FEDERAL UNIVERSITY OF SANTA CATARINA
JOINVILLE TECHNOLOGICAL CENTER
AUTOMOTIVE ENGINEERING UNDERGRADUATE PROGRAM

ARIEL GUSTAVO AVI

MUTLI-OBJECTIVE OPTIMIZATION OF SUSPENSION KINEMATICS

Joinville
2021

ARIEL GUSTAVO AVI

MUTLI-OBJECTIVE OPTIMIZATION OF SUSPENSION KINEMATICS

Graduation Thesis submitted to the Automotive Engineering Undergraduate Program of Federal University of Santa Catarina to obtain the title of Bachelor in Automotive Engineering.
Advisor: Prof. Andrea Piga Carboni, Dr.
Co-Advisor: Claude Rouelle, MSc.

Joinville
2021

ARIEL GUSTAVO AVI

MUTLI-OBJECTIVE OPTIMIZATION OF SUSPENSION KINEMATICS

The present work at Graduation level was judged adequate to obtain of the title of Bachelor in Automotive Engineering and was approved in its final form by the Automotive Engineering Undergraduate Program of Federal University of Santa Catarina.

Examination committee:

Prof. Andrea Piga Carboni, Dr.
Committee President

Prof. Sérgio Junichi Idehara, Dr.
Federal University of Santa Catarina

Prof. Thiago Antonio Fiorentin, Dr.
Federal University of Santa Catarina

Mohammad Behtash, PhD Candidate.
OptimumG LLC

Joinville, 2021.

AKNOWLEDGEMENTS

I would like to thank my partner, Suelen, whose passion about psychology inspired me to pursue my dream in automotive engineering and today is my best friend and lover. The change of paths wouldn't be possible without the unmeasured support of my beloved parents, Tadeu and Adriana and my sister Kaliandra. It is a huge privilege to be in the position that I am today and it would not be possible without them.

I am also thankful for my friends and colleagues that accompanied me through the remarkable journey on Formula SAE during my graduation, even on the most difficult moments. Yago for being my closest friend during the graduation and the one who motivated me to push my limits, physically and intellectually. Our friendship cannot be summarized in a single paragraph. Pedro Stella for hopping on this research group at the first stages, and becoming a close friend. Notable names that are worth mentioning here: André, Serpa, Caio, Fabio, Fernanda, Zane, Gustavo, Keila, Leonardo, Lucas (A.K.A. Negot), Meg, Paulinho, Pedro (BIRL), Kiemo, Renan (A.K.A. DK) , Nomura. They all have been part of my formation as an individual and engineer. I carry good memories of the times — and nights — we spent together at the workshop, the competition and on every *gueri* we had. #Fahrwek4ever.

This work would not be possible without my advisor, Prof. Dr. Andrea Piga Carboni, who came up with the idea and not only helped me through this work but also kept me motivated. This work represents roughly 18 months of research since the first concept was sketched by ourselves. I have learned a lot from and with him. Furthermore, the work has evolved a lot since the first lines of code were laid on the screen, thanks to his persistence on improvement.

My gratitude to Claude Rouelle, my co-advisor, who believed in my work and is a source of inspiration for me since I attended to his first seminar. The effort he puts on teaching young students is outstanding and admirable. His constant search for improvement and share of knowledge were fundamental contributions for me to achieve the quality present in this work. I would also like to thank my colleagues at OptimumG, with whom I have learned a lot in the past few months and contributed greatly to this work, Bruno, Mo and Girish. It is a pleasure to work with them.

Lastly, I am grateful for the support that OptimumG provided for this work.

"He who follows another is always behind"

Anonymous

RESUMO

O design de cinemática de suspensão pode ser um processo desafiador. Em processos de desenvolvimento para veículos de competição, onde a agenda de desenvolvimento é estreita e a qualidade da cinemática é crucial, o trabalho requer ainda mais maturidade. O designer de suspensão precisa conciliar os requisitos de empacotamento dos componentes no veículo e percepções do piloto enquanto tenta fazer o melhor uso dos pneus em todo tempo possível. Recursos computacionais estão constantemente se tornando mais baratos e podem ser utilizados para resolver esse problema relacionado à cinemática auxiliado por ténicas de otimização. O processo de otimização pode gerar novas soluções de engenharia que não são óbvias ao entendimento humano e dificilmente seriam alcançadas pelo time de engenharia. Dada a natureza multi-objetivo desse tipo de problema e a falta de convexidade entre as funções objetivo, esse trabalho propõe a integração de uma otimização multi-objetivo evolucionária a um solver cinemático de suspensão. A otimização acarreta em um conjunto de soluções subótimas que são colocadas em ordem por um conjunto de funções-peso e fatores de escala. Esse processo oferece uma excelente razão tempo-benefício, uma vez que potencialmente pode reduzir semanas de trabalho para algumas horas de esforço computacional. Esse trabalho ilustra o poder do processo de otimização com um estudo de caso onde uma suspensão do tipo Duplo-A é completamente sintetizada pelo algoritmo de otimização, que é composto por 19 funções-objetivo divididas entre 4 movimentos diferentes.

Palavras-chave: Suspensão. Cinemática. Otimização. Algorithmo genético.

ABSTRACT

Suspension kinematics design can be a challenging process. In racing and motorsports development processes, where the design schedule is tighter and the quality of kinematics are critical, the job requires even more maturity. The suspension designer must conciliate vehicle packaging and driver perception requirements whilst trying to make the best use of the tires at all possible times. Computational resources are constantly getting cheaper and can be used to solve the kinematics issue aided by optimization techniques. The optimization process can generate new engineering solutions that are not obvious to the human understanding and would be hardly achieved by the engineering team. Given the multi-objective nature of this type of problem and the lack of convexity between the objective functions, this work proposes the integration of an Evolutionary Multi-Objective Optimization (EMOO) to a suspension kinematics solver. The optimization yields a set of sub-optimal solutions that are ranked by a set of weighting functions and scaling factors. This process offers an excellent time-to-benefit ratio, once it can potentially reduce weeks of workload to a few hours of computational effort. This work illustrates the power of such optimization process with a case study where a Double A-Arm suspension system is completely synthesized by the optimization algorithm, which is composed by 19 objective functions split in 4 different movements.

Keywords: Suspension. Kinematics. Optimization. Genetic algorithm.

LIST OF FIGURES

Figure 1 – Example of a G-G diagram of a common passenger car operating on normal conditions. 20
Figure 2 – Adapted SAE tire reference coordinates. 21
Figure 3 – Different tire coordinate systems and sign conventions. 22
Figure 4 – Difference between camber angle and inclination angle. 23
Figure 5 – General Magic Formula behavior. 24
Figure 6 – Fitted Magic Formula tire model (SAE Coordinates). 25
Figure 7 – ISO Coordinate Systems for vehicle. 26
Figure 8 – Camber and toe angle definitions. 27
Figure 9 – Kinematic roll and pitch axes representation 29
Figure 10 – Anti-Dive definition. 30
Figure 11 – Anti-Lift on front definition. 30
Figure 12 – Anti-Lift on rear definition. 30
Figure 13 – Anti-Squat definition. 31
Figure 14 – Double Wishbone mechanism with direct coil-over actuation. 32
Figure 15 – General construction of a double wishbone. 33
Figure 16 – King Pin geometry and reference. 34
Figure 17 – Kinematic steering of a four-wheel vehicle. 36
Figure 18 – Slip angle at the peak F_y shift due to vertical load shift (Adapted ISO Coordinates). 37
Figure 19 – Pro-Ackerman, parallel and anti-Ackerman steering systems and their correlations to the position of the tie rod. 37
Figure 20 – Graphical representation of the Newton-Raphson method. 39
Figure 21 – An example of divergence due to a bad initial guess in the Newton-Raphson algorithm. 40
Figure 22 – Vector connecting two points P_i and P_j located on different bodies. . 42
Figure 23 – Spatial spherical joint representation. Points P_i and P_j must always coincide. 43
Figure 24 – Pareto optimal set for a cantilever beam optimization. 45
Figure 25 – Local and global minimum of the function $z = 10(x^2 \sin x(-(y-10)^2 + 100))$, $0 < [x; y] < 20$. 47
Figure 26 – Generic Evolutionary Algorithm flowchart. 48
Figure 27 – Examples of a ranked selection of a small population for two functions. 50
Figure 28 – Uniform crossover representation. 52
Figure 29 – Graphical representation of the Linear-BLX-α crossover 52
Figure 30 – Graphical representation of the Voluminal-BLX-α crossover 53
Figure 31 – Suspension pickup points change. 57

Figure 32 – Suspension output change for a heave motion (from -50mm to +50mm). 58
Figure 33 – Algorithm speed comparison between different programming languages. 59
Figure 34 – Base solver classes. 60
Figure 35 – Vehicle assembly classes. 61
Figure 36 – Suspension simulation classes. 61
Figure 37 – Double A-Arm kinematic points identification. 62
Figure 38 – Example of an iterative suspension kinematics design process. . . . 64
Figure 39 – Proposed optimization workflow. 65
Figure 40 – OptimumGenetics template class diagram. 68
Figure 41 – Boundaries class diagram. 69
Figure 42 – Genetic operators implementation. 69
Figure 43 – Suspension factory class implementation. 70
Figure 44 – Suspension evaluation class implementation. 70
Figure 45 – Baseline suspension system model in OptimumKinematics (isometric view) 71
Figure 46 – Suspension model used in parametric analysis (Front View). 72
Figure 47 – Convergence graphs between runs 1 and 2. 75
Figure 48 – Convergence graphs between runs 3 and 4. 75
Figure 49 – Convergence graphs between runs 5 and 6. 76
Figure 50 – Single objective optimization results and KPIs comparison. 77
Figure 51 – Optimization run 5 resulting system overlaid with objective function. 78
Figure 52 – Convergence graphs between runs 7 and 8. 79
Figure 53 – Runs 7 and 8 objectives comparison. 79
Figure 54 – Convergence graphs between runs 9 and 10. 80
Figure 55 – Runs 9 and 10 objectives comparison. 81
Figure 56 – Convergence graphs between runs 11, 12 and 13. 82
Figure 57 – Runs 11, 12 and 13 objectives comparison. 83
Figure 58 – Convergence graphs between runs 14 and 15. 84
Figure 59 – Runs 14 and 15 objectives comparison. 84
Figure 60 – Multi-Objective Optimization (MOO) results comparison. 85
Figure 61 – Objective results on run 15. 86
Figure 62 – F_y vs α plot for the proposed GT3 front tires (SAE coordinates). . . 89
Figure 63 – F_y vs α plot for the proposed GT3 rear tires (SAE coordinates). . . . 90
Figure 64 – Peak slip angle α shift with relation to the normal load F_z. 90
Figure 65 – Tire peak lateral force F_y with slip angle F_z variation for different values of inclination angle γ. 91
Figure 66 – G-G diagram of a GT3 race car for a full lap in Sebring. 92
Figure 67 – Convergence graphs of case study. 98

Figure 68 – Final result comparison of objectives. 99
Figure 69 – Heave motion evaluation objectives. 101
Figure 70 – Vehicle suspension roll gradient for several laps. 102
Figure 71 – Roll motion evaluation objectives. 103
Figure 72 – Vehicle suspension pitch gradient for several laps. 104
Figure 73 – Pitch motion evaluation objectives. 105
Figure 74 – Vehicle speed and steering wheel angle channels of a GT3 car on a flying lap in Sebring. 106
Figure 75 – Steering motion evaluation objectives. 108
Figure 76 – Suspension optimization comparison (Front View). 109
Figure 77 – Suspension optimization comparison (Side View). 110
Figure 78 – Suspension optimization comparison (Top View). 110
Figure 79 – Full set of parameters of the baseline suspension used in the case study extracted from OptimumKinematics Excel exporter. 118

LIST OF TABLES

Table 1 – Template parameters for the kinematics optimization. 67
Table 2 – Design space defined for the Genetic Algorithm (GA) parametric study. 73
Table 3 – Fixed parameters for the GA parametric study. 73
Table 4 – Truncation selection for reproduction variance analysis. 74
Table 5 – Tournament selection for reproduction variance analysis with generational selection for replacement. 74
Table 6 – Tournament selection for reproduction variance analysis with steady state selection for replacement. 76
Table 7 – Ranked selection for reproduction vs population size variance analysis setup. 78
Table 8 – Tournament selection for reproduction vs steady state selection for reproduction selection size variance analysis setup. 80
Table 9 – Ranked selection for reproduction and replacement variance analysis setup. 81
Table 10 – Ranked selection for reproduction with steady state selection for replacement variance analysis setup. 82
Table 11 – Final result of runs 7 to 15. 85
Table 12 – Baseline suspension system pickup points. 88
Table 13 – Common parameters of a GT3 car 92
Table 14 – Weight transfer estimation for the proposed conditions. 93
Table 15 – Front suspension box boundaries definition for case study. 94
Table 16 – Front suspension spherical boundaries definition for case study. . . . 95
Table 17 – Rear suspension box boundaries definition for case study. 96
Table 18 – Rear suspension spherical boundaries definition for case study. . . . 97
Table 19 – GA setup used in case study. 97
Table 20 – Final indicators of the optimization run for the case study. 98
Table 21 – Final result objectives fitness 100
Table 22 – Optimized system pickup points. 109
Table 23 – System points used in parametric analysis data. 117

LIST OF ABBREVIATIONS AND ACRONYMS

EMOO	Evolutionary Multi-Objective Optimization
GA	Genetic Algorithm
KPI	Key Performance Indicator
MOGA	Multi-Objective Genetic Algorithm
MOO	Multi-Objective Optimization
OS	Operating System
SOO	Single Objective Optimization
SQP	Sequential Quadratic Programming
SUMT	Sequential Unconstrained Minimization Technique
UML	Unified Modelling Language

LIST OF SYMBOLS

F_y	Tire lateral force
α	Tire slip angle
F_z	Tire normal force
γ	Tire inclination angle
F_x	Tire longitudinal force
M_z	Tire self-aligning torque
κ	Tire slip ratio
V_x	Longitudinal component of the velocity vector of the wheel
r_e	Effective rolling radius
Ω	Wheel angular velocity
V_y	Lateral component of the velocity vector of the wheel
y	Output variable of the Magic Formula
D	Peak value
C	Shape factor
B	Stiffness factor
x	Input variable of the Magic Formula with horizontal shift
E	Curvature factor
Y	Output variable of the Magic Formula with vertical shift
X	Input variable of the Magic Formula
S_V	Vertical shift
S_H	Horizontal shift
$\%AD$	Anti-Dive percentage
$\%BT_F$	Braking torque at front percentage
L	Wheelbase length
θ_{F_D}	Angle between the line that connects the front wheel contact patch and the side view instant center and the ground plane
h_{CG}	Center of gravity height w.r.t. ground
$\%AL_F$	Anti-Lift Front percentage
$\%DT_F$	Driving torque at front percentage
θ_{F_L}	Angle between the line that connects the front wheel center and the side view instant center and the ground plane
$\%AL_R$	Anti-Lift Rear percentage
θ_{R_L}	Angle between the line that connects the rear wheel contact patch and the side view instant center and the ground plane
$\%AS$	Anti-Squat percentage
θ_{R_D}	Angle between the line that connects the rear wheel center and the side view instant center and the line parallel to the ground plane passing through the wheel center

δ_1	Steering angle of the inner wheel
δ_2	Steering angle of the outer wheel
t_F	Vehicle front track length
c_{factor}	Steering rack c-factor
D_{SR}	Steering rack displacement
ϕ_p	Pinion rotation
Φ_q	Jacobian matrix of constraints
Φ	Generic mathematical constraint
q	Generalized coordinate
α_{BLX}	Scaling coefficient for the crossover variations
p_s	Selection pressure
g_0	Overall fitness value of the best individual
$\bar{g}$	Average overall fitness values of all individuals in the current population
f	Objective fitness value
$\bar{f}$	Overall fitness value of the individual
σ_g	Gaussian mutation standard deviation
F_d	Downforce
$C_\ell A$	Downforce coefficient multiplied by the frontal area
ρ_{air}	Air density
V	Vehicle speed
W	Vehicle total weight
m	Vehicle total mass
g	Gravity
W_F	Front axle weight
m_F	Vehicle total mass distribution (% front)
$C_{\ell_{dist}}$	Downforce distribution (% front)
W_R	Rear axle weight
ΔW	Total weight transfer
A_y	Vehicle lateral acceleration
t_R	Vehicle rear track length
$LLTD$	Lateral load transfer distribution (% front)
ΔW_F	Weight transfer (front)
ΔW_R	Weight transfer (rear)
δ_{Ack}	Ackerman angle

CONTENTS

1 INTRODUCTION . . . 17
1.1 OBJECTIVES . . . 18
1.1.1 Main objective . . . 18
1.1.2 Specific objectives . . . 18
2 THERORETICAL FOUNDATION . . . 19
2.1 THE PROBLEM IMPOSED BY RACING . . . 19
2.2 TIRES . . . 20
2.2.1 Introduction . . . 20
2.2.2 Longitudinal dynamics . . . 21
2.2.3 Lateral dynamics . . . 22
2.2.4 The Magic Formula tire model . . . 23
2.3 SUSPENSION KINEMATICS . . . 24
2.3.1 Motions . . . 25
2.3.2 Wheel relative angles . . . 26
2.3.3 Instant centers of rotation . . . 27
2.3.3.1 Anti-Dive . . . 28
2.3.3.2 Anti-Lift . . . 29
2.3.3.3 Anti-Squat . . . 31
2.4 THE DOUBLE A-ARM SUSPENSION . . . 31
2.5 RACK AND PINION STEERING SYSTEM . . . 32
2.5.1 King Pin geometry . . . 33
2.5.2 Tie rods location and the Ackerman geometry . . . 35
2.5.3 Steering ratio and the C-Factor . . . 38
2.6 NUMERICAL METHODS ON SPATIAL KINEMATICS . . . 38
2.6.1 The Newton-Raphson algorithm . . . 38
2.6.2 Point collection and distance constraints . . . 39
2.6.3 The multi-body approach to suspension kinematics . . . 41
2.7 OPTIMIZATION . . . 44
2.7.1 Design variables . . . 44
2.7.2 Multi objective optimization and the Pareto-optimal set . . . 45
2.7.3 Metaheuristics . . . 46
2.7.4 Genetic algorithm . . . 47
2.7.4.1 Fitness function . . . 49
2.7.4.2 Selection operators . . . 49
2.7.4.2.1 Truncation selection . . . 49
2.7.4.2.2 Deterministic Tournament . . . 49
2.7.4.2.3 Ranked selection . . . 50

2.7.4.2.4 Steady state replacement *50*
2.7.4.2.5 Generational replacement *51*
2.7.4.2.6 Elitist replacement *51*
2.7.4.3 Crossover methods 51
2.7.4.3.1 Uniform crossover *51*
2.7.4.3.2 Linear BLX-α crossover *51*
2.7.4.3.3 Voluminal BLX-α crossover *53*
2.7.4.4 Mutation methods 53
2.7.4.4.1 Uniform mutation *54*
2.7.4.4.2 Gaussian mutation *54*
2.7.4.5 Selection pressure 54
2.8 STATE OF THE ART 54
3 METHODOLOGY 56
3.1 THE PROBLEM IN SUSPENSION KINEMATICS DESIGN 56
3.2 THE STARTING POINT: OPTIMUMKINEMATICS 57
3.3 PROGRAMMING LANGUAGES 58
3.4 PROGRAM STRUCTURE 59
3.5 KINEMATIC MODEL 62
3.5.1 Outputs 63
3.6 GENETIC ALGORITHM 64
3.6.1 Optimization workflow 64
3.6.2 Design variables 65
3.6.3 Evaluation and objective functions 65
3.6.4 Operators 66
3.6.5 Key performance indicators 66
3.6.6 Implementation 67
3.7 CASE STUDY 69
4 PARAMETRIC ANALYSIS 72
4.1 GENETIC ALGORITHM PARAMETRIC ANALYSIS 72
4.1.1 Single objective optimization parameters 73
4.1.2 Multiple objectives optimization parameters 77
4.1.3 Remarks 85
5 CASE STUDY 87
5.1 PROBLEM DESCRIPTION 87
5.2 TIRES 87
5.3 WEIGHT TRANSFER ANALYSIS 88
5.4 DESIGN SPACE 93
5.5 GA CONFIGURATION 93
5.6 CONVERGENCE 94

5.7 EVALUATION AND RESULTS 95
5.7.1 Heave 97
5.7.2 Roll 100
5.7.3 Pitch 103
5.7.4 Steering 105
5.8 RESULTING SYSTEM 107
6 CONCLUSIONS 111
REFERENCES 112
APPENDIX A – KINEMATIC OUTPUTS CALCULATIONS 116
APPENDIX B – PARAMETRIC ANALYSIS DATA 117
APPENDIX C – CASE STUDY BASELINE SYSTEM DATA 118

1 INTRODUCTION

Automobile racing is structured in many categories. No matter how distinct these modalities are, there are common grounds to all of them. In general, the objective in racing is to cover a specific track length or circuit in the shortest possible time (SEWARD, 2014).

As Seward (2014) affirms that, in order to accomplish this objective, the driver must accelerate the vehicle to the highest speed, break as late as possible and in the smallest distance and also spend less time in cornering while maximizing the exit speed. This implies that the competitive driver will spend no time cruising and will exploit the whole vehicle potential.

Every innovation in racing vehicle development has resulted in the expansion of the acceleration limits of the vehicle and its exploitation through improvements of control and driving techniques (MILLIKEN, W. F.; MILLIKEN, D. L., 1995). Under this perspective, the steering and suspension systems play a vital role on the success of a new racing prototype, since the steering system is the only real feedback source to the driver and the suspension movements have an affect on the tires' performance, thus, the overall vehicle's performance

William F. Milliken and Douglas L. Milliken (1995) point that the suspension development must be done hand-to-hand with the tires and their characteristic behaviours, whereas the suspension must maintain the tires in optimal contact with the ground at all possible times. Using mathematical models that describes the tire behaviour, it is possible to optimize the suspension to maximize the forces generated at the tire-ground contact in each one of the axles, which, therefore, expands the limits of acceleration, both lateral and longitudinal.

Many tire models have been developed during the last century, each one with its own specific purpose. Different levels of precision and complexity can be introduced in the distinct use categories, involving completely different approaches (PACEJKA, 2006).

A racing vehicle should be, ideally, developed to enhance the driver-vehicle system performance. Unfortunately, to the moment, there are no easily exploitable driver models. That is why the optimization of a racing vehicle refers mostly about the vehicle itself (MASTINU et al., 2007).

Mastinu et al. (2007) clarify that one of the biggest technical challenges is to achieve maximum speed in cornering, while tire forces in front and rear axles are balanced to maintain vehicle handling. Due to this, the adjustment of tire characteristics and suspension behaviour is of crucial importance.

Even though some real world problems can be reduced to a single objective, it is really difficult to define all of the criteria in function of a single objective. The

establishment of multiple objectives usually gives a clearer description of the problem (ABRAHAM; GOLDBERG, 2006).

According to Abraham and Goldberg (2006), the Multi-Objective Optimization (MOO) is available for at least two decades and its application in real problems is continuously increasing. Evolutionary algorithms can be employed as tools of MOO and are distinguished by a population of different solutions, whose reproduction operator allows the process of combining existing solutions to generate new ones.

The suspension system and its influence on tire behaviour and driver reactions is far too complex to be given a single objective on an optimization process. Under this point of view, this work proposes the implementation of an Evolutionary Multi-Objective Optimization (EMOO) algorithm as a tool to improve an existing system or even design a brand new one.

1.1 OBJECTIVES

1.1.1 Main objective

Present an application capable of analysing and optimizing the kinematic behaviour of a vehicle's suspension system.

1.1.2 Specific objectives

- Implement a kinematics analysis software capable of handling tridimensional suspension systems,
- Model a Double A-Arm suspension system using the implemented software,
- Implement an EMOO,
- Develop a case study on the optimization of a whole vehicle kinematics,

2 THERORETICAL FOUNDATION

This chapter covers the main topics of fundamental importance to the adequate comprehension of the theme. It is initiated with the elucidation of the topics through the justification and necessity of the proposed project in automotive applications. The chapter deepens on tire behavior and its correlations with the suspension and steering system, finishing with an overview of optimization methods and the detailing of the Genetic Algorithm (GA), used in this work.

2.1 THE PROBLEM IMPOSED BY RACING

The technical global objective in motorsport is the accomplishment of a vehicular configuration capable of covering a given track extension in minimum time or at maximum average speed, when operated manually by a driver that uses techniques under his or her limitations. An important principle of racing is that the vehicle's speed should never be constant, if not limited by regulation or the vehicle's top speed (MILLIKEN, W. F.; MILLIKEN, D. L., 1995).

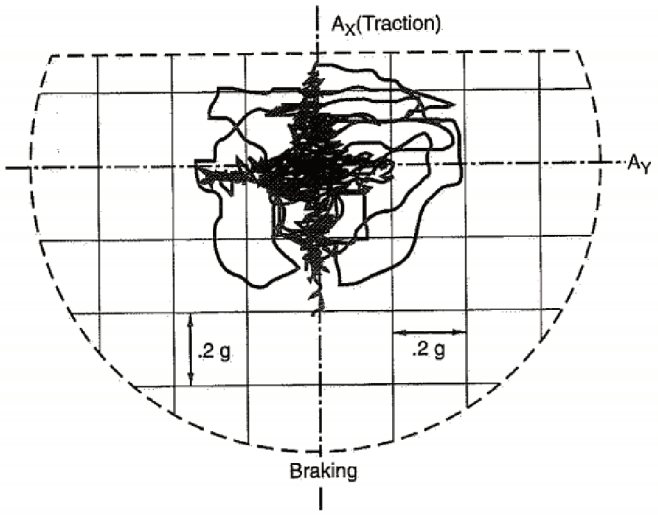
The problem imposed by racing, according to William F. Milliken and Douglas L. Milliken (1995), can be synthesized as the extension of the drive-vehicle system capabilities to spend the most possible time at the friction limits imposed by the contours of the G-G diagram. The diagram's limits are defined by the powertrain and tire forces, disposing of any load transfer effects, suspension movements, balance and brake bias. These simplifications swell the grip potential, establishing a limit, given on the diagram. Thus, the G-G diagram works as an upper boundary of the vehicle's grip factor, where the engineers and the driver must work together to exploit the inner region. An example of a G-G diagram is given in Figure 1.

This way, the system can be evaluated through the analysis of this diagram's limits. Conceptually, the friction envelope can be applied to the vehicle by collapsing all four wheels in a single equivalent contact point. In fact, a real vehicle should not be able to constantly reach the limits imposed by the G-G diagram.

The diagram presented on Figure 1 show in brighter lines the frontier that limits the vehicle's accelerations. It is possible to observe that the upper portion, which represents the longitudinal acceleration of the vehicle is limited by the powertrain characteristics. The disarranged lines represent the accelerometer channel's data on X and Y axes. Therefore, the suspension system exploits the delimited area on the G-G diagram, once it optimizes the use of the tires. The more disperse the data in the diagram, the better the driver is using the available tire grip of the car.

The pair analysis, presented by William F. Milliken and Douglas L. Milliken (1995) is a more practical alternative to the development of new racing prototypes when compared to the G-G diagram. Normally the objective of this approach is to maximize the

Figure 1 – Example of a G-G diagram of a common passenger car operating on normal conditions.



Source: William F. Milliken and Douglas L. Milliken (1995)

lateral force generated at the tires on both front and rear axles.

Through pair analysis it is possible to explore the changes on functional parameters due to project condition changes on a simplified manner. The pair analysis shows the relationship of the suspensions' kinematics and the tires work conditions, whose characteristics will be presented next.

2.2 TIRES

This section introduces the concept of tire modeling, how they can be exploited through computer simulations and the correlation of their characteristics with suspension kinematics.

2.2.1 Introduction

The tires have a crucial importance in the dynamic behavior of a vehicle, as all the effort made to accelerate, brake and change the car's directions are transferred to the ground through the tires.

Given the fact that the tires have such a complex structure, they must be studied apart from the whole vehicle for their characteristics to be fully comprehended. The force generation mechanisms in the tires are separated in longitudinal, lateral and self-aligning torque, introduced in this section. Despite their relevance in racing applications, combined tire forces and moments will not be discussed in this work, in behalf of simplification. In addition, the overturn moment of the tires are not discussed in this work as well.

