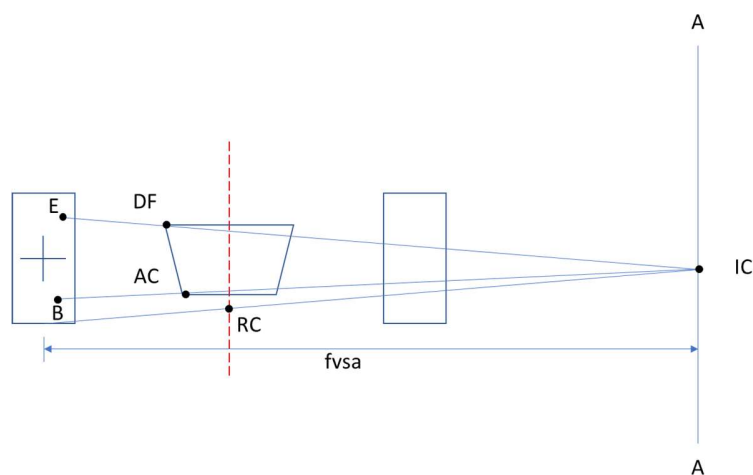


Ilustración 3-8: Vista frontal para el diseño de una suspensión de doble brazo de A.

Debido a que la altura del punto **AC** y la distancia fvsa están fijos el rango del centro de *roll* está definido por el desplazamiento vertical y horizontal de los puntos **B** y **AC** respectivamente.

Dentro de esta tesis se ha supuesto que se diseña para el vehículo de FSAE del equipo de la UNAM, por lo que se revisaron los datos de las llantas que actualmente posee el equipo (Hoosier 18.0x6.0-10 R25B). De los datos se concluyó que el valor máximo de *camber* antes que la fuerza lateral provista por esta comience a decaer. Por otro lado, el reglamento indica un desplazamiento vertical de la llanta mínimo de 5 [cm], considerando ambas restricciones se propondrá un *camber* estático de -2° y una ganancia de *camber* de 1° por un desplazamiento vertical del neumático de 2.5 [cm], por lo que el *ride camber* queda de $0.4 [^\circ/\text{cm}]$.

Se puede calcular el brazo de oscilación en la vista frontal o *fvsa* por las siglas en inglés, con la siguiente fórmula:

$$fvsa = \frac{1}{\tan(\text{ganancia de camber})} \quad \text{Ec. 3.46}$$

Al sustituir los valores previamente encontrados en la ecuación Ec. 3.47 se encuentra el siguiente valor.

$$fvsa = \frac{1}{\tan(0.4)} = 143.237 \text{ cm} = 1.432 \text{ m}$$

Se prosigue con el cálculo de la altura de **AC**, está será la mínima posible pues se pretende diseñar un vehículo de competencia. Se comienza con 2.5 [cm] pues el reglamento [33] exige 5 [cm] de desplazamiento mínimo útil vertical de la suspensión. Después se continúa con los elementos que se encuentran debajo de este punto, para el caso del equipo de la UNAM sería el piso plano con un espesor de 1 [cm] considerando una tolerancia de manufactura. Al mismo tiempo se supondrá que el punto **AC** se encuentra al centro del tubo inferior de chasis por lo que se agrega el radio de este elemento, $\frac{1}{2} [\text{in}]$. Finalmente se agregan 3.5 [cm] de tolerancia por la manufactura del chasis, lo cual se resume en la Ilustración 3-9.

2.5	Desplazamiento reglamentario
1	Espesor del piso plano y tolerancia
$\frac{1}{2} \times 2.54$	Radio del tubo inferior
3.5	Tolerancia de manufactura del chasis
<hr/>	
8.27 [cm]	Total

Ilustración 3-9 Cálculo de la altura mínima del punto AC

Es decir:

$$z_{AC} = 0.083 \text{ m} \quad \text{Ec. 3.47}$$

Para definir el punto **AC** en un plano aún es necesario determinar su coordenada en el “eje y”, esta distancia no será fija. Su valor mínimo esta dado por: la plantilla del habitáculo, los tubos y la tolerancia de manufactura (ver Ilustración 3-5), se suman estos valores y luego se dividen entre dos porque el sistema coordenado se encuentra al centro del *track*, también es necesario considerar el espacio para la junta y la

soldadura de los anclajes a chasis, esta distancia será 2.5 [cm], entonces para el eje delantero es 25.08 [cm] y 27.58 [cm] para el eje trasero.

El valor máximo de **AC** será limitado por el *track* menos una distancia tal que permita girar a las llantas sin colisionar, esta distancia se ha aproximado con el radio de la llanta.

$$y_{AC_{M\acute{a}x}} = \frac{1}{2}(tr - t_{\phi}) \quad \text{Ec. 3.48}$$

Por otra parte, es obtenido el rango de altura del punto B, este se encuentra restringido por el rin y el rodamiento con sus respectivos espacios libres necesarios, ver Ilustración 3-10.

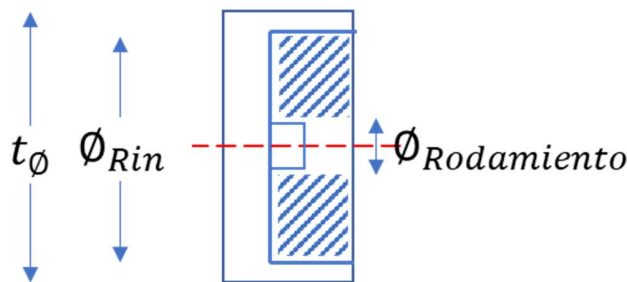


Ilustración 3-10 Diagrama simplificado del espacio interior del rin

El cálculo de la altura mínima de **B** se realiza de la siguiente forma:

$$z_{B_{Min}} = \frac{t_{\phi} - \phi_{Rin}}{2} + 0.03$$

Donde ϕ_{Rin} se refiere al diámetro interior del rin, se agregan 0.03 [m], 0.02 [m] para el espacio del tornillo, rótula y su housing y 0.01 [m] por el movimiento del brazo.

Para el caso del equipo de la UNAM: $t_{\phi} \approx 0.46$ [m] y $\phi_{Rin} \approx 0.24$ [m] lo que significa que $z_{B_{Min}} = 0.14$ [m].

Para terminar el cálculo del rango del punto **B**, su altura máxima se obtiene de la siguiente forma:

$$z_{B_{M\acute{a}x}} = \frac{t_{\phi} - \phi_{Rodamiento}}{2} - 0.02$$

Donde $\phi_{Rodamiento}$ se refiere al diámetro de la carcasa que contiene al rodamiento seleccionado por el sistema de USM, además se agrega 0.02 [m] de espacio para el tornillo.

Con valor de $\phi_{Rodamiento} = 0.092$ [m], $z_{B_{M\acute{a}x}}$ es igual a 0.164 [m].

Finalmente, para delimitar el punto **B** en el plano XZ es necesaria su posición en el "eje y" esta se encuentra limitada por el sistema de USM, dicho sistema necesita un espacio de diseño para colocar el disco de frenos y demás componentes, por lo anterior:

$$y_B = \frac{tr}{2} - 0.031 \quad \text{Ec. 3.49}$$

Es posible el cálculo de la altura del centro de *roll* con las siguientes formulas.

$$z_{RC} = \frac{z_{IC} tr}{2 f v s a} \quad \text{Ec. 3.50}$$

$$z_{IC} = z_B + \frac{z_{AyC} - z_B}{y_{AyC} - y_B} (y_{IC} - y_B) \quad \text{Ec. 3.51}$$

$$y_{IC} = \frac{1}{2} tr - f v s a \quad \text{Ec. 3.52}$$

Las ecuaciones Ec. 3.50 y Ec. 3.51 se obtienen por medio de geometría analítica y la Ilustración 3-8.

Al sustituir las ecuaciones Ec. 3.47, Ec. 3.49 y Ec. 3.52 en la ecuación Ec. 3.51 es obtenida la siguiente expresión en función de z_B , y_{AyC} y tr .

$$z_{IC} = z_B + \frac{0.083 - z_B}{y_{AyC} - \left(\frac{tr}{2} - 0.034\right)} \left(\frac{1}{2} tr - f v s a - \left(\frac{tr}{2} - 0.034\right)\right)$$

Para el rango de z_B obtenido la menor altura de centro de *roll*, $z_{RC_{min}}$, se encontrará con $z_B = z_{B_{máx}}$ y $y_{AC} = y_{AC_{máx}}$, es decir, $z_B = 0.164$ y $y_{AC_{máx}} = \frac{1}{2}(tr - t_\phi) = \frac{tr}{2} - 0.23$, por lo que la ecuación para la altura mínima del RC en función del *track* queda como:

$$z_{RC_{min}} = -0.142 tr$$

Para $z_{RC_{máx}}$ se sustituye $z_B = z_{B_{min}}$ y $y_{AC} = y_{AC_{min}}$.

$z_{RC_{máx}}$	
Eje delantero	Eje trasero
$y_{AC_F} = y_{AC_{F_{min}}} = 0.250$	$y_{AC_R} = y_{AC_{R_{min}}} = 0.276$
$z_B = z_{B_{min}} = 13.987$	
$z_{RC_{máx F}} = \frac{(-0.0834 + 0.049 tr_F) tr_F}{-0.563 + tr_F}$	$z_{RC_{máx R}} = \frac{(-0.086 + 0.049 tr_R) tr_R}{-0.0613 + tr_R}$

3.2.5: RIGIDEZ DE ROLL

La rigidez de *roll* puede ser modificada por medio de las barras de torsión o ARB por sus siglas en inglés, por lo que este valor puede ser modificado a voluntad del diseñador, puede ser tan grande como este lo desee y solo limitada por su habilidad de empaquetamiento, en caso de que la ARB tenga que ser montada en la parte inferior del vehículo, su altura debe ser considerada en el cálculo de altura mínima del punto AC.

De acuerdo con Milliken [30] la rigidez a *roll* se puede calcular como:

$$c_\phi = \frac{IR^2 k_s k_t}{k_t + IR^2 k_s} \frac{tr^2}{2} \quad \text{Ec. 3.53}$$

Donde

- k_s es la constante de resorte de la suspensión
- k_t es la constante de resorte de la llanta
- IR es la razón de instalación, en inglés *installation ratio*.

Se continua con la obtención de la constante elástica de la llanta por lo cual se analizaron los datos provistos por el TTC®, este valor es función de la velocidad, la temperatura, el *camber* y la presión del neumático. Se considerará que las llantas se encuentran a 12 psi y 40 [km/hr], por ser las más cercana a las condiciones de pista, finalmente el valor de *camber* será considerado 4° pues este sería el peor escenario, dado que a estos valores disminuye la constante elástica de la llanta. Entonces el valor de la constante elástica de la llanta es de 595.217 [lb/in] que en sistema internacional es equivalente a 104,238.487 [N/m].

Para la c_φ mínima k_s será la menor constate que se posea, en el caso del equipo de la FI esta es 150 [lb/in] $\approx 26,269.02525$ [N/m].

La rigidez máxima esta solamente delimitada por la barra de torsión que sea posible empaquetar en el vehículo, siguiendo los cálculos propuestos por Aldo Espejel [2] la rigidez proporcionada por la barra de torsión se puede calcular en función de su diámetro con la siguiente expresión:

$$c_{\varphi ARB} = \frac{E \pi \varnothing_{ARB}^4}{64(1 + \nu_P)L_{ARB}}$$

Donde:

- $c_{\varphi ARB}$ es la rigidez torsional de la barra.
- E es el módulo de elasticidad de la ARB.
- \varnothing_{ARB} diámetro de la ARB.
- ν_P módulo de Poisson.
- L_{ARB} longitud de la ARB.

La rigidez de un eje del vehículo con barra de torsión de acuerdo con Milliken [30] se puede calcular con:

$$c_{\varphi ARB} = \frac{c_\varphi k_t \left(\frac{tr^2}{2}\right)}{k_t \left(\frac{t^2}{2}\right) - c_\varphi} - k_s IR^2 \frac{tr^2}{2} \quad \text{Ec. 3.54}$$

De la ecuación Ec. 3.54 se pude despejar c_φ y obtener la siguiente expresión

$$c_\varphi = \frac{k_t tr^2 (2C_{\varphi ARB} + IR^2 k_s tr^2)}{2(2C_{\varphi ARB} + IR^2 k_s tr^2 + k_t tr^2)}$$

Ahora bien, se supondrá una ARB con 3 [cm] de diámetro y longitud de 0.3 [m] hecha de Aluminio 7075 T6 instalada en una suspensión con IR unitario y k_s máximo, es decir $\varnothing_{ARB} = 0.03$ [m], $L_{ARB} = 0.3$ [m], $E =$

72,000 [MPa], $v_p = 0.33$, $IR = 1$, $k_s = 600$ [lb/in]. El único parámetro faltante sería el *track* y debido a que se busca el máximo c_ϕ se debe de tomar el *track* máximo, este se encuentra con el *wheelbase* mínimo, como se vio en secciones anteriores.

3.3: RESTRICCIONES DE IGUALDAD

Estas serán los parámetros considerados fijos dentro del análisis, debido a que su modificación resulta complicada pues son la suma de una gran cantidad de otros parámetros en los que intervienen diversos sistemas del vehículo.

- Masa total: 290.299 [kg]

Esta puede ser tomada del prototipo anterior o puede contemplar la estimación de pérdida o ganancia de masa que el equipo haga, para fines de este trabajo se toma la del vehículo de FSAE del 2019.

- Masa del piloto: 70 [kg]
- Distribución de carga delantera/trasera: 48.281/51.719%
- Altura del centro de gravedad: 29.372 [cm]
- Gravedad: $9.8.1 \left[\frac{m}{s^2} \right]$

También son necesarios el IR de cada eje y el gradiente de *roll* deseado, el primero es la relación entre el desplazamiento del resorte y el de llanta, este afecta la magnitud de las fuerzas que experimentan los brazos de suspensión, el chasis y el knuckle para evitar perjudicar a alguno se ocupará un $IR = 1$, lo cual es posible utilizando el recorrido completo del amortiguador el cual es de 5 [cm] con el desplazamiento mínimo del neumático reglamentado de 5 [cm], el segundo será 1.5 [°/g] por las recomendaciones de Milliken [30].

3.4: PROGRAMACIÓN DEL ALGORITMO GENÉTICO PARA DETERMINAR PARÁMETROS PRIMITIVOS

Para comprobar que fuera posible la optimización de las variables deseadas por medio de las funciones anteriores utilizando la metodología de algoritmos genéticos, se desarrolló un programa en lenguaje Python®, comúnmente utilizado en aplicaciones de inteligencia artificial. Durante el desarrollo de este trabajo se utilizó la versión 3.6.8 del intérprete de Python® y Spyder® 3.3.4 como editor de código.

Se utilizó la metodología de COGA-II, *Compressed-objective genetic algorithm II*, presentada en el trabajo de Boonlong [19], esta metodología puede ser vista como la transformación de un problema con M objetivos a uno con un único objetivo, la maximización de la puntuación ganadora.

Además de los operadores descritos en el trabajo de Boonlong [19], se agregó una función para revisar la factibilidad de cada individuo, si el individuo no es factible se repite el operador hasta que este sea realizable, si después de intentar 350 veces toma el individuo original. El número anterior de iteraciones fue determinado experimentalmente al realizar múltiples ensayos, donde fue notado que al rebasar dicho número llegaba a miles de iteraciones sin cumplir con la función de factibilidad. La función anterior es necesario debido a que solo los límites de uno de los parámetros son fijos, los demás dependen de este o

de alguno que debe de ser calculado previamente. La Ilustración 1-20 muestra el diagrama simplificado del procedimiento de COGA-II.

Se prefirió una codificación real en lugar de binaria, pues de haber utilizado el segundo método sería necesario decodificar cada individuo después de cruzarlo o mutarlo para revisar su factibilidad y en caso de que no fuera realizable se tendría que volver a codificar y repetir dicho proceso, gastando así más tiempo de cómputo. Evitar lo anterior con codificación binaria es posible con una función de factibilidad utilizando números binarios, pero esta sería más complicada de entender y en su caso de modificar.

El *wheelbase*, *tracks* y rigideces a *roll* se encuentran en un intervalo continuo de números positivos, pero para la altura de centros de *roll* es necesario utilizar números negativos el uso de codificación binaria hubiera complicado esta y aumentado la longitud del cromosoma.

Respecto al cálculo de aptitudes de los individuos, este es realizado con las funciones previamente expuestas en la sección de funciones objetivo, sin ser ponderadas, por lo que se genera un vector en \mathbb{R}^5 para cada individuo.

El operador de cruce utilizado fue BLX- α con un valor de 0.5 mientras que la mutación fue no uniforme, debido a que mostraron mejores resultados en la comparación de operadores realizada por Herrera et. al. en [28]

3.5: DISMINUCIÓN DEL ERROR DE LA RAZÓN DE INSTALACIÓN

Para lograr el IR deseado se propone la utilización de un programa en Wolfram Mathematica 11.1[®] que modifica las distancias entre el punto del balancín a la push-rod y al amortiguador al igual que el ángulo entre estos dos, esto se muestran en la Ilustración 3-11.