Tires are naturally susceptible to deformation and they depend on it to generate lateral and longitudinal forces. The interaction between the tires and the road generate the lateral and longitudinal forces needed to handle the car (MILLIKEN, W. F.; MILLIKEN, D. L., 1995). The main dynamic characteristics of the tires are presented in the following sections, based on the fundamentals presented by Pacejka (2006).

The tire longitudinal and lateral forces and the self-aligning moment, F_X, F_Y and M_Z, respectively, are results of the lateral and longitudinal slip conditions inputs, α and κ, respectively, and the tire inclination angle γ, which will be presented in the next sections. These forces calculations can be generalized for steady state conditions by the set of functions (PACEJKA, 2006):

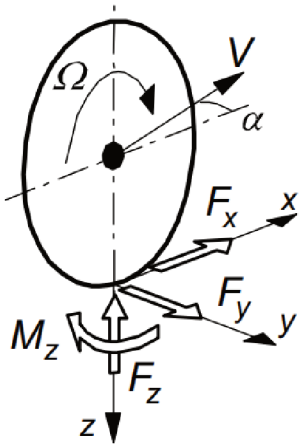
$$F_X = f(\alpha, \kappa, \gamma, F_Z) \tag{1}$$

$$F_Y = f(\alpha, \kappa, \gamma, F_Z) \tag{2}$$

$$M_Z = f(\alpha, \kappa, \gamma, F_Z) \tag{3}$$

The tire reference coordinates and angles used in this chapter (Adapted SAE) are shown in Figure 2 while other coordinate systems for tire modelling and analysis are illustrated in Figure 3.

Figure 2 – Adapted SAE tire reference coordinates.

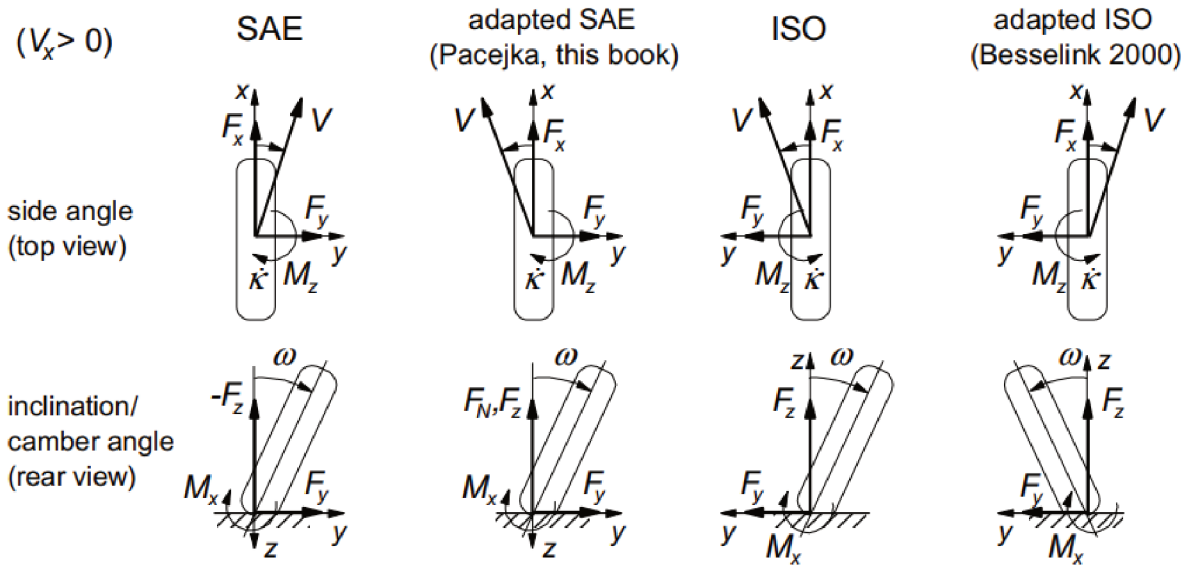


Source: Pacejka (2006)

2.2.2 Longitudinal dynamics

The starting situation where all slip components are zero can be defined as the wheel rolling freely, on its upright condition, without applying a driving torque, over a flat level road surface along a straight line at zero side slip. When the wheel motion from this

Figure 3 – Different tire coordinate systems and sign conventions.



Source: Pacejka (2006)

zero slip condition, the slip is accompanied by a build-up of additional tire deformation and partial sliding in the contact patch. Such behavior results on additional horizontal forces and aligning torque are generated. When an acceleration or braking torque is applied about the wheel axis, a longitudinal slip arises that is defined in Equation (4) (PACEJKA, 2006):

$$\kappa = -\frac{V_X - r_e\Omega}{V_X} \tag{4}$$

where κ is the slip ratio, V_X the longitudinal speed, r_e the effective tire radius and Ω the angular velocity.

The sign is taken such that for a driving force, κ assumes a positive value and generates a positive longitudinal force F_X. Similarly, the value of κ becomes negative and generates a negative F_X for braking forces. At wheel lock, $\kappa = 1$ and for very slippery roads, κ becomes a very large value (higher than 1) for acceleration torques. For braking torques, the κ will only be less than -1 if the car is moving forward and the vehicle is in reverse gear.

2.2.3 Lateral dynamics

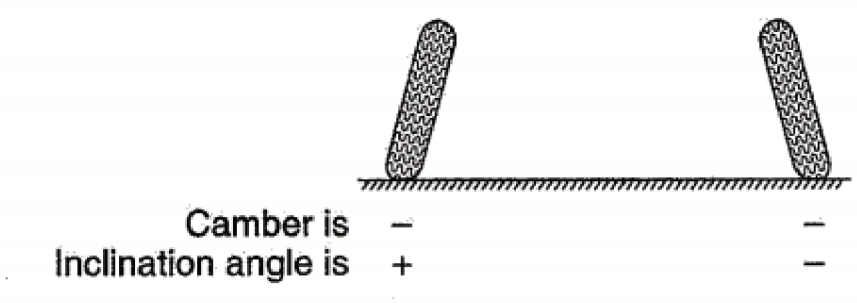
Lateral wheel slip is defined as the ratio of the lateral and the forward velocity of the wheel, which is equivalent to minus the tangent of sleep angle α, as shown in Equation (5). The negative sign on the right side of the equation is taken such as positive values of α generate positive values for F_y. If the coordinate system of the tire is changed, this sign is changed as well.

$$\tan \alpha = -\frac{V_y}{V_x} \tag{5}$$

where α is the slip angle and V_y is the lateral velocity of the tire.

In the Adapted SAE coordinate system, the inclination angle γ is defined positive when looking from behind the wheel is tilted to the right. The inclination angle differs from the camber angle — introduced in subsection 2.3.2 — in result of the reference of such angles. For instance, the camber uses the chassis to define the sign convention, whilst the inclination angle is arbitrary. This is illustrated in Figure 4, considering the Adapted SAE coordinate system.

Figure 4 – Difference between camber angle and inclination angle.



Source: William F. Milliken and Douglas L. Milliken (1995)

2.2.4 The Magic Formula tire model

Tires have such complex behavior and structure that nowadays no theory has yet been propounded. The tire characteristics still present a challenge to devise a theory which coordinates the vast mass of empirical data and give some guidance to the manufacturer and user. From the vehicle dynamicist perspective, the mechanical behavior of the tire needs to be investigated systematically in terms of its reactions to the inputs associated with the wheel motions and road conditions (PACEJKA, 2006).

The widely used *Magic Formula* model is a semi-empirical model, whose development was started in mid-eighties. In these models the combined slip situation was modelled from a physical point of view. The general form of the model is:

$$y = D \sin\left(C \arctan\left(Bx - E(Bx - \arctan\left(Bx\right)\right)\right)\right) \tag{6}$$

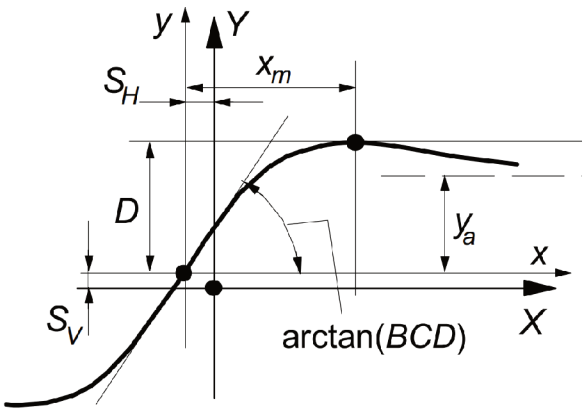
where:

$$Y(X) = y(x) + S_V \tag{7}$$

$$x = X + S_H \tag{8}$$

Each of the factors above can be found in the list of symbols. The above function typically produces a curve that is similar to the one shown in Figure 5.

Figure 5 – General Magic Formula behavior.



Source: Pacejka (2006)

The formula is capable of accurately reproduce the characteristics that match the measured curves for the side force F_y, the longitudinal force F_x and if desired also for the self-aligning torque M_z as functions of their respective slip quantities. The accuracy is illustrated by overlaying the tire test data and proper coefficients, as shown in Figure 6. The full set of the Magic Formula equations will not be introduced in this work, but are fully described in Pacejka (2006). The fitted tire model is shown in continuous lines while the tire raw data is shown in noisy points in Figure 6. The model was fitted using OptimumTire[1] modeling software.

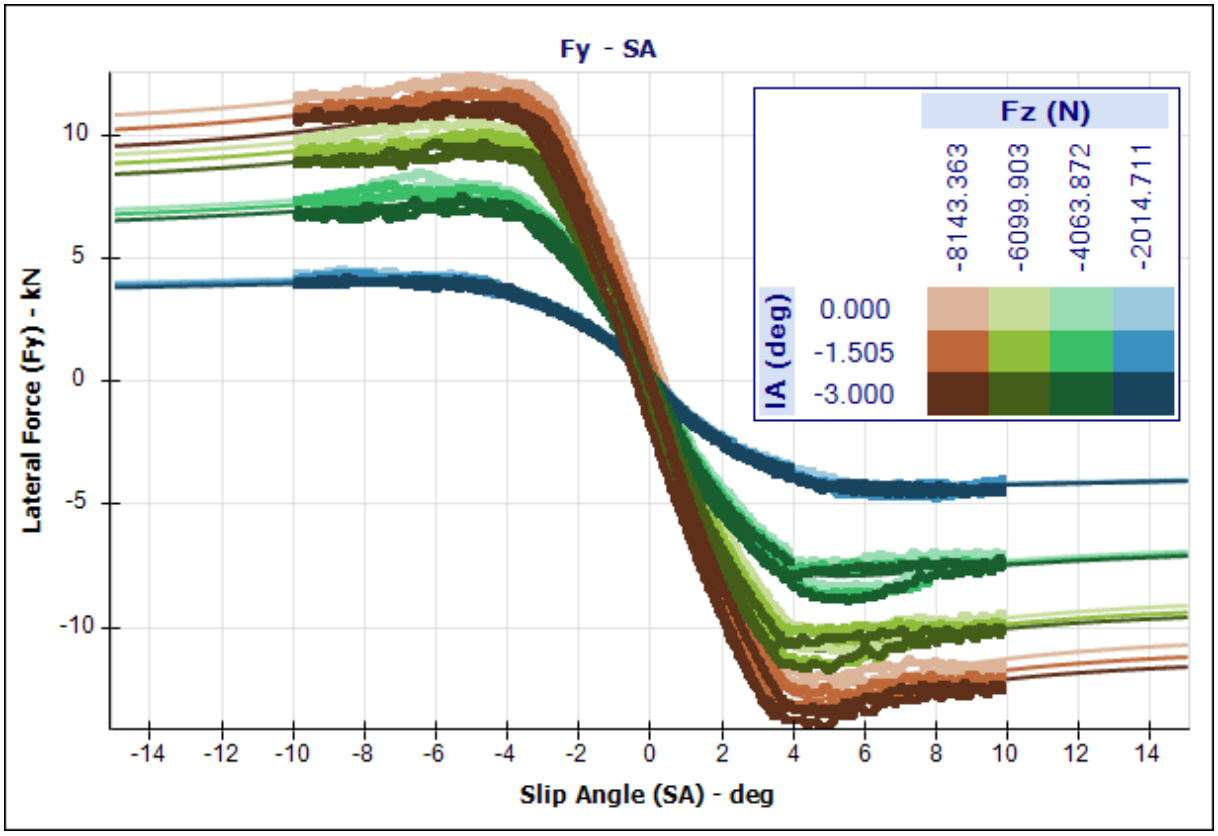
With such model, the suspension designer can exploit the tire characteristics and therefore define better and more accurate objectives — such as ideal camber and Ackerman values — for the suspension mechanism, which is approached in the next section.

2.3 SUSPENSION KINEMATICS

The suspension is the system that links the chassis (suspended mass) to the wheels (non-suspended masses) and allows relative motion between these two parts and the ground (JAZAR, 2013). According to Seward (2014), the main purpose of designing a race car kinematics is to maintain the wheels — hence, the tires — at the optimal angle to the road surface at all possible times.

[1] OptimumG, LLC, more information available at `https://optimumg.com/product/optimumtire/`

Figure 6 – Fitted Magic Formula tire model (SAE Coordinates).



Source: Author (2021)

Furthermore, the kinematic design of the suspension needs to take into account the effects that will be introduced by its elastic elements, such as springs and anti-roll bars. Highly influenced by the position of the instant centers influence, the load transfer distribution and response time need to be estimated prior to the design phase.

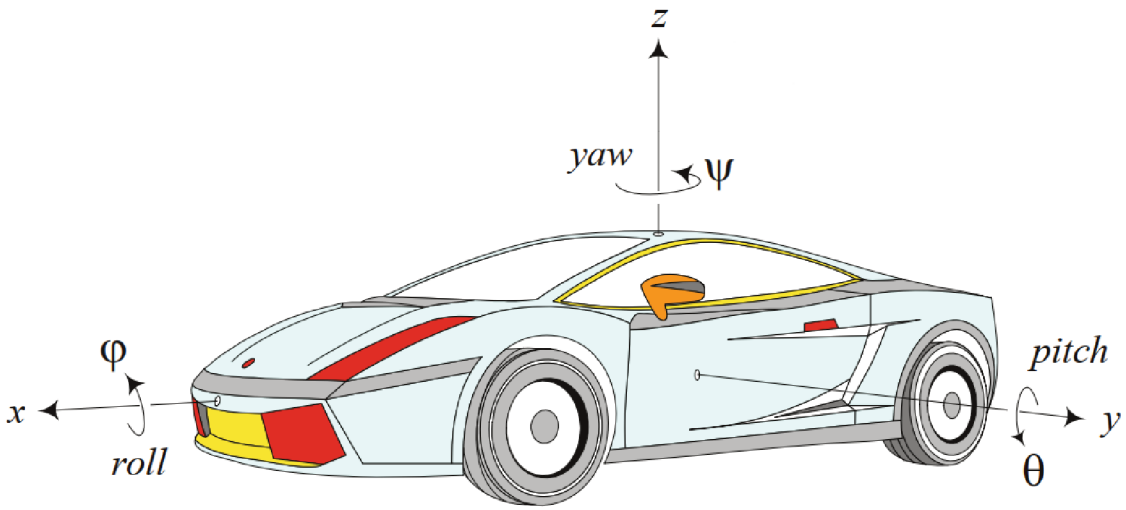
2.3.1 Motions

The main vehicle movements controlled by the suspension are heave, pitch and roll. Defined by Dixon (2009), pitch is the angular movement of the vehicle around the Y axis, with positive values for the front lower than the rear, by the ISO coordinates, knowing the Y axis points to the left of the car.

The roll movement is defined by the rotation of the vehicle around the X axis, longitudinal to the vehicle, where in both SAE and ISO coordinates the direction of the X axis points towards the travel direction. Therefore, for a vehicle with positive roll, the right side is lower than the left. Figure 7 shows the Cartesian axes and the relative movements of the chassis in the ISO coordinate system.

The relative movements described above are highly dependent on the vehicle suspension system. Usually, the pitch is caused by longitudinal accelerations, while the roll movement is provoked by lateral accelerations. However, due to the coupling and

Figure 7 – ISO Coordinate Systems for vehicle.



Source: Adapted from Jazar (2013)

the possible asymmetry of the suspension mechanism, longitudinal accelerations can cause roll movements and lateral accelerations can provoke pitch movements.

2.3.2 Wheel relative angles

As mentioned previously, the suspension system is responsible for maintaining the wheel at the optimal angle and position at all possible times. The two main angles that define the wheels angular positions with relation to the chassis (and possibly the ground) are camber and toe.

The camber angle — or camber only —- is the angle between the wheel mid plane and the vertical (DIXON, 2009). Seward (2014) defines that the camber angle is said to be positive if the top of the wheel is leaning outwards the vehicle, as shown in Figure 8a.

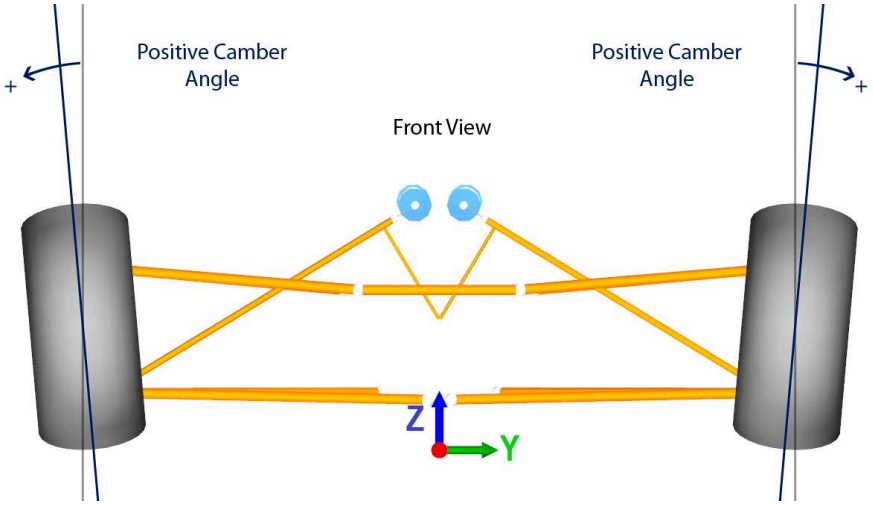
As seen in section 2.2, the camber angle — previously referenced as inclination angle — affects the lateral tire performance, making it a crucial parameter for a competitive vehicle. To optimize a vehicle's behavior in a turn, the suspension should provide a slightly negative camber in roll, as dictated by the tire model.

The toe angle is the angle between the mid plane of the car and the direction of the wheel. It is said that a vehicle has *toe-in* if the wheels are pointing inwards the vehicle direction. Similarly, *toe-out* happens when the wheel directions diverge (JAZAR, 2013). The toe angle is shown in Figure 8b. The vehicle is usually steered by a change in the steering angle of the front wheels, initially provoked by the driver. However, the suspension mechanism geometry can also cause this effect in the wheels steering angle. This effect of steering caused by the vertical movement of the wheel is known

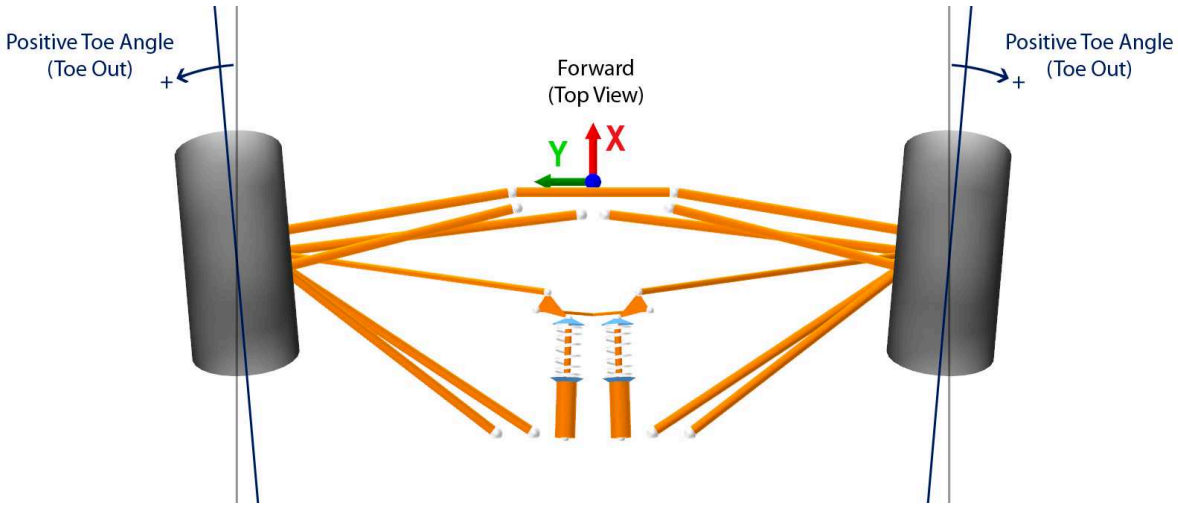
as bump steer (DIXON, 2009).

Figure 8 – Camber and toe angle definitions.

(a) Camber angle.



(b) Toe angle.



Source: OptimumG (2019)

In agreement with Jazar (2013), toe settings affect three major performances: tire wear — hence, tire temperature — , straight-line stability and corner entry handling. For minimum tire-wear and power loss, the desired toe is zero, whilst different toe setups can be used in racing applications to the driver's and the performance engineer's desires accordingly.

2.3.3 Instant centers of rotation

In planar kinematics, at a given instant, the velocity of any point of the body can be expressed as a rotation around a given point. Similarly, in spatial kinematics, the same can be assumed for a body rotating around an axis. These entities are called the instant center and instant axis of rotation (JAZAR, 2013).

The instant centers are analysed within the kinematics studies, once they play a big role on vehicle handling and ride comfort (PAPAIOANNOU et al., 2020). This happens mostly because of the said *geometric* and *elastic* load transfers, as a result of the difference of the suspension elements' stiffness and load paths.

In accordance to the instant axis definition, the roll axis is the axis around which the vehicle's body rotates in an axis roughly perpendicular to the ground's X axis. Similarly, the pitch axis is the axis around which the vehicle's body rotates in an axis roughly perpendicular to the ground's Y axis. These axes are represented in Figure 9.

The "classic" method to calculate the roll center is shown by William F. Milliken and Douglas L. Milliken (1995), Jazar (2013) and Seward (2014). However, it is important to note that the concept of instant centers — that later will be expanded to roll and pitch centers — are not more than a simplification of reality for better human understanding. Dixon (2009) points out that the idea of a single roll center for the suspension is just an approximation. This concept has been criticized in the past but is still used. Originally, the roll center idea was essentially a simplification to facilitate the manual calculation of the vehicle's lateral dynamics. That purpose diminished with the introduction of computers.

Extending the roll center to the lateral view, it's possible to calculate the pitch axis, which correlate to the anti-features: anti-dive, anti-squat and anti-lift. Similarly to lateral dynamics, the anti-features change the amount of load going through the springs on longitudinal dynamics. William F. Milliken and Douglas L. Milliken (1995) show that, for example, anti-dive reduces the bump deflection on forward braking while anti-squat reduces the amount of rebound travel on forward acceleration on rear wheel drive cars. However, such geometries drastically increases the load on the suspension elements, such as the wishbones.

Even though they are widely used, specially on aerodynamic cars, the usage of these features can yield negative effects in some cases, such as roughness on very wavy roads, vibration during braking (DIXON, 2009). In this work, the anti-dive, anti-squat and front and rear anti-lift are defined as following.

2.3.3.1 Anti-Dive

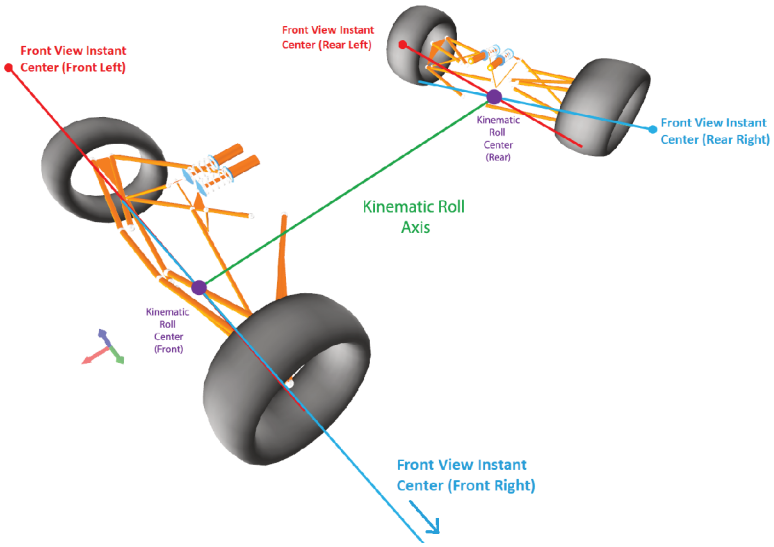
The anti-dive calculation is represented in Figure 10 and defined in Equation (9).

$$\%AD = \%BT_F \frac{L \tan \theta_{F_D}}{h_{CG}} \tag{9}$$

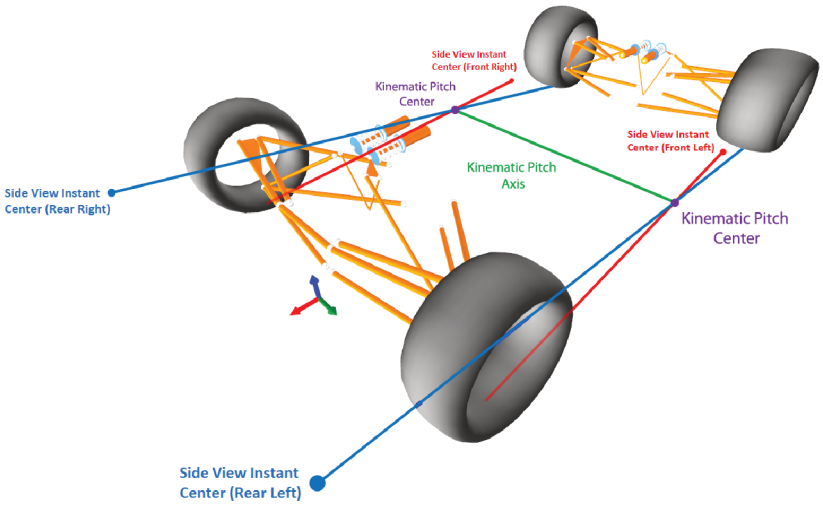
where $\%AD$ is the Anti-Dive percentage and $\%BT_F$ is the braking torque distribution in the front axle, in percentage as well. The remaining variables are defined in Figure 10.

Figure 9 – Kinematic roll and pitch axes representation

(a) Kinematic Roll Axis.



(b) Kinematic Pitch Axis.



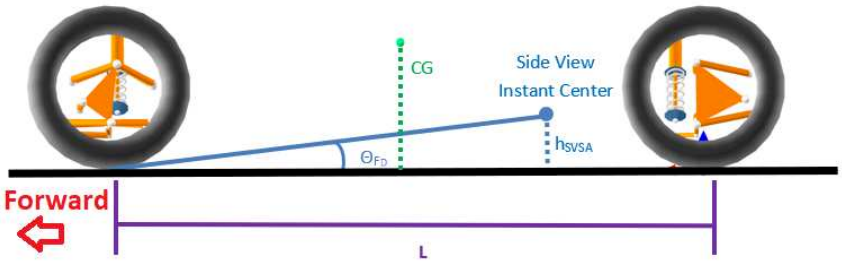
Source: OptimumG (2019)

2.3.3.2 Anti-Lift

The anti-lift calculation for the front axle is represented in Figure 11 and defined in Equation (10).

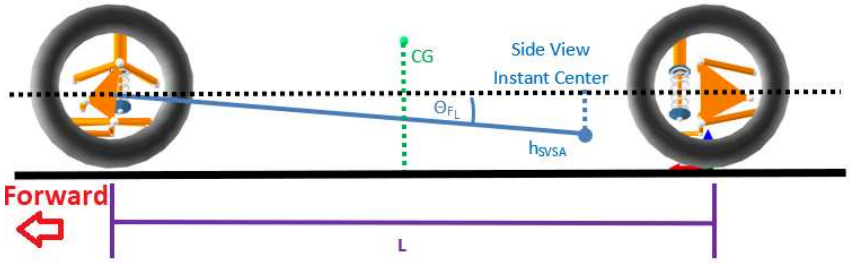
$$\%AL_F = \%DT_F \frac{L \tan \theta_{F_L}}{h_{CG}} \tag{10}$$

Figure 10 – Anti-Dive definition.