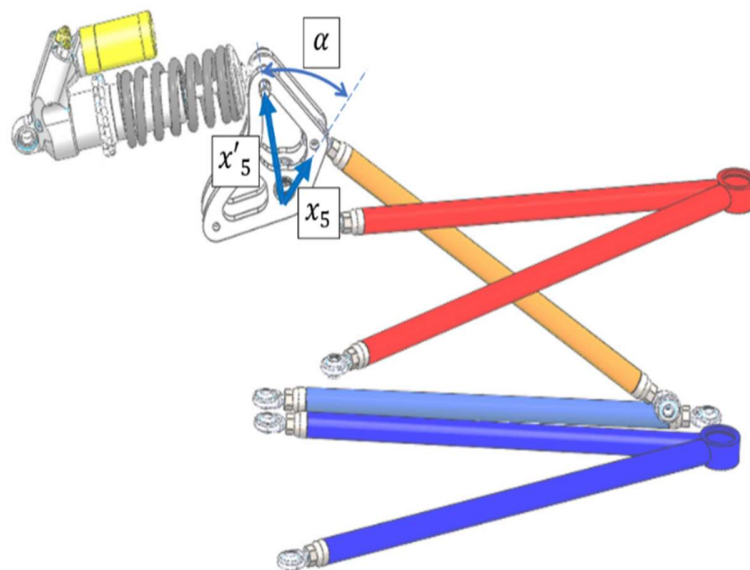
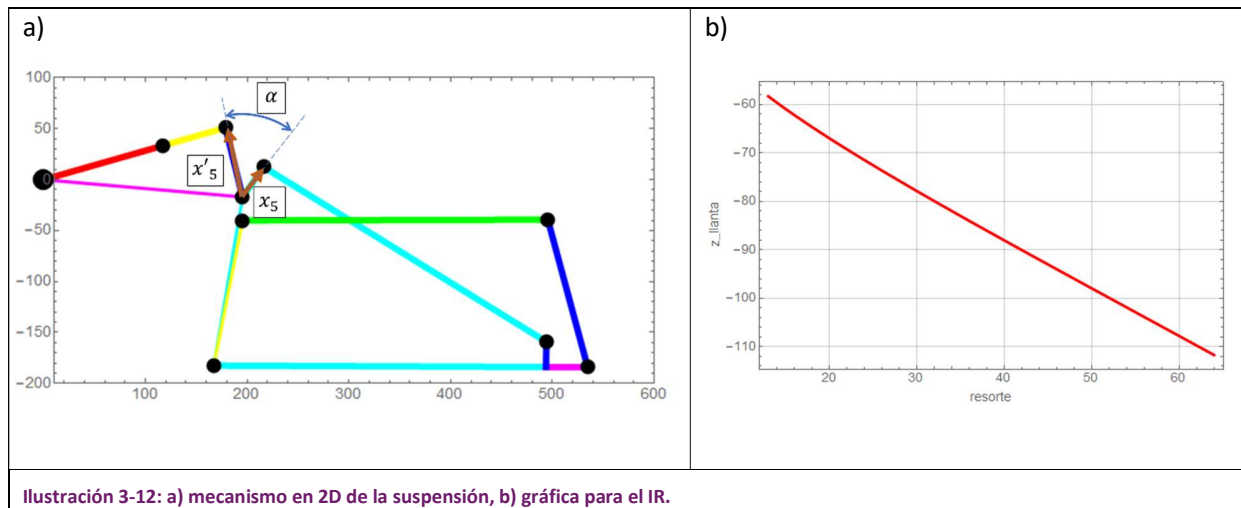


Ilustración 3-11: Dibujo auxiliar para la optimización del IR

Posteriormente, se calcula el movimiento del mecanismo de suspensión en 2D para determinar el "IR" de ese conjunto de valores, lo anterior se ejemplifica en la Ilustración 3-12.



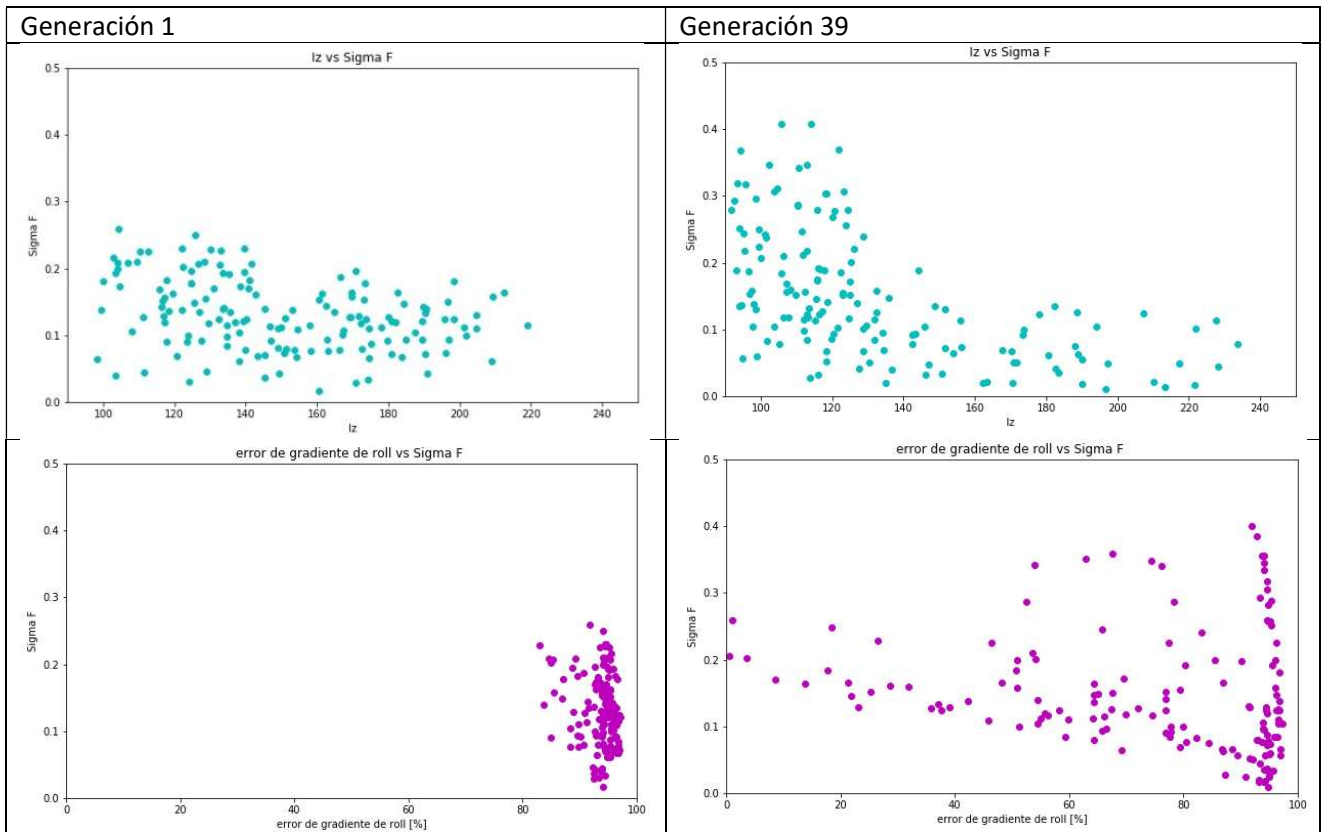
Se emplea el procedimiento de un algoritmo genético para modificar los valores, este tiene como objetivo minimizar el error al IR deseado. De manera similar que para el caso de los parámetros primitivos, se utilizó codificación real y los mismos operadores de cruce y mutación. Dado que este algoritmo de programación no es multiobjetivo, no fue necesario un ranking para el ordenamiento de los individuos. En su lugar, se utilizó la aptitud sin ninguna ponderación. Finalmente, en lugar de que la condición de término fuera por convergencia, se utilizaron dos condiciones para detener el programa:

- 1) Que el programa llegara a la generación máxima previamente estipulada.
- 2) Que encuentre "n" individuos con un error menor al establecido.

Para fines de este trabajo se proponen 30 individuos y 0.001 como tolerancia aceptable. Se propusieron un máximo de 100 generaciones con 50 individuos cada una. Y con base en la condición de término que se haya obtenido (presentado), el diseñador podrá seleccionar un individuo óptimo en los dos posibles resultados que genera el programa: una lista de individuos de la última generación ideales o un conjunto de individuos que se encuentren dentro de la tolerancia propuesta. La última parte del programa permite animar el mecanismo del individuo deseado. El algoritmo completo se encuentra en el Anexo 5.

Capítulo 4: RESULTADOS

Con fines comparativos se presentan las gráficas de I_z contra σ_F , así como el error de gradiente de Roll contra σ_F . En las gráficas, se puede observar el aumento de individuos en valores menores de I_z y error de gradiente de roll, respectivamente.



Además de las gráficas el algoritmo provee de un archivo Excel® con los datos de los individuos de la última generación. Éste incluye la puntuación ganadora (*winning score*) y la distancia euclidiana entre el conjunto de soluciones. Para facilitar la selección de un individuo se dio un formato condicional a la columna de WS (puntuación ganadora) y distancia euclidiana para indicar visualmente la competitividad del individuo. Como se puede observar en la siguiente imagen, se seleccionó el individuo siete (7) como caso óptimo dado que tiene una distancia euclidiana aceptable respecto a los individuos con puntuaciones ganadoras similares, como se puede observar en la Ilustración 4-1

ID	WB [mm]	Front track [mm]	Rear track [mm]	Front RC [mm]	Rear RC [mm]	Front phi [Nm/°]	Rear phi [Nm/°]	WS	Distancia Euclidean
13	1599.0562	1134.536049	1201.652423	-115.6593425	-116.6810273	484479.1102	17661.73352	57.9299	21.48343842
97	1546.4456	1134.746974	1198.753616	-111.2851327	-106.9254459	588620.1506	191862.5689	54.814	24.15809791
70	1614.1142	1135.551039	1181.560904	-99.72425173	-96.40028822	605485.7803	99842.05726	54.5925	23.59755316
48	1704.7413	1121.794723	1180.317338	-102.6662372	-80.587282	547627.6922	52595.67732	53.0139	22.34054801
6	1784.4489	1124.822206	1190.292222	-120.2513841	-85.65207676	434022.6607	15806.69923	45.3765	21.37261513
7	1569.518	1124.464094	1175.0037	-109.8522984	-83.27939097	141945.4545	16006.4974	43.9454	17.31731132
3	1789.2038	1126.392113	1190.321529	-120.3871671	-122.9992887	330281.8143	15597.57289	42.5977	19.72138721
63	1839.2384	1137.366395	1189.283152	-114.3692099	-85.49366269	429701.4735	79115.80324	40.5472	22.39290262
67	1682.8253	1128.814714	1214.710879	-83.11374303	-111.4268695	365151.6971	86324.48759	37.7031	17.92849644
12	1539.7405	1130.679468	1177.296013	-109.9182986	-96.26947732	91500.26126	17491.2477	36.1626	18.03729903
84	1806.2986	1149.389294	1211.415494	-120.5522757	-123.8963732	417389.0586	132573.6254	35.9958	20.85291656

Ilustración 4-1: Tabla de resultados del algoritmo genético.

A continuación, se presenta una tabla comparativa de los valores del vehículo del UNAM Motorsport del año 2019 y de los valores propuestos.

	2019	Propuestos	Diferencia
Wheelbase [m]	1.6556	1.5695	-0.0861
Track delantero [m]	1.2446	1.1245	-0.1201
Track trasero [m]	1.2192	1.1750	-0.0442
Rigidez a roll delantera [Nm/rad]	45,802.5899	141,945.454	+96,142.8641
Rigidez a roll trasera [Nm/rad]	46,946.2701	16,006.497	+30,939.8031
Centro de roll delantero [mm]	-32.181	-109.852	-77.671
Centro de roll trasero [m]	-10.963	-83.279	-72.316

Con objeto de visualizar la forma geométrica que tomaría la suspensión con los valores propuestos se muestran

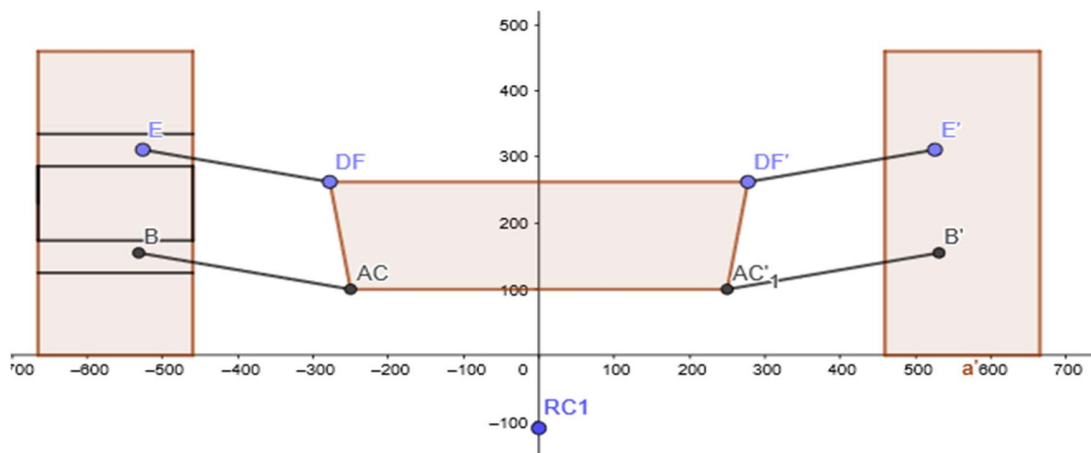


Ilustración 4-2: Propuesta para suspensión delantera

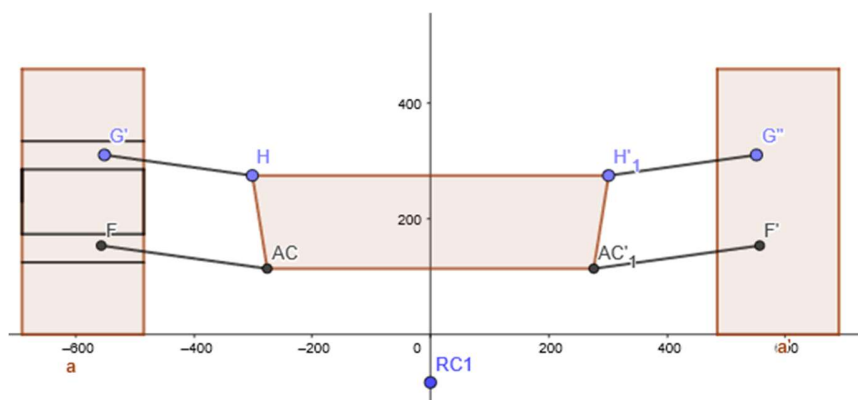
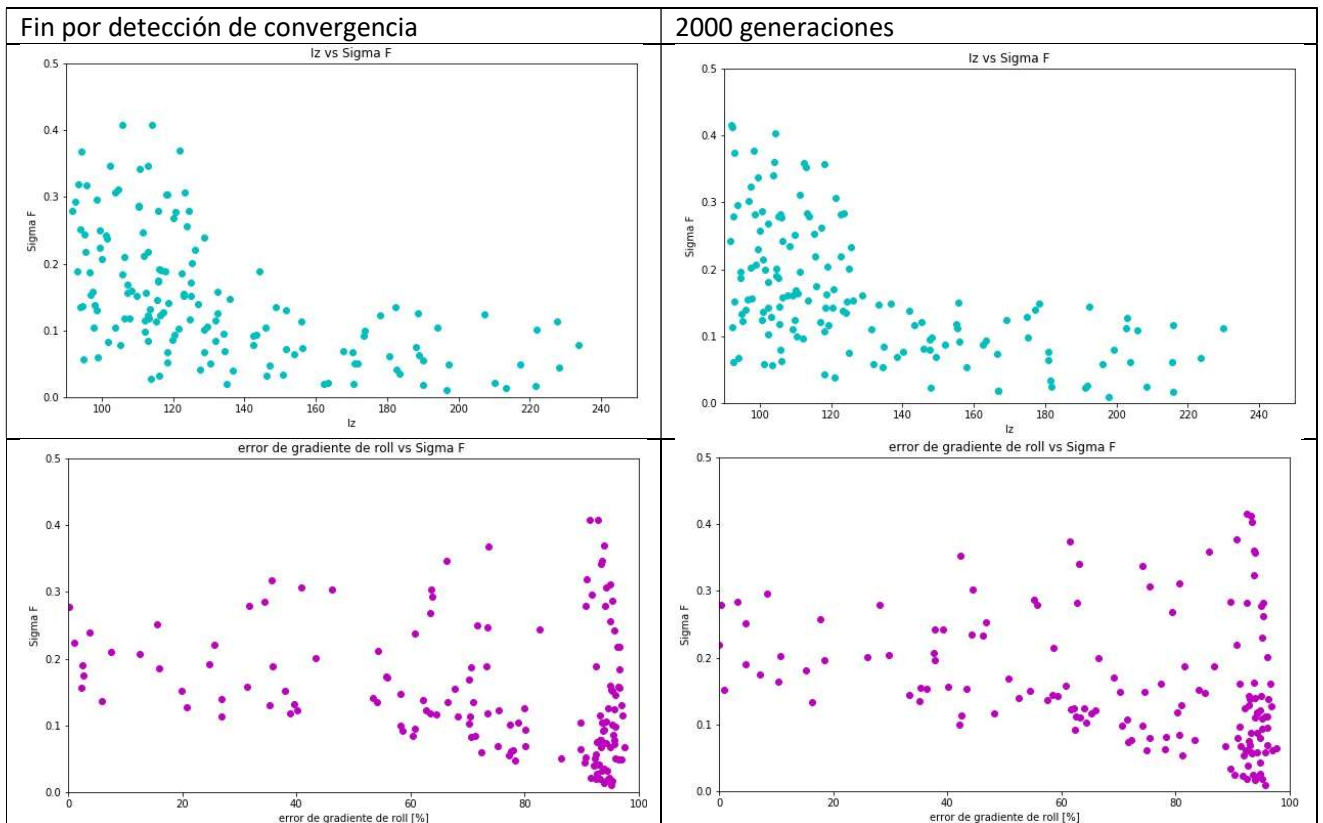


Ilustración 4-3: Propuesta de suspensión trasera

En las ilustraciones Ilustración 4-2 e Ilustración 4-3 se observan líneas horizontales en las llantas izquierdas, las interiores representan el espacio para el rodamiento y su carcasa, mientras que las exteriores representan el espacio interior del rin.

Como un medio para comprobar que la función de convergencia no esté deteniendo el algoritmo de forma prematura, se realizaron una corrida del algoritmo a 2000 generaciones. Esto permitió comparar la tendencia que describen las gráficas de distribución.



Capítulo 5: DISCUSIÓN Y CONCLUSIONES

Del resultado iterativo del algoritmo propuesto, la tendencia descrita en las gráficas anteriores sugiere que existe una gran similitud entre las generaciones 2000 y 39. Como un medio para analizar de forma más clara esta tendencia, se construyeron los histogramas de cada función objetivo en ambas generaciones, los cuales se muestran a continuación:

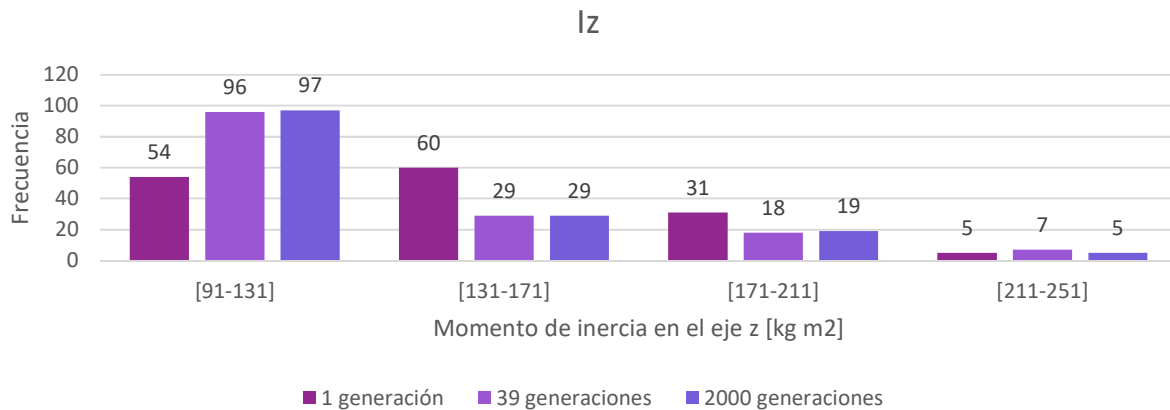


Ilustración 5-1: Histogramas del momento de inercia de iteraciones con 1, 39 y 2000 generaciones.

Para obtener un vehículo maniobrable, se recomienda que éste tenga un I_z bajo; es decir, los individuos que se encuentran en rangos bajos de I_z se pueden considerar como mejores individuos. Al observar la gráfica anterior, se puede observar que aumentan más de un 40% los individuos que se encuentran dentro del rango de 91 a 131 [kg m²] de la iteración con una sola generación a las iteraciones de 39 y 2000 generaciones.

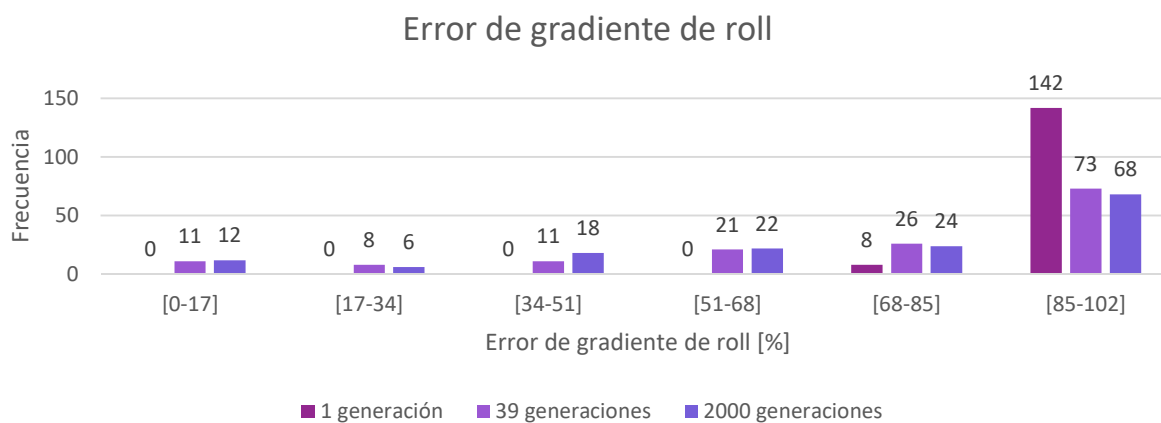


Ilustración 5-2: Histogramas del error de gradiente de roll de iteraciones con 1, 39 y 2000 generaciones.

Se busca el menor error al gradiente de roll, siguiendo las recomendaciones de diseño de la literatura [30], pero en la gráfica anterior, Ilustración 5-2, se observa que la mayoría de los individuos se encuentran en el último rango, 85-102 [%], sin embargo, en las iteraciones con 39 y 2000 generaciones se logra encontrar al menos diez individuos en el primer rango, 0-17 [%].

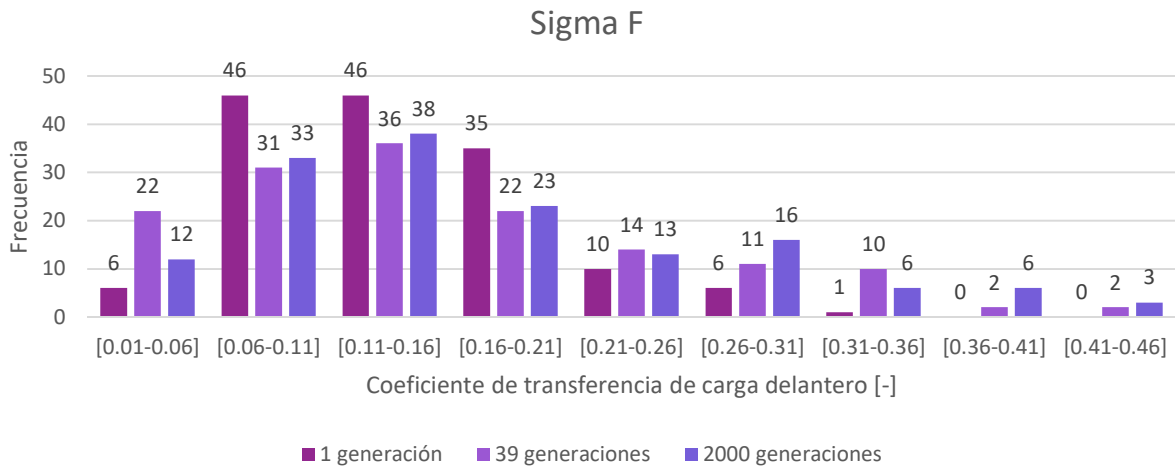


Ilustración 5-3: Histogramas del error de gradiente de roll de iteraciones con 1, 39 y 2000 generaciones.

Un menor valor de σ_F provee de una menor transferencia de carga, lo que causa una mayor fuerza lateral de los neumáticos, permitiendo una mayor aceleración lateral. Si se observa la gráfica, Ilustración 5-3, se puede notar que en el primer rango, 0.01-0.06, existe una mayor cantidad de individuos de las iteraciones con 39 y 2000 generaciones que los individuos en la iteración con una sola generación, sin embargo en el segundo rango existe una mayor concentración de individuos en la primer iteración que en las dos restantes. Respecto al resto de los rangos se nota una gran similitud entre 39 y 2000 generaciones.

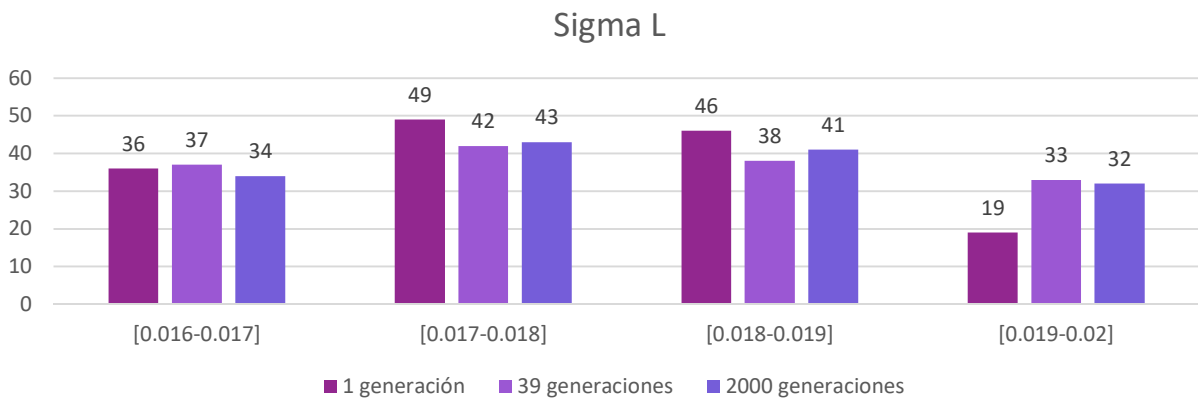


Ilustración 5-4: Histogramas del error de gradiente de roll de iteraciones con 1, 39 y 2000 generaciones

En los histogramas del coeficiente de carga longitudinal no se distinguen diferencias significantes entre las iteraciones con 1, 39 y 2000 generaciones, esto probablemente debido al pequeño rango en el que se permitió variar el *wheelbase*, para evitarlo se puede aumentar el valor superior de este.

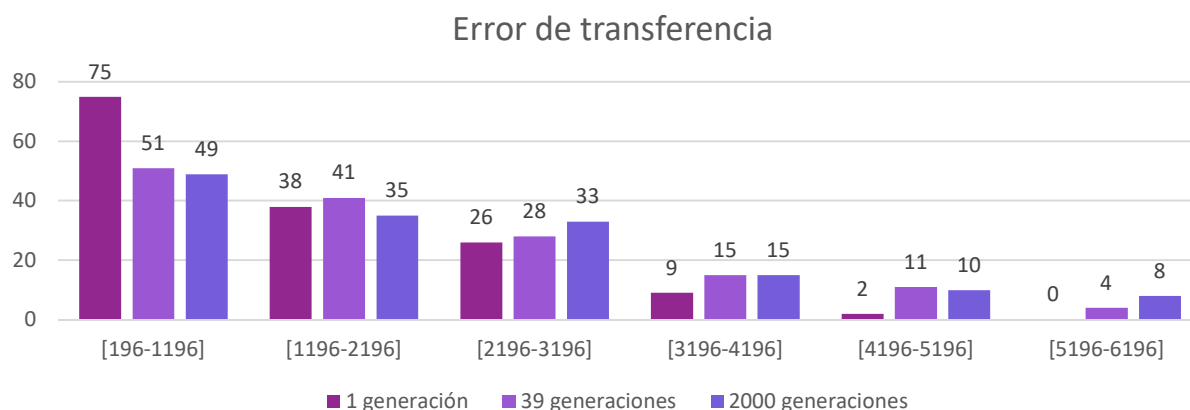


Ilustración 5-5: Histogramas de error de transferencia de iteraciones con 1, 39 y 2000 generaciones

En el objetivo de reducción de diferencia en las cargas de los neumáticos debido a la transferencia de carga es la única función objetivo que muestra un comportamiento inferior en las iteraciones con 39 y 2000 generaciones, pero después de la selección del individuo ganador se puede modificar manualmente la distribución de la rigidez a roll de ambos ejes para mejorar este parámetro.

Con base a los histogramas anteriores se puede concluir que los resultados de la iteración detenida al detectar una convergencia; las iteraciones con 39 generaciones y con 2000 generaciones, éstas tienen un comportamiento muy semejante, por lo que se concluye que la función de detección de convergencia reduce el tiempo de computo sin comprometer los resultados.

Fueron realizadas diferentes corridas de prueba, en estas se observó que el algoritmo es interrumpido por la función de detección de convergencia entre las generaciones 20 y 90, lo que indica que la velocidad de convergencia depende de la aptitud de la primera generación, la cual es aleatoria, de estas pruebas también se concluyó que para reducir aún más el tiempo de computo la detección de convergencia puede comenzar a partir de la vigésima generación.

Dependiendo de la cantidad de generaciones el programa puede demorar de 5 a 20 minutos, lo cual representa una reducción importante del tiempo de decisión de los valores primitivos, debido a que esto tomaba un par de días.

Con fines de comparación se introdujo al individuo con los valores del vehículo de UNAM Motorsport del 2019 a la última generación, para el cálculo de la puntuación ganadora y distancia euclidiana dentro de esta, y así poder comparar los resultados con el del individuo seleccionado. Es importante remarcar que la puntuación ganadora es una función diseñada para comparar un grupo de individuos y asignar valores más altos a los que presenten una mejor aptitud. La puntuación ganadora del individuo seleccionado fue de 37.7, mientras que el del individuo diseñado por medio del procedimiento tradicional es fue de -34.6. Mientras que la distancia euclidiana promedio es muy similar, 17.8 y 17.5 para el seleccionado con el proceso propuesto y el del proceso tradicional realizado para el prototipo del 2019 respectivamente.

A partir de los datos provistos por el programa se logró un diseño de suspensión delantera y trasera que cumple con las especificaciones de la competencia y las restricciones del equipo de FSAE de la UNAM, por lo que es posible utilizar esta metodología para reducir el tiempo necesario para el diseño de una suspensión tipo doble brazo de A de un coche tipo fórmula, a pesar de que es un buen modelo este puede ser mejorado por medio de la inclusión de:

- El fvsa, para poder modificar la ganancia de *camber* y tener mayor libertad en la posición del RC.
- Considerar el cambio de peso por el cambio de dimensiones.

Además de la inclusión de una interfaz gráfica de usuario para facilitar el uso del programa.

ANEXO 1: FÓRMULA SAE

Es una competencia estudiantil con diferentes sedes alrededor del mundo. Su objetivo es desafiar a estudiantes para diseñar, fabricar, desarrollar y competir con vehículos tipo fórmula [33]. Las pruebas durante la competencia se dividen en dos grupos, estáticas y dinámicas que otorga un máximo de 325 y 675 puntos respectivamente.

Las pruebas estáticas son 3, presentación de negocios, reporte de costos y evaluación del diseño, ésta última comienza antes de la competencia per se con el envío de dos documentos, reporte de diseño (design report) y la hoja de especificaciones (spec sheet). El primero además de tener dibujos del vehículo debe de incluir la descripción de las características del vehículo, análisis y pruebas. En el segundo se debe de completar un documento provisto por la competencia con los valores de las especificaciones más relevantes de todos los sistemas. Dicha prueba termina con la evaluación de los jueces respecto al cumplimiento de los objetivos de diseño, eficiencia de costos y entendimiento del diseño, por lo que los miembros del equipo están presentes y les es permitido llevar material de soporte que ayude a la presentación y discusión del desarrollo del vehículo. Esta presentación es calificada con base en la hoja de puntuaciones del evento de diseño (Formula SAE Design Judging Score Sheet) provista por la competencia, esta tiene 8 categorías, solo en una de ellas el sistema de suspensión tiene una influencia total y puede proveer de 25 puntos, en otras 3 tiene una influencia indirecta fabricabilidad (*manufacturability*), estética y creatividad, estas proveen un total de 30 puntos.