Source: OptimumG (2019)

Figure 11 – Anti-Lift on front definition.

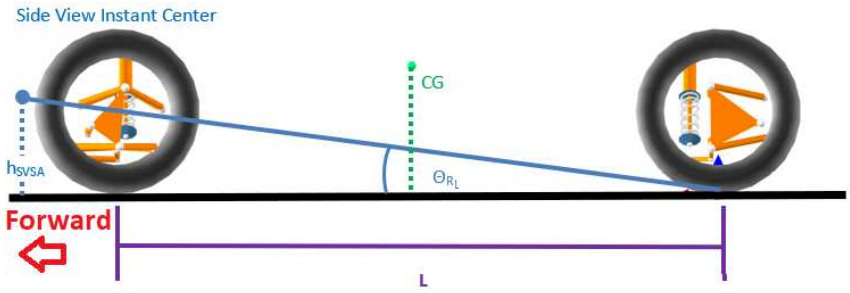


Source: OptimumG (2019)

where $\%AL_F$ is the Anti-Lift percentage in the front axle and $\%DT_F$ is the driving torque distribution in the front axle, in percentage as well. The remaining variables are defined in Figure 11.

Similarly, for the rear axle, the anti-lift calculation is geometrically represented in Figure 12 and defined in Equation (11).

Figure 12 – Anti-Lift on rear definition.



Source: OptimumG (2019)

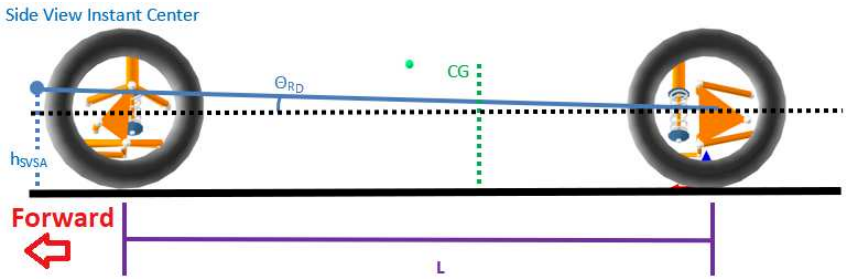
$$\%AL_R = (1 - \%BT_F)\frac{L\tan\theta_{R_L}}{h_{CG}} \tag{11}$$

where $\%AL_R$ is the Anti-Lift percentage in the rear axle. The remaining variables are defined in Figure 12.

2.3.3.3 Anti-Squat

Lastly, the anti-squat calculation is represented in Figure 13 and defined in Equation (12).

Figure 13 – Anti-Squat definition.



Source: OptimumG (2019)

$$\%AS = (1 - \%DT_F)\frac{L \tan \theta_{R_D}}{h_{CG}} \tag{12}$$

where $\%AS$ is the Anti-Squat percentage in the rear axle. The remaining variables are defined in Figure 13.

As stated by Mohan et al. (2008), the "classic" method of roll and pitch centers calculation is limited to the suspension type and the omission of the steering (toe) angle of the wheel. Therefore, this work uses the approach presented by this later method.

If the roll center happens to coincide with the suspended mass center of mass, there is no moment, therefore, the vehicle does not roll. In an intuitive way, bringing the roll center closer to the center of mass is not recommended, due to the fact that this type of geometry causes high scrub values, increases the non-damped loads in the suspension linkages and potentiates the phenomenon known as jacking (SEWARD, 2014), introduced in the next item.

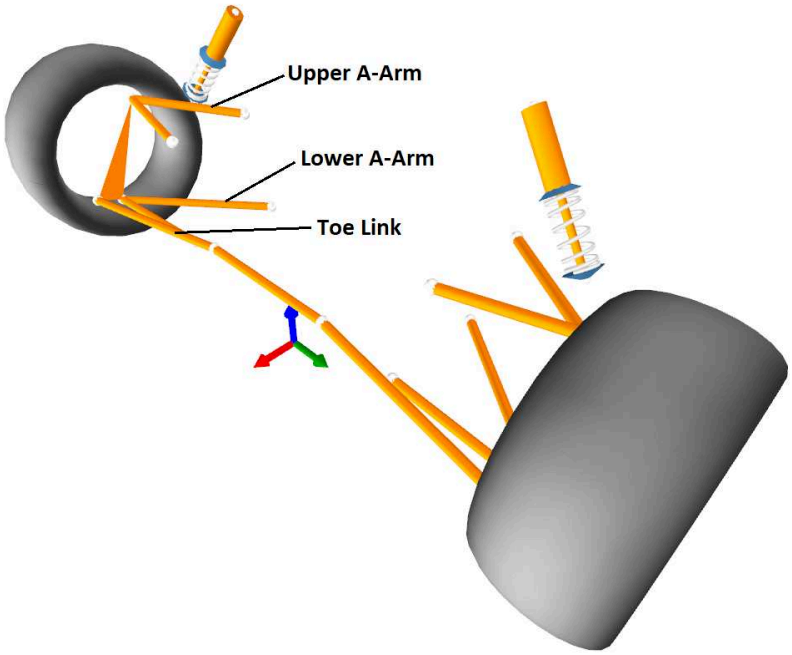
2.4 THE DOUBLE A-ARM SUSPENSION

There are many construction types for an independent suspension, however, double A-arm and McPherson strut suspensions are the simplest and the most common designs. Kinematically, the double-A arm or double wishbone system — also called short-long arm — is a spatial four-bar mechanism, with the chassis being the ground link and the coupler being the wheel set (JAZAR, 2013).

The actuation system can also assume different configurations, such as direct actuation or push/pull rods. Figure 14 shows an example of double wishbone mechanism with direct actuation.

Even though there are many types of suspension system, this work will focus on the double wishbone, because it is the most widely used in racing applications. It is

Figure 14 – Double Wishbone mechanism with direct coil-over actuation.



Source: Author (2021)

also versatile, which means that it not only works for both front and rear axles, but also allows the designer to combine different actuation and anti-roll bar types with the same base system.

In a double A-arm suspension system, the steering system's properties are more dependent on the connections of the steering mechanism and can be separated from the basic system of the suspension arms. With this, the suspension arms – also known as A-arms or wishbones – are used to control parameters such as roll center, camber gain, anti-dive and caster variation (DIXON, 2009).

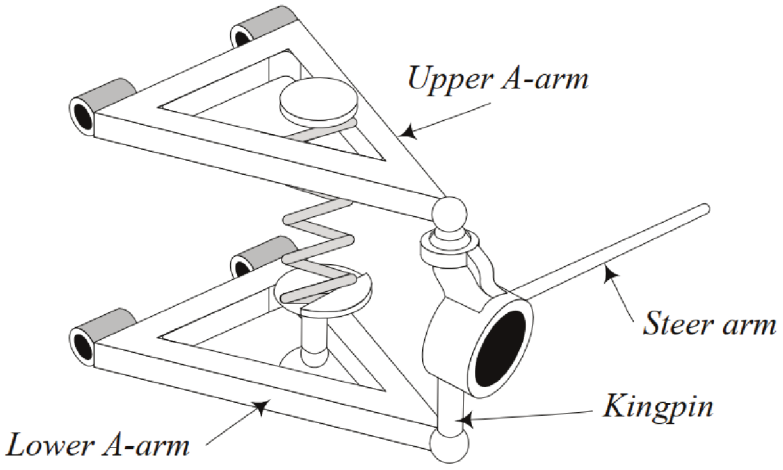
2.5 RACK AND PINION STEERING SYSTEM

The steering mechanism demonstrated in this work will be the rack and pinion, once they are invariably used in competition vehicles. The usual requirement for the steering system is that it has an accurate and quick response system, so that the driver can reach the limiting stops in less than half a turn on the steering wheel. However, this parameter can result in the driver making great efforts (SEWARD, 2014).

In agreement with Dixon (2009), the directional control of a vehicle is usually carried out by steering the front wheels, that is, rotating the wheels around a roughly vertical axis. Wheel steering is mainly the result of the steering wheel movement performed by the driver, with a smaller portion attributed to the suspension characteristics, such as the bump steer.

The direction of each wheel is controlled by the steering arm, shown in Figure 15. The steering arm is rigidly connected to the steering upright and can even be part of it. In the rack-and-pinion system, the rotation movement of the steering wheel is transmitted to the pinion and converted into translational movement by the rack. The connection of the box to the steering arm is made through an ideally rigid bar, called a drag link or tie rod.

Figure 15 – General construction of a double wishbone.



Source: Jazar (2013)

2.5.1 King Pin geometry

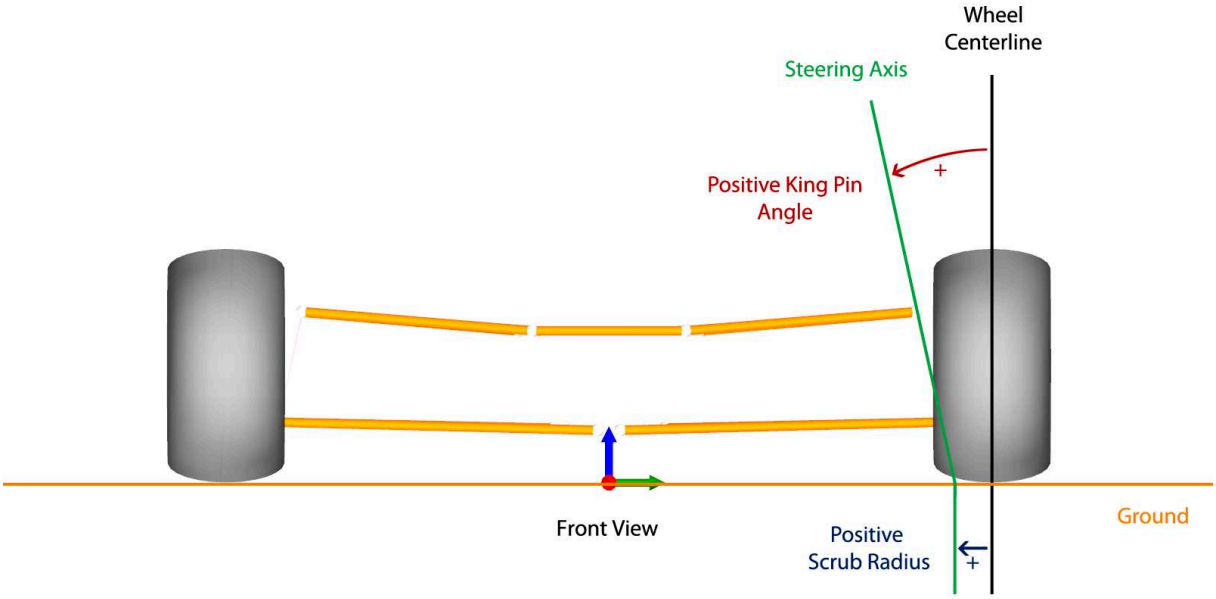
When steered, the wheel rotation in old vehicles was realized around an axis known as king pin. In modern suspension mechanisms, such as the double wishbone, the steering axis is a virtual projection between the the upper and lower wishbones' spherical joints (MILLIKEN, W. F.; MILLIKEN, D. L., 1995).

The kingpin and its inclinations can be analysed separately in front and side views. Notated by William F. Milliken and Douglas L. Milliken (1995), the angle formed between the steering axis and the tire's mid plane is known as king pin inclination on front view and is illustrated in Figure 16a. Similarly, the angle between the vertical axis plane and the steering axis is known as caster angle on side view, as shown in Figure 16b.

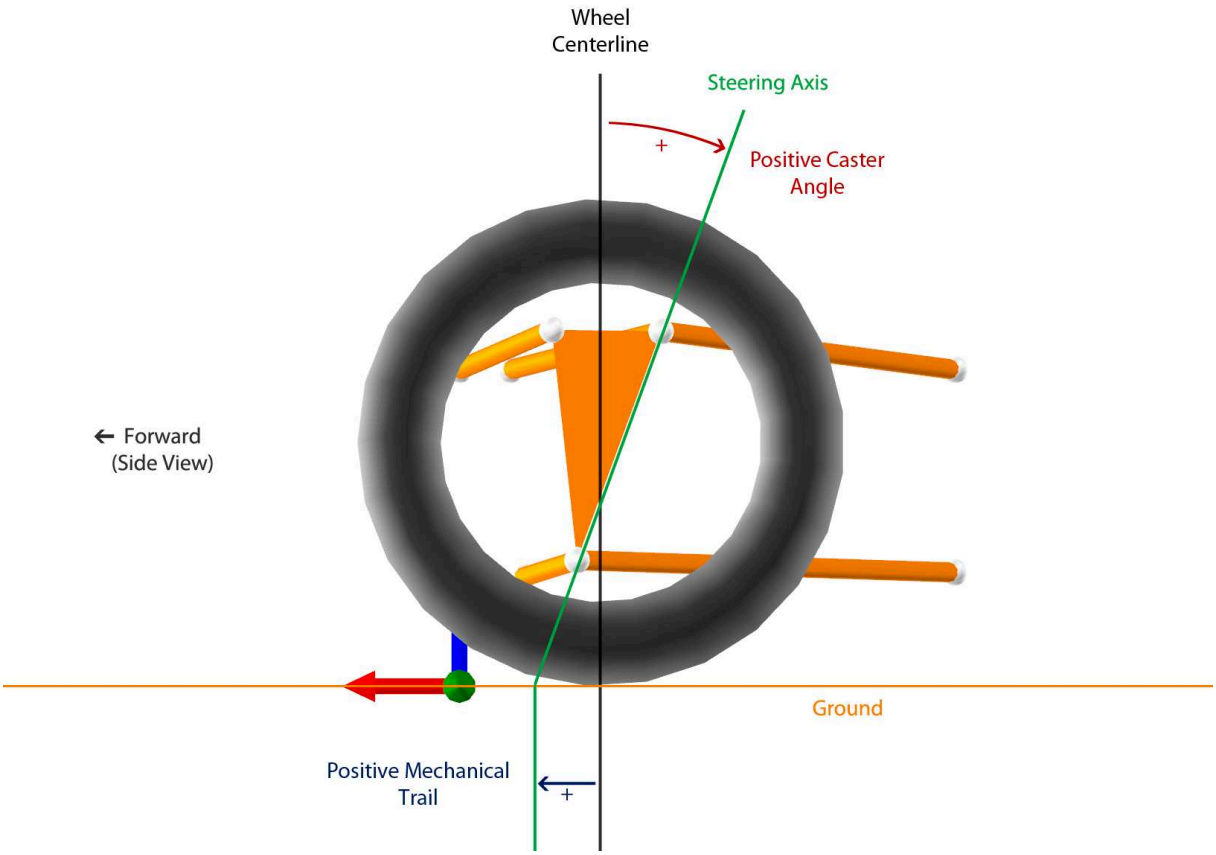
The scrub radius — also represented in Figure 16a — is obtained by extending the steering axis to the ground and measuring the distance from the point coinciding with the ground plane to the tire's mid plane. This parameter influences mostly the longitudinal loads and alignment efforts transmission, generated in the tire, to the steering system. The king pin inclination and scrub radius must maintain a compromise between

Figure 16 – King Pin geometry and reference.

(a) King pin inclination and scrub radius references.



(b) Caster angle and mechanical trail references.



Source: OptimumG (2019)

packaging and performance requirements (MILLIKEN, W. F.; MILLIKEN, D. L., 1995). The factors of fundamental consideration are:

- The higher the king pin inclination, the greater the effect of steer jacking. For example, a high king pin inclination with a caster angle equal to zero, the vehicle is lifted when steered to either sides,
- The king pin inclination influences in the camber variation by steering, also known as steer-camber,
- The greater the spindle length, the greater the driver sensibility, through moments on the steering wheel, to variations on the track,
- Longitudinal forces on the tires, originated by breaking or traction, generate moments around the steering axis and tend to change the direction of the tire force, inducing an angle of convergence proportional to the compliance of the system.

As for the caster (or mechanical to some authors) trail, shown in Figure 16b, the centroid of the tire-ground contact point follows behind the steering axis through the side view. The greater this distance, the greater the lever arm between the steering axis and the tire, consequently, undergoes greater moments induced by lateral forces generated in the tires (MILLIKEN, W. F.; MILLIKEN, D. L., 1995). Some considerations when choosing the caster angle and trail are:

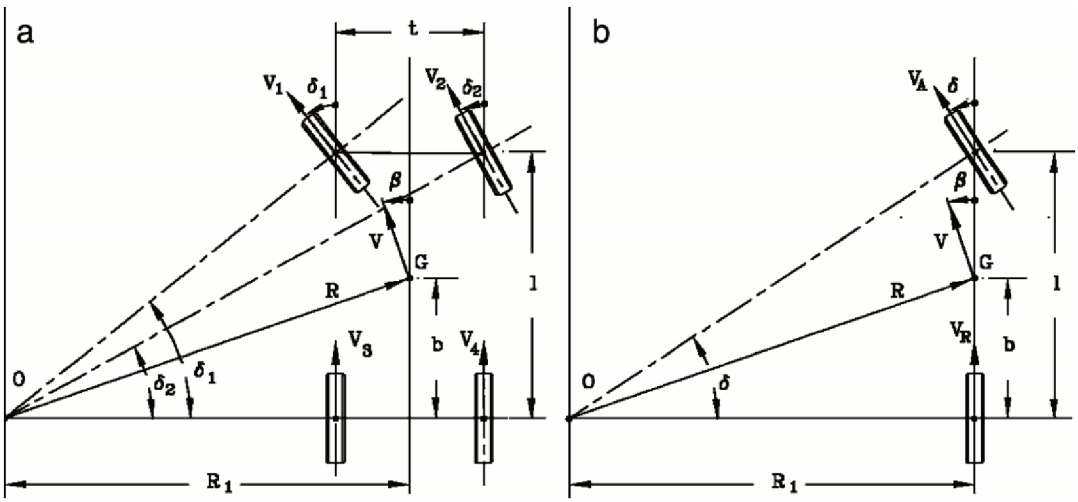
- The greater the trail, the greater the steering efforts,
- The caster angle, like the king pin inclination, causes the chassis lift or jacking. However, unlike king pin inclination, steering has an asymmetric behavior in relation to this effect, that is, while for one direction of steering the vehicle raises on one side, for another, the same side descends,
- As well as shown for king pin inclination, the caster angle has an influence on the steer-camber,
- The self-aligning torque can effectively change the mechanical trail and the driver's feeling about the feedback provided by the tires,
- The mechanical trail, when measured perpendicularly, gives a better representation of the moments generated by the tires around the steering axis.

2.5.2 Tie rods location and the Ackerman geometry

The position of the tie rod relative to the wheel set influences parameters such as the bump steer, Ackerman geometry and steering ratio. These properties will be

defined below. Due to the construction characteristics of ordinary vehicles, with four wheels and front steering, when they are steered, for them not to have relative lateral slip between the inner and outer wheel during cornering, it is necessary that the two steering wheels have differential steering angles. In this condition, the centre of the vehicle's radius of curvature is perpendicular to the rear axle, as shown in Figure 17.

Figure 17 – Kinematic steering of a four-wheel vehicle.



Source: Genta and Morello (2008)

A geometry capable of meeting the requirement to maintain a constant centre of rotation over the entire steering course is determined by Ackerman geometry. Denoted by Genta and Morello (2008), the Ackerman geometry can be attributed to a mechanism capable of steering the wheels according to Equation (13):

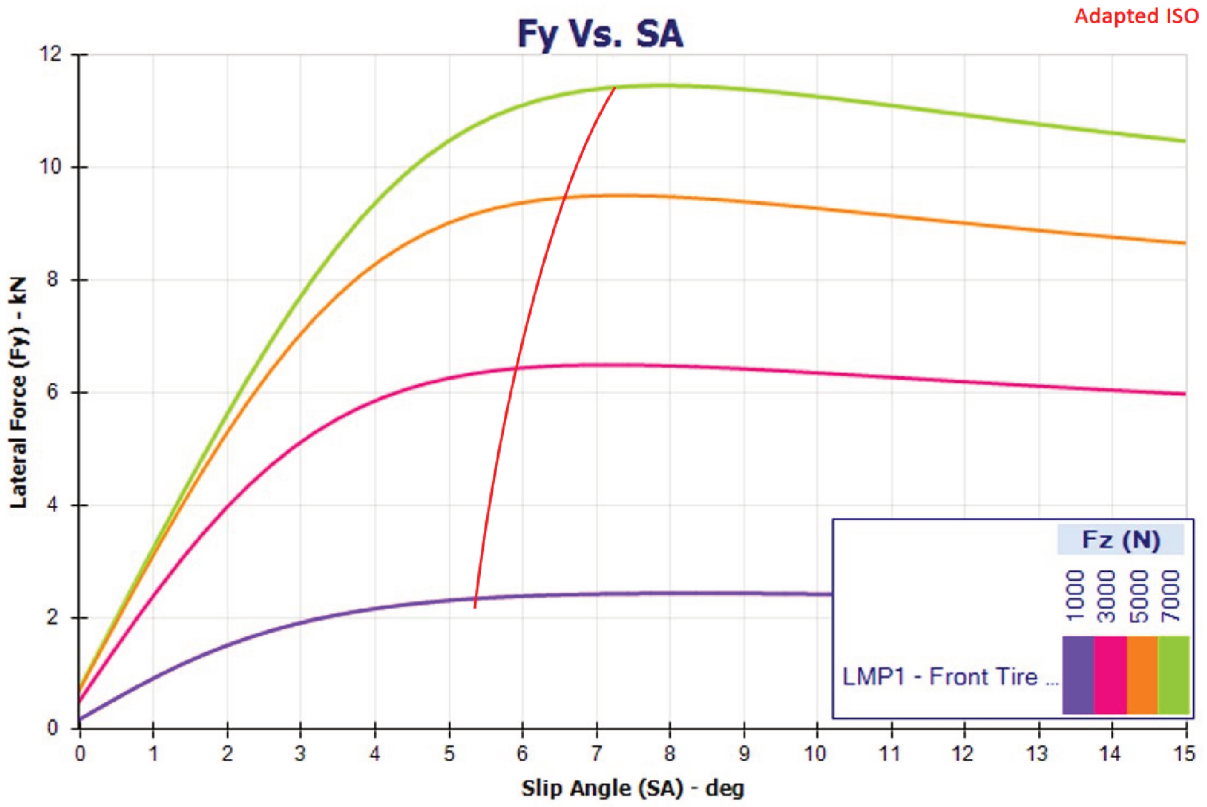
$$\cot \delta_1 - \cot \delta_2 = \frac{t_F}{L} \tag{13}$$

where δ_1 and δ_2 are the steering angle of the inner and outer wheels respectivelly. t_F is the front track width and L is the wheelbase. These variables are shown in Figure 17.

The tire performance curves usually show the displacement of the slip angle at the peak lateral force caused by the change in normal load. The effect can be detrimental specially when the vehicle is subject to high load transfers (JAZAR, 2013). This effect is illustrated in Figure 18.

As this phenomenon is naturally seen in racing prototypes, parallel and anti-Ackerman (also known as reverse Ackerman) steering geometries are commonly used in these applications. Figure 18 shows a tire that suggests the use of an anti-Ackerman geometry. According to William F. Milliken and Douglas L. Milliken (1995), it is possible to calculate the correct amount of reverse Ackerman when the tire and the loading values are known. This type of geometry the vehicle very hard to handle at low speeds, as when pitting. The definition of the system geometry is given by the relative positioning

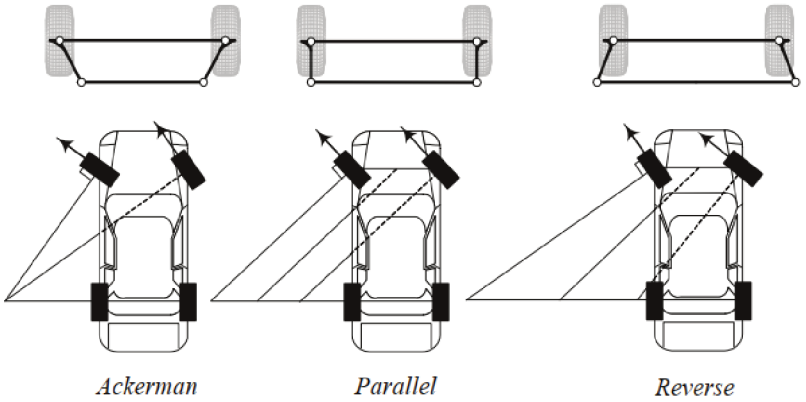
Figure 18 – Slip angle at the peak F_y shift due to vertical load shift (Adapted ISO Coordinates).



Source: OptimumG (2020)

between the steering housing, wheel and tie rods, presented by William F. Milliken and Douglas L. Milliken (1995). These placements are illustrated in Figure 19.

Figure 19 – Pro-Ackerman, parallel and anti-Ackerman steering systems and their correlations to the position of the tie rod.



Source: Jazar (2013)

Figure 19 affirms yet another compromise between the steering design and the components packaging in the chassis.

2.5.3 Steering ratio and the C-Factor

In addition to changing the Ackerman effect, the position of the steering housing and the tie rod outer joint change the steering ratio, once the movement of the mechanism is non-linear. The steering ratio is defined as the ratio between the steering wheel angle and the corresponding wheel steering angle. For parallel systems this value is the same for the inner and outer wheels, since the sides are symmetrical, however, the steering ratio value will be different between the steering wheels in Ackerman and reverse Ackerman geometry (MILLIKEN, W. F.; MILLIKEN, D. L., 1995).

Another convenient definition for the steering system, proposed by William F. Milliken and Douglas L. Milliken (1995) is the c-factor, which translates the linear amount that the rack moves for each rotation on the steering wheel and is given by equation Equation (14).

$$c_{factor} = \frac{\Delta D_{SR}}{\Delta \phi_p} \tag{14}$$

where c_{factor} is the c-factor, D_{SR} is the rack travel (given in *mm* or *in*) and ϕ_p is the pinion rotation in revolutions. The standard definition units definition for the c_{factor} is *mm*/*rev*. This number is very useful to determine the pinion size at the design stage and also serves as a parameter for defining commercial steering racks.

Pointing out the geometric parameters of the suspension and their correlations, the need for a methodology or process that assists in understanding the kinematic and dynamic behaviour of the system is striking.The problem is scaled when the objective is to optimize all the variables described and establish a commitment to all of them. Thus, the use of computational tools for analysis and optimization is justified in the development of new suspension and steering mechanisms.

2.6 NUMERICAL METHODS ON SPATIAL KINEMATICS

This section presents two approaches to solve and analyze spatial mechanisms. The first one is a method based on distance constraint between points that are contained in a single set and the other is the generic multi-body approach, following the work of Nikravesh (1988). Both methods rely on the Newton-Raphson root-finding algorithm, introduced next.

2.6.1 The Newton-Raphson algorithm

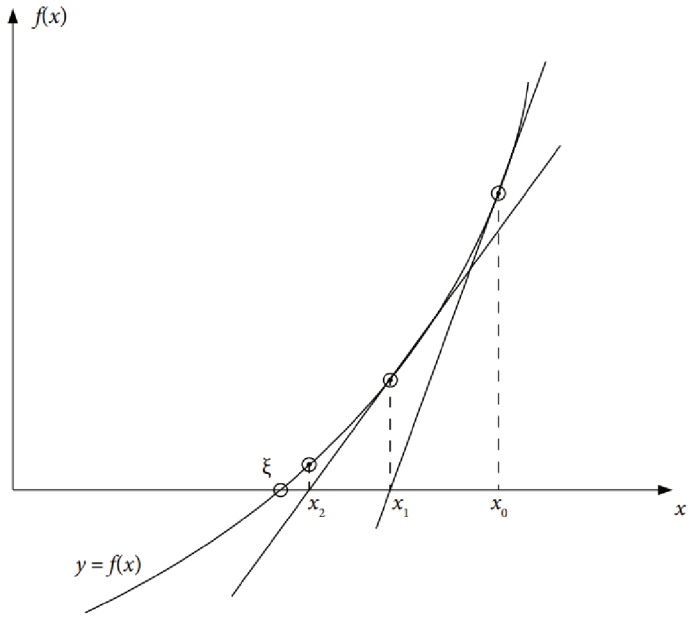
According to Boyd and Vandenberghe (2018), the Newton-Raphson algorithm is a variation from the Gauss-Newton, for when the problem consists of a system with *n* non-linear equations and *n* dependent variables. It is an iterative method and the

general form of the algorithm follows Equation (15):

$$x^{(k+1)} = x^{(k)} - \Phi_q{}^{-1} f(x^{(k)}) \tag{15}$$

where k is the k-th iteration of the solution process. The Newton-Raphson method approximates the answer of step $k+1$ by the linearization of the solution to the root on step k. The geometric representation of the algorithm for $n = 1$ is illustrated in Figure 20.