Después de las pruebas estáticas se realiza una inspección técnica con el objetivo de revisar el cumplimiento del reglamento en el vehículo, a pesar de que esta prueba no es puntuada de no ser aprobada no le es permitida la participación al equipo en las pruebas dinámicas.

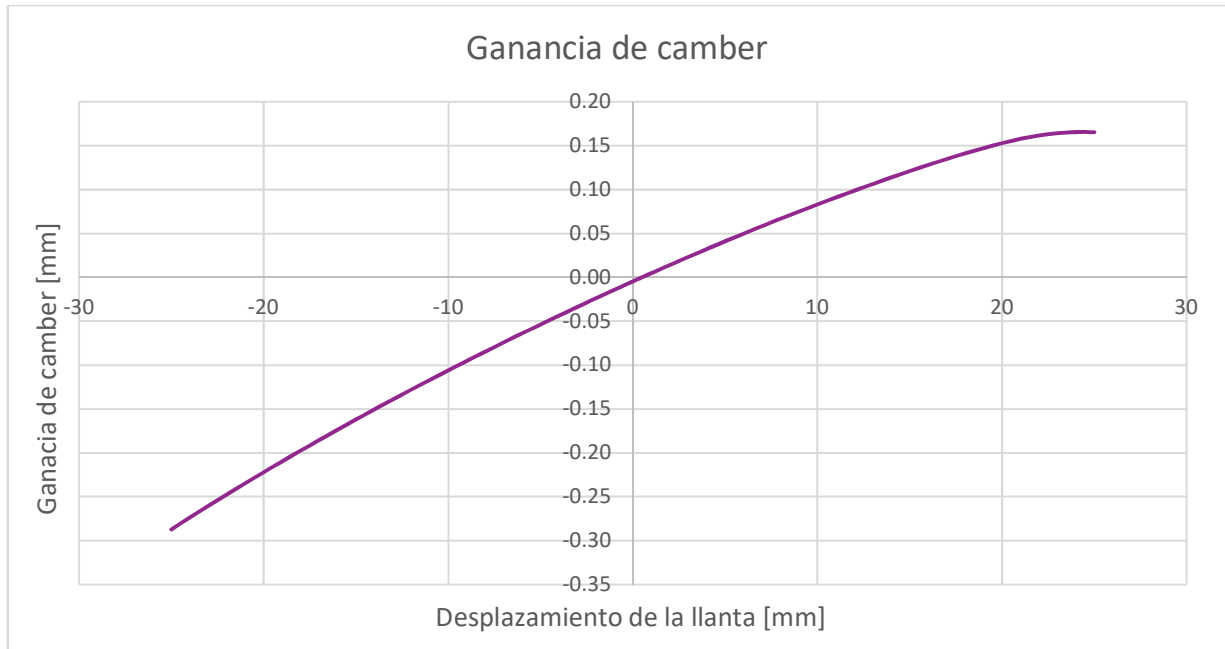
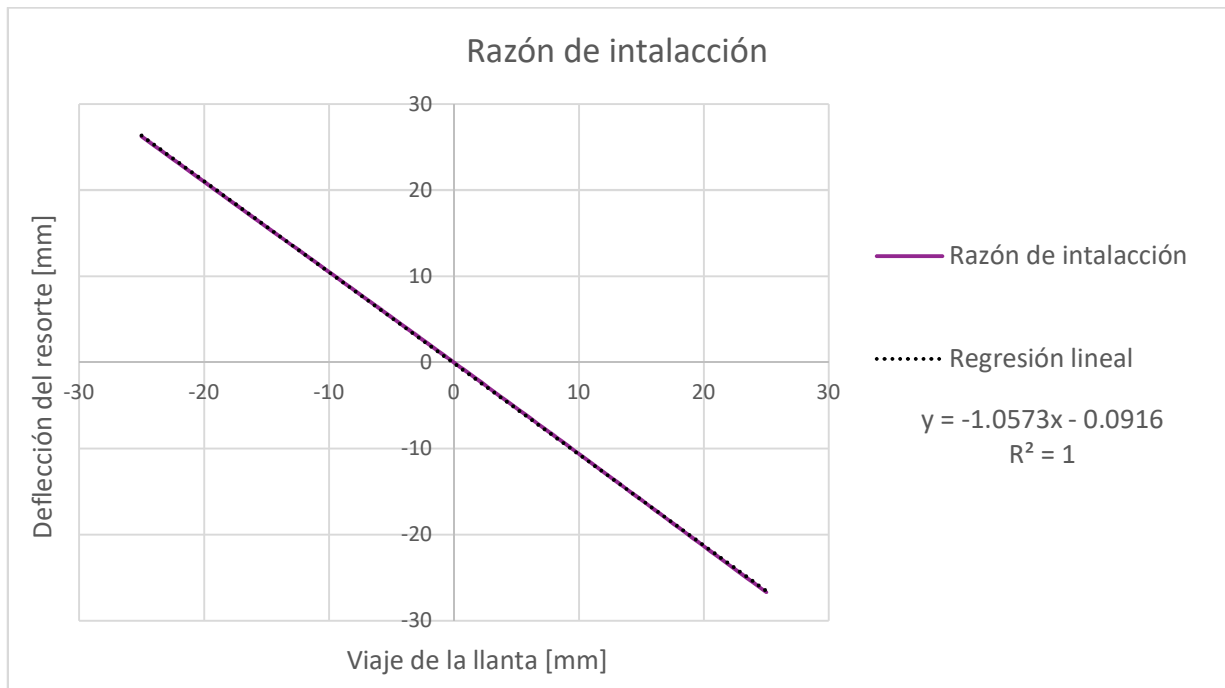
Las pruebas dinámicas son 5:

- Aceleración: recta de 75 [m] con un ancho de 4.9 [m].
- *Skipad*: pista en forma de 8, el diámetro interior de ambas circunferencias es de 15.25 [m] con ancho de 3[m].
- *Autocross*: Serie de rectas, curvas y zigzagueos con un ancho de pista mínimo de 3.5[m] y diámetro mínimo de 9 [m].
- Resistencia (*Endurance*): Pista similar a la de *autocross* pero diseñada para una velocidad promedio mayor y es recorrida en varias ocasiones hasta llegas a 22 [km].
- Eficiencia de energía: El consumo de combustible es medido durante la prueba de resistencia.

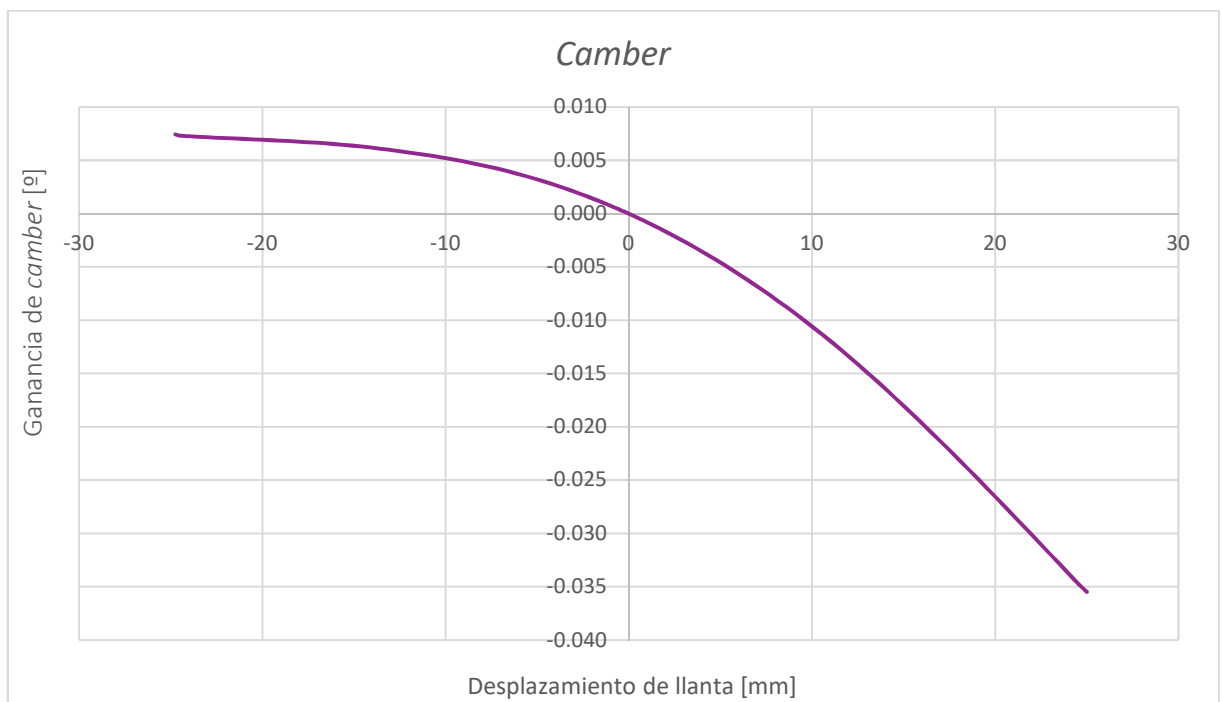
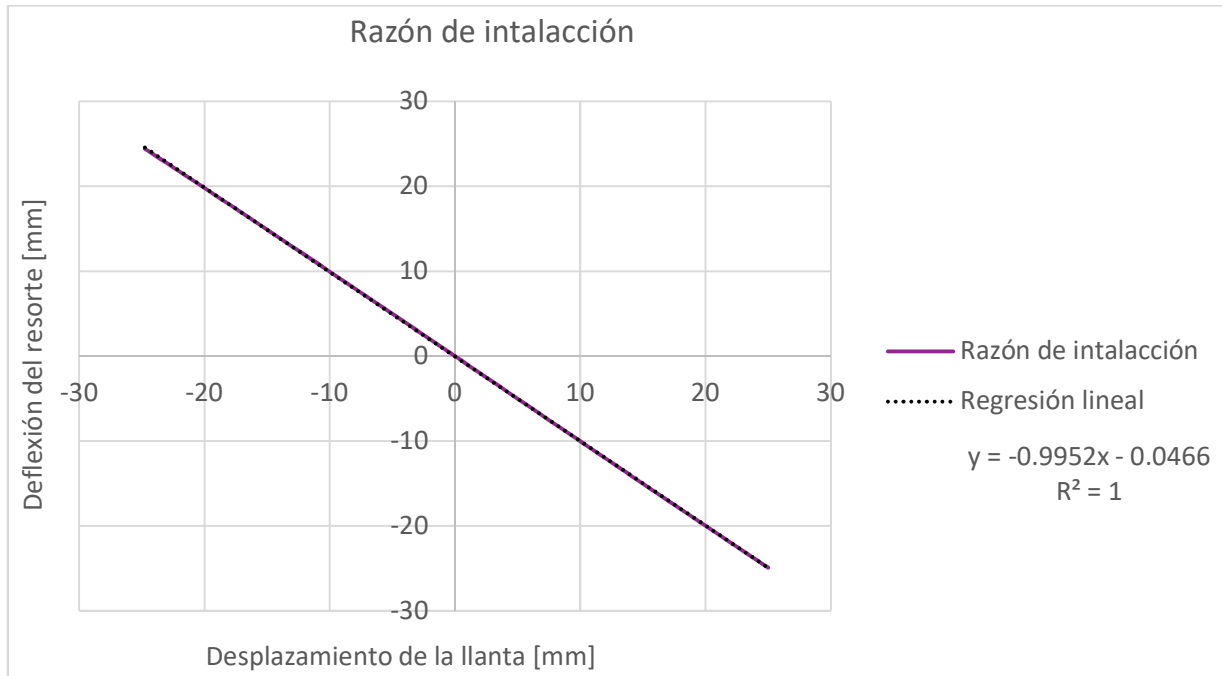
ANEXO 2: SUSPENSIÓN 2017

Punto	x	y	z
Suspensión delantera			
Eje de balancín	27.89	214.053	506.680
Centro del balancín	27.94	217.590	510.220
Chasis – amortiguador	192.38	153.78	574.22
Amortiguador – balancín	17.50	153.78	574.22
Inferior interno delantero	-115	220	136
Inferior externo	-6.1	586	145
Inferior interno trasero	115	220	136
Push – balancín	-33.52	225.82	502.68
Push – brazo inferior	-6.1	586	145
Tie rod interior	-3.18	131.0	158
Tie rod exterior	35.3	570.79	167.07
Superior interno delantero	-115	240	301
Superior externo	6.1	586	300
Superior interno trasero	115	240	301
Suspensión trasera			
Eje de balancín	1624.1	234.5	308
Centro del balancín	1619.1	234.5	308
Chasis – amortiguador	1619.1	40	325
Amortiguador – balancín	1619.1	214	376
Inferior interno delantero	1430.4	245	143
Inferior externo	1619.1	576.6	149.05
Inferior interno trasero	1623.2	235	143
Push – balancín	1619.1	254.6	339
Push – brazo inferior	1619.1	576.6	149.05
<i>Track rod</i> interior	1656.6	235	147.4
<i>Track rod</i> exterior	1660	590	152.4
Superior interno delantero	1356.7	262	309
Superior externo	1632.1	576.6	314.15
Superior interno trasero	1554.4	252	309

Suspensión delantera



Suspensión trasera



Resultados del programa de razón de roll y ride:

$k_s \left[\frac{lb}{in} \right]$	$\omega_F [Hz]$	$\omega_R [Hz]$	$K_{WF} \left[\frac{N}{m} \right]$	$K_{WR} \left[\frac{N}{m} \right]$	$K_{RF} \left[\frac{N}{m} \right]$	$K_{RR} \left[\frac{N}{m} \right]$	$c_{\phi F} \left[\frac{Nm}{\circ} \right]$	$c_{\phi R} \left[\frac{Nm}{\circ} \right]$	$\nabla roll \left[\frac{\circ}{g} \right]$
150	2.9085	2.9543	24920	26637	19984	21073	336.86	345.52	-1.0782
250	3.5291	3.5733	41533	44365	29422	30830	561.43	575.87	-0.6469
350	3.9515	3.9911	58146	62152	36888	38461	786	806.22	- 0.46207
450	4.2635	4.2975	74759	79910	42942	44593	1010.6	1036.6	- 0.35939
600	4.6071	4.6328	99678	106550	50142	51823	1347.4	1382.1	- 0.26954

$k_s \left[\frac{lb}{in} \right]$	$\nabla roll \left[\frac{\circ}{g} \right]$	$c_{\phi F_{ARB}} \left[\frac{Nm}{\circ} \right]$	$c_{\phi R_{ARB}} \left[\frac{Nm}{\circ} \right]$
150	-1.0782	226.67	195.94
250	-0.6469	114.38	75.915
350	-0.46207	2.0951	-44.109
450	-0.35939	-110.19	-164.13
600	-0.26954	278.61	-344.17

ANEXO 3: DISEÑO GEOMÉTRICO DE UNA SUSPENSIÓN DOBLE BRAZO DE A

En la ilustración 3-8 se muestran los puntos básicos para el diseño de una suspensión, como dentro de los objetivos de esta tesis no está la optimización del movimiento de pitch se ocupará solamente la vista frontal y los puntos internos serán considerados como uno solo, es decir los puntos D y F serán el punto DF mientras que A y C pasarán a ser AC, como se observa en la Ilustración 3-9., lo anterior significa que se considerará la altura de los puntos D y F idénticas, misma condición para A y C, debido a esto no existe una configuración anti-sumergirse (anti-dive) o anti-agacharse (anti-squat).

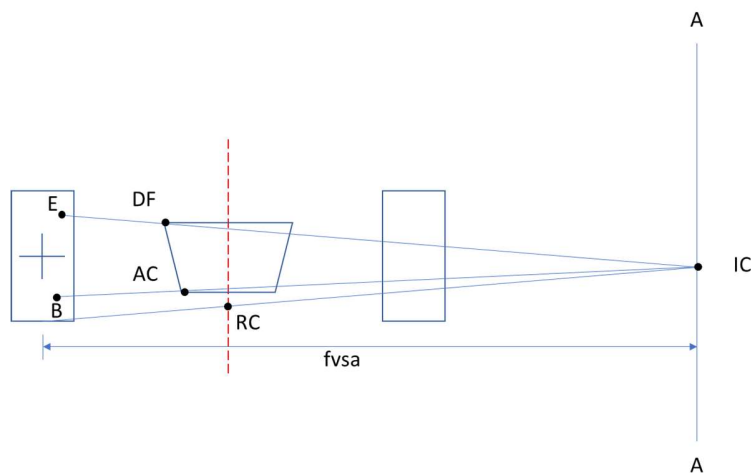


Ilustración 3-9 Cálculo de la altura mínima del punto AC.

Al observar los vehículos de competencia se puede notar que son más bajos, es decir su piso se encuentra más cerca del pavimento que su similar de calle, esto para proveer mayor estabilidad y aprovechar los efectos aerodinámicos. Debido a que el diseño de esta tesis toma como ejemplo un vehículo para la competencia de FSAE se seguirá la tendencia de un piso bajo, entonces el primer paso del diseño será calcular la altura mínima del punto AC conforme a las restricciones de la competencia y el espacio para los componentes aerodinámicos. Para terminar de ubicar el punto AC es necesario conocer la “coordenada y” la cual se encuentra definida por el espacio del habitáculo.

Después se propone una altura para el centro de *roll*. Para una suspensión simétrica el centro de *roll* se encuentra en el plano XZ, es decir en el centro del vehículo lo cual hace que este punto quede totalmente definido en la vista frontal.

De acuerdo a la literatura, la fuerza lateral provista por las llantas aumenta de forma lineal con el aumento de “*camber*” hasta un punto donde comienza a bajar de manera abrupta [32], dicho valor depende de las características de cada neumático, si este valor es conocido se puede calcular una ganancia de *camber* tal que permita proveer la mayor fuerza lateral, es decir sin que se sobrepase dicho valor ni llegue a positivos.

Conociendo la ganancia de *camber* deseada ahora se puede calcular el brazo de oscilación en la vista frontal, *fvsa*, por las siglas en inglés, con la siguiente fórmula:

$$f_{vsa} = \frac{1}{\tan(\text{ganancia de camber})} \quad \text{Ec. 0.1}$$

Además de estar relacionado con la ganancia de *camber* el f_{vsa} es la distancia de centro de la huella de contacto de la llanta con el centro instantáneo de rotación de la suspensión IC, por sus siglas en inglés, por consiguiente, es posible generar un eje A-A a una distancia igual al f_{vsa} sobre el cual se encontrará el IC.

El RC es encontrado con la intersección entre las líneas creadas entre el IC de cada lado y el centro de la huella de contacto de su llanta [32]. Ahora bien, en lugar de encontrar el RC por medio de los centros instantáneos de rotación se encontrará el IC por medio de la intersección entre el eje A-A y la línea generada entre el RC y el centro de la huella de contacto del neumático.

Esta ahora se ha encontrado el IC, el RC y uno solo de los puntos de la suspensión, por lo cual se prosigue a encontrar los demás.

Para el punto B se genera una línea con el IC y el punto AC, la cual se prolonga hasta interceptar con el plano de USM (*unsprung mass*), por lo cual es indispensable conocer las generalidades del empaquetamiento de este, en específico la distancia a la que se encuentra el centro del rodamiento del centro de la llanta.

Para el punto E es decir el punto exterior del brazo superior, se seguirá la recomendación de una distancia grande de *kingpin* para disminuir la fuerza en los brazos de suspensión [30] por lo que la altura del punto E será limitada por el rin y su “coordenada y” por el *camber* estático previamente definido para provechar la mayor cantidad de fuerza lateral.

La relación entre los tamaños de los brazos determina la forma de la curva de *camber a ride* [30].

Brazo superior respecto al inferior	Curva de <i>camber</i>
Igual	Recta
Más largo	Convexa con curvatura hacia <i>camber</i> positivo
Más corto	Cóncava con curvatura hacia <i>camber</i> negativo

La curva ideal tiene *camber* negativo progresivo con el desplazamiento hacia arriba de la llanta y menos con el desplazamiento hacia abajo [30], por lo que como primera iteración podemos proponer un punto cualesquiera sobre la línea que une E con IC, siempre y cuando se mantenga que el brazo superior sea más corto y se respete el espacio interior reglamentado del habitáculo.

Finalmente se debe recordar que la proyección de la línea que une los puntos de la *tie/track-rod* debe pasar por el IC, para evitar que las llantas giren con un desplazamiento vertical sin que el piloto gire el volante, dicho efecto es indeseado.

ANEXO 4: PROGRAMA PARA OPTIMIZACIÓN DE PARÁMETROS PRIMITIVOS DE SUSPENSIÓN

```

from random import randint, randrange,
uniform, choice
from math import sqrt, pi, floor
from copy import copy, deepcopy
import matplotlib.pyplot as plt
import pandas as pd

# Datos para el genético
maxGen = 20000 #Generaciones máximas
minGen = 20
popRecordLen = 100 #Cantidad de generaciones
a recordar, imágenes a generar al final
popLen = 150
imagdirectory = "imagenesP7"
resultfile = "ResultadosP7"
#Datos del coche
hcg = 0.293724 #altura del cg en m
m = 290.299 #masa del coche con piloto en kg
mp = 70 #masa del piloto en kg
mu = 10.178 #1/4 de la masa no suspendida en
kg
ms = m - 4*mu #masa suspendida en kg
mF = 140.16 #masa frontal en kg
mR = m - mF
#Aero
AE_F = 0.7 #Espacio delantero para
aerodinámica
AE_R = 0.25 #Espacio trasero para aerodinámica
#Datos de las llantas
tdim = 18.1*0.0254 #diámetro de la llanta en m
tw = 8.1*0.0254 #ancho de la llanta en m
kt = 104238.4868 #Rigidez de la llanta en N/m
#rigidez de los resortes
ks_min = 26269.02525
ks_max = 105076.101
#límites de WB y tracks
l_min = 1.53 #WB mínimo en m
l_max = 1.84 #WB máximo en m
tF_min = 1.12 #track delantero mínimo en m
tR_min = 1.17 #track trasero mínimo en m
#Datos de la pista
rho_ext = 4.5 #radio externo de la pista
rho = 2.75 #Radio al centro pista de skipad del
reglamento en m
#Densidades
rho_p = 950 # Densidad de una persona[kg/m3]
rho_s = 7850 # Densidad del acero [kg/m3]
#Gradiente de roll deseado
roll_gradient_Des = -0.02618 #rad/g
IR_F = 1 #Installation ratio para el diseño
IR_R = 1
freq_Des = 3 # Frecuencia deseada a la velocidad
V_freq = 44.704
#Cálculos para el lz
#Volumenes
V1 = mp/rho_p + (ms - mp)/rho_s #Volumen del
prisma/chasis
V2 = pi* tdim**2 * tw /4 #Volumen de USM
lzcg2 = mu/12 * (3/4 * tdim**2 + tw**2)
#Momento de las llantas
#gravedad
g = 9.81 #m/s**2
yACminF = 0.25 # "coordenada y" mínima del
punto AC delantero
yACminR = 0.276
zAC = 0.083

# Calculos previos
zBmax =(tdim-dimbea)/2 -0.02
zBmin = (tdim - rinDim)/2 + 0.03
# Calculo de cphi_max
cphi_ARB_max =
(E*pi*dim_ARB**4)/(64*(1+vp)*L_ARB)
a = mR/m * l_min
tF_max = 2*sqrt(rho_ext**2-(a + tdim/2 +
AE_F)**2) - 2*rho - tw - 0.2
cphi_max = 6 * kt*tF_max**2*(2*cphi_ARB_max
+ (IR_F+IR_R)/2** 2
*ks_max*tF_max**2)/(2*(2*cphi_ARB_max+
(IR_F+IR_R)/2 **2
*ks_max*tF_max**2+kt*tF_max**2))

#Genera la población
def PopGen(popsiz):
    #Crea una lista con n individuos con n igual a
    popsiz
    for i in range (popsiz):
        #WB aleatorio

```



```

L =
randint(int(l_min*1000),int(l_max*1000))/1000
#Numero aleatorio para la generación del WB
b = (mF / m) * L # Dist entre el cg y el tR
a = L - b # Dist entre el cg y el tF
#Generación de tF
tFmax = 2*sqrt(rho_ext**2-(a + tdim/2 +
AE_F)**2) - 2*rho - tw - 0.2 #Calcula el máx de
tF dependiendo de L
tF= randint(int(tF_min*1000),int(tFmax*
1000))/1000
#Generación de tR
tRmax= 2*sqrt(rho_ext**2-(b + AE_R +
tdim/2)**2) - 2*rho - tw - 0.2 #Calcula el max de
tR dependiendo de L
if tRmax > tF/0.75:
tRmax = tF/0.75
if tR_min < tF*0.75:
tR_min1 = tF*0.75
else:
tR_min1 = tR_min
tR= randint(int(tR_min1*1000),int(tRmax*
1000))/1000
#Generación de RC delantero y trasero
yBF = tF/2 - 0.031
yBR = tR/2 - 0.031
IC_Fmin = zBmax + (zAC - zBmax)*(tF/2 -
fvsa - yBF)/((tF-tdim - 0.180)/2-yBF)
IC_Rmin = zBmax + (zAC - zBmax)*(tR/2 -
fvsa - yBR)/((tR-tdim - 0.180)/2-yBR)
RC_F_min = (IC_Fmin * tF)/(2*fvs)
RC_R_min = (IC_Rmin * tR)/(2*fvs)
IC_Fmax = zBmin + (zAC - zBmin)*(tF/2 - fvsa
- yBF)/(yACminF-yBF)
IC_Rmax = zBmin + (zAC - zBmin)*(tR/2 -
fvsa - yBR)/(yACminR-yBR)
RC_F_max = (IC_Fmax * tF)/(2*fvs)
RC_R_max = (IC_Rmax * tR)/(2*fvs)
RC_F= randint(int(RC_F_min*100000),
floor(RC_F_max*100000))/100000
RC_R= randint(int(RC_R_min*100000),
floor(RC_R_max*100000))/100000
cphi_F = randrange(int((52119.2*IR_F**2
*tF**2)/(3.9811+IR_F**2)) + 1, int(cphi_max -
1),5)
cphi_R = randrange(int((52119.2*IR_R**2
*tR**2)/(3.9811+IR_R**2)) + 1, int(cphi_max - 1)
,5)

```

```

#Generación de un individuo
pop.append([[L, tF, tR, RC_F,
RC_R,cphi_F,cphi_R], [], 0, 0, 0])
return (pop)
#Cálculo de Fitness de una población
def Fitness(pop):
for ind in pop:
L = ind[0][0]
tF = ind[0][1]
tR = ind[0][2]
RC_F = ind[0][3]
RC_R = ind[0][4]
cphi_F = ind[0][5]
cphi_R = ind[0][6]
#Distancias entre el cg y los g dependiendo
de la distribución de carga
b = (mF / m) * L # Dist entre el cg y el tR
a = L - b # Dist entre el cg y el tF
#-----Calculo de lz-----
x1 = (a*V1 + 4*a*V2 - 2*L*V2)/V1
y2 = tF/2
y4 = tR/2
lzcg1 = ms/12 * ( ( (tF+tR)/2 - tdim - tw)**2
+ L**2 ) #lz del prisma/chasis
Dz1 = x1 - a
Dz2 = sqrt( a**2 + y2**2 )
Dz4 = sqrt( b**2 + y4**2 )
lz1 = lzcg1 + ms * Dz1**2
#-----lz
lz = lz1 + 4*lzcg2 + 2*mu*(Dz2**2 + Dz4**2)
m1 = (RC_R - RC_F)/L
hp = abs (m1 * a - hcg + RC_F) / sqrt(m1**2
+ 1)
SigmaF = (1/tF)*((cphi_F/(cphi_F + cphi_R -
m*g*hp))*hp + (b/L) * (RC_F)) #factor delantero
SigmaR = (1/tR)*((cphi_R/(cphi_F + cphi_R -
m*g*hp))*hp + (a/L) * (RC_R)) #factor trasero
ErrorTrans = abs (1/2*g*(mF - mR)-
m*g*(SigmaF-SigmaR)) + abs (1/2*g*(mF-
mR)+m*g*(SigmaF-SigmaR)) + abs (1/2*g*(mF-
mR)-m*2*g*(SigmaF-SigmaR))+abs (1/2*g*(mF-
mR)+m*2*g*(SigmaF-SigmaR))
SigmaL = hcg/(L*g)
roll_gradient_Cal = (-m*hp*g)/(cphi_F+
cphi_R-m*g*hp)
roll_gradient_error = abs(roll_gradient
_Des-roll_gradient_Cal)*100/abs
(roll_gradient_Des)

```

```

    fit = [Iz, SigmaF, ErrorTrans, SigmaL,
roll_gradient_error]
    ind[1] = fit
    return (pop)
def Dom(fitA,fitB):
    dom = False
    #Revisa cada objetivo
    for i in range(len(fitA)):
        if fitA[i] < fitB[i]:
            #Significa que A domina a B
            dom = True
        if fitA[i] > fitB[i]:
            dom = False
            break
    return(dom)
def DomEq(fitA,fitB):
    domEq = True
    #Revisa cada objetivo
    for i in range(len(fitA)):
        #Si A es menor que B en el objetivo i
        if fitA[i] > fitB[i]:
            domEq = False
            break
    return(domEq)
# PARETO
def Pareto(popWfit):
    #Ordenar una lista de listas
    popWfit.sort(key = lambda x: x[1][0]) #Se
ordena en forma ascendente el 1. objetivo
    noOpt = [] #Se inicializa la lista de no óptimos
    #Agrega el primer elemento a la frontera
    paretoSet = copy([popWfit[0]])
    for j in range(1, len (popWfit)):
        #Se revisa contra cada elemento de la
frontera actual
        for i in paretoSet:
            pareto = True #Se inicializa como
verdadero que pertenece a la frontera
            #Si algún elemento de la frontera domina
al elemento a revisar
            if Dom(i[1],popWfit[j][1]):#los elementos
1 porque ahi esta la lista de fitness
            if pareto == True:
                #Se agrega ese elemento a la frontera
                paretoSet.append(popWfit[j])
            return (paretoSet, noOpt)
#Función para encontrar los ind extremos
def ExtremInd(paretoSet, M, N):

```

```

    ind_ext = [] #inicializamos la lista de ind
extremos
    for i in range (M):
        # ordeno los elementos
        paretoSet.sort( key = lambda x: x[1][i])
        if (paretoSet[0] in ind_ext) == False :
            ind_ext.append(paretoSet[0])
        if (paretoSet[N - 1] in ind_ext) == False :
            ind_ext.append(paretoSet[N - 1])
    return(ind_ext)
#Determina el ranking de los ind dependiendo
cuantos ind los dominan
def RankNoPareto(noOpt, paretoSet):
    for i in noOpt:
        aux1 = 0
        for j in paretoSet:
            if Dom(j[1],i[1]) == True:
                aux1 = aux1 + 1
        rank = len(paretoSet) + aux1
        i[2] = -len(paretoSet) + 1 #Winning Score
        i[3] = rank
    return (noOpt)
def WS(paretoSet):
    N = len (paretoSet)
    M = len (paretoSet[0][1])
    V = []
    for k in range(M): #Para cada objetivo k
        V.append(0) #Se agrega un cero a la lista V
    for i in range(N - 1):
        j = i + 1
        while j < N:
            rhoj = 0
            infe = 0
            sup = 0
            eq = 0
            for p in range(M): #Para cada objetivo
                if paretoSet[i][1][p] < paretoSet[j][1][p]:
                    sup = sup + 1 #Cantidad de objetivos
sup de i sobre j
                if paretoSet[i][1][p] > paretoSet[j][1][p]:
                    infe = infe + 1 #Cantidad de objetivos
inf de i sobre j
                if paretoSet[i][1][p] ==
paretoSet[j][1][p]:
                    eq = eq + 1 #Cantidad de objetivos
iguales entre i y j
            for k in range(M): #Para cada objetivo k
                if paretoSet[i][k] < paretoSet[j][k]:

```

```

    rhoij= (sup + infe)/(2*sup)
    if paretoSet[i][k] > paretoSet[j][k]:
        rhoij= (sup + infe)/(2*infe)
    if paretoSet[i][k] == paretoSet[j][k]:
        rhoij= 1
    V[k]= V[k] + rhoij
    j = j + 1
SumV = sum(V) #Se hace la sumatoria de V
# Ahora calculamos WF de cada k
WFk = []
for v in V:
    WFk.append(v/SumV)
for i in paretoSet: #Calcular el WS de cada ind
    WinScore = 0
    q = 0
    for j in paretoSet: #Para el wij
        for k in range(M):
            if i[1][k] > j[1][k]:
                q = 1
            if i[1][k] < j[1][k]:
                q = -1
            if i[1][k] == j[1][k]:
                q == 0
            wij = WFk[k]*q
            WinScore = WinScore + wij
        i[2] = WinScore
        if i[0][1] >=1.27:
            i[2] = i[2] - N/3
        if i[0][2] >=1.27:
            i[2] = i[2] - N/3
    return (paretoSet)
def Ranking (popWfit):
    aux0 = Pareto(popWfit)
    paretoSet = aux0[0]
    noOpt = aux0[1]
    N = len (paretoSet) #Numero de ind no
    dominados
    M = len (paretoSet[0][1]) #Cantidad de
    objetivos
#-*- Ranking para NO pareto *-*-
RankNoPareto(noOpt, paretoSet)
# Selecciona los indi extremos de cada objetivo
ind_ext = ExtremInd(paretoSet, M, N)
rank = 1
ind_ext.sort( key = lambda x: x[2], reverse =
True ) #Los ordenamos de mayor winning score
for i in ind_ext:
    i[3] = rank

    rank = rank + 1
    paretoSet.remove(i)
# Los ind no extremos se ordenan de mayor a
menor winning score
    paretoSet.sort( key = lambda x: x[2], reverse =
True )
    rank = len(ind_ext) + 1
    for i in paretoSet:
        i[3] = rank
        rank = rank + 1
    return(popWfit)
#Torneo binario
def Tournament(pop, MatPoolSize):
    matPool = []
    popWranks = deepcopy(pop)
    for i in popWranks: #aseguramos que el
    elemento 4 de todos los ind sea cero
        i[4] = 0
    while len(matPool) < MatPoolSize
        ind1 = choice( popWranks )
        ind2 = choice( popWranks ) #estos números
son los ind seleccionados para el torneo
        while ind1 == ind2:
            ind2 = choice( popWranks ) #si lo es se
vuelve a generar hasta que sea diferente
            if ind1[3] < ind2[3]:#Si el ind 1 tiene un rank
más alto que el 2
                matPool.append(ind1)
            if ind1[3] > ind2[3]:
                matPool.append(ind2)
            if ind1[3] == ind2[3]
            if len(matPool) == 0:
                matPool.append(choice( [ind1, ind2]
))#Escoge un ind entre el 1 y 2 aleatoriamente
            else:
                if ind1[4] > ind2[4]:
                    matPool.append(ind1)
                if ind1[4] < ind2[4]:
                    matPool.append(ind2)
                if ind1[4] == ind2[4]:
                    matPool.append(choice( [ind1, ind2] ))
#Se elimina el ganador de la población
popWranks.remove( matPool [ -1 ] )
#Se agrega la distancia
for i in popWranks:
    i[4] = i[4] + abs(matPool[ -1 ][3] - i[3])
    return (matPool)
#Función para revisar la factibilidad del individuo

```

```

def Feas(ind):
    #ind solo los valores del ind
    feas = False
    #Revisar si L es permitido
    if ind[0] <= l_max and ind [0] >= l_min: #Revisa
el WB
    L = ind [0] #asigna el valor que tiene el
individuo al WB
    b = (mF / m) * L #dist entre el cg y el tR
    tRmax= 2*sqrt(rho_ext**2-(b + AE_R +
tdim/2)**2)- 2*rho - tw - 0.2 #Calcula el max de
tR dependiendo de L
    if ind [2] >= tR_min and ind[2] <= tRmax
    tR = ind[2] #Si pasa la condición anterior
ind[2] posee un track factible
    a = L - b # dist entre el cg y el tF
    tFmax = 2*sqrt(rho_ext**2-(a + tdim/2 +
AE_F)**2) - 2*rho - tw - 0.2 #Calcula el max de tF
dependiendo de L
    if ind[1] >= tF_min and ind[1] <= tFmax
and ind[1]<tR/0.75 and ind[1]>tR*0.75:
#Factibilidad del track delantero
    tF = ind [1] # si pasa la condición
anterior tiene un tF posible
    yBF = tF/2 - 0.031
    IC_Fmin = zBmax + (zAC - zBmax)*(tF/2 -
fvsa - yBF)/(1/2*(tF-tdim-0.08)-yBF)
    RC_F_min = (IC_Fmin * tF)/(2*fvsa)
    IC_Fmax = zBmin + (zAC - zBmin)*(tF/2 -
fvsa - yBF)/(yACminF-yBF)
    RC_F_max = (IC_Fmax * tF)/(2*fvsa)
    if ind[3] >= RC_F_min and ind[3] <=
RC_F_max:# Factibilidad del RC delantero
    yBR = tR/2 - 0.031
    IC_Rmin = zBmax + (zAC -
zBmax)*(tR/2 - fvsa - yBR)/(1/2*(tR-tdim-0.080)-
yBR)
    RC_R_min = (IC_Rmin * tR)/(2*fvsa)
    IC_Rmax = zBmin + (zAC -
zBmin)*(tR/2 - fvsa - yBR)/(yACminR-yBR)
    RC_R_max = (IC_Rmax * tR)/(2*fvsa)
    if ind[4]>= RC_R_min and ind[4] <=
RC_R_max:
        if ind [5] >=
(52119.2*IR_F**2*tF**2)/(3.9811+IR_F**2) and
ind[5] <= cphi_max + 1 and ind[6] >=
(52119.2*IR_R**2*tR**2)/(3.9811+IR_R**2) and
ind [6] <= cphi_max + 1:

```

```

        feas = True
    return(feas)

# Cruza BLX
def CrossBLX(matPool, qOffs):
    i = 0
    popOffs = []
    while len(popOffs) < qOffs: # Se generan n
hijos con n igual a 1/2 de la longitud de la
población
        #Selección de dos ind diferentes
        n1 = choice( matPool ) #Selecciona un ind de
mating Pool
        n2 = choice( matPool ) #Selecciona un ind de
mating Pool
        while n1 == n2: # A segura que sean
diferentes
            n2 = choice( matPool )
        #Inicializa los hijos
        o1 = deepcopy(n1) #Se inicializa el hijo 1
        o2 = deepcopy(n2) #Se inicializa el hijo 2
        alpha = 0.5
        l = n1[0].copy()
        feas1 = False
        feas2 = False
        cmax = n1[0].copy()
        cmin = n1[0].copy()
        j = 0
        k = 0
        for j in range (len(n1[0])):
            cmin[j] = min(n1[0][j], n2[0][j]) #Se calcula
el vector de mínimos
            cmax[j] = max(n1[0][j], n2[0][j]) #Se
calcula el vector de máximos
            l[j] = abs(n1[0][j] - n2[0][j]) #Se calcula el
vector de intervalos
            while feas1 == False: #Se sigue el
procedimiento hasta encontrar un ind valido
                for i in range (len(o1[0])): #Para todos los
elementos del individuo
                    #Cada elemento es remplazado con un
número aleatorio en el rango generado entre los
padres y el alpha
                    o1[0][i]= uniform((cmin[i] - l[i]*alpha) ,
(cmax[i] + l[i]*alpha))
                    feas1 = Feas (o1[0]) #Se revisa la
factibilidad

```

```

    j += 1 #Es el contador para que no esté
infinitamente en esta función
    if j == 350: #En caso de llegar a 100
intentos y no lograr un elemento factible
        o1 = copy(n1) #El hijo uno se hace igual
al padre
        break #Se sale del while
    while feas2 == False: #Se sigue el
procedimiento hasta encontrar un ind valido
        for q in range (len(n2[0])): #Para todos los
elementos del individuo
            o2[0][q] = uniform((cmin[q] -
l[q]*alpha) , (cmax[q] + l[q]*alpha))
            feas2 = Feas(o2[0]) #Se revisa la
factibilidad
            k += 1 #Es el contador para que no esté
infinitamente en esta función
            if k == 350: #En caso de llegar a 100
intentos y no lograr un elemento factible
                o2 = copy(n2) #El hijo uno se hace igual
al padre
#Para indicar que no se logró un elemento
factible
                break #Se sale del while
            popOffs.extend([o1, o2]) #Los agrega a los
hijos a la lista de hijos
            return(popOffs)
#Función para la mutación NO uniforme
def NoUniform(chro, valor_min, valor_max,
actualGen, maxGen):
    b = 5 #Valor designado por el usuario
    tau = randint(0,1) #Genera 0 o 1
    if tau == 0: #Si el valor fue 0 se hace los
siguiente
        #Se calcula un delta para el elemento a
mutar
        delta = (valor_max - chro)*(1 -
pow(uniform(0,1),pow(1-actualGen/maxGen,
b)))
        chro_new = chro + delta
    else:
        #Se calcula un delta para el elemento a
mutar
        delta =(chro - valor_min)*(1 -
pow(uniform(0,1) ,pow(1- actualGen/maxGen,
b)))
        chro_new = chro - delta
    return (chro_new)

```

```

#Mutación NO uniforme para generar una lista
de individuos factibles
def Muta(popOffs, actualGen, maxGen): #Se
solicita el individuo a mutar, la generación actual
y la generación máxima
    popMut = []
    feas = False #Se inicializa la factibilidad en
Falso para que entre al while
    i = 0 #Contador para los intentos de
mutación
    while feas == False:
        ind_new = deepcopy(ind)
        ind_new[0][0] =
NoUniform(ind_new[0][0],l_min, l_max,
actualGen, maxGen) #Muta el WB
        #Para la mutacion del tR
        b = (mF / m) * ind_new[0][0] #dist entre el
cg y el tR dependiendo del WB
        tRmax= 2*sqrt(rho_ext**2-(b + AE_R +
tdim/2)**2)- 2*rho - tw - 0.2
        ind_new[0][2] =
NoUniform(ind[0][2],tR_min, tRmax, actualGen,
maxGen) #Muta el tR
        #Para la mutación del track delantero
        a = ind_new[0][0] - b # dist entre el cg y el
tF dependiendo del WB
        tFmax = 2*sqrt(rho_ext**2-(a + tdim/2 +
AE_F)**2) - 2*rho - tw - 0.2
        # tFmax= 2*sqrt(20.25-(a + 0.92987)**2) -
5.90574 #Calcula el max de tF dependiendo de L
        if tFmax > ind_new[0][2]/0.75:
            tFmax = ind_new[0][2]/0.75
        if tF_min < ind_new[0][2]*0.75:
            tF_min1 = ind_new[0][2]*0.75
        else:
            tF_min1= tF_min
        ind_new[0][1] = NoUniform(ind[0][1],
tF_min1, tFmax, actualGen, maxGen) #Muta el tF
        tF = ind_new[0][1]
        tR = ind_new[0][2]
        #Para la mutation del RC delantero
        yBF = tF/2 - 0.031
        IC_Fmin = zBmax + (zAC - zBmax)*(tF/2 -
fvsa - yBF)/(1/2*(tF-tdim - .08)-yBF)
        RC_F_min = (IC_Fmin * tF)/(2*fvsa)
        IC_Fmax = zBmin + (zAC - zBmin)*(tF/2 -
fvsa - yBF)/(yACminF-yBF)
        RC_F_max = (IC_Fmax * tF)/(2*fvsa)

```

```

    ind_new[0][3] =
NoUniform(ind[0][3],RC_F_min,RC_F_max,
actualGen,maxGen)
    #Para la mutacion del RC trasero
    yBR = tR/2 - 0.031
    IC_Rmin = zBmax + (zAC - zBmax)*(tR/2 -
fvsa - yBR)/(1/2*(tR-tdim - .08)-yBR)
    RC_R_min = (IC_Rmin * tR)/(2*fvsa)
    IC_Rmax = zBmin + (zAC - zBmin)*(tR/2 -
fvsa - yBR)/(yACminR-yBR)
    RC_R_max = (IC_Rmax * tR)/(2*fvsa)
    ind_new[0][4] = NoUniform(ind[0][4],
RC_R_min, RC_R_max, actualGen,maxGen)
    ind_new[0][5] = NoUniform(ind[0][5],
(52119.2*IR_F**2*ind[0][1]**2
)/(3.9811+IR_F**2) + 1 , cphi_max - 1 ,
actualGen, maxGen)
    ind_new[0][6] = NoUniform(ind[0][6],
(52119.2*IR_R**2*ind[0][2]**2)/(3.9811+IR_R*
*2) + 1 , cphi_max - 1 , actualGen, maxGen)
    feas = Feas(ind_new[0])
    i = i + 1
    if i == 350: #Si llega a 300 repeticiones sin
llegar a un individuo valido
        if Feas(ind[0]) == True: # Y el individuo
original es valido
            print('Tomo el no mutado')
            ind_new = ind
        else:
            ind_new = PopGen(1)
            print('genero otro')
            break
    popMut.append(ind_new)
    return(popMut)
def Truncated(Rt, popLen):
    # Rt es un conjunto para truncar con padres e
hijos
    #Encontrar los ind no dominados
    aux0 = Pareto(Rt) #Se obtiene la frontera de
pareto
    paretoSet = aux0[0] # El primer elemento
contiene a los ind de conjunto de Pareto
    noOpt = aux0[1] #El segundo elemento
contiene a los ind no óptimos
    indWrack = []
    pop = []
    N = len (paretoSet) #Numero de ind no
dominados

```

```

    M = len (paretoSet[0][1]) #Cantidad de
objetivos
    # Si hay menos o igual ind no dominados que
la longitud de la población
    if N <= popLen:
        pop.extend(paretoSet) #Agrega todos los
elementos no dominados
        if len(pop) < popLen:
            #ranking de los ind dominados para
seleccionar los ind dominados que entran a la
nueva generación
            indWrack = RankNoPareto(noOpt,
paretoSet)
            indWrack.sort (key = lambda x: x[3])
#ordena asendentemente
            for i in range (popLen - N): #Agrega los ind
faltantes
                pop.append(indWrack[i])
            # Si hay más ind dominantes que los deseados
en la población
            else :
# if N > popLen:
                #Calcular Winning score
                WS(paretoSet)
                #Encontrar los ind extremos
                ind_ext = ExtremInd(paretoSet, M, N)
                #Encontrar los valores extremos
                extVal = []
                for i in range( M ):
                    ind_ext.sort( key = lambda x: x[1][i])
                    extVal.append( [ind_ext[0][1][i], ind_ext[-
1][1][i]]#[min,max]
                    #Eliminar los extremos del Pareto set y los
agrega a la población
                for i in ind_ext:
                    pop.append(i)
                    paretoSet.remove(i)
                # Agregamos ind hasta llegar al popLen
                while len(pop) < popLen :
                    #Calculo de la dist Euclidiana entre cada
ind de la pop y del pareto set
                    for i in paretoSet:
                        di = [] # Lista con las distancia entre
cada par de ind
                        for j in pop:
                            aux0 = 0 #variable auxiliar para la
sumatoria en la distancia.
                            for k in range(M):

```

```

        aux0 = aux0 + ((i[1][k] -
j[1][k])/(extVal[k][1]-extVal[k][0]))**2
        if aux0 == 0:
            c = c + 1
        di.append(sqrt(aux0)) #se agrega di
entre cada i y j
        if c == 0 :
            i[4] = min(di)
        else :
            i[4] = -(c - 1) * sqrt(M)
        #Después de calcular todas las distancias,
ordenar por la distancia descendente
        paretoSet.sort(key = lambda x: x[4],
reverse = True )
        aux0 = [] #Lista aux con las distancias mas
grandes de popLen - len(pop) ind
        for i in range (popLen - len(pop)):
            aux0.append(paretoSet[i]) #agrega n
ind faltantes a una lista auxiliar

        #Se ordenan respecto al WS
aux0.sort(key = lambda x: x[2], reverse =
True)
        pop.append(aux0[0]) #Se agrega el ind
paretoSet.remove(aux0[0]) #Se elimina el
ind
        return(pop)
def FunC (xp, xpp):
    aux1 = 0
    for i in xp: #Revisar los ind de la ultima
generacion
        for j in xpp:
            if DomEq(i[1],j[1]) == True:
                aux1 = aux1 + 1
                break
    result = aux1 / len(xpp)
    print(aux1)
    return(result)

#Detección de convergencia
def ConDet(popRecord):
    conv = False
    k = randint(-len(popRecord)+1, -2) #selecciona
una generacion anterior aleatoria
    A = popRecord[-1] #Generacion Actual
    B = popRecord[-k]
    Conv = FunC(A,B)
    Conv2 = FunC(B,A)

```

```

    print(Conv, Conv2)
    if FunC(A,B) < FunC(B,A):
        conv = True
    return (conv)
# Distancia euclidiana ponderada en el espacio
de variables
def DistEucl(pop):
    ind_ext = [] #inicializamos la lista de ind
extremos
    for i in range(len(pop[0][0])):
        # ordeno los elementos
        pop.sort( key = lambda x: x[0][i])
        ind_ext.append([pop[0][0][i],pop[-1][0][i] )
    for i in pop:
        dp = 0 # Distancia promedio
        for j in pop:
            aux0 = 0 #variable auxiliar para la
sumatoria en la distancia.
            for k in range(len (i[0])):
                aux0 = aux0 + ((i[0][k]-
j[0][k])/(ind_ext[k][0]-ind_ext[k][1]))**2
            dp = dp + sqrt (aux0)
        i.append(dp/len (i[0])) #se agrega la
distancia promedio
    return (pop)
# Graficas iniciales
def GraphGen(pop,genNum):
    #Graficas iniciales
    try:
        os.stat(imagdirectory)
    except:
        os.mkdir(imagdirectory)
    Ll = []
    tFl = []
    tRl = []
    lzl = []
    SigmaFl = []
    ErrorTransl = []
    SigmaLl = []
    roll_gradient_errorl = []
    for ind in pop:
        Ll.append(ind[0][0])
        tFl.append(ind[0][1])
        tRl.append(ind[0][2])
        lzl.append(ind[1][0])
        SigmaFl.append(ind[1][1])
        ErrorTransl.append(ind[1][2])
        SigmaLl.append(ind[1][3])