Figure 20 – Graphical representation of the Newton-Raphson method.



Source: Gupta (2019)

As shown by Figure 20, the iterative process is repeated until it reaches a termination condition: normally when the solution residuals reach a predefined tolerance. Another termination condition is set for when the algorithm reaches a maximum number of iterations, that is, the algorithm was not able to converge for a solution within m iterations. This later issue can happen even when the system has a solution, but is given a bad initial guess, as represented in Figure 21. On both plots the initial guess is $x = 1$, but due to the slight difference in the function curvature, the right one diverges.

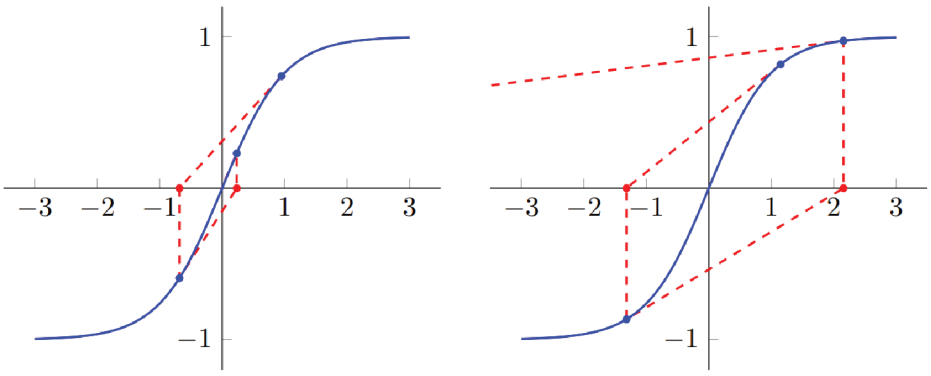
2.6.2 Point collection and distance constraints

A distance constraint between two points A and B in space can be described by Equation (16):

$$\Phi_{AB} = (x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 - d^2 = 0 \tag{16}$$

where Φ is the distance constraint between points $\mathbf{A}$ and $\mathbf{B}$ and d^2 is the constant squared distance between the points $\mathbf{A}$ and $\mathbf{B}$. The above equation must be satisfied

Figure 21 – An example of divergence due to a bad initial guess in the Newton-Raphson algorithm.



Source: Boyd and Vandenberghe (2018)

for any position of the system that contains points A and B. If the system contains more than two constraints, more equations will need to be satisfied in order to find a feasible solution. However, these equations are implicit and non-linear, making it difficult to solve them using simple methods. This set of equations can potentially present more than one feasible solution, given by its quadratic nature. In order to solve this problem efficiently, the Newton-Raphson method is chosen. The general problem to be solved is:

$$\Phi = 0 \tag{17}$$

where Φ is the vector of constraints.

The Jacobian matrix of constraints Φ_q for this problem can be obtained from the partial derivatives of the constraints. Each row i of the Jacobian represents one geometric constraint in the system, whilst each column j represents one dependent coordinate. The Jacobian matrix is then partitioned, removing the columns associated with the degrees of freedom of the system , resulting in a square, possibly invertible matrix (NIKRAVESH, 1988). Taking the partial derivatives of the above equation with respect to the x-coordinate of A:

$$\frac{\partial \Phi_{AB}}{\partial x_A} = -2(x_B - x_A) \tag{18}$$

Expanding this to the other coordinates and assuming a matrix form, yields:

$$\frac{\partial \Phi_{AB}}{\partial q} = \begin{bmatrix} -2(x_B - x_A) \\ -2(y_B - y_A) \\ -2(z_B - z_A) \\ 2(x_B - x_A) \\ 2(y_B - y_A) \\ 2(z_B - z_A) \end{bmatrix}^T \tag{19}$$

where q is the vector of generalized coordinates. Generalizing for this problem:

$$\Phi_q = \left[\frac{\partial \Phi_i}{\partial q} \right] \tag{20}$$

Equations 18 through 20 show that it is rather easy to build a Jacobian matrix for such a problem. Computationally, it takes only two floating points operations for each item in the matrix. However, to be invertible, it must be a square, non-singular matrix. Practically, the matrix becomes singular when there are redundant constraints or when the constraints form a structure that does not allow any movement on the independent coordinates. Generally, this method is stable for spatial kinematics and converges quite fast, given the quadratic convergence rate provided by the Newton-Raphson algorithm. Moreover, there are many modifications to this method that can raise the convergence rate, thus, making the algorithm faster, specially for optimization applications (MCDOUGALL; WOTHERSPOON, 2014).

This geometric approach to spatial kinematics is simple and has a straightforward implementation. However, when building a more complex system — such as a suspension mechanism — , this method is not much intuitive from the developer perspective and can get really complex when complex joints — the universal or Hooke joint for example — must be modelled. This ultimately ends up requiring a lot of testing mechanisms to ensure that the model represents reality. A feasible solution to these problems is the multi-body approach, which has a solid theoretical background and is scalable. This approach is briefly introduced next and derives from the work of Nikravesh (1988).

2.6.3 The multi-body approach to suspension kinematics

A mechanical system is defined as a collection of bodies in which some or all of the bodies can move relative to one another. Such systems can be either simple or complex. While motion of some systems can rely on a 2-dimensional plane, other systems need a 3-dimensional modelling to provide exploitable results, such as a wheel carrier-steering assembly (NIKRAVESH, 1988).

Nikravesh (1988) shows that a single body in space can be fully described with 6 independent coordinates — 3 translations and 3 rotations. Moreover, Euler Parameters can be used to avoid singularities when rotating the bodies, adding one dependent

variable to the body. The generalized vector of coordinates q_i for body i represented with Euler Parameters is denoted in Equation (21) (NIKRAVESH, 1988).

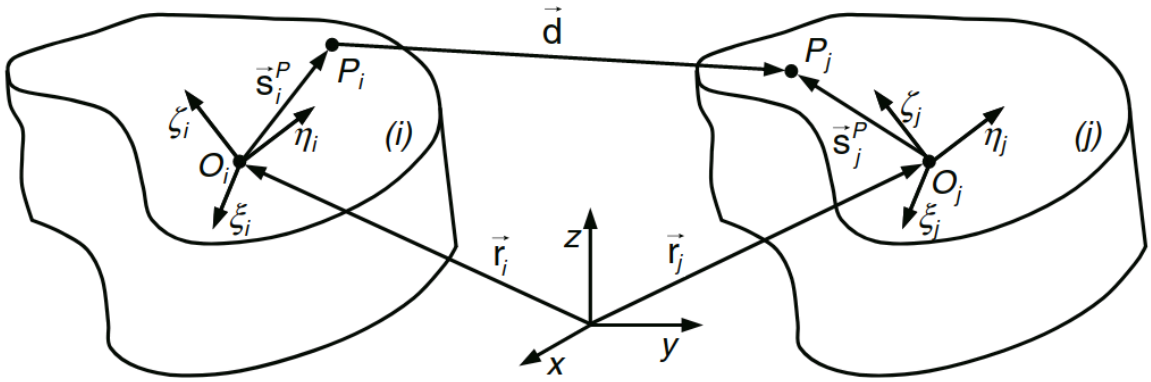
$$q_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ e_{0i} \\ e_{1i} \\ e_{2i} \\ e_{3i} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{p}_i \end{bmatrix} \tag{21}$$

where x_i, y_i and z_i are the position components of the body i and e_{0i}, e_{1i}, e_{2i} and e_{3i} are the Euler Parameters (scalar first). The vectors $\mathbf{r}_i$ and $\mathbf{p}_i$ represent the translation components and the rotation respectively, in a more compact form. The Euler constraint is given in Equation (22). This expression must be derived for each coordinate and added to the Jacobian matrix for each body that composes the system. Thus, for a system of m bodies, m Euler constraints shall be appended to the system of equations.

$$e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1 \tag{22}$$

The relative motion between two bodies will depend on the type of constraints attributed to each body, as shown in Figure 22.

Figure 22 – Vector connecting two points P_i and P_j located on different bodies.



Source: Flores (2015)

These types of constraints modelling can be then combined to build virtual joints that represent real world application joints. The two main constraints that form the joints are parallel (or normal) constraints and the spherical constraint. The parallel constraint is a constraint that specify that two vectors defined in different bodies remain perpendicular at all times and its expression is given in Equation (23), which yields one equation.

$$\Phi^{(n1,1)} = \mathbf{s}_i^T \mathbf{s}_j = 0 \tag{23}$$

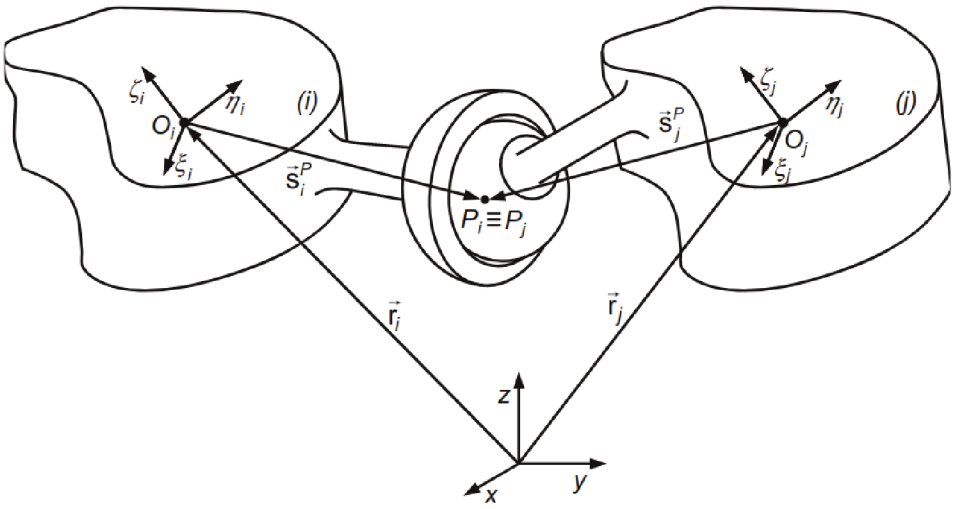
Furthermore, the spherical constraint yields three equations and its constraint vector can be written as Equation (24) shows. The physical correlation is illustrated in Figure 23.

$$\Phi^{(s,3)} \equiv \mathbf{r}_j^P - \mathbf{r}_j^P = \mathbf{r}_j + \mathbf{s}_j^P - \mathbf{r}_i - \mathbf{s}_i^P = \mathbf{0} \tag{24}$$

Both constraints shown can be combined to form different types of joint, such as the Hooke — also known as universal — , revolute and prismatic joints. They just depend on vectors defined in the joined bodies, thus, can be easily implemented and expanded.

The constraints can also be algebraically derived to build the Jacobian matrix thanks to the use of Euler Parameters, dropping the need of a numerical differentiation approach. Nikravesh (1988) provides the mathematical formulations that address this problem.

Figure 23 – Spatial spherical joint representation. Points P_i and P_j must always coincide.



Source: Flores (2015)

The process of solving the system position for a displacement in one of the independent coordinates is very similar to the geometric method. The system is displaced in a desired coordinate by a given amount, the Jacobian matrix is built and partitioned on the displaced coordinate. The Newton-Raphson algorithm is then applied to solve the system of non-linear equations until a given tolerance is achieved. The constrained system is defined by Equation (25).

$$\mathbf{\Phi} \equiv \mathbf{\Phi}(\mathbf{q}) = \mathbf{0} \tag{25}$$

where $\mathbf{q}$ denotes the vector of body-coordinates defined in Equation (21) and $\mathbf{\Phi}$ represents a function describing the kinematic constraints. Non-holonomic constraint equations in will not be discussed in this work. The iterative equation for the solution process is given in Equation (26).

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} - \mathbf{\Phi_q}^{-1}\mathbf{\Phi}(\mathbf{q}^{(k)}) \tag{26}$$

Generally, the multi-body approach is much superior and robust than the geometric one presented earlier in this work. It can also provide a better physical understanding of complex mechanisms for the developer than the purely geometric approach, as it consists of joints and bodies instead of just points and distances. The multi-body approach can also be extended with joint reaction forces, time-domain simulations, elastic components and ultimately enhanced with flexible bodies.

On the other hand, the geometric approach is much simpler, easier to implement and it is more efficient for small and simple systems, where only the kinematics are analysed. This work uses the geometric approach, as it was derived from the solver of a worldwide used suspension kinematics software, OptimumKinematics[2].

2.7 OPTIMIZATION

When the engineer can simulate the physical behaviour of a project through a validated model, the subsequent task is to determine the model's parameters that correspond to the desired performance of the problem. Normally these response characteristics are conflicting, where making one better, worsens the other, which makes necessary the search of a compromise between the different outputs (MASTINU et al., 2007).

As stated in past sections, the project of a suspension system that holds the ideal parameters cannot be described in a single objective. In single objective optimizations (SOO), the search space is well defined. When other input parameters result in conflicting outputs, a single solution is non-existent, giving place to a set of possible solutions of similar quality (ABRAHAM; GOLDBERG, 2006).

2.7.1 Design variables

In general, a mathematical system contains a set of parameters. The usual optimization goal is to find the set of parameters that best suit a given set of objectives. The optimization procedure changes the set of parameters in order to find the best

[2] OptimumG, LLC, more information available at `https://www.optimumg.com/software/optimumkinematics/`

possible solution within the feasible set of design variables, also known as design space. The design space can contain either bounded — they are limited to a set or range of values — or unbounded variables — they can assume any feasible value (MASTINU et al., 2007).

2.7.2 Multi objective optimization and the Pareto-optimal set

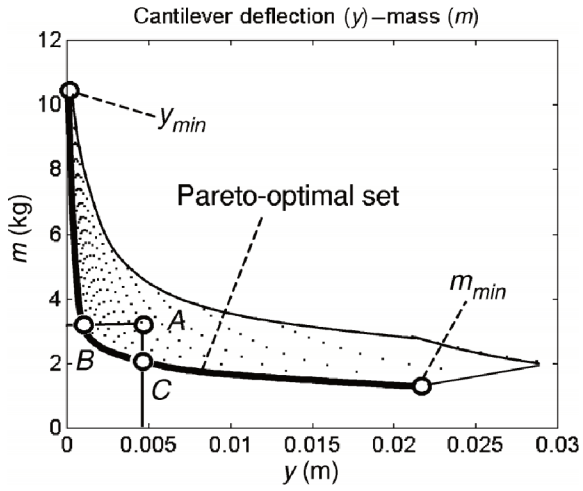
Denoted by Miettinen (2001), a generic MOO problem is of the form:

$$\begin{aligned} &\text{minimize} \quad f(\mathrm{x}) = \{f_1(\mathrm{x}), f_2(\mathrm{x}), \ldots, f_k(\mathrm{x})\} \\ &\text{subject to} \quad \mathrm{x} \in \mathrm{S} \end{aligned} \tag{27}$$

where k is the number of objective functions $f_i : \Re^n \rightarrow \Re$ to be minimized simultaneously. The vector of design variables $\mathrm{x} = (x_1, x_2, \ldots, x_n)^T$ belong to the non-empty feasible region S, which is also denominated design space. A MOO does not usually yield a single solution, but a set of equally optimal solutions, called the Pareto set.

To demonstrate the Pareto set in practical terms, Mastinu et al. (2007) illustrates the Pareto set obtained from a cantilever optimization which has the objectives of minimize mass and deflection at the same time. As shown in Figure 24, there is no single optimal solution, but a set of solutions that are equally optimal.

Figure 24 – Pareto optimal set for a cantilever beam optimization.



Source: Mastinu et al. (2007)

Figure 24 shows an optimization of a cantilever beam, where the objectives are: minimize the mass and the deflection (maximize stiffness). The points B and C represent equally optimal solutions, as they belong to the Pareto front, while A is a sub-optimal solution. m_{min} is the minimum mass with a feasible solution and y_{min} is the minimum deflection within the feasible solution set. The figure shows that for such optimization, there is not a unique solution, but a set of solutions that show that a compromise must be met.

An optimal solution on a multi-objective optimization problem is a solution that is not dominated by any other in the search space. This solution is not unique and is denominated Pareto-optimal solution. The collection of all Pareto-optimal solutions form the Pareto optimal set. A solution that is Pareto-optimal — also called non-dominated solution — means that it is a solution that is not the worst solution in any objective and yet is the best in at least one, with relation to any other solution in the set (ABRAHAM; GOLDBERG, 2006).

The methods used to determine the Pareto optimal set can be based in many optimization techniques. This work is focused on stochastic techniques, that is, it uses random search to find the best solution for a set of objectives. Metaheuristics techniques can also be used to find this solution and are introduced next.

2.7.3 Metaheuristics

Optimization problems can be separated in two basic types: discrete and continuous. The arrival of metaheuristics brought the conciliation between these two domain types, because they can be applied to every type of combinatorial problems and can be expanded to continuous problems (DRÉO et al., 2006). The metaheuristics methods have some common characteristics such as:

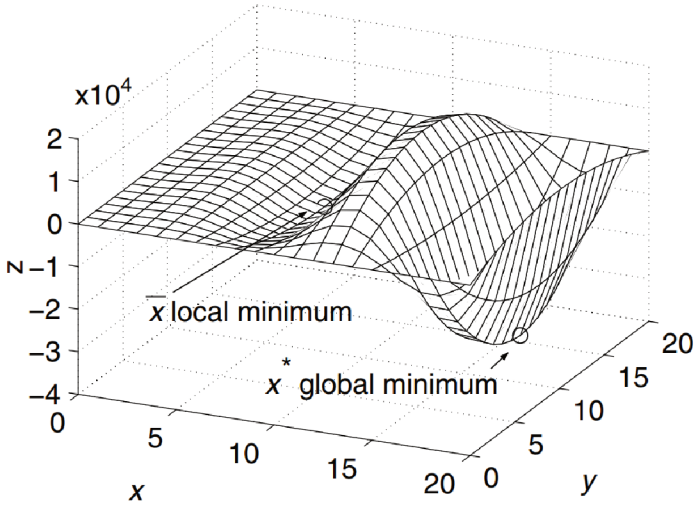
- They are, overall, stochastic, to counter the possibilities combinatory explosion.
- Generally, of discrete origin, establishing the advantage of being direct, once they are independent of objective function gradient computation.
- Their common disadvantages are the difficulty of finding the optimal method parameters and the huge computational time.

Yet mentioned by Dréo et al. (2006), the methods are not mutually excluded, since there is no way to evaluate the effectiveness of each one. Moreover, the emergent tendency is the hybridization of different methods, to use the benefits of the specific advantages of each approach, through the combination of their concepts.

Classical gradient-based iterative algorithms — such as Sequential Unconstrained Minimization Technique (SUMT) and Sequential Quadratic Programming (SQP) — can present the huge disadvantage of trapping the solutions in local minima, whilst metaheuristics allow the search mechanism to overcome these local minima, allowing the acceptance of worse solutions, in order to find a global minimum (MASTINU et al., 2007).

Some methods, like simulated annealing and tabu search, use the idea of neighbourhood search that allows the algorithm to overcome the local minima region. The local minima concept is shown through the function drawn on Figure 25.

Figure 25 – Local and global minimum of the function $z = 10(x^2 \sin x(-(y-10)^2 + 100))$, $0 < [x; y] < 20$.



Source: Mastinu et al. (2007)

According to Dréo et al. (2006), the quoted methods enable the solution to be extracted from the local minima to search other minimal results, entering on other valley of the solution gradient. The distributed metaheuristics, as evolutionary algorithms, also allows the search algorithm to leave the local minima through the mutation operators.

2.7.4 Genetic algorithm

Genetic algorithms (GA) belong to the evolutionary algorithms group and the solution search is done by mutation and recombination of configurations. As the name itself suggests, GA mimics the evolution process and use the ideas of natural selection — the fittest individuals are more likely to survive and reproduce — and genetics — recombination and mutation — to control the evolution process (ZÄPFEL et al., 2010).

Denoted by Zäpfel et al. (2010), a population, on genetic algorithms' context is a set of solutions, generally called individuals. Every individual is a unique solution, formed by genes that determine their characteristics. The values that constitute the genes are denominated alleles.

As defined by Dréo et al. (2006), a set of N points in a search space, chosen at random, constitute the initial population. Each individual from the population has a fitness value that measures how well adapted this individual is to a given objective function. An evolutionary algorithm consists of a gradual evolution of the population's composition, of a constant size, through successive generations.

Along generations, the objective is to integrally improve the fitness of the individuals in the population. The result is obtained through the simulation of the main mechanisms that the living beings are subjected, the Darwin theory, selection and

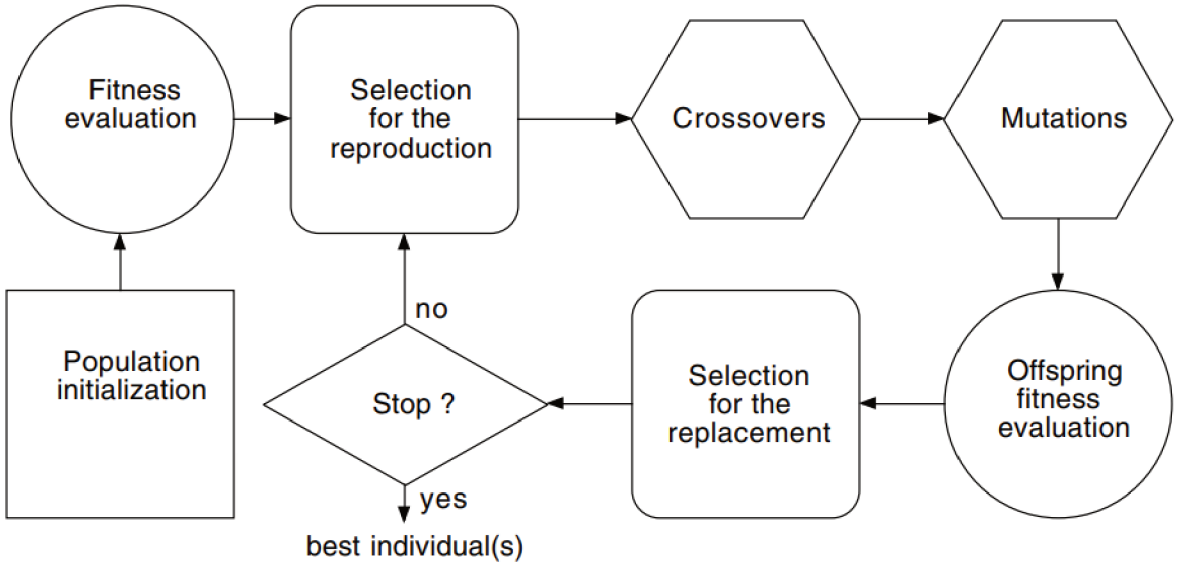
reproduction.

The selection supports the reproduction and the survival of the fittest individuals. The reproduction allows the mixing and recombination of each individual's characteristics that constitute the offspring with new potentials.

Individuals are chosen on initial or previous populations by a selection operator. This operator compares the adaptation of the individuals in a generation with relation to the global objective. The selected individuals are recombined by a crossover operator, generating a new offspring. The new generation can be mutated, that is, a random perturbation is applied to the solutions (ZÄPFEL et al., 2010).

The new descendants are then evaluated with relation to the objective functions, selected and reproduced using the same techniques described above, substituting the replacement set of individuals of the past generations. A flowchart in Figure 26 shows the principles of a genetic algorithm proposed by Dréo et al. (2006).

Figure 26 – Generic Evolutionary Algorithm flowchart.



Source: Dréo et al. (2006)

Because they deal with a population of instances of solutions, the evolutionary algorithms are particularly dedicated to find a set of many solutions, when a compromise of objective functions must be reached, satisfying many global optima. This way, they provide a sample of solutions that involve many objectives, possibly conflicting (DRÉO et al., 2006).

Genetic algorithms are very useful specially when the project variables assume discrete values. Unfortunately, they are not so easy to implement, when compared to other methods, when complex optimization problems are being dealt with. In fact, genetic algorithms require an exact definition of a parameter that influences the efficiency of the search (MASTINU et al., 2007).

2.7.4.1 Fitness function

Also known as evaluation function, the fitness function associates one or multiple fitness (depending if the problem is single or multi objective) values to each individual in order to determine the number of times it will be selected, be it for reproduction or for replacement. The quality of such function can greatly improve the efficiency of a genetic algorithm (DRÉO et al., 2006).

Depending on the underlying implementation of the genetic algorithm and the problem that is being solved, the fitness function may or may not correspond to an objective function. Still, it is at least some kind of quality monitoring tool, once solutions with better fitness values usually are preferred in the selection steps (ZÄPFEL et al., 2010).

2.7.4.2 Selection operators

From one generation to another, in GA, the population is clustered in two distinct groups: the reproduction group and the replacement group. One is not necessarily the complement of the other. The reproduction group contains the individuals that will be reproduced using the crossover and mutation techniques described later in this chapter.

The replacement group contains the individuals that will be removed from the current population and will be replaced by the offspring generated by the reproduction group. The groups are clustered using selection operators, which use their fitness values as a parameter to select the individuals. The most used selection operators are introduced next.

2.7.4.2.1 Truncation selection

The truncation selection is probably the simplest and most common selection operator. It selects the *n* best individuals in the population, being n a parameter chosen by the user. It can be easily implemented either for reproduction as for replacement. This method can lead to fast convergence but may trap the solution in a local minimum (DRÉO et al., 2006).

2.7.4.2.2 Deterministic Tournament

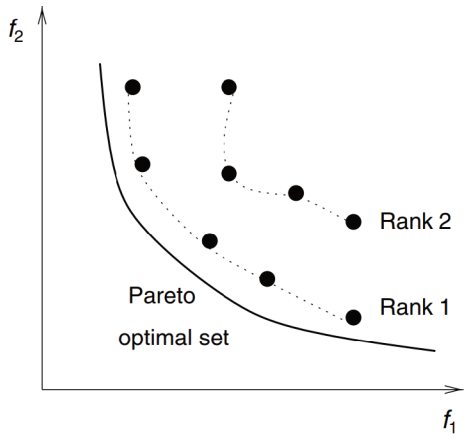
The deterministic tournament consists in choosing k individuals at random for a tournament and choose the best one among them. In fact, there will be as many tournaments as selected individuals when using deterministic tournament. As stated by Dréo et al. (2006), this method can be applied for either reproduction and replacement.

2.7.4.2.3 Ranked selection

The ranked selection is a method that ranks individuals according to their fitness dominance, that is, sorting several Pareto sets for every generation. This method should only be used on multi-objective optimization cases. A dominated individual is an individual that has worse fitness values in all objective functions with relation to another individual. A non-dominated individual is an individual that is not dominated by any other individual in the set, which can be considered a Pareto solution for that specific set of individuals.

If an individual has a better fitness value in at least one, but not in every, objective function, than another individual and both are non-dominated, they belong to the same rank. The rank 1 individuals are the Pareto front for the current population. This method can be used to select individuals for reproduction and replacement. The ranked selection is shown in Figure 27 for a simple minimization of two arbitrary functions f_1 and f_2. The plot shows two total ranks for the population.

Figure 27 – Examples of a ranked selection of a small population for two functions.



Source: Mastinu et al. (2007)

2.7.4.2.4 Steady state replacement

The steady state selection for replacement is the complement of truncation selection, which selects the n worst individuals to be replaced, with n being a small number, as mentioned by Dréo et al. (2006), one or two individuals. This strategy is useful when the representation of the solution is distributed on several individuals, that is, when the design variables are really disperse through the population, but the population has a low overall fitness standard deviation.

2.7.4.2.5 Generational replacement

The generational replacement is the simplest replacement selection. The whole population gets replaced, such as that the offspring contains only child individuals. According to Dréo et al. (2006), the canonical GA uses a generational replacement.

2.7.4.2.6 Elitist replacement

An elitist strategy consists in keeping at least the best individual through generation g to generation $g+1$. There are various elitist strategies in GA, but, in accordance with Dréo et al. (2006), the current alternatives are those that keep the parents from generation g in generation $g+1$.

2.7.4.3 Crossover methods

Generally, a crossover operator uses two parents to generate one or two offspring. Of stochastic nature, they can provide different results when applied to the same set of parents. Since the evolutionary algorithms (EA) are not subjected to biological constraints, more then two parents can be matted to generate a new individual (DRÉO et al., 2006).

Due to the continuous nature of the search space, a purely uniform crossover would converge the solution too quickly and would be extremely dependent on population initialization. Uniform crossover is effective only when high mutation rates are used as well. The BLX-α crossover (DRÉO et al., 2006) variations are more effective when working with continuous problems. Both methods are explained in detail next.

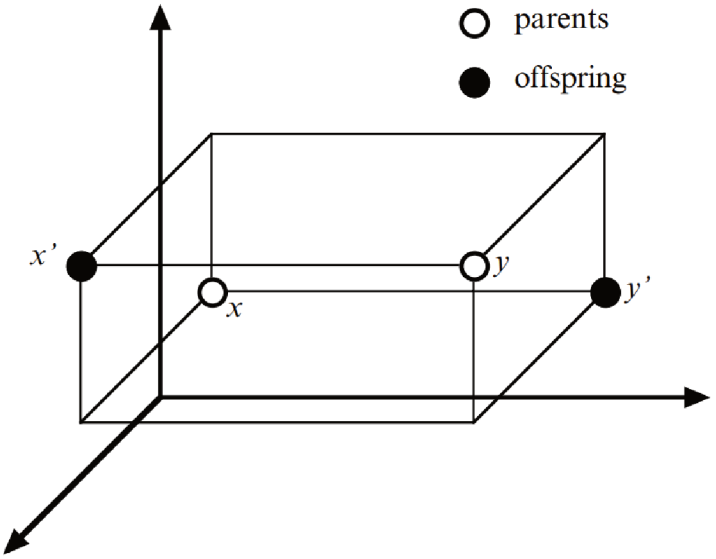
2.7.4.3.1 Uniform crossover

The uniform crossover is the simplest recombination method. It exchanges the design variables values from one individual with another to generate a third. The main problem with this crossover is, with a low mutation rate, the solution converges quickly, and the individuals become the same in a fast pace (DRÉO et al., 2006). Figure 28 shows an individual's genome of genes x' and y' resulting from an uniform crossover from a parents with genes x and y.

2.7.4.3.2 Linear BLX-α crossover

Also known as arithmetic crossover and intermediary recombination, the linear-BLX-α crossover creates a new individual along a line that connects the parent's design variables. An alpha constant is added to the operation in order to exploit the neighbourhood (DRÉO et al., 2006). It is geometrically easy to represent and implement, which

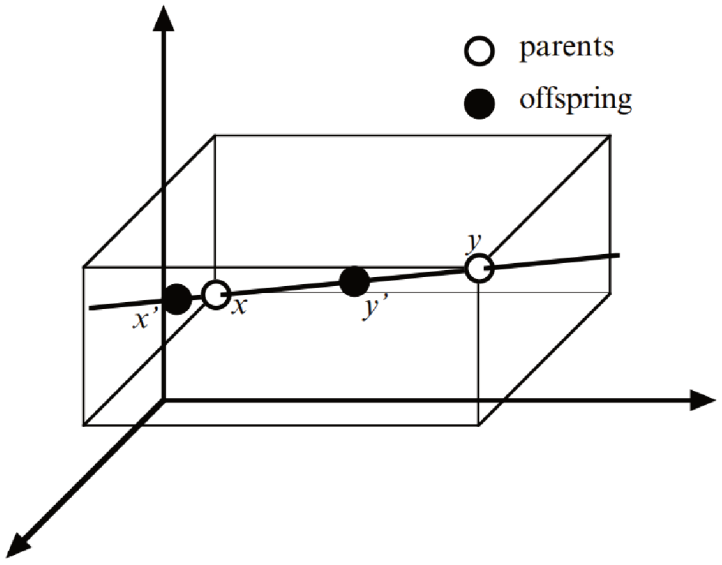
Figure 28 – Uniform crossover representation.



Source: Dréo et al. (2006)

makes a good choice for this problem. A graphical representation of this operator is shown in Figure 29.

Figure 29 – Graphical representation of the Linear-BLX-α crossover



Source: Dréo et al. (2006)

Let $\vec{p_i}$ and $\vec{p_j}$ be points drawn in the design space for individuals *i* and *j* respectively. The vector that describes a movement in space from $\vec{p_i}$ ti $\vec{p_j}$ is $\vec{v} = \vec{p_j} - \vec{p_i}$ and the midpoint is $\vec{p} = \frac{\vec{p_i}+\vec{p_j}}{2}$. Let *u* be a random scalar drawn in the interval [–1, 1] and α_{BLX} a scalar defined by the user. The offspring's allele will be:

$$\vec{p_o} = \vec{\bar{p}} + \frac{\vec{v}}{2}\alpha_{BLX}u \tag{28}$$

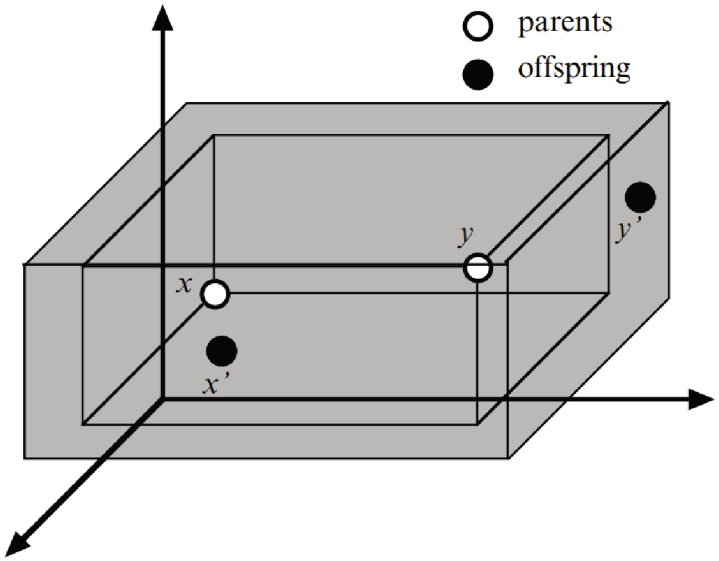
where $\vec{p_o}$ is the allele of a specific gene of the offspring.

This way, α_{BLX} is a parameter that can allow or forbid the individual to exceed the parent's boundaries, in the same line segment that binds them both. The higher the value of alpha, more likely a new individual will be taken out of its parent's range and slower will be the convergence. If alpha is zero, the new individual will always inherit the midpoint of the two coordinates, converging fast.

2.7.4.3.3 Voluminal BLX-α crossover

The voluminal BLX-α crossover method is much alike the linear version previously introduced. It differs from the linear approach by choosing new offspring uniformly inside a hyper-rectangle with sides parallel to the coordinate axes. The parents and the coefficient α_{BLX} define a one of its longest diagonals, as shown in Figure 30.

Figure 30 – Graphical representation of the Voluminal-BLX-α crossover



Source: Dréo et al. (2006)

2.7.4.4 Mutation methods

Mutation consists of adding a small numerical value to each component of the individual, according to a zero-average distribution, with a variance possibility increasing with time. According to Dréo et al. (2006), this way, the operator leaves the population's centroid unchanged. The mutation operator can also be used to keep the population diversity at a desired value.

2.7.4.4.1 Uniform mutation

A uniform mutation adds to an individual a random value of uniform probability, in a hyper-cube. This method can trap the solution to local minima, if the peak on the objective function is broader then the hyper-cube edge lengths.

2.7.4.4.2 Gaussian mutation

Mentioned by Dréo et al. (2006), the Gaussian mutation is the most widely used for GA continuous problems solving. It adds to an individual a Gaussian random variable, with zero-average and a user-defined standard deviation, given by the Gaussian probability distribution. Other Gaussian methods can also be implemented, but since they are an extension to this one, they are not introduced on this work.

2.7.4.5 Selection pressure

During the evolution of the population, it is important to monitor some statistics to have a notion of what is happening during the optimization. The EMOO is a rich source of statistical data, such as population mean fitness, fittest individual fitness, selection pressure and more.

The selection pressure is a great indicator of the population's diversity, as defined by Equation (29). It measures how close the population is to the best individual.

$$p_s = \frac{g_0}{\bar{g}} \tag{29}$$

where p_s is the value of selection pressure, g_0 is the best — lowest in value — overall fitness and $\bar{g}$ is the average fitness of all individuals in the population.

When the selection pressure comes close to 1, that means that all individuals are very alike or even have the same configuration. The individuals having the best fitness values are reproduced more often than the others. If the variation operators, such as mutation, are inhibited, the best individual shall reproduce faster than the others until the whole population is equal or nearly equal to it. This is caused by high selection pressures. With a high selection pressure, there is a great risk of premature convergence (DRÉO et al., 2006).

2.8 STATE OF THE ART

In literature, different approaches have been employed for this problem: mathematical programming and metaheuristic methods. In general, mathematical programming methods offer a guarantee of optimality while metaheuristic methods do not. Sancibrian et al. (2010) introduces a synthesis method based on gradient determination using exact differentiation to search for optimal solution. The proposed method

formulates the objective function as a measurement of the synthesis error between the generated and desired mechanisms. Seven functional parameters are considered in the optimization. Raghavan (2004) proposes an algorithm for the synthesis of the tie-rod joints location in a suspension, in order to achieve linear changes in toe-angle during jounce and rebound. On the other hand, heuristic methods can handle large and complex optimization problems while mathematical programming methods can face convergence problems when the size of the optimization problem increases (RODRIGUEZ et al., 2018).

Arikere et al. (2010) present a MOO approach for the design of a double wishbone suspension. Three multi-objective optimization methods are proposed: the weighted-sum method, the min-max method and Multi-Objective Genetic Algorithm (MOGA). Pareto-optimal solutions to the mechanism synthesis problem are generated. Thus, the designer can choose from the set of solutions, considering the minimization of two functional parameters: camber and toe. Cheng and Lin (2014) apply robust optimization based on particle swarm optimization for the double wishbone suspension design. Multi-objective approach is employed, and a set of Pareto solutions is proposed. Afkar et al. (2012) propose a GA for the optimization of ride comfort, handling and stability of vehicle. Moreover, sensitivity analysis and variations of geometric parameters of suspension system resulted from bump and vehicle roll inputs are presented for the optimal case.

3 METHODOLOGY

This chapter approaches the conception and development process of the kinematic analysis and optimization software. It starts with the problem highlight, elucidates the basic requirements of such application and presents the proposed structure. It approaches the parametric study that was run to study the GA parameters and ends with the presentation of a case study that is used in this work to show the potential of this type of optimization.

3.1 THE PROBLEM IN SUSPENSION KINEMATICS DESIGN

When designing a new suspension system, there are a set of requirements imposed to the design of such system. These requirements can range from packaging and manufacturing tolerance to load path control and kinematic behavior. This list can grow even bigger if the project belongs to a racing prototype, where tire behavior, driver characteristics and regulations come to the picture.

The Double A-Arm suspension is composed by 8 pickup points for a quarter of a vehicle — not including wheels, actuation or anti-roll systems — which yields one degree of freedom: the vertical wheel movement. Another degree of freedom can be obtained if the system has a steering mechanism. Expanding this quarter vehicle to the four corners — considering asymmetric systems — the solution sums up 32 points, where each point has three Cartesian coordinates — X, Y and Z — totaling 96 variables in the system configuration. Although it is shown in section 2.3 that specific points have more or less influence on different characteristics of the suspension system, the design process of a new suspension system or the improvement of an existing one can become not only a challenging but also a time- and resource-consuming task.

In order to demonstrate the struggle that the suspension designer goes through, a simple comparative study was run in OptimumKinematics. A generic Double A-Arm suspension was taken as the subject of this example. The outer ball joint Z coordinate of the upper A-Arm of the front suspension is moved from 593.910 mm to 650.000 mm. These changes, illustrated in Figure 31, are arbitrary, since the objective of this study is just to show the complexity of the design process. The system is then submitted to a heave motion ranging from -50 mm (bump) to +50 mm (rebound) from the initial position.

As Figure 31 shows, only one parameters of the 24 available coordinates — considering symmetry — that constitute the Double A-Arm suspension was changed. However, three different outputs of the heave motion simulation for this suspension are shown in Figure 32: camber angle, toe angle and roll center height. As mentioned in section 2.3, camber and toe have a direct influence in tire behavior and the vehicle performance, while the kinematic roll center has an indirect effect in the dynamics of

Figure 31 – Suspension pickup points change.

(a) Baseline front suspension. (b) Modified front suspension.

Source: Author (2021)

the car. In this case, if the designer was aiming to keep the baseline roll center value and increase camber gain for both wheels, while eliminating bump steer, only one target would be achieved.

This means that regarding the change in only one specific kinematic parameter curve — in this example, increase the camber variation in heave — while keeping all the other kinematic parameter curves the same, the suspension designer would probably need to change many other — if not all — pickup points. In addition, it is too difficult for a human to find the best compromise between a set of kinematic objectives within the time demands of modern suspension projects. Therefore an optimization procedure would not only reduce the development time, but also increase the quality of the final suspension.

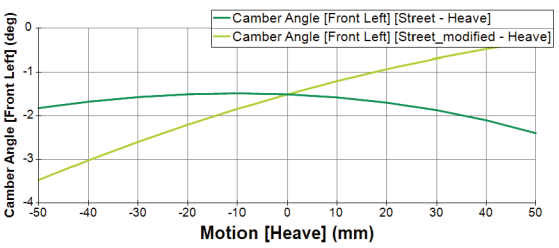
This example also shows how complex the design of a double wishbone system can be, once that a change in one coordinate of the system can be propagated to many different kinematic curves. Additionally, it shows the multi-objective nature of this type of problem.

3.2 THE STARTING POINT: OPTIMUMKINEMATICS

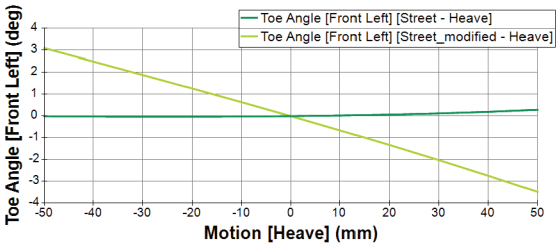
Previously mentioned on chapter 2, OptimumKinematics is a powerful tool that allows suspension designers analyse suspension systems. It has a wide range of functionalities that allow the user to compare different types of suspension, providing a good overview and a broad understanding of the overall objectives.

The software is used world-wide by many different customers, ranging from students to competition and passenger cars manufacturers. This work uses OptimumKinematics' core solver architecture as its baseline to build the models and analyze them over a range of motions.

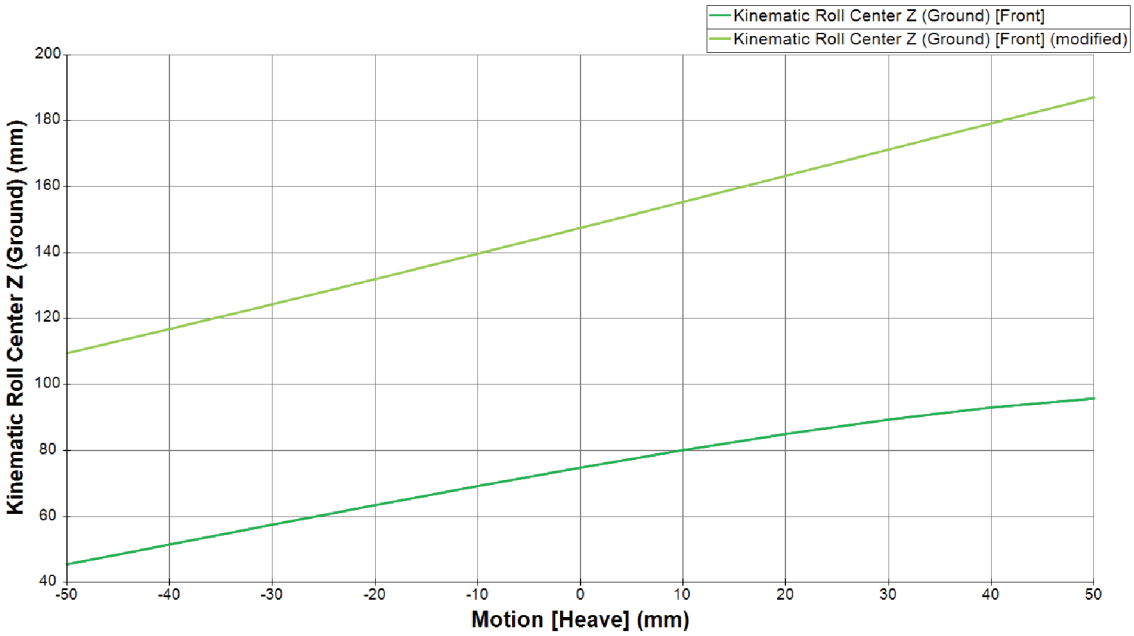
Figure 32 – Suspension output change for a heave motion (from -50mm to +50mm).



(a) Camber variation (Front Left).



(b) Toe variation (Front Left).



(c) Front roll center height change comparison.

Source: Author (2021)

However, an improvement to the software structure is proposed in this work, aiming calculation speed and portability. The software — including its kernel — is currently written in VB.NET, part of the Microsoft .NET Framework[1], which is not the fastest available language for numerical computation and, in addition, does not support cross-platform compilation.

3.3 PROGRAMMING LANGUAGES

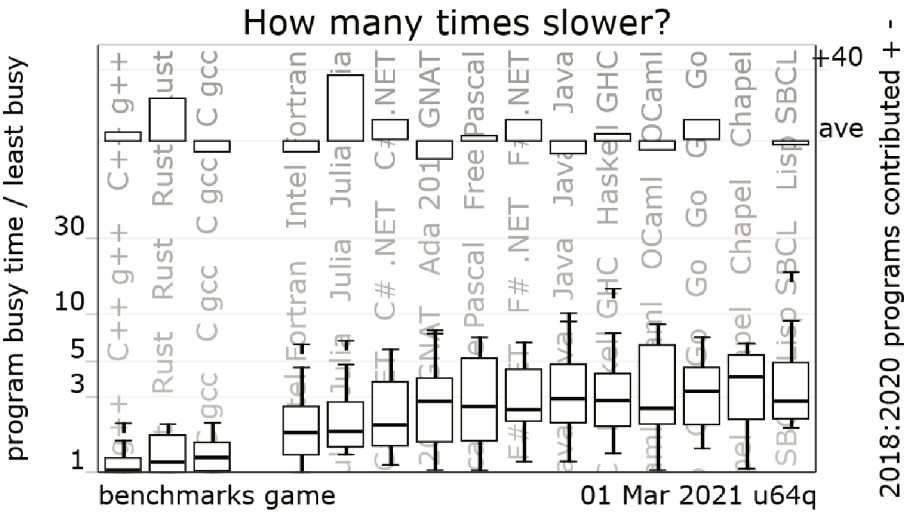
For a program to run, its source text — human readable — has to be processed by a compiler, which are translated into machine-language, not readable by humans. An executable program is created for a specific hardware or system combination, thus it is not directly portable from a Windows machine to a Mac or Linux machine. Still,

[1] More information available at: `https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework`. Accessed on March 25th, 2021.

depending on the language, the source code can be ported to other systems and therefore can be compiled for different systems. This is what makes a programming language portable (STROUSTRUP, 2018).

For this reason, a benchmark of programming languages that would find common grounds between the needs described above was done. Debian (2021) shows that in most algorithms — ranging from floating point operations to string processing — the C++ language stands out at the top of the fastest languages for computation speed, shown in Figure 33. The candles represent the deviation between the different algorithms that are being used to benchmark each programming language. Besides the computational speed, programs written in C++ can be compiled for many different platforms and Operating System (OS). Ultimately, ISO (2020) provides some standard definitions for the C++ language and is updated every three years.

Figure 33 – Algorithm speed comparison between different programming languages.



Source: Debian (2021)

Figure 33 shows the time taken for set of algorithms run in each language, represented by the candles. The bars on the top shows the contribution of the community with new algorithms. As it can be noted, C++ is the language that takes the less time, with a slight difference between Rust and C. C++ is not only faster but supports object-oriented programming, thus, it is the language of choice.

3.4 PROGRAM STRUCTURE

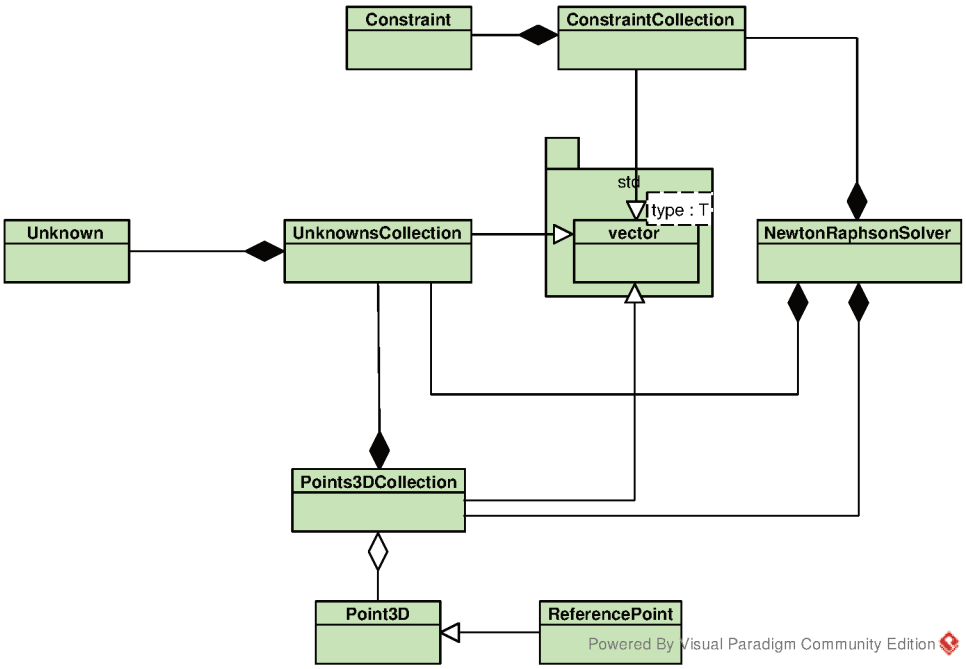
Nowadays, software engineering has become an effective engineering discipline. As a result of the constantly increasing in software and engineering complexity, approaching problems using an object-oriented perspective is more effective and communicative than just the common procedural approach. Moreover, object-oriented

programming complements the task of generalizing the problem into more abstract concepts. The Unified Modelling Language (UML) is a communication and mapping tool based on visualization that allows the software development team share the same concepts and transfer knowledge between themselves more efficiently (RUMPE, 2016).

OptimumKinematics' solver was previously written in VB.NET and the translation to C++ was proposed for this work. The software structure shown in this section is an improvement proposal over the previously existent implementation of OptimumKinematics.

The UML Class diagram was used to aid in the identification of the key classes already present in the former implementation of OptimumKinematics' geometry solver. The program is divided in classes, in a way that, the higher the layer, the more dependent it is on the lower classes. The lowest classes are the geometric entities, such as Point3D, whilst the top layers are the ones whose functionalities are closer to the end user. Figure 34 shows a class diagram of the base solver, which has its functionality explained in subsection 2.6.2.

Figure 34 – Base solver classes.

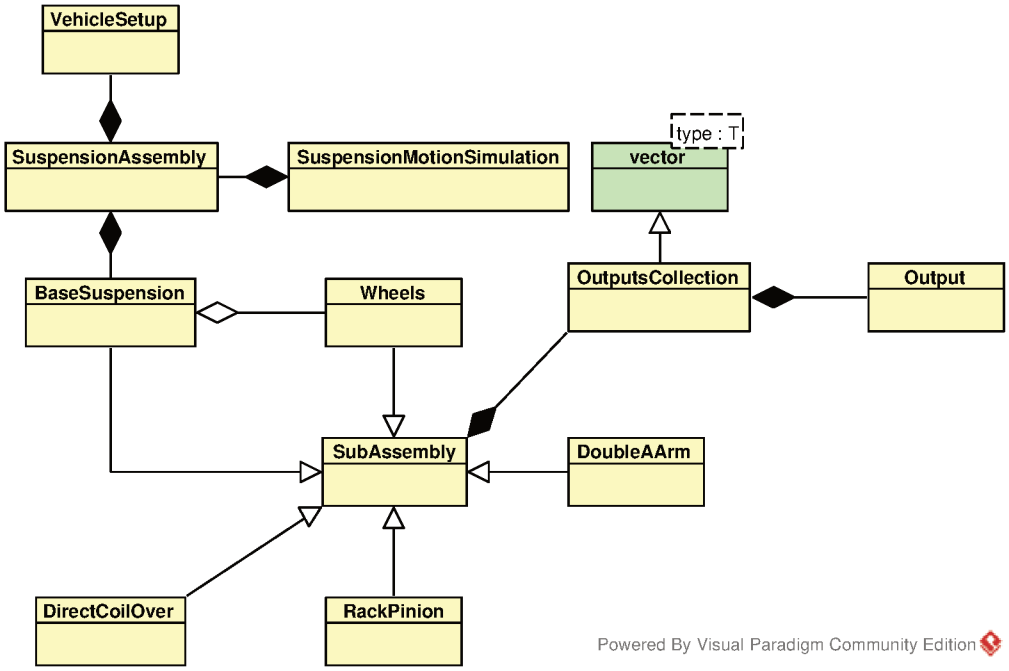


Source: Author (2021)

Furthermore, Figure 35 shows the classes that hold information regarding the vehicle assembly. A *SuspensionAssembly* is constituted of two independent suspension systems, the front and rear suspensions, of type *BaseSuspension*.

Lastly, the SuspensionAssembly class is combined with the *Motion* into a *SuspensionMotionSimulation* class, shown in Figure 36. The Motion class holds information

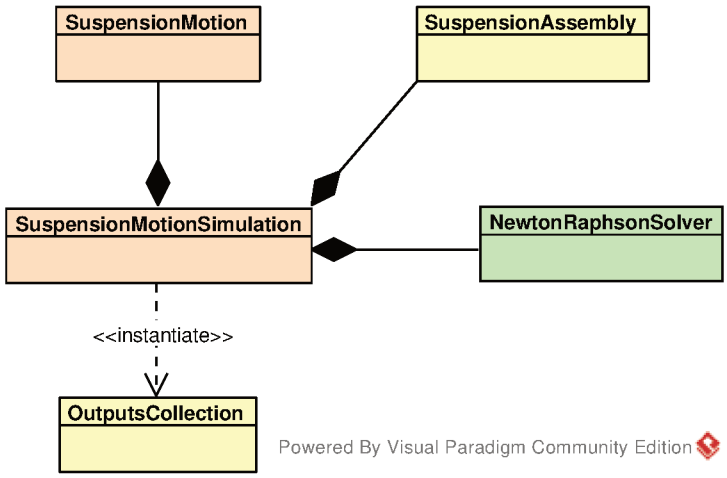
Figure 35 – Vehicle assembly classes.



Source: Author (2021)

regarding the motion that the system will be subjected to through the simulation, while the SuspensionAssembly contains all the characteristics of the system.

Figure 36 – Suspension simulation classes.