```

```

    roll_gradient_errorl.append(ind[1][4])
plt.figure()
plt.title('L vs SigmaL')
plt.xlabel('L')
plt.ylabel('Sigma L')
plt.plot(Ll, SigmaLl,'bo')
plt.axis([1.5, 1.85, 0.015, 0.02])
plt.savefig("{}L_SigmaL_gen.jpg".format(imagdir
ectory,genNum))
plt.close()
plt.figure()
plt.title('tF vs SigmaF')
plt.xlabel('tF')
plt.ylabel('Sigma F')
plt.plot(tFl, SigmaFl, 'go')
plt.axis([1, 2.5, 0, 0.5])
plt.savefig("{}tF_SigmaF_gen.jpg".format(imagdi
rectory,genNum))
plt.close()
plt.figure()
plt.title('tR vs Error de Transferencia')
plt.xlabel('tR')
plt.ylabel('Error de transferencia')
plt.plot(tRl, ErrorTransl, 'ro')
plt.axis([1, 2.9, 0, 6000])
plt.savefig("{}tR_ErrorTransf_gen.jpg".format(i
magdirectory,genNum))
plt.close()
plt.figure()
plt.title('Iz vs Sigma F')
plt.xlabel('Iz')
plt.ylabel('Sigma F')
plt.plot(Izl, SigmaFl,'co')
plt.axis([90, 250, 0, 0.5])
plt.savefig("{}Iz_SigmaF_gen.jpg".format(imagdi
rectory,genNum))
plt.close()
plt.figure()
plt.title('error de gradiente de roll vs Sigma F ')
plt.xlabel('error de gradiente de roll [%]')
plt.ylabel('Sigma F')
plt.plot(roll_gradient_errorl, SigmaFl,'mo')
plt.axis([0, 100, 0, 0.5])
plt.savefig("{}RollGradError_SigmaF_gen.jpg".fo
rmat(imagdirectory,genNum))
plt.close()

```

```
#-----GENETICO-----
```

```

figs = {}
axs = {}
popRecord = [] #Inicializa el historial de
poblaciones
con = False #Inicializa el estado de la
convergencia a falso
actualGen = 1 #Inicializa la generación actual
como
pop = PopGen(popLen) #Generar una población
con popLen individuos
Fitness(pop) #Agrega el fitness a la población
GraphGen(pop,"Inicial") #Hace las gráficas de la
población inicial
popRecord.append(pop) #Agrega la población al
historial
#Ciclo que se hace mientras la convergencia sea
falso y aun no se llegue a las generaciones
máximas
while con == False and actualGen < maxGen :
# Agrega los rankings a los ind
    Ranking(pop) #lista con el ind y el ranking
# Genera una población sablista con los ind a
cruzar
    matPool = Tournament(pop, popLen/2) # Lista
con los elementos ganadores del torneo
# Genera una lista de los elementos cruzados
    popOffs = CrossBLX(matPool, popLen) #Lista
de hijos
# Genera una lista de los elementos mutados
    popMut = Muta(popOffs, actualGen, maxGen)
#Lista P
# Agregas el fitness de los hijos mutados
    Fitness(popMut)
# Mezcla, agraga los hijos a la poblacion
    Rt = deepcopy(pop)
    Rt.extend(popMut) #A + P = R
# -----Truncar
# Asignar ranking a la poblacion
    pop = Truncated(Rt, popLen) #A_t+1
#Calcula fitness de actualGen + 1
    popWfit = []
    popWfit = Fitness(pop)
    if len(popRecord) >= popRecordLen:
        popRecord.pop(0) #Elimina la pop mas vieja
        popRecord.append(pop) #Agrega la nueva
poblacion al record

```



```

    actualGen = actualGen + 1 #Aumenta la
generación
    print ("\n Generacion actual ", actualGen)
#Imprime la generacion actual
# -----Revisar convergencia
    if actualGen > minGen:
        con = ConDet(popRecord)
#-----Termina el ciclo-----
print ("Haciendo las gráficas")
for gen in popRecord:
    genNum = popRecord.index(gen) + 1
    GraphGen(gen,genNum)
#Hace el WS igual a cero
for i in pop:
    i[2] = 0
#Encuentra los ind de Pareto
aux1 = Pareto(pop)
paretoSet = aux1[0]
noOpt = aux1[1]
for i in noOpt:
    i.extend(["NP","-"])
for i in paretoSet:
    i.append("P")
#Calcula la distancia euclidiana de la última
generación
DistEucl(paretoSet)
pop = paretoSet + noOpt
WB = []
tF = []
tR = []
RC_F = []
RC_R = []
cphi_F = []
cphi_R = []
lz = []
SigmaF = []
ErrorTrans = []
SigmaL = []
roll_gradient_error = []
pareto = []
WS = []

```

```

distEuc = []
for i in pop:
    WB.append(i[0][0]*1000)
    tF.append(i[0][1]*1000)
    tR.append(i[0][2]*1000)
    RC_F.append(i[0][3]*1000)
    RC_R.append(i[0][4]*1000)
    cphi_F.append(i[0][5])
    cphi_R.append(i[0][6])
    lz.append(i[1][0])
    SigmaF.append(i[1][1])
    ErrorTrans.append(i[1][2])
    SigmaL.append(i[1][3])
    roll_gradient_error.append(i[1][4])
    pareto.append(i[5])
    WS.append (i[2])
    distEuc.append(i[6])
data = {'WB [mm]': WB,
        'Front track [mm]': tF,
        'Rear track [mm]': tR,
        'Front RC [mm]': RC_F,
        'Rear RC [mm]': RC_R,
        'Front phi [Nm/°]': cphi_F,
        'Rear phi [Nm/°]': cphi_R,
        'lz': lz,
        'Sigma F': SigmaF,
        'Error Trans': ErrorTrans,
        'Sigma L': SigmaL,
        'roll gradient error': roll_gradient_error,
        'Pareto': pareto,
        'WS': WS,
        'Distancia Euclideana': distEuc}
df = pd.DataFrame(data, columns = ['WB [mm]',
'Front track [mm]', 'Rear track [mm]', 'Front RC
[mm]', 'Rear RC [mm]', 'Front phi [Nm/°]', 'Rear
phi [Nm/°]','lz', 'Sigma F','Error Trans','Sigma
L','roll gradient error','Pareto','WS','Distancia
Euclideana'])
df.to_excel('{}\xlsx'.format(resultfile),
sheet_name = resultfile

```

ANEXO 5: PROGRAMA PARA LA DISMINUCIÓN DEL ERROR DEL IR DESEADO

Datos

```
(*--Datos del genetico-----*)
popLen = 50; (*ind en la poblacion*)
offLen= 20;(*hijos por generacion*)
 $\alpha$ Cross = 0.5; (*constante cruza*)
maxGen =100; (*generación max*)
bMut = 5; (*constante mutación*)
tolDes = 0.001; (*tolerancia de IR*)
indTol = 30; (*ind en tolerancia*)
ai={x5min,x5pmin, $\alpha$ min };(*Val mín*)
bi = {x5max,x5pmax, $\alpha$ max};(*máx*)
x5inicial = 88.84;
x5pinicial = 123.87;
 $\theta$ 1inicial = 11.45*Degree;
 $\theta$ 2inicial = 191.53*Degree;
 $\theta$ 3inicial =92*Degree;
 $\theta$ 4inicial=290.3*Degree;
 $\theta$ 5inicial=231.01*Degree;
 $\theta$ 7inicial=19.52*Degree;
(*----A variar en el programa----*)
x5min=85.64; x5max =90.04;
x5pmin=11567; x5pmax =130.07;
 $\alpha$ min=20*Degree;
 $\alpha$ max=80*Degree;
(*-----Fijos-----*)
x1=260.53;(*mm*)
x1p=30.92; x2=255.91;
x3=166.2; x4=521.27;
x4p=39.73; x7=120.5; l={0,0,0};
j={226.97,-50.68,0};
a={245.39,-388.38,0};
d={216.29,-561.17,0};
Funciones
(*--genera una poblacion----*)
PopGen[popLen1_,pop1_] :=
For[i=1,i<=popLen1,i=i+1,
  x5=RandomReal[{x5min,x5max}];
  x5p=RandomReal[{x5pmin,x5pmax}];
   $\alpha$ =RandomReal[{ $\alpha$ min, $\alpha$ max}];
  pop1[i]={x5,x5p, $\alpha$ }
(*-----Matriz de rotación-----*)
Rz[ $\theta$ _] := {{Cos[ $\theta$ ],-
Sin[ $\theta$ ,0],{Sin[ $\theta$ ,Cos[ $\theta$ ,0},{0,0,1}};
(*-----Fitness-----*)
Fitness[pop1_,popLen1_, popWfit1_] :=
For[i=1,i<= popLen1,i++,
```

```
x5=pop1[i][[1]];
x5p=pop1[i][[2]];
 $\alpha$  =pop1[i][[3]];
r1={x1,0,0}; r1p={x1p,0,0};
r2={x2,0,0}; r3={x3,0,0};
r4={x4,0,0}; r4p={x4p,0,0};
r5={x5,0,0}; r5p={x5p,0,0};
r6={x6,0,0}; r7={x7,0,0};
R1=Rz[ $\theta$ 1].r1; R1p=Rz[ $\theta$ 1].r1p;
R2=Rz[ $\theta$ 2].r2; R3=Rz[ $\theta$ 3].r3;
R4=Rz[ $\theta$ 4].r4; R4p=Rz[ $\theta$ 1+ $\beta$ + $\pi$ ].r4p;
R5=Rz[ $\theta$ 5].r5; R5p=Rz[ $\theta$ 5+ $\alpha$ ].r5p;
R6=Rz[ $\theta$ 7].r6; R7=Rz[ $\theta$ 7].r7;
Rlj=j-l; Rjd=d-j; Rda=a-d;
(*Ecuacion de Cinematicas*)
Pos1=R7+R6+R5p-Rlj;
Pos2=-R5+R4+R4p-R1-Rjd;
Pos3=R1+R1p+R3+R2-Rda;
x6=x6inicial;
PosInicial=FindRoot[{
  Pos1[[1]]==0, Pos1[[2]]==0,
  Pos2[[1]]==0, Pos2[[2]]==0,
  Pos3[[1]]==0, Pos3[[2]]==0,
  { $\theta$ 1, $\theta$ 1inicial}, { $\theta$ 2, $\theta$ 2inicial},
  { $\theta$ 3, $\theta$ 3inicial}, { $\theta$ 4, $\theta$ 4inicial},
  { $\theta$ 5, $\theta$ 5inicial}, { $\theta$ 7, $\theta$ 7inicial},
  MaxIterations->100];
(*solucion total *)
 $\theta$ 1k= $\theta$ 1/.PosInicial;
 $\theta$ 2k= $\theta$ 2/.PosInicial;
 $\theta$ 3k= $\theta$ 3/.PosInicial;
 $\theta$ 4k= $\theta$ 4/.PosInicial;
 $\theta$ 5k= $\theta$ 5/.PosInicial;
 $\theta$ 7k= $\theta$ 7/.PosInicial;
For[k=x6inicial,k>= x6final,k=k-1,
  x6=k;
  SolPos[k]=FindRoot[{
    Pos1[[1]]==0, Pos1[[2]]==0,
    Pos2[[1]]==0, Pos2[[2]]==0,
    Pos3[[1]]==0, Pos3[[2]]==0,
    { $\theta$ 1, $\theta$ 1k}, { $\theta$ 2, $\theta$ 2k}, { $\theta$ 3, $\theta$ 3k},
    { $\theta$ 4, $\theta$ 4k}, { $\theta$ 5, $\theta$ 5k}, { $\theta$ 7, $\theta$ 7k},
    MaxIterations->100];
   $\theta$ 1k= $\theta$ 1/.SolPos[k];
   $\theta$ 2k= $\theta$ 2/.SolPos[k];
   $\theta$ 3k= $\theta$ 3/.SolPos[k];
```

```

    04k=04/.SolPos[k];
    05k=05/.SolPos[k];
    07k=07/.SolPos[k];
    For[k=x6inicial,k>= x6final,k=k-1,
    RRwheel[k]=Rlj+Rjd+R1+R1p+1/2*
R3/.SolPos[k];
tablaR=Table[{k,RRwheel[k][[2]]},{k,x6final,x6inicial,1}];
    lm=LinearModelFit[tablaR,x,x] ;
m=Abs[(x6final-x6inicial)/(lm[x6final]-lm[x6inicial]));
    errorM = Abs [m-1];
    popWfit1[i]={pop1[i],errorM};
T2D[R_]:=R[[1]],R[[2]];
Genético
selcInd = {};
PopGen[popLen,pop];
Fitness [pop,popLen, popWfit];
For [actualGen=1, actualGen<= maxGen &&
Length[selcInd]<= indTol,actualGen++,
(*----Torneo binario ---*)
For[i=1,i<= offLen*2,i++,
ind1= popWfit[RandomInteger [{1,popLen}]];
ind2 = popWfit [RandomInteger [{1,popLen}]];
If[ind1[[2]]<=ind2[[2]],
MaPool[i] =ind1];
If[ind1[[2]]>ind2[[2]],
MaPool[i] =ind2]; ];
(*-----CrossBMX-----*)
For [i = 1, i<= offLen,i++,offs[i]={};
ind1= MaPool[RandomInteger[{1,offLen*2}]
ind2 = MaPool[RandomInteger[{1,offLen*2}]
For [k = 1, k<=Length[ind1[[1]]],k++,
inter = ind1[[1]][[k]]- ind2[[1]][[k]];
cmin = Min[{ind1[[1]][[k]], ind2[[1]][[k]]];
cmax = Max[{ind1[[1]][[k]], ind2[[1]][[k]]];
offs[i]=Append[offs[i],RandomReal[{cmin-
inter*αCross, cmax-inter*αCross}]]; ]
(*Mutation *)
For [i = 1, i<= offLen ,i++, offsMut[i]={}
For [k = 1, k= Length[ind1[[1]]],k++,
τ = RandomInteger[{0,1}];
ci =offs[i][[k]];
r=RandomReal[{0,1}];
If[τ==0,
cin=ci+(bi[[k]]-ci)*(1-r^((1-
actualGen/maxGen)^bMut));];

```

```

    If[τ==1,
        cin=ci-(ci-ai[[k]])*(1-r^((1-
actualGen/maxGen)^bMut));];
offsMut[i] = Append[offsMut[i],cin]];
Fitness [offsMut,offLen, offWfit];
(*-----Ordenar y trincar-----*)
popAux = {};
For [i=1, i<= popLen,i++,
popAux= Append[popAux,popWfit[i]];
If [popWfit[i][[2]]<tolDes,
AppendTo[selcInd,popWfit[i]] ];
For [i=1, i<= offLen,i++,
popAux= Append[popAux,offWfit[i]];
If [offWfit[i][[2]]<tolDes &&
MemberQ[selcInd,{selcInd,offWfit[i]}]==False ,
selcInd = Append[selcInd,offWfit[i]]]
Sort[popAux,#1[[2]]<#2[[2]]&];
For [i=1, i<= popLen,i++,
pop[i]= popAux[[i]]];]
Animación
01i=01/.PosInicial;
02i=02/.PosInicial;
03i=03/.PosInicial;
04i=04/.PosInicial;
05i=05/.PosInicial;
07i=07/.PosInicial;
For[i=x6inicial,i>= x6final,i=i-1,
x6=i;
SolPos[i]=FindRoot[{
Pos1[[1]]==0, Pos1[[2]]==0, Pos2[[1]]==0,
Pos2[[2]]==0, Pos3[[1]]==0, Pos3[[2]]==0},
{01,01i},{02,02i},{03,03i},{04,04i},{05,05i},{07,07
i},MaxIterations->15];
01i=01/.SolPos[i];
02i=02/.SolPos[i];
03i=03/.SolPos[i];
04i=04/.SolPos[i];
05i=05/.SolPos[i];
07i=07/.SolPos[i];>(*cierra For*)
Animate[ x6=i; cero={0,0};
RR1=T2D[R1]; RR1p=T2D[R1p];
RR2=T2D[R2]; RR3=T2D[R3];
RR4=T2D[R4]; RR4p=T2D[R4p];
RR5=T2D[R5]; RR5p=T2D[R5p];
RR6=T2D[R6]; RR7=T2D[R7];
RRLj=T2D[Rlj]; RRJD=T2D[Rjd];
RRDA=T2D[Rda];
linea1=Line[{cero,RR7}]/.SolPos[i];
linea2=Line[{RR7,RR7+RR6}]/.SolPos[i];