Source: Author (2021)

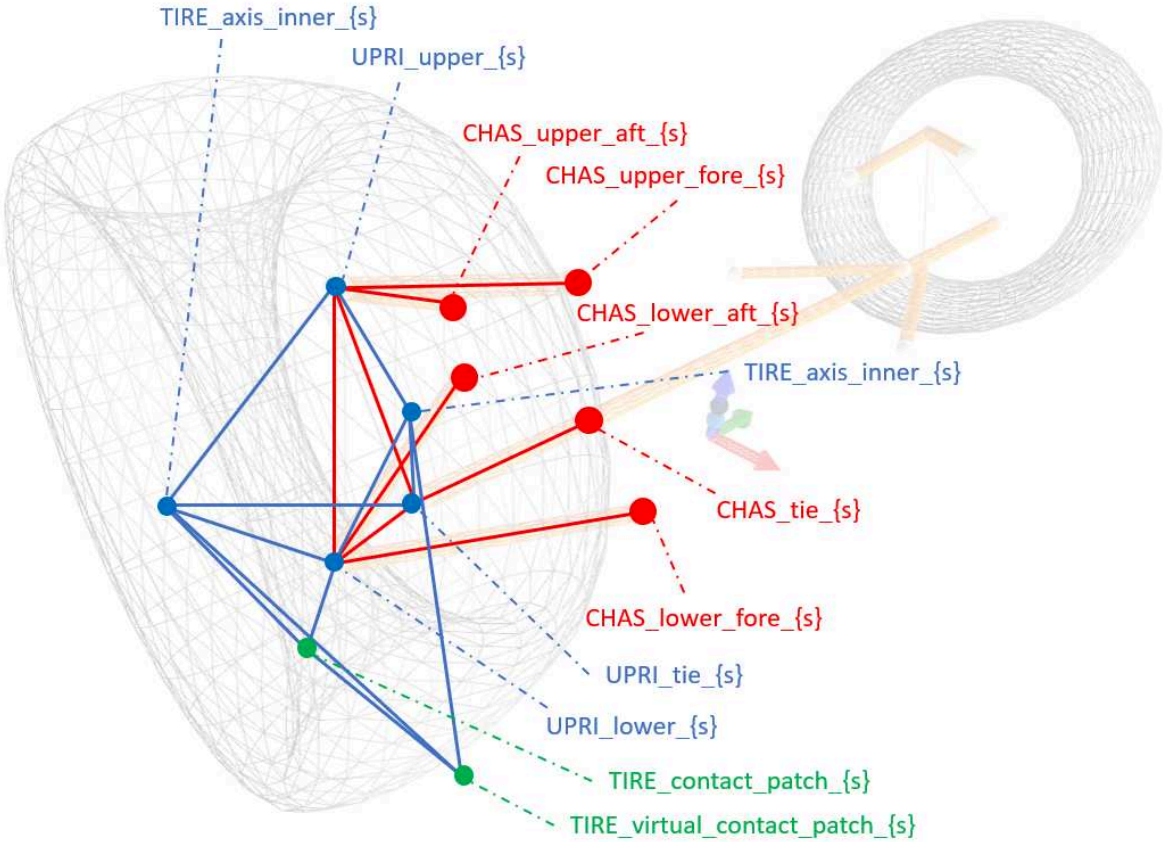
A proper class diagram allows the development team to identify performance bottlenecks, scalability possibilities and other significant improvement opportunities, such as memory management. Additionally, the class diagram allowed this work to happen with a smoother flow, avoiding unidentified problems during the implementation phase.

3.5 KINEMATIC MODEL

The suspension system is modeled via a kinematic approach: a set of points are subject to a set of constraints. The number of dependent variables must be the same as the number of constraints of the system. To this extent, the points and constraints of the model are shown in Figure 37 respectively. The points are represented by the circles and the constraints by the lines.

It is important to note that the points which are bound to the chassis — shown in red in Figure 37 — have their coordinates imposed through the simulation process, that is, they can't be moved by the solver. Similarly the tire contact patch and virtual ground points — shown in green — have their Z coordinate imposed, so they can only move in the X and Y directions. Finally, the blue points are the ones that have their degrees of freedom imposed exclusively by the constraints.

Figure 37 – Double A-Arm kinematic points identification.



Source: Author (2021)

Each point has a name — or key — as shown in Figure 37. The *s* between the curly braces represent the side of the car that holds the point. The expression can be substituted by *l* or *r*, that represent the left and the right side of the car, respectively.

The end user does not need to determine all the points shown in Figure 37, once the wheel points are calculated using other more commonly used parameters, such as track width, tire diameter, rim width and the lateral, longitudinal and vertical offsets. The reference distances — that must be maintained throughout the whole simulation process — are calculated based on the points coordinates at the beginning of the simulation process.

The kinematic simulation consists of moving a set of points that are bound to a virtual body — i.e. the chassis or the contact patch points — in a given direction and finding the new coordinates that satisfies the set of distance constraints. The heave motion moves the inboard points in the Z direction.

Both pitch and roll motions consist in rotating the chassis points around a given axis, defined by a line that connects two points. The pitch and roll axes used in the simulation can be set fixed by the user or can be determined using the suspension system's current roll and pitch centers. If the later approach is chosen, the user must be aware that the delta steps in the simulation will directly affect the behavior of the simulation in the later steps. This method can also prevent the simulation to complete, since the kinematic centers can, theoretically, achieve infinite values.

Lastly, the steering motion displaces the tie rod point bounded to the chassis. The amount in X, Y, and Z coordinates depend on the steering system type. This work considers only the rack and pinion steering system, which yields a steering motion that moves the tie rod inner points in the direction of the line that connects both left and right tie rod inboard points. The scalar value of the displacement is given by Equation (14).

The system shown in Figure 37 has 19 constraints and 19 variables — or degrees of freedom — which yield a 19×19 Jacobian matrix which needs to be inverted in each step of the Newton-Raphson algorithm. If the opposite side is considered, the matrix size is enlarged to 38×38. If reference points, actuation systems and anti-roll bars are added, the system's size increases quite rapidly and will add computational cost to the process.

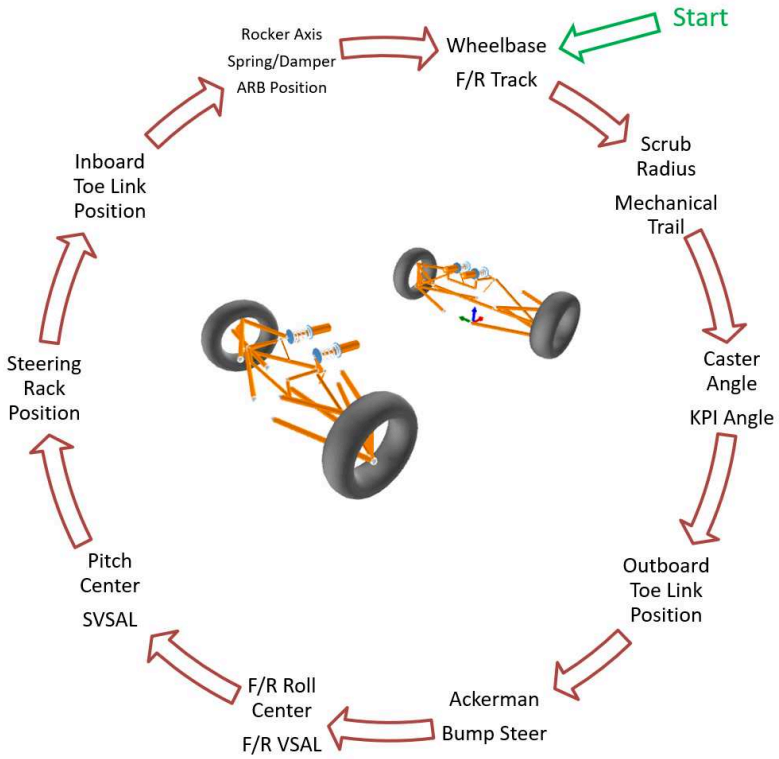
3.5.1 Outputs

The output channels calculate and store the system's information for each simulation step. These channels are utterly important, because they describe the system with information that is meaningful to the designer. During a simulation process, the configuration that describes the system is saved for each step. When the post-processing is executed, the output values — such as camber and toe angle, roll center position, scrub radius and mechanical trail — are calculated and stored in ordered vectors. The calculations of the output channels used in this work are given in Appendix A.

3.6 GENETIC ALGORITHM

As shown in section 3.1, it is quite common to iterate over a set of suspension systems when developing a new system or even improving an existing system. When designing a new system from scratch, a set of tools must be used together to accomplish a common result, usually established at the design time, therefore a set of desired output parameters are monitored over each iteration. An example of an iterative suspension design process is shown in Figure 38.

Figure 38 – Example of an iterative suspension kinematics design process.



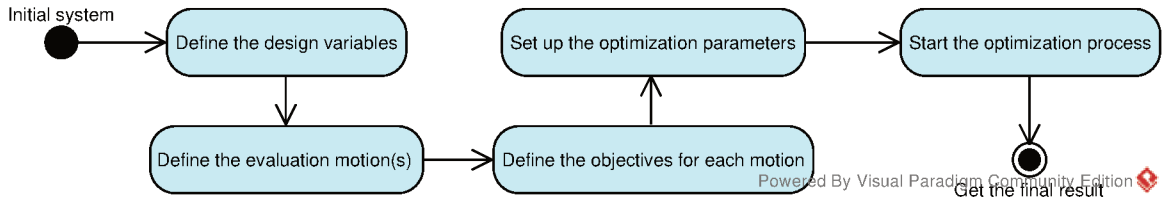
Source: Adapted from OptimumG (2020)

Nevertheless, the design process shown in Figure 38 has no limit on the number of iterations that the design team may go through. These iterations can take a considerable amount of time until a common convergence point between all the objectives and constraints are achieved, if ever achieved. Thus, the focus of the optimization presented in this work is to reduce the time spent on this phase.

3.6.1 Optimization workflow

Given the problem illustrated above, an optimization workflow is proposed in this work. It is graphically represented in Figure 39 and its details will be explained next.

Figure 39 – Proposed optimization workflow.



Source: Author (2021)

3.6.2 Design variables

The design space in this work is defined as a virtual boundary where a design variable can be created. In practical terms, it is a volume in 3-dimensional space where the algorithm performs the search of the pickup point that best matches the objective functions. The boundaries in this problem can be either *hard* or *soft*. A hard boundary is a boundary that does not allow the variable to leave its imposed limits.

Whenever a design variable leaves that region it is immediately returned to the region in the nearest possible coordinates. Similarly, the soft boundary does not limit the design variable, so it can float freely as much as the evaluation functions allow. Thus, the soft boundary is only used to initialize the population.

Two boundary geometries were implemented in this work: a box and a sphere. The box is a simple boundary defined by an upper and lower set of coordinates. Geometrically, the upper and lower boundaries are points which define the diagonal of the volume created in a Cartesian coordinate system. The sphere, on the other hand, is defined by a point and its radius. Even though the implementation allows the use of soft boundaries, it is not recommended, because it can generate suspension systems that are conceptually acceptable but impossible to be fabricated.

3.6.3 Evaluation and objective functions

Despite they have a strong connection, the evaluation and objective functions are treated separately in this work. An evaluation function is seen here as the movement of a suspension system. This movement can be any combination of heave, roll, pitch and steering.

Accordingly, the objective functions are the output channels of the evaluation function. The algorithm was structured like this in a way that the user can set different objectives for different movements. For example, it allows the algorithm to find a targeted camber gain in heave and in steering, separately.

Each objective function has its own weight function and a scaling value. The weight function represents the degree of importance of an output value in a given moment of the simulation. On the other hand, the scaling factor defines the degree of

importance of the function with relation to all other objective functions being used.

Additionally, this factor grants a mean of normalizing the objective functions, which is indispensable, once the optimization numerically compares many parameters that have different orders of magnitude. The weighting approach allows the multi-objective problem to be described as a single objective, otherwise, the optimization would yield several suspension systems.

The GA implemented in this work has two types of fitness values: the objective and the overall fitnesses. The objective fitness is the fitness value for a given objective. Thus, for an optimization that involves n objectives, there will be n objective fitnesses. The mathematical expression that describes the objective fitness if given by Equation (30).

$$f_i = \sqrt{\frac{\sum_{s=0}^{k}((out_s - obj_s)w_s)^2}{k}} S_i \tag{30}$$

where f_i is the objective i fitness value, k is the number of steps in the evaluation function, out_s is the simulation output value at step s, obj_s is the objective value at step s, w_s is the weight function value at step s and S_i is the scaling factor for objective i. The overall fitness is simply the average of all objective fitnesses of the given optimization problem, as shown in Equation (31).

$$\bar{f}_j = \sum_{i=0}^{n} \frac{f_i}{n} \tag{31}$$

where $\bar{f}_j$ is the overall fitness value of the j-th individual, n is the number of objectives.

3.6.4 Operators

All the selection, crossover and mutation operators introduced in subsection 2.7.4 where implemented in this work. However, different operators — and possibly the combination of these operators — yield different behavior in the results, ultimately resulting in a sub-optimal solution. In order to determine the best set of operators for this specific application of the GA, a parametric analysis was run. Several setup options were created and the results analysed in terms of Key Performance Indicator (KPI), presented next.

3.6.5 Key performance indicators

The first KPI used in the variance analysis is the convergence rate. The convergence rate is taken as the number of generations that the algorithm takes to find a solution which has an overall fitness that has a numerical value less than 101% of the best result of a given GA setup. This error margin was obtained empirically. This KPI aids on the identification of the setup that yields the solution in a faster manner.

In complement to the convergence rate, the time to converge is also monitored. The time to converge is the time that the algorithm takes to solve the problem. Since it is dependent on the CPU where the program is run, the absolute value of this KPI is not important, but the comparison between the other setups, as long as they are run in the same machine, with the same number of threads.

The selection pressure is also taken as a KPI that measures the diversity of the population for each setup. As mentioned in subsection 2.7.4, a high selection pressure can yield sub-optimal values, once it makes the algorithm more prone to local-minimum trapping.

This analysis was separated into two major groups: the single-objective group and the multi-objective group. The grouping was done because the single objective and multi-objective optimizations behavior can vary a lot based on the number of objectives.

3.6.6 Implementation

The GA was implemented in C++ as a template library. C++ templates are classes that are created based on the particular implementation for a particular problem. This is specially advantageous, once the GA can be extended to other problems, not only kinematics optimization. The GA is here nominated OptimumGenetics, as it will become a product of OptimumG, intended for internal usage. The general class structure of OptimumGenetics is shown in Figure 40.

Table 1 summarizes the template classes that are used in OptimumGenetics. However, in spite of several classes are generically implemented and can be used without any modifications, some classes need to be overrode to explore the full potential of the library.

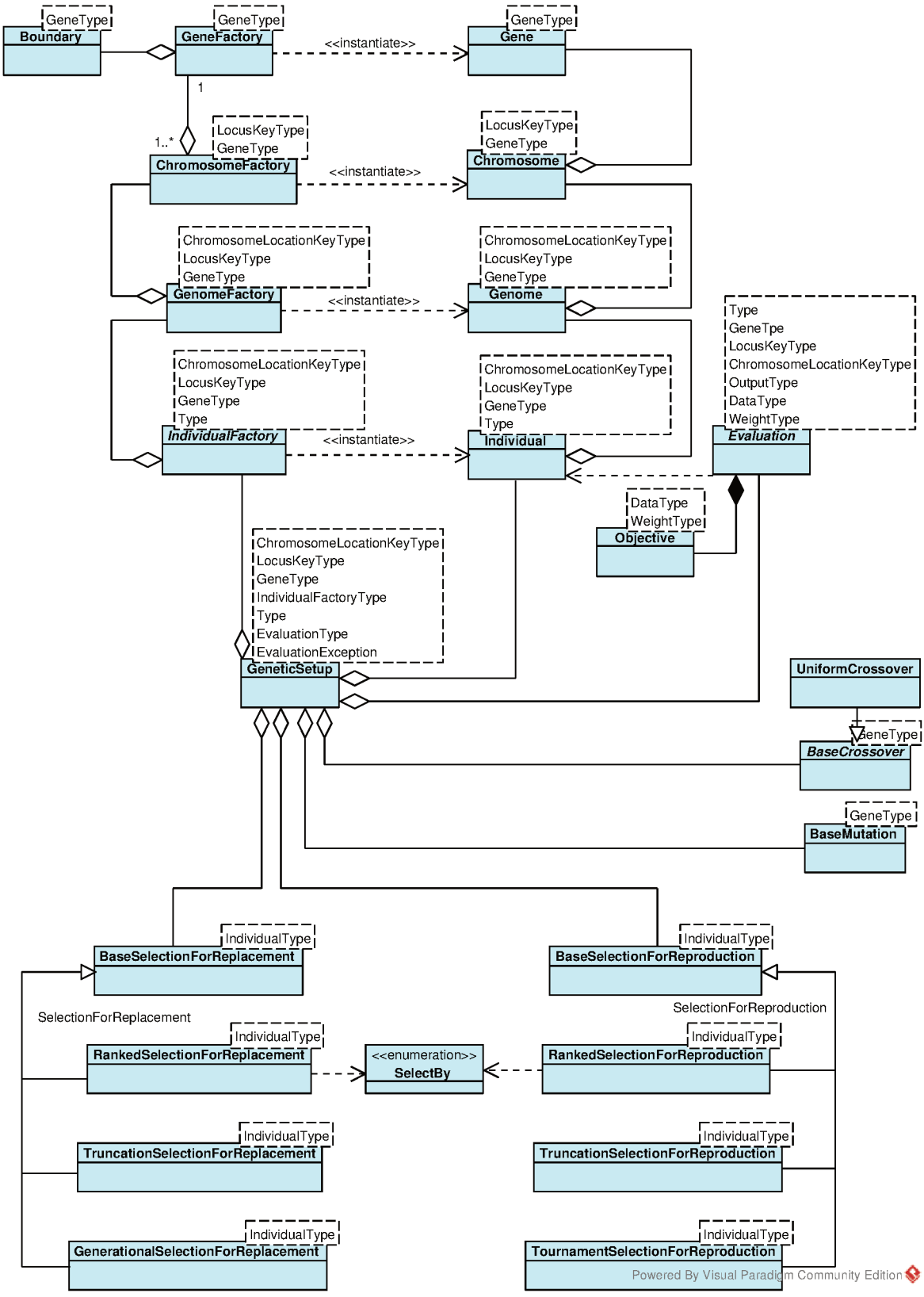
Table 1 – Template parameters for the kinematics optimization.

Template name	**Assigned class**
GeneType	Point3D
LocusKeyType	string
ChromosomeLocationType	SuspensionPosition
Type	SuspensionAssembly
OutputType	OutputsCollection
DataType	vector<double>
EvaluationException	SimulationFailedException

Source: Author (2021)

The first classes that need a specific implementation are the boundaries. The boundaries define the limits of the problem. As mentioned previously, there are two implementations of boundaries: the box and the spherical boundaries. The class diagram that define these classes is shown in Figure 41. The crossover and mutation operators

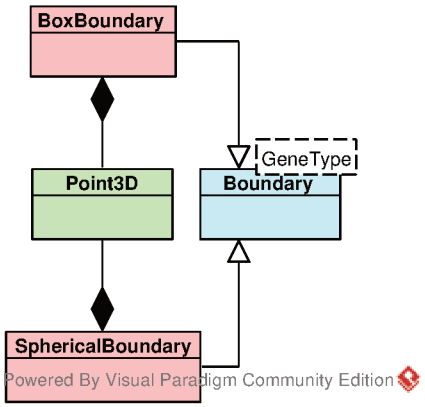
Figure 40 – OptimumGenetics template class diagram.



Source: Author (2021)

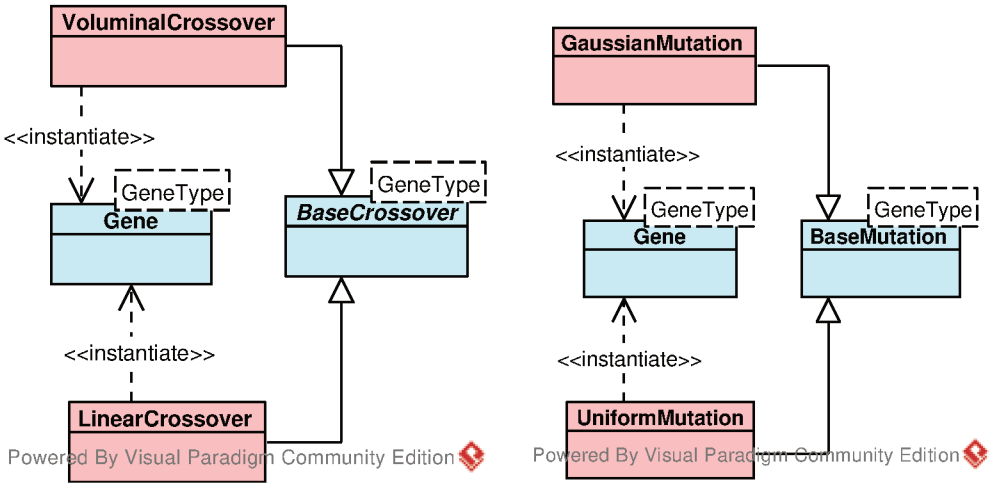
must also be defined specifically for each problem. The operators implemented for the kinematics optimization is shown in the diagrams in Figure 42.

Figure 41 – Boundaries class diagram.



Source: Author (2021)

Figure 42 – Genetic operators implementation.



(a) Crossover operators (b) Mutation operators

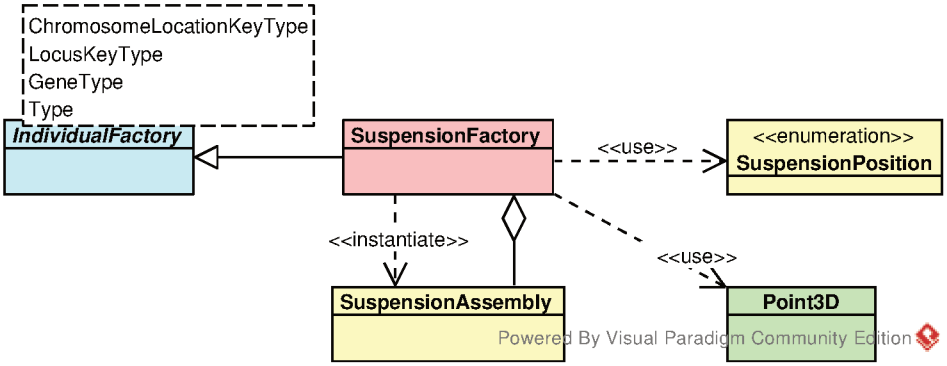
Source: Author (2021)

Furthermore, the class that clones and alters the suspension system based on its genes — pickup points — also have a specific implementation, shown in Figure 43. The evaluation implementation is probably the most complex one of the specific classes and is shown in Figure 44.

3.7 CASE STUDY

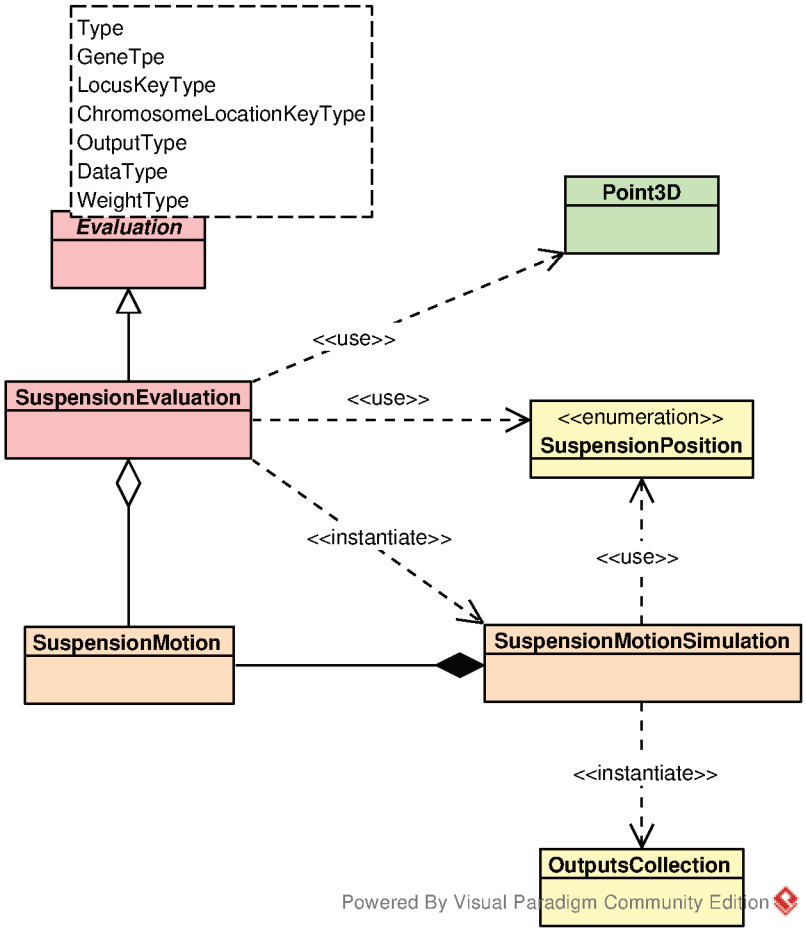
A case study is used to demonstrate the potential of the optimization proposed in this work. The case study consists of an optimization of a whole vehicle, with a Double A-Arm suspension on both axles.

Figure 43 – Suspension factory class implementation.



Source: Author (2021)

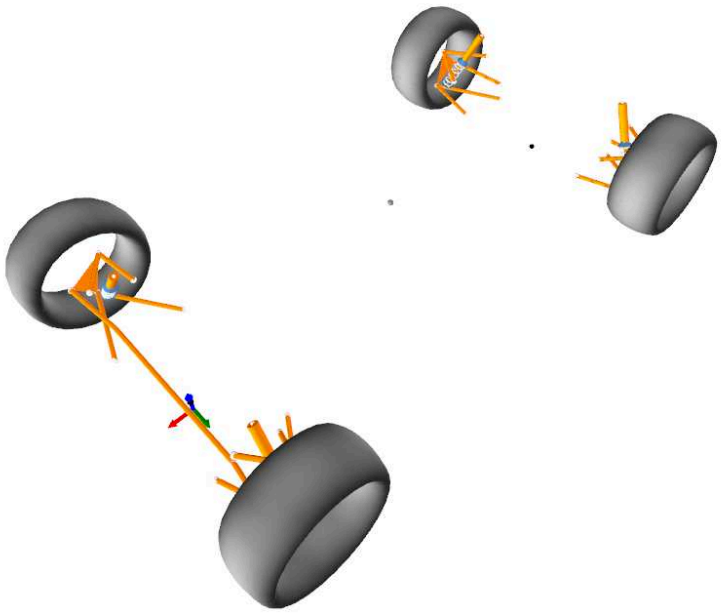
Figure 44 – Suspension evaluation class implementation.



Source: Author (2021)

The chosen initial system is derived from a GT3-spec race car suspension and its kinematic model on OptimumKinematics is shown in Figure 45. The average parameters of a GT3 vehicle are used in order to estimate the expected load transfers. Additionally, a tire model is used to determine the optimal camber in roll and steering as well as the Ackerman angle.

Figure 45 – Baseline suspension system model in OptimumKinematics (isometric view)



Source: Author (2021)

The vehicle kinematics are then analyzed in four motions, independently. They are heave, roll, pitch and steering. A set of objectives was created to describe the desired kinematic behavior of the system, totaling 19 objective functions, explained in detail in the next chapter. Moreover, the design space was chosen arbitrarily due to the lack of information of the packaging restrictions and components dimensions. However, the removal of the physical constraints allows the algorithm to perform broader searches, ultimately yielding a better solution than with the extra constraints.

4 PARAMETRIC ANALYSIS

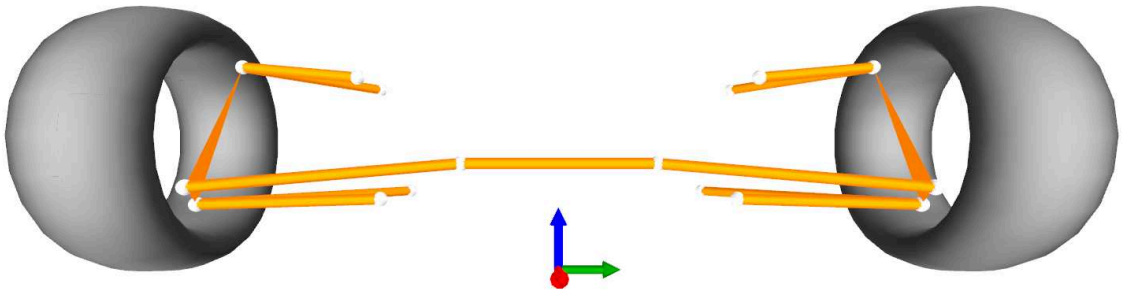
This chapter introduces the details and display the results of the parametric analysis. As previously mentioned, the parametric analysis of the genetic algorithm is of major importance, since the quality of results obtained from the optimization depend on a proper selection of GA parameters, which are further investigated in this chapter.