```

```

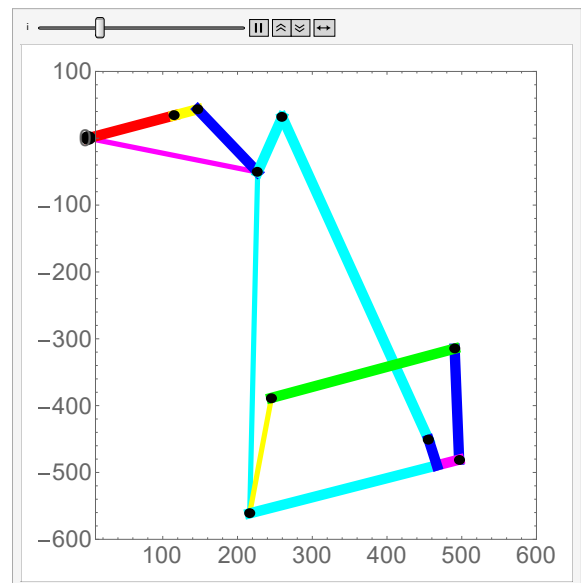
linea3=Line[{RRLJ,RR7+RR6}]/.SolPos];
linea4=Line[{RR7+RR6+RR5p,RR7+RR6+RR5p-
RR5}]/.SolPos[i];
linea5=Line[{RRLJ-RR5,RRLJ-
RR5+RR4}]/.SolPos[i];
linea6=Line[{RRLJ+RRJD,RRLJ+RRJD+RR1}]/.SolPos[i];
linea7=Line[{RRLJ+RRJD+RR1,RRLJ+RRJD+RR1+RR
1p}]/.SolPos[i];
linea8=Line[{cero,RRLJ}]/.SolPos[i];
linea9=Line[{RRLJ,RRLJ+RRJD}]/.SolPos[i];
linea10=Line[{RRLJ+RRJD,RRLJ+RRJD+RRDA}]/.SolPos[i];
linea11=Line[{RRLJ+RRJD+RRDA,RRLJ+RRJD+RRD
A-RR2}]/.SolPos[i];
linea12=Line[{RRLJ-RR5+RR4,RRLJ-
RR5+RR4+RR4p}]/.SolPos[i];(*R4p*)
linea13=Line[{RRLJ+RRJD+RRDA-RR2,RRLJ-
RR5+RR4+RR4p+RR1p}]/.SolPos[i];
puntoL=Point[{0,0}];
puntoK=Point[RR7]/.SolPos[i];
puntoI=Point[RR7+RR6]/.SolPos[i];
puntoJ=Point[RRLJ]/.SolPos[i];
puntoH=Point[RRLJ-RR5]/.SolPos[i];
puntoG=Point[RRLJ-RR5+RR4]/.SolPos[i];
puntoD=Point[RRLJ+RRJD]/.SolPos[i];
puntoA=Point[RRLJ+RRJD+RRDA]/.SolPos[i];
puntoB=Point[RRLJ+RRJD+RRDA-RR2]/.SolPos[i];
puntoE=Point[RRLJ-
RR5+RR4+RR4p+RR1p]/.SolPos[i];
pernoL=Graphics[{PointSize[0.035],puntoL}];
pernoK=Graphics[{PointSize[0.025],puntoK}];
pernoI=Graphics[{PointSize[0.025],puntoI}];
pernoJ=Graphics[{PointSize[0.025],puntoJ}];
pernoH=Graphics[{PointSize[0.025],puntoH}];
pernoD=Graphics[{PointSize[0.025],puntoD}];
pernoA=Graphics[{PointSize[0.025],RGBColor[0,0
,0],puntoA}];
pernoB=Graphics[{PointSize[0.025],RGBColor[0,0
,0],puntoB}];
pernoG=Graphics[{PointSize[0.025],puntoG}];
pernoE=Graphics[{PointSize[0.025],puntoE}];
barra1=Graphics[{AbsoluteThickness[10],RGBCol
or[1,0,0],linea1}];
barra2=Graphics[{AbsoluteThickness[10],RGBCol
or[1,1,0],linea2}];
barra3=Graphics[{AbsoluteThickness[10],RGBCol
or[0,0,1],linea3}];

```

```

barra4=Graphics[{AbsoluteThickness[10],RGBCol
or[0,1,1],linea4}];
barra5=Graphics[{AbsoluteThickness[10],RGBCol
or[0,1,1],linea5}];
barra6=Graphics[{AbsoluteThickness[10],RGBCol
or[0,1,1],linea6}];
barra7=Graphics[{AbsoluteThickness[10],RGBCol
or[1,0,1],linea7}];
barra8=Graphics[{AbsoluteThickness[5],RGBColo
r[1,0,1],linea8}];
barra9=Graphics[{AbsoluteThickness[5],RGBColo
r[0,1,1],linea9}];
barra10=Graphics[{AbsoluteThickness[5],RGBCol
or[1,1,0],linea10}];
barra11=Graphics[{AbsoluteThickness[10],RGBC
olor[0,1,0],linea11}];
barra12=Graphics[{AbsoluteThickness[10],RGBC
olor[0,0,1],linea12}];
barra13=Graphics[{AbsoluteThickness[10],RGBC
olor[0,0,1],linea13}];
Frame->True,ImageSize->500,
PlotRange->{{10,600},{-600,100}},
AspectRatio->Automatic,GridLines->None,
BaseStyle->{24,FontFamily->
"Arial"},{i,x6final,x6inicial,1},AnimationRunning->
False]

```



ANEXO 6: SUSPENSIÓN 2019

Puntos de suspensión.

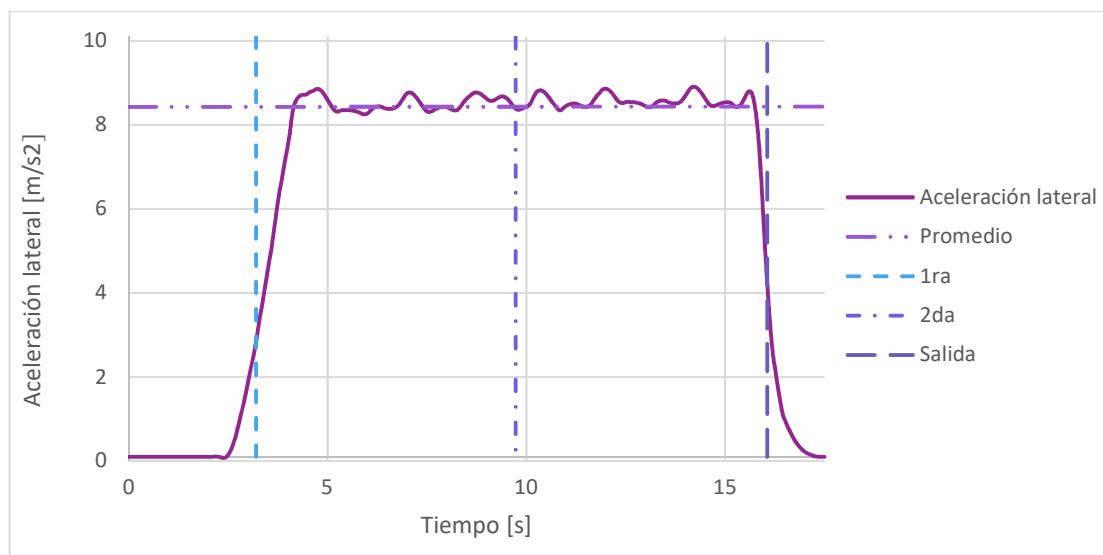
Nombre	X	y	z
Suspensión delantera			
Centro del balancín	6.1	220	528
Chasis – amortiguador	6.1	63.93	586.48
Amortiguador – balancín	6.1	242.71	607.4
Inferior interno delantero	-115	220	136
Inferior externo	-6.1	586.17	145
Inferior interno trasero	115	220	136
Push – balancín	6.1	289.26	555.13
Push – brazo superior	6.1	506.63	333.64
Tie rod interior	15	131	165
Tie rod exterior	53.5	549.12	179
Superior interno delantero	-115	240	301
Superior externo	6.1	561	300
Superior interno trasero	115	240	301
Centro de la llanta	0	622.3	228.6
Suspensión trasera			
Centro del balancín	1605.6	275.06	350
Chasis – amortiguador	1605.6	26.96	387.94
Amortiguador – balancín	1605.6	201.98	430
Inferior interno delantero	1430.4	245	143
Inferior externo	1619.1	576.6	149
Inferior interno trasero	1623.2	235	143
Push – balancín	1605.6	275.06	435
Push – brazo superior	1605.6	485.34	337.84
Ride height	1623.2	235	143
Track rod interior	1658.6	235	143
Track rod exterior	1662.52	569.6	149
Superior interno delantero	1356.7	262	309
Superior externo	1632.1	576.6	314.15
Superior interno trasero	1556.72	262.12	308.54
Centro de la llanta	1625.6	609.6	228.6

Parámetros del vehículo.

	Delantero	Trasero	Total
Masas			
Masa con piloto [kg]	140.16	150.139	290.299
Masa del piloto [kg]	-	-	70
Masa de USM [kg]	10.178	10.178	-
Masa de las llantas [kg]	3.731251	3.731251	-

Masa suspendida [kg]	119.804	129.783	249.587
Llantas			
Llanta	Hoosier 18x6-10	Hoosier 18x6-10	-
Diámetro de la llanta [m]	0.4597	0.4597	-
Ancho de la llanta [m]	0.2057	0.2057	-
Rigidez vertical [N/mm]	104.2384693	104.2384693	-
$I_{xx} = I_{yy}$ [kg m ²]	0.211	0.211	-
I_{zz} [kg m ²]	0.351	0.351	-
Dimensiones			
Track [m]	1.2446	1.2192	-
Wheelbase [m]	a = 0.841	b = 0.785	1.626
Altura del CG [m]	-	-	0.293724
Suspensión			
Gradiente de roll [rad/g]	-	-	-
Rigidez a roll [Nm/rad]	45,802.59	46,946.270	-
IR [1]	0.982	1.015	-
Altura de RC [mm]	-32.154	-10.262	-
Longitud del resorte [m]	0.093504	0.0885	-

Resultados de simulación de *skipad* en Adams-Car[®]



BIBLIOGRAFÍA

- [1] T. Gillespie, *Fundamentals of Vehicle Dynamics*, Society of Automotive Engineers, Inc, 1992.
- [2] A. G. Espejel Arroyo, *Rediseño de un sistema de suspensión para un auto de competencia mediante Adams/Car y Matlab*, Distrito Federal: UNAM, 2015.
- [3] R. Stone y J. Ball, *Automotive Engineering Fundamentals*, Warrendale: SAE International, 2004.
- [4] J. Reimpell, H. Stoll y J. Betzler, *The Automotive Chassis*, Oxford: Butterworth-Heinemann, 2001.
- [5] R. Jazar, *Vehicle Dynamics: Theory and Application*, New York: Springer, 2008.
- [6] J. Hurel Ezeta, A. Mandow y A. García Cerezo, «Los Sistemas de Suspensión Activa y Semiactiva: Una revisión,» *Revista Iberoamericana de Automática e Información industrial*, vol. 10, pp. 121-132, 2013.
- [7] J. Hurel Ezeta, A. Mandow y A. García Cerezo, «Los Sistemas de Suspensión Activa y Semiactiva: Una Revisión,» *Revista Iberoamericana de Automática e Informática industrial*, vol. 10, pp. 121-131, 2013.
- [8] E. Guglielmino, T. Sireteanu, C. Stammers, G. Ghita y M. Giuclea, *Semi-active Suspension Control: Improved Vehicle Ride and Road Friendliness*, London: Springer, 2008.
- [9] H. B. Pacejka, *Tyre and Vehicle Dynamics*, Butterworth-Heinemann, 2005.
- [10] M. Blundell y D. Harty, *Multibody Systems Approach to Vehicle Dynamics*, New York: Elsevier Butterworth-Heinemann, 2004.
- [11] R. Alkhatib, G. Nakhaie Jazar y M. Golnaraghi, «Optimal design of passive linear suspension using genetic algorithm,» *Journal of Sound and Vibration*, vol. 275, pp. 665-691, 23 Agosto 2004.
- [12] G. Verros, S. Natsiavas y C. Papadimitriou, «Design Optimization of Quarter-car Models with Passive and Semi-active Suspensions under Random Road Excitation,» *Journal of Vibration and Control*, vol. 11, pp. 581-606, 1 Mayo 2005.
- [13] Ö. Gündoğdu, «Optimal seat and suspension design for a quarter car with driver mode using genetic algorithms,» *International Journal of Industrial Ergonomics*, pp. 327-332, Abril 2007.
- [14] S. Ming-Yuan, C. Juing-Shian y L. Ming-Tang, «Design of Immune-Algorithm Based Adaptive Fuzzy Controllers for Active Suspension System,» *Advances in Mechanical Engineering*, p. 8, 28 Abril 2014.

- [15] M. P. Nagarkar, G. J. V. Patil y R. N. Z. Patil, «Optimization of nonlinear quarter car suspension--seat--driver model,» *Journal of Advanced Research*, vol. 7, pp. 991-1007, 2016.
- [16] M. P. Nagarkar, Y. J. Bhalerao, G. J. V. Patil y R. N. Z. Patil, «Multi-Objective Optimization of Nonlinear Quarter Car Suspension System--PID and LQR Control,» *Procedia Manufacturing*, vol. 20, pp. 420-427, 2018.
- [17] N. Nariman-Zadeh, M. Salehpour, A. Jamali y E. Haghgoo, «Pareto optimization of a five-degree of freedom vehicle vibration model using a multi-objective uniform-diversity genetic algorithm (MUGA),» *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 543-551, 2010.
- [18] V. Goga y M. Klůčik, «Optimization of Vehicle Suspension Parameters with use of Evolutionary Computation,» *Procedia Engineering*, vol. 48, pp. 174-179, 2012.
- [19] K. Boonlong, «Multiobjective Optimization of a Vehicle Vibration Model Using the Improved Compressed-Objective Genetic Algorithm with Convergence Detection,» *Advances in Mechanical Engineering*, vol. 5, p. 131495, 2013.
- [20] G. Papaioannou y D. Koulocheris, «An approach for minimizing the number of objective functions in the optimization of vehicle suspension systems,» *Journal of Sound and Vibration*, vol. 435, pp. 149-168, 24 Noviembre 2018.
- [21] D. E. Goldberg, *Genetic algorithms in search, optimization & machine learning*, Ontario: Addison-Wesley Publishing Company, Inc., 1989.
- [22] M. Rahul, S. Narinder y S. Yaduvir, «Genetic Algorithms: Concepts, Design for Optimization of Process Controllers,» *Computer and Information Science*, vol. 4, 3 2011.
- [23] A. Kumar, «Encoding Schemes in Genetic Algorithm,» *International Journal of Advanced Research in IT and Engineering*, vol. 2, nº 3, pp. 1-7, Marzo 2013.
- [24] Jyotishree, «Knowledge based operation and problems representation in genetic algorithms,» 2012.
- [25] S. N. Sivanandam y S. N. Deepa, *Introduction to Genetic Algorithms*, Springer-Verlag Berlin Heidelberg, 2008.
- [26] A. Moraglio y R. Poli, «Geometric crossover for the permutation representation.,» *Intelligenza Artificiale*, vol. 5, pp. 49-63, 1 2011.
- [27] N. Soni y T. Kumar, «Study of Various Mutation Operators in Genetic Algorithms,» *International Journal of Computer Science and Information Technologies*, vol. 5, nº 3, pp. 4519-4521, 2014.

- [28] F. Herrera, M. Lozano y J. L. Verdegay, «Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis,» *Artificial Intelligence Review*, vol. 12, pp. 265-319, 01 8 1998.
- [29] S. Patil y M. Bhende, «Comparison and Analysis of Different Mutation Strategies to improve the Performance of Genetic Algorithm,») *International Journal of Computer Science and Information Technologies*, vol. 5, nº 3, pp. 4669-4673, 2014.
- [30] W. Milliken y D. Milliken, *Race Car Vehicle Dynamics*, Warrendale: Society of Automotive Engineers, Inc, 1995.
- [31] Kaz Technologies, «Tech Tips & Seminar Downloads :Kaz Technologies,» [En línea]. Available: <https://www.kaztechnologies.com/wp-content/uploads/2015/06/A-Guide-To-Your-Dampers-Chapter-from-FSAE-Book-by-Jim-Kasprzak-Updated.pdf>. [Último acceso: 11 Septiembre 2019].
- [32] C. Smith, *Tune to Win*, Fallbrook: Aero Publishers, Inc, 1978.
- [33] SAE International, «FSAE online,» SAE, 25 Julio 2018. [En línea]. Available: <https://www.fsaeonline.com/cdsweb/gen/DownloadDocument.aspx?DocumentID=64b861c2-980a-40fc-aa88-6a80c43a8540>. [Último acceso: 2019 Enero 10].
- [34] Hoosier, «Hoosier racing tire,» [En línea]. Available: <https://www.hoosiertire.com/assets/Master5.pdf>. [Último acceso: 29 Julio 2019].