4.1 GENETIC ALGORITHM PARAMETRIC ANALYSIS

In order to determine the ideal set of parameters for the GA, 15 different optimization setups were created and run on the same machine. The setups are divided into two major groups: runs 1 to 6 comprise the Single Objective Optimization (SOO) setups while runs 7 to 15, the MOO setups.

A generic suspension system, illustrated in Figure 46, is arbitrarily chosen for this analysis. The design space is defined in two variables: the inner and outer ball joint positions of the tie rods. The system is set as symmetrical, which means that only the left side is changed by the optimization and then the system is mirrored by the *XZ* plane.

Figure 46 – Suspension model used in parametric analysis (Front View).



Source: Author (2021)

The design space boundaries are defined in millimeters and displayed in Table 2. Only the front axle is considered in this study. The remaining pickup points coordinates that compose the baseline system in this study can be found in Appendix B.

The design variables on Table 2 were chosen based on previous experience for being the most sensitive suspension parameters that change the later proposed objectives. Before setting the objectives, a set of GA parameters is fixed as shown in Table 3.

The number of generations is fixed for this study because the objective of the parametric analysis is not do determine the needed generations to optimize a system, but to present a set of parameters that converge towards an optimal solution faster.

Table 2 – Design space defined for the GA parametric study.

Boundary	Pickup point		Box boundary Upper (mm)	Box boundary Lower (mm)	Spherical Boundary Center (mm)	Spherical Boundary Radius (mm)
Box	CHAS_tie_l	$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$	$\begin{bmatrix} 63.3 \\ 203.0 \\ 233.2 \end{bmatrix}$	$\begin{bmatrix} 73.3 \\ 213.0 \\ 243.2 \end{bmatrix}$	-	-
Sphere	UPRI_tie_l	$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$	-	-	$\begin{bmatrix} 133.8 \\ 750.0 \\ 191.2 \end{bmatrix}$	50

Source: Author (2021)

Table 3 – Fixed parameters for the GA parametric study.

Parameter	Value	Units
Number of generations	2000	-
Crossover type	Voluminal	-
α_{BLX}	2	-
Mutation rate	10	%
Mutation type	Gaussian	-
σ_g	2	mm

Source: Author (2021)

Despite the importance of the crossover and mutation operators in the GA setup, they will not be studied at this time.

Through empirical analysis and intuition, the voluminal crossover and Gaussian mutation have demonstrated to be better suited for this type of problem. The $\alpha_{BLX} = 2$ was chosen such as even with a crossover operation, the resulting point from the crossover operation will not restrained inside its parents genes. It gives the GA a lower selection pressure, maintaining the population diversity through several generations, when compared to when $\alpha_{BLX} \leq 1$, ultimately avoiding local minimum traps.

According to Dréo et al. (2006), common mutation rates range from 1 to 5%. However, this study uses 10%, because for low mutation strengths, such as 2 mm of standard deviation for the Gaussian mutation, the operator can maintains a lower selection pressure without generating too disperse individuals, specially for further generations. Moreover, the Gaussian distribution allows the algorithm to escape local minima in a lower rate.

4.1.1 Single objective optimization parameters

The single objective optimization parametric study is based on a bump steer optimization, where the main objective is to minimize the toe angle variation along the described motion. A heave motion ranging from -50 to +50 mm from the static position

is used as an evaluation function. No weight function nor scaling factor are applied to the following problems.

The first parameter variation studied was the population size combined with truncation selection for reproduction and generational selection for replacement. The details of these runs are shown in Table 4. The truncation selection for reproduction was not tested against the steady state selection for replacement because the algorithm would keep selecting the same individuals through many generations, potentially trapping the solution into a sub-optimal configuration.

Table 4 – Truncation selection for reproduction variance analysis.

	Run number	
	1	**2**
Population size	50	200
Selection for reproduction	Truncation	
Selection size	10	
Selection for replacement	Generational	

Source: Author (2021)

The results comparison of runs 1 and 2 are shown in Figure 47. As the graph suggests, the population size does not influence in the convergence behavior of the optimization. The closer the optimization gets to the final solution, more spread can be seen in the average fitness, which reaches values very close to 1. Figure 47 also shows that the mutation operator can keep a minimum diversity in the population on later generations.

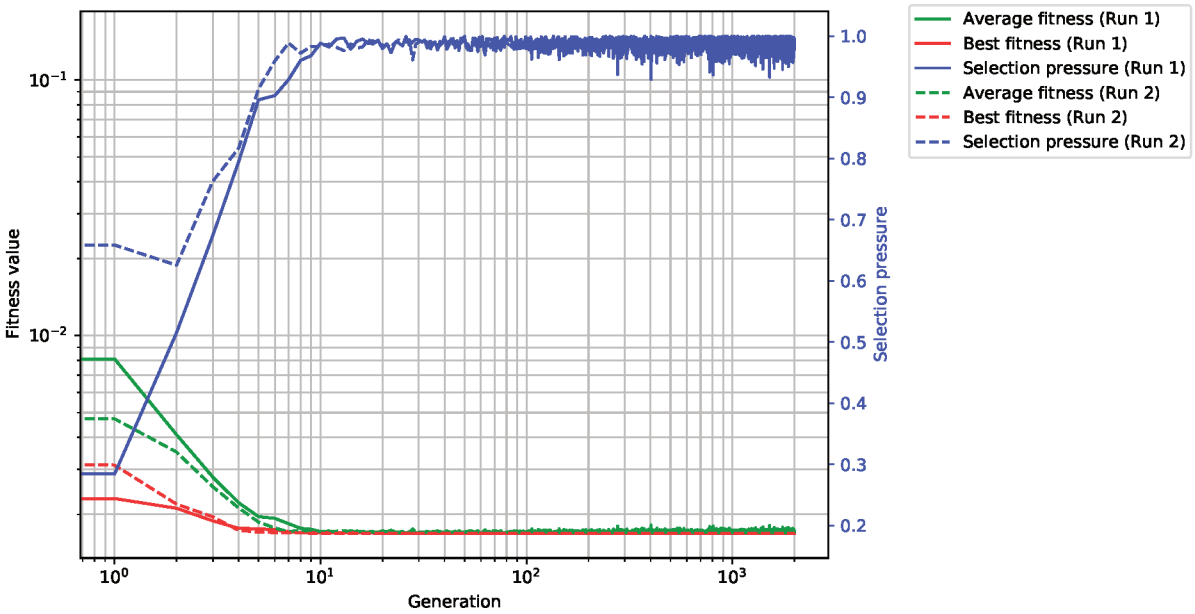
The second parameter to study for the single objective case is the tournament selection for reproduction. It is combined with generational and steady state selection for replacement methods. Since the tournament selection has an additional parameter, the tournament size, the runs were split into four, divided in two comparisons, as shown in Table 5 and Table 6.

Table 5 – Tournament selection for reproduction variance analysis with generational selection for replacement.

	Run number	
	3	**4**
Population size	200	200
Selection for reproduction	Tournament	
Selection size	10	10
Tournament size	5	10
Selection for replacement	Generational	

Source: Author (2021)

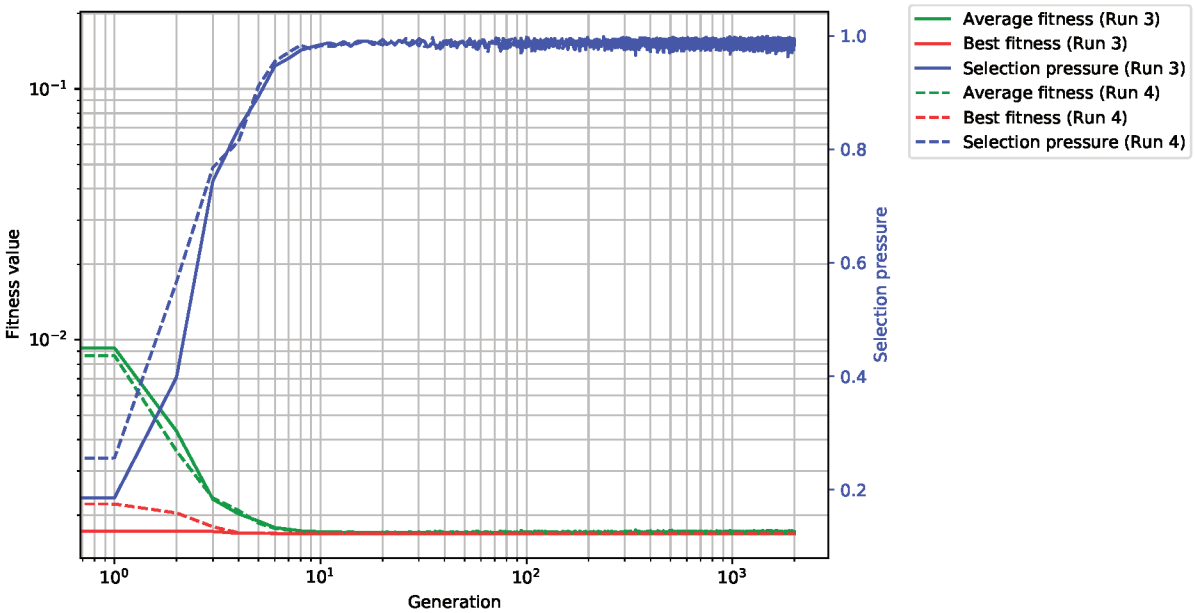
Figure 47 – Convergence graphs between runs 1 and 2.



Source: Author (2021)

The results from runs 3 and 4 are shown in Figure 48. As the graph shows, there is not much difference in the tournament size for this population with regards to the convergence rate. In both runs, the average fitness is decreasing in roughly the same rate, taking into account the stochastic nature of the GA.

Figure 48 – Convergence graphs between runs 3 and 4.



Source: Author (2021)

The results comparison between runs 5 and 6 are shown in Figure 49. Once

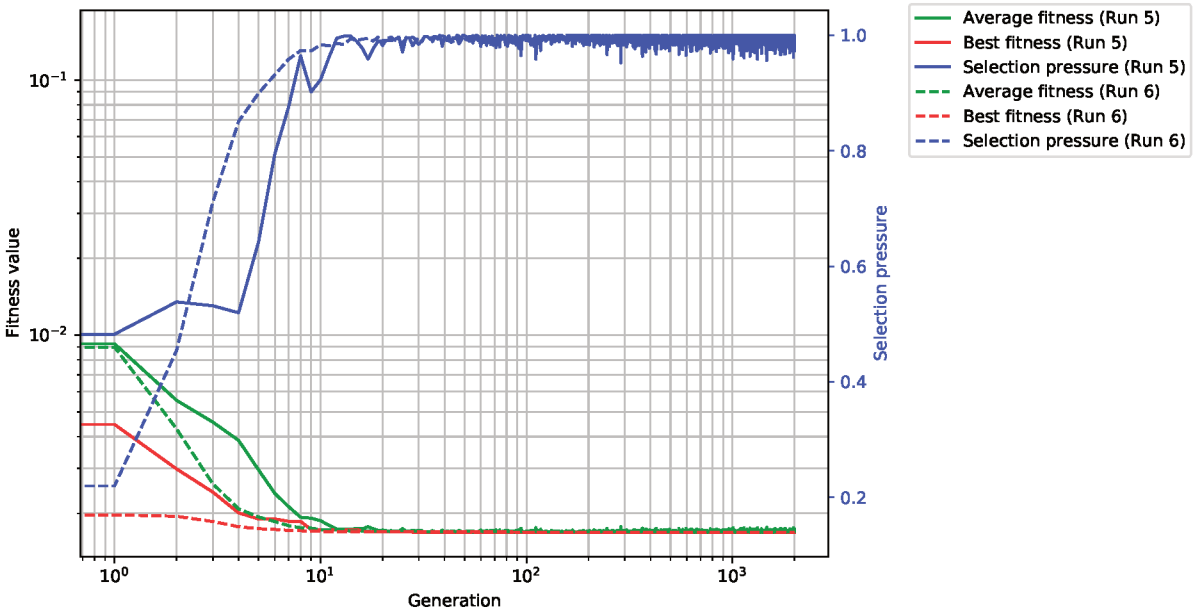
Table 6 – Tournament selection for reproduction variance analysis with steady state selection for replacement.

	Run number	
	5	**6**
Population size	50	200
Selection for reproduction	Tournament	
Selection size	10	10
Tournament size	5	10
Selection for replacement	Steady state	
Selection size	25	100

Source: Author (2021)

again, the tournament selection for reproduction does not show expressive differences when both are using steady state replacement — even with different population sizes.

Figure 49 – Convergence graphs between runs 5 and 6.



Source: Author (2021)

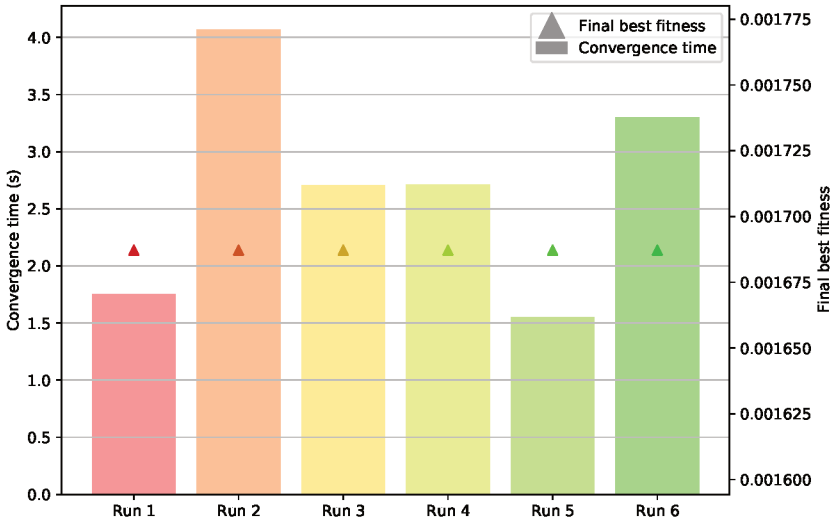
Figure 50 summarizes the optimization results of this study done for a single objective optimization. The plot shows that there is not a visual difference between the result's quality with relation to any of the setups used. However, it is possible to identify a difference in the convergence time. The convergence time is measured as the time needed (in seconds) for the GA to find an individual that has an overall fitness within 101% of the overall fitness of the best individual found at the end of the optimization.

Even though every setup yields an acceptable sub-optimal solution before 5 seconds, run number 5 stands out as the fastest one to achieve such value. However,

given the stochastic nature of such optimization, it is recommended for future studies that these runs are repeated several times and an average value is considered.

All setups yielded very similar solutions. The resulting system's optimized function is overlaid with its objective function in Figure 51. The result shows that there is no better solution for the problem in the given design space for this problem. In order to improve the objective, the designer must change the design space. Still, the design space also affects the vehicle's packaging and manufacturing, thus the suspension designer must be aware of such restriction prior to the optimization.

Figure 50 – Single objective optimization results and KPIs comparison.



Source: Author (2021)

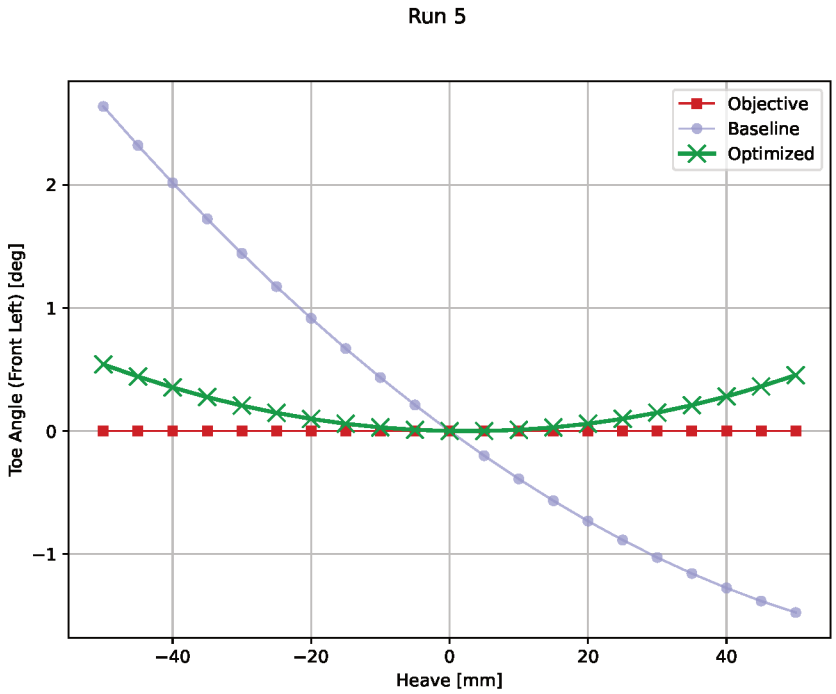
4.1.2 Multiple objectives optimization parameters

As previously mentioned, when running a MOO, there is usually no single final solution, but a set of solutions that are equally optimal. OptimumGenetics weight functions and scaling factors provide an interface to the user that allows the MOO to be treated as an SOO.

However, when the user does not have a clear definition of the objective functions and the sensitivity of each objective with relation to the overall solution, the optimization results can provide information regarding the sensitivity of the system with respect to the given objectives. This section shows another set of parameters that are sensitive to the MOO process.

Following the philosophy of subsection 4.1.1, the design space is kept for this section as well, whilst two other objectives contained in another evaluation are considered. The added objectives are: set a constant steering ratio throughout the whole steering motion — -270 to 270 degrees in the steering wheel — and provide a parallel

Figure 51 – Optimization run 5 resulting system overlaid with objective function.



Source: Author (2021)

Ackerman geometry, that is, set the Ackerman angle to zero through the whole steering motion.

Similarly to subsection 4.1.1, several setups were created in order to define the best set of parameters for this problem. Table 7 show the input parameters for runs 7 and 8, which are based on the ranked selection for reproduction and generational selection for replacement. The variable parameter of this run is the population size. The results comparison between runs 7 and 8 are shown in Figure 52.

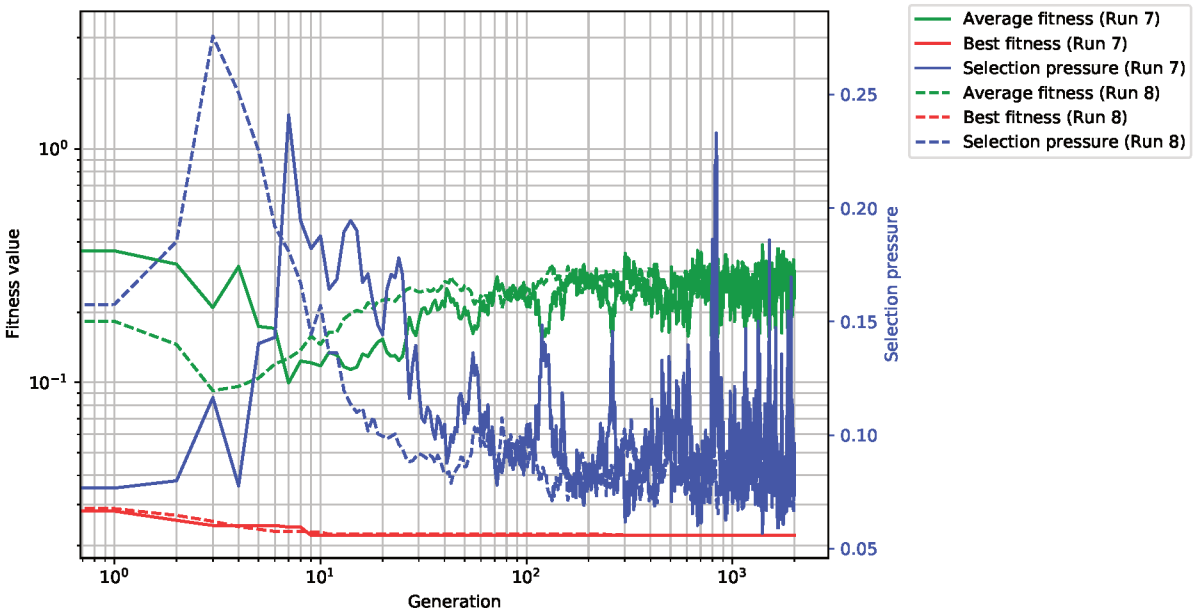
Table 7 – Ranked selection for reproduction vs population size variance analysis setup.

	Run number	
	7	**8**
Population size	50	200
Selection for reproduction	Ranked	Ranked
Selection size	1 Rank	1 Rank
Selection for replacement	Generational	Generational

Source: Author (2021)

Differently from previous results, runs 7 and 8 show a noticeable variation on the population average fitness, specially on later generations. Moreover, the selection pressure is kept under 0.25 for later generations, which is an indication of a diverse population.

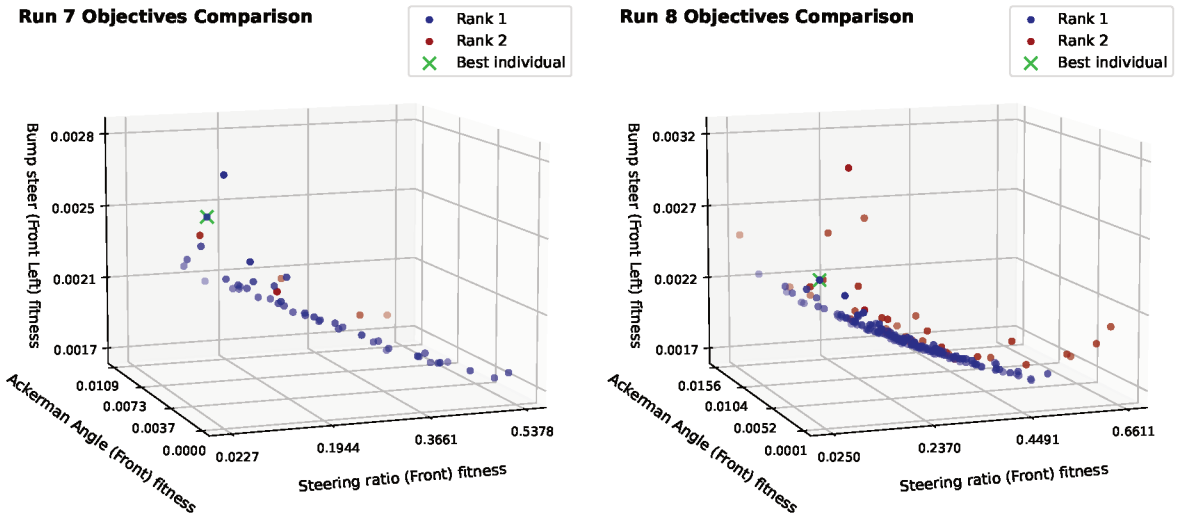
Figure 52 – Convergence graphs between runs 7 and 8.



Source: Author (2021)

These effects happen because the selection operators do not consider the individuals' overall fitness — they are instead chosen for reproduction by their objective functions — and then the whole generation is replaced after. Since weighting and scaling factors are equal to every objective, most of the variation is possibly coming from only one of the objectives. Figure 53 allows the visual demonstration of such effects.

Figure 53 – Runs 7 and 8 objectives comparison.



(a) Run 7 (b) Run 8

Source: Author (2021)

The steering ratio fitnesses as shown Figure 53a and Figure 53b, have a variance

of about two orders of magnitude with relation to the other objectives for both setups. This is a huge indication that the objectives should be scaled, in order to produce a better solution. It is also interesting to note that the objectives yield a very clear Pareto set, denoted by Rank 1.

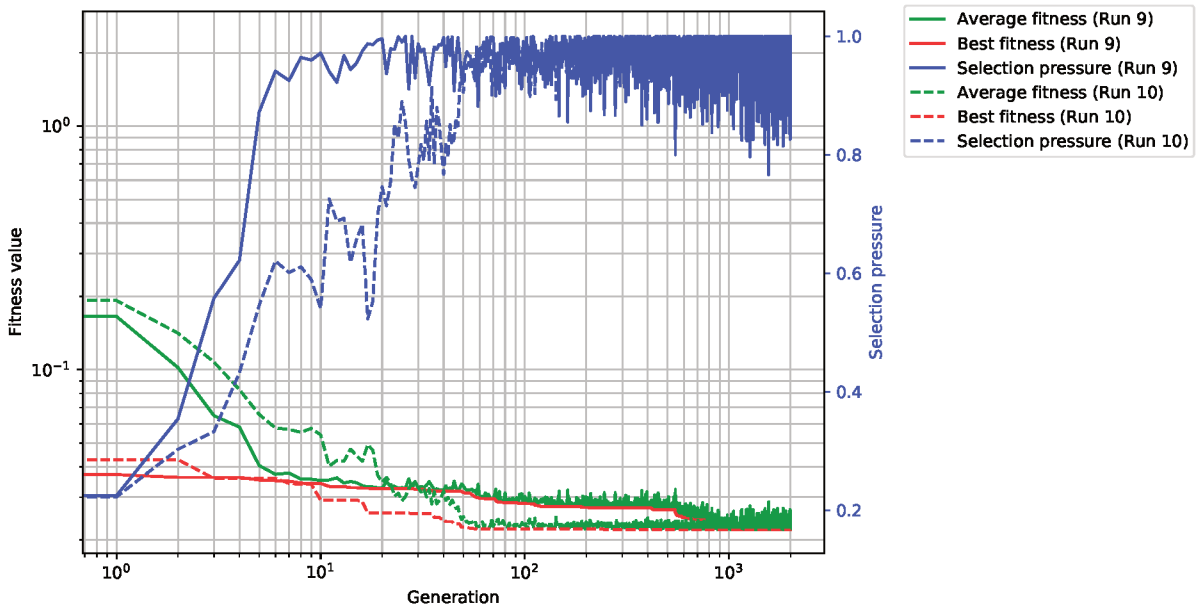
Table 8 shows the variation of the objectives for a tournament selection for reproduction with relation to the variation of the size of the selection for replacement set (runs 9 and 10). The convergence plots for runs 9 and 10 are found in Figure 54.

Table 8 – Tournament selection for reproduction vs steady state selection for reproduction selection size variance analysis setup.

	Run number	
	9	**10**
Population size	200	200
Selection for reproduction	Tournament	Tournament
Selection size	10	10
Tournament size	5	5
Selection for replacement	Steady state	Steady state
Selection size	20	80

Source: Author (2021)

Figure 54 – Convergence graphs between runs 9 and 10.



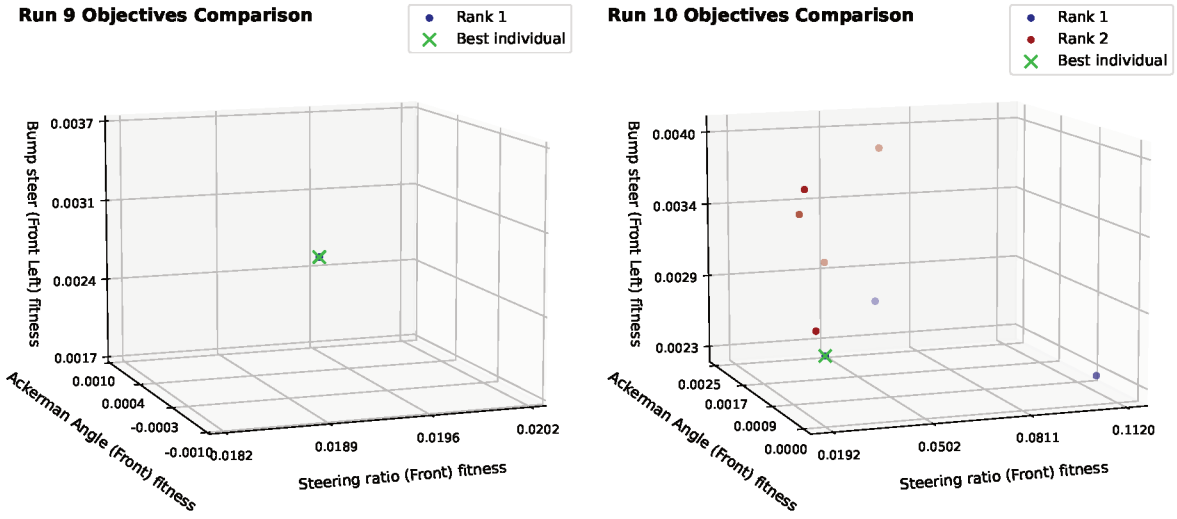
Source: Author (2021)

As shown in Figure 54, the selection pressure achieves values very close to 1 in later generations and the mutation operator guarantees that the population has a

minimum diversity, specially on generations past 1000. However, this method is more sensitive with relation to the weights and scaling factors input by the user.

Even though runs 9 and 10 use a higher population number, they yield sub-optimal values, possibly trapped in a local minimum. Figure 55 illustrates the objective functions for the last population. As the figure suggests, the algorithm has a strong tendency of concentrating the values in a particular region of the feasibility range.

Figure 55 – Runs 9 and 10 objectives comparison.



(a) Run 9 (b) Run 10

Source: Author (2021)

Runs 11, 12 and 13 comprise ranked selection for both reproduction and replacement. The variation parameter is the number of ranks chosen for both operators, as described by Table 9. The convergence of these runs are shown in Figure 56.

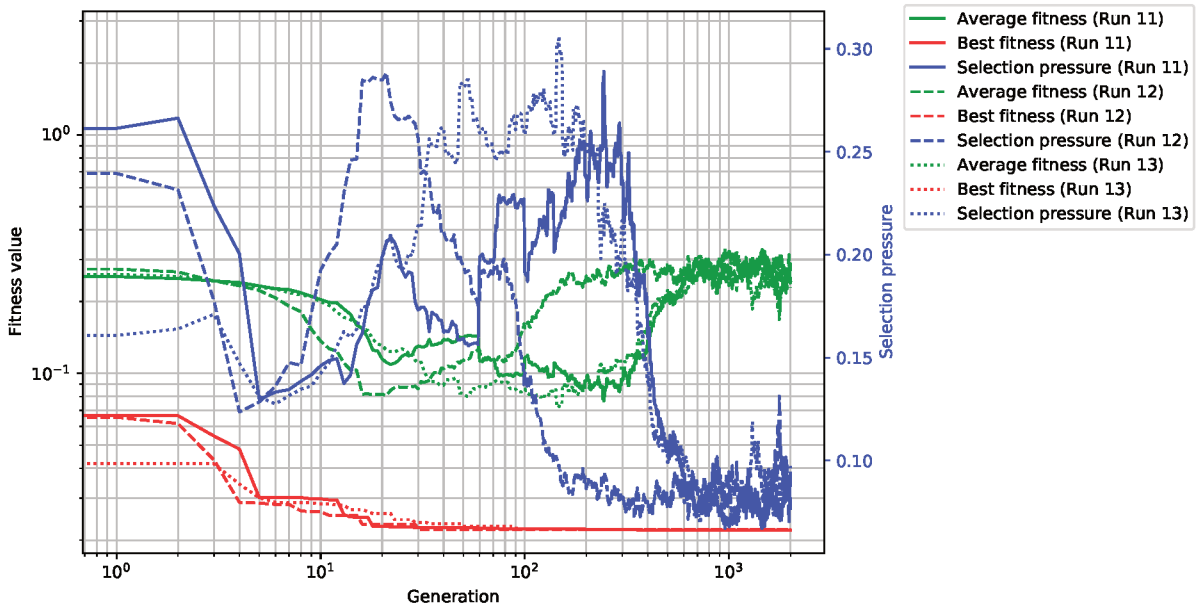
Table 9 – Ranked selection for reproduction and replacement variance analysis setup.

	Run number		
	11	**12**	**13**
Population size	200	200	200
Selection for reproduction	Ranked	Ranked	Ranked
Selection size	1 Rank	1 Rank	2 Ranks
Selection for replacement	Ranked	Ranked	Ranked
Selection size	1 Rank	2 Rank	1 Rank

Source: Author (2021)

In correlation to Figure 52, runs 11, 12 and 13 have selection operators that do not depend on the overall fitness of the population. This causes the low selection pressure seen in later generations — meaning that the population has a high diversity — and the large variation of average fitness. Such similarity is shown by Figure 57.

Figure 56 – Convergence graphs between runs 11, 12 and 13.



Source: Author (2021)

Figure 57 shows the similarity to runs 7 and 8. All setups yielded a clear Pareto set (denoted by Rank 1). Still, run 13 resulted in only one rank, which represents all the Pareto solutions of the optimization.

At last, runs 14 and 15 compare the ranked selection associated with a steady state replacement, as shown in Table 10. The results are shown in Figure 58.

Table 10 – Ranked selection for reproduction with steady state selection for replacement variance analysis setup.

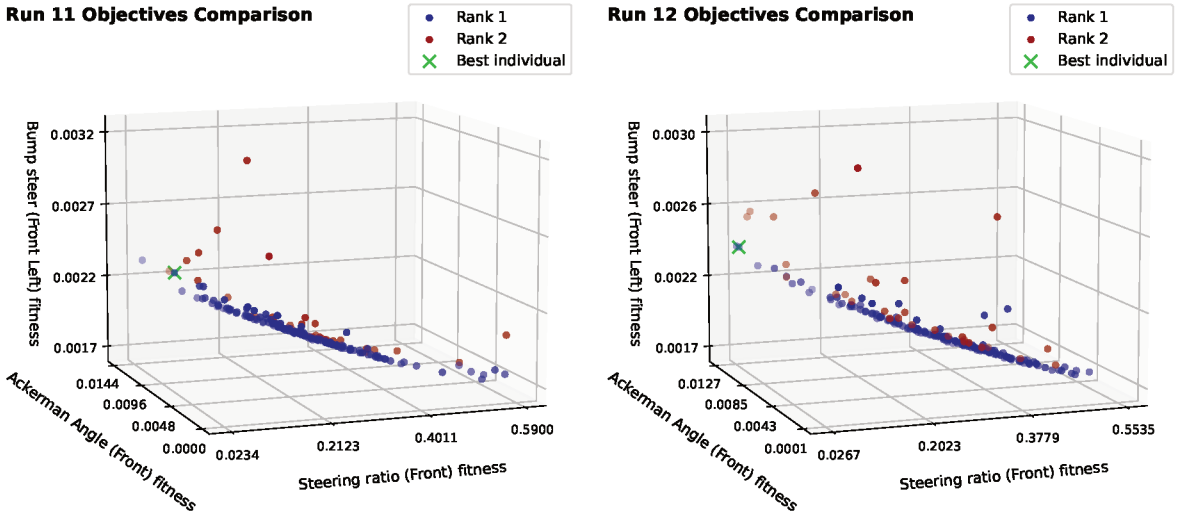
	Run number	
	14	**15**
Population size	200	200
Selection for reproduction	Ranked	Ranked
Selection size	1 Rank	1 Rank
Selection for replacement	Steady state	Steady state
Selection size	100	10

Source: Author (2021)

Figure 58 demonstrate a more stable solution, which has a selection for reproduction operator that does not depend on the user-defined weight and scale factors. Nonetheless, the final solution is still dependent on the weight functions and scaling factors. Figure 59 show the correlation between the objectives.

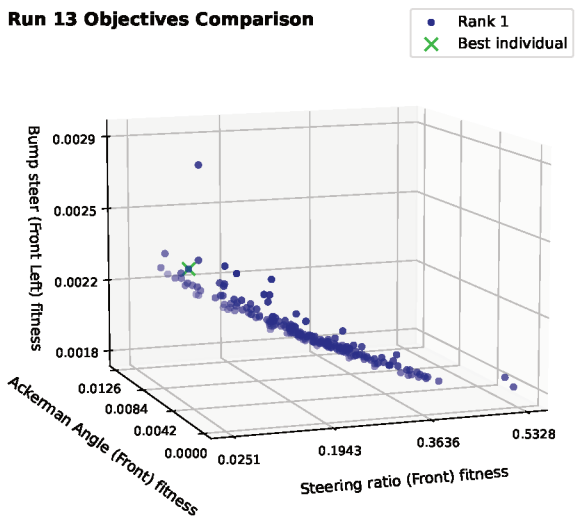
Exposed by Figure 59, run 14 is more aggressive towards the final objective — the overall fitness — whilst run 15 parameters allow the optimization to keep a certain

Figure 57 – Runs 11, 12 and 13 objectives comparison.



(a) Run 11

(b) Run 12



(c) Run 13

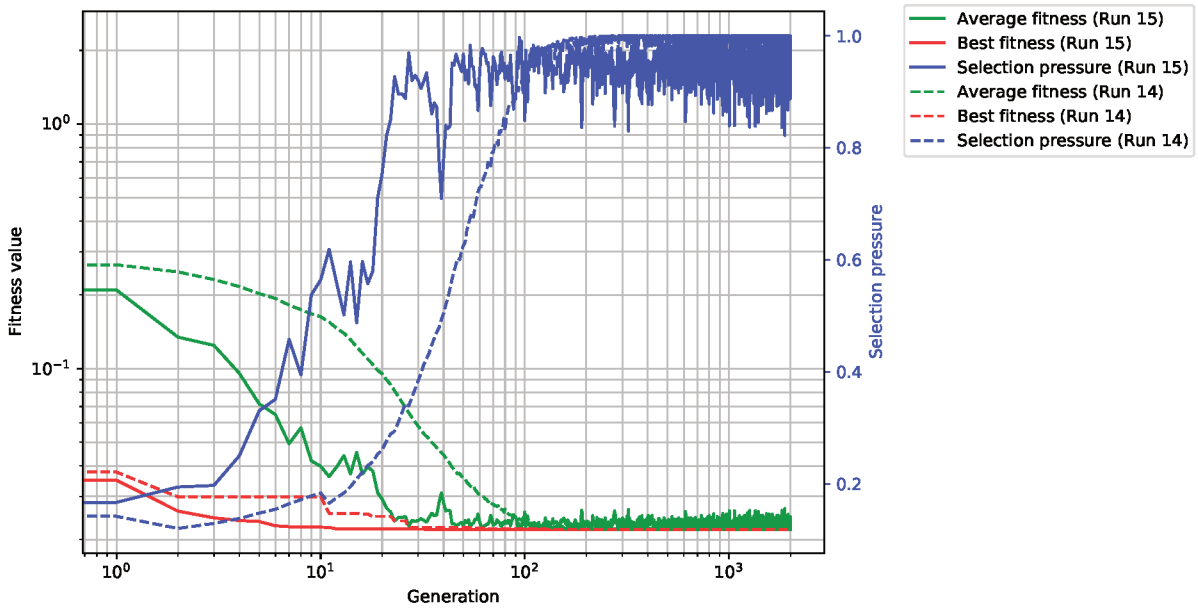
Source: Author (2021)

degree of diversity, possibly allowing the solution to escape local minima traps. Run 14 yielded one set of optimal solutions that are concentrated in a small region, while run 15 has a greater capability of exploring the neighborhood.

Figure 60 compares the setups and the results in terms of quality — overall fitness — and convergence time. From the chart, run 7 provided a faster convergence rate but a sub-optimal solution. On the other hand, run 15 did not exceed run 7's convergence time by too much and still provided a much better solution, based on the user-defined weighting functions and scaling factors.

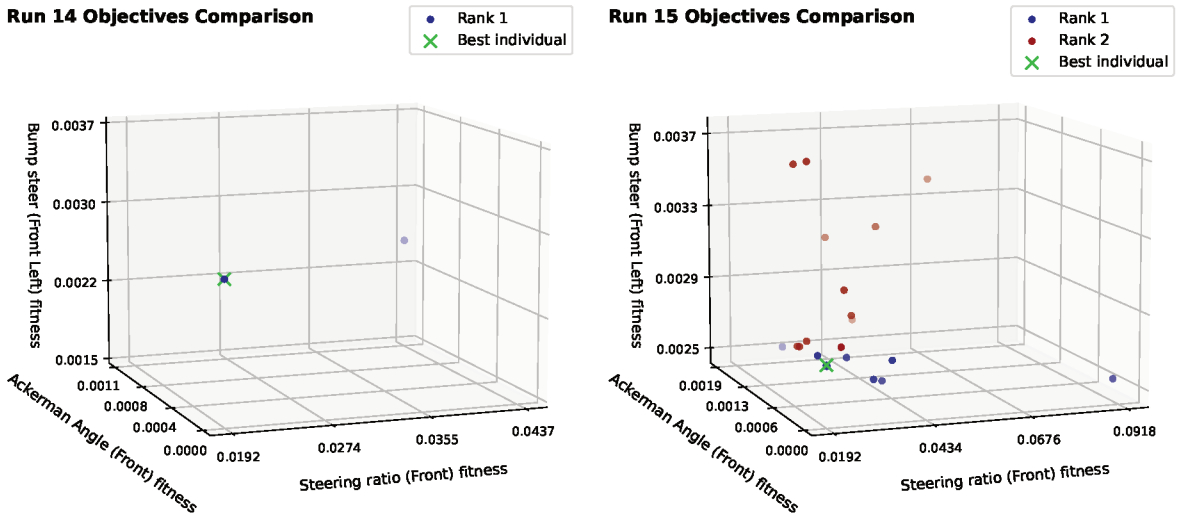
Despite that the results generated from runs 7 to 15 have very different numerical results and convergence behaviors, the average solution is very similar. The resulting

Figure 58 – Convergence graphs between runs 14 and 15.



Source: Author (2021)

Figure 59 – Runs 14 and 15 objectives comparison.



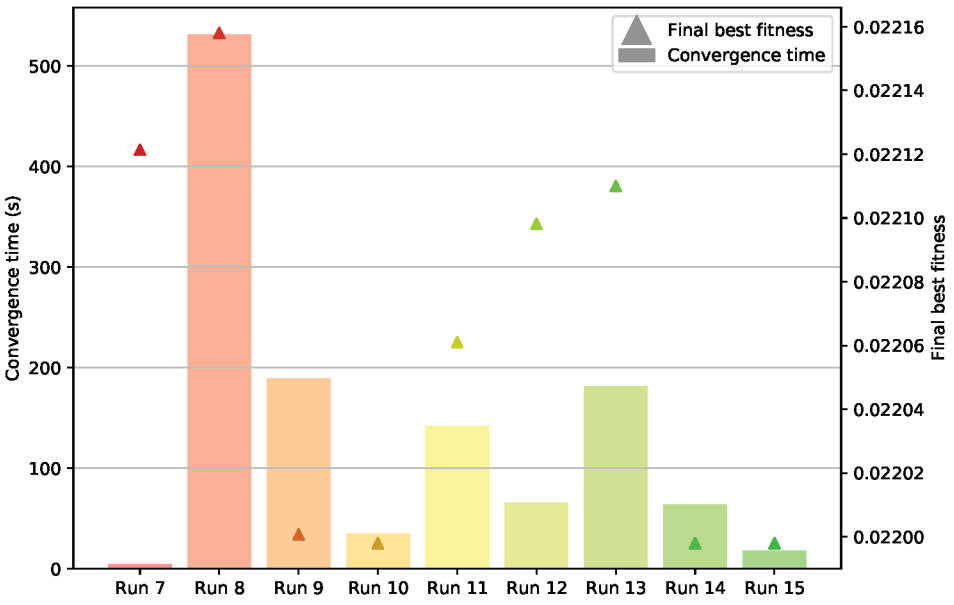
(a) Run 14 (b) Run 15

Source: Author (2021)

coordinates of such optimization are shown by the resulting coordinates of the design variables in Table 11 (units in millimeters). The three objective functions overlaid with their respective initial and final solutions of run 15 are shown in Figure 61. The visual difference of the objective results between the other multi-objective setups is so subtle that cannot be visualized in a chart.

Figure 61 show that even though two of the three objectives were drastically

Figure 60 – MOO results comparison.



Source: Author (2021)

Table 11 – Final result of runs 7 to 15.

		Run number								
Point		**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
CHAS_tie_I	X	73.3	63.8	73.3	73.3	71.6	67.9	63.3	73.3	73.3
	Y	213.0	212.6	213.0	213.0	212.6	206.6	213.0	213.0	213.0
	Z	233.2	233.4	233.2	233.2	233.2	233.2	233.2	233.2	233.2
UPRI_tie_I	X	118.2	117.8	118.1	118.1	118.1	117.9	117.8	118.1	118.1
	Y	768.6	766.5	768.3	768.2	768.1	767.4	766.6	768.2	768.2
	Z	234.7	235.6	235.0	235.0	235.0	235.3	235.5	235.0	235.0

Source: Author (2021)

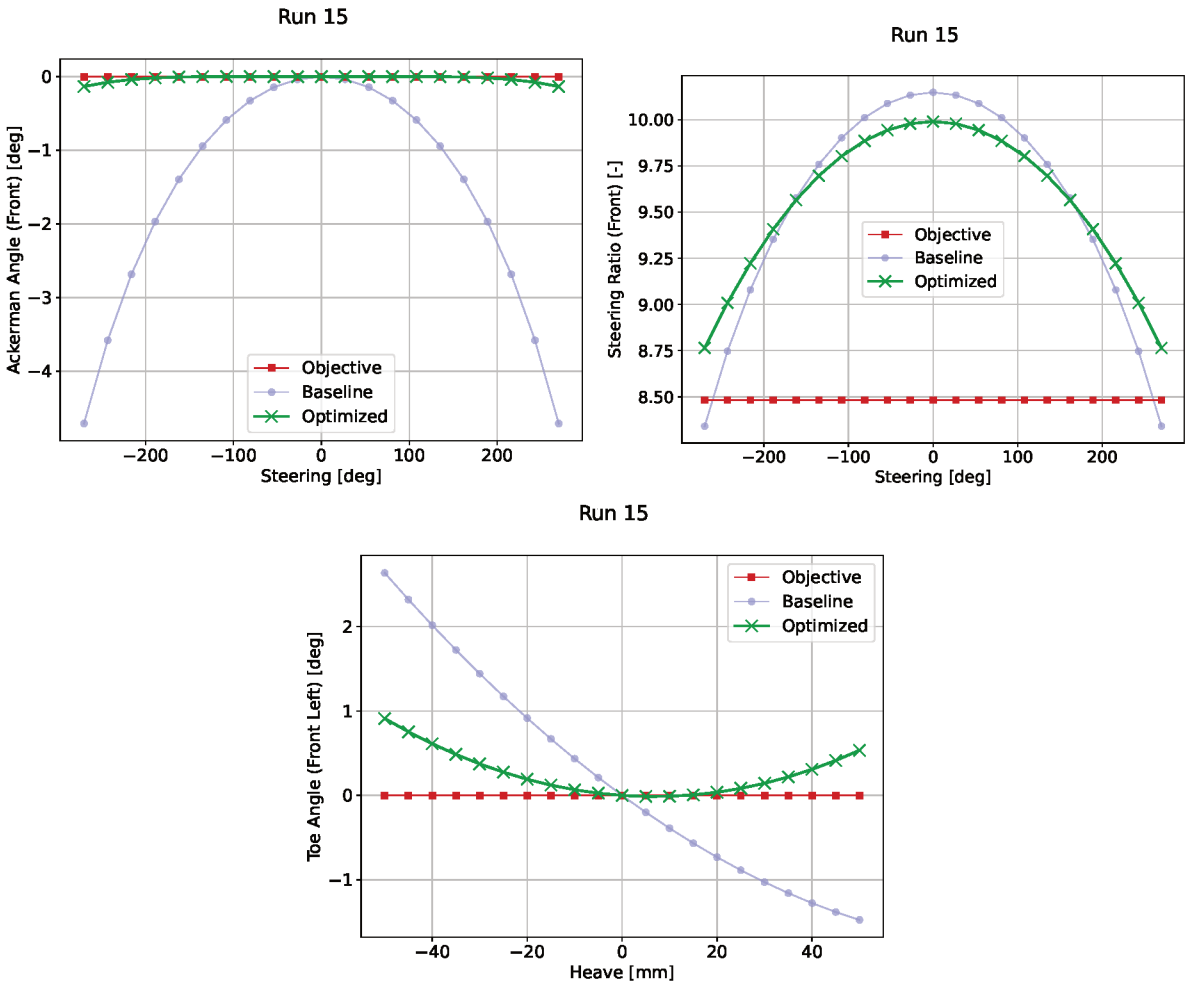
improved by the optimization algorithm. The steering ratio objective was probably not reached due to the narrow design space defined for this study. It shows, however, the best compromise between all the objectives, given the set of weights and scaling factors defined by the user.

4.1.3 Remarks

This study shows that the parameters of the GA can have a huge influence on the final result of the optimization. In addition, to yield a single solution to a MOO problem using this implementation, the user must have a good understanding of the parameters sensitivity and feasibility. Such feeling may come with practice and experience.

Thus, this kind of implementation may not be well suited for novice suspension designers. Nevertheless, it also provides an indication of which set of parameters are

Figure 61 – Objective results on run 15.



Source: Author (2021)

best suited depending on the set of objectives and design space.

5 CASE STUDY

This chapter introduces the case study conducted to illustrate the potential of the optimization. It begins elucidating the problem and displaying the basic information regarding the vehicle to be designed, such as initial pickup points, total mass and load transfer. Furthermore, a specific tire model for the car under optimization is used to determine some of the kinematics parameters. The chapter covers all the objective functions used in the optimization and their respective results.

5.1 PROBLEM DESCRIPTION

The baseline vehicle used in this work is a GT3-spec race car. The information about this type of car is provided by OptimumG. The make and model of the car are maintained confidential, however, the other available parameters will be shown in further sections. The pickup points coordinates in millimeters are displayed in Table 12. The remaining parameters used for the simulation can be found in Appendix C.

The origin of the system of coordinates is taken in the front axle, coincident to the ground plane and at the car's plane of symmetry (mid plane). The X coordinate is positive towards the forward direction of the vehicle, the Y direction is positive to the left side of the vehicle and Z is positive to values above the ground. Only the left coordinates are shown because the system is symmetrical, thus, the Y points are reflected about the XZ plane.

The OptimumKinematics model of this system was already introduced by Figure 45. The colored vectors represent the origin of the coordinate system. The X, Y and Z coordinates are represented by the red, green and blue arrows respectively. This case study does not take into consideration the vehicle's packaging due to the lack of access to the car's CAD model. Even though it lacks a little bit on the sense of reality, it gives more freedom to the optimization model as the design space can be widened, fundamentally making this case study more interesting.

5.2 TIRES

The GT3-spec tire model is used in this work was provided by OptimumG. The data was gathered on a tire testing facility and fitted using OptimumTire tire modeling software, as mentioned in section 2.2. Since it is not relevant to this work, the fitting process is not covered. Figure 62 and Figure 63 show the tire models, in continuous lines, overlaid with the tire's data. Both plots show a good correlation between the model and respective data. The data used in this case study is analyzed at constant pressure — 1.8bar — considering the huge complexity of using multiple pressure values.

The fitted model is then exploited in search of the optimal tire position that

Table 12 – Baseline suspension system pickup points.

Point name	X(mm)	Y (mm)	Z (mm)
Front suspension			
CHAS_lower_fore_l	137.4	416.9	110.0
CHAS_lower_aft_l	-249.3	415.5	146.6
CHAS_upper_fore_l	82.2	553.2	452.5
CHAS_upper_aft_l	-120.3	560.3	365.1
UPRI_lower_l	-1.9	820.1	145.3
UPRI_upper_l	-66.6	752.5	445.6
CHAS_tie_l	64.99	472.3	240.0
UPRI_tie_l	110.9	786.9	287.9
Rear suspension			
CHAS_lower_fore_l	-2454.3	437.8	194.5
CHAS_lower_aft_l	-2827.3	431.4	125.5
CHAS_upper_fore_l	-2451.6	546.4	389.4
CHAS_upper_aft_l	-2761.5	541.5	445.6
UPRI_lower_l	-2573.0	792.6	183.2
UPRI_upper_l	-2592.3	768.5	454.0
CHAS_tie_l	-2816.5	435.0	248.8
UPRI_tie_l	-2780.0	810.1	270.8

Source: Author (2021)

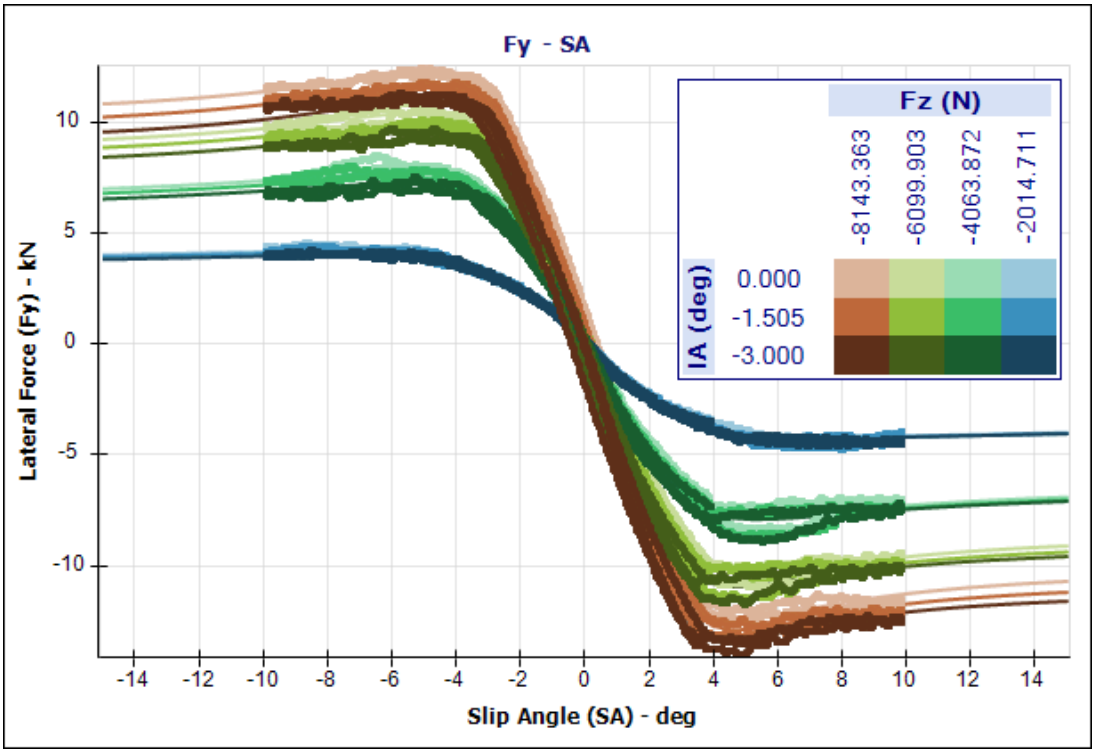
generates the maximum lateral force for most situations. Figure 64 shows the slip angle α at peak F_y shifting. For example, the front axle shown in Figure 64a suggests the use of a pro-Ackerman steering geometry. The plots also show a tendency that as the camber angle increases, the lower the Ackerman is required. Still, this effect is more relevant for lower normal loads.

Moreover, the model can be used to leverage the potential of lateral force increase due to camber. Figure 65 shows the variance of the peak lateral force due to normal load variation. Three different inclination angles γ (IA) are shown. Regarding the use of the SAE coordinates, a positive γ represents a negative camber on the outside wheel, when it is mounted on a car that is inside a turn. Comparing Figure 65a and Figure 65b, it is possible to note that the rear tires are a slightly more sensitive to the γ variation than the front tires, indicating a possible preference on the camber variance objective, which will be discussed later in this text.

5.3 WEIGHT TRANSFER ANALYSIS

In order to determine the kinematics parameters based on the tires, a simplified steady-state weight transfer analysis is used to determine the vertical load in each tire. The load transfer model is presented by William F. Milliken and Douglas L. Milliken

Figure 62 – F_y vs α plot for the proposed GT3 front tires (SAE coordinates).



Source: Author (2021)

(1995) and demonstrated over the next equations. The *downforce* equation generated by the car is represented in Equation (32).

$$F_d = \frac{C_\ell A \rho_{air} V^2}{2} \tag{32}$$

where F_d is the downforce value, in newtons, $C_\ell A$ is the downforce (negative lift) coefficient multiplied by the frontal area, in m^2, ρ_{air} is the air density and V is the vehicle speed, in m/s. Adding the downforce to to the vehicle weight yields the vehicle load at the given conditions, as denoted in Equation (33).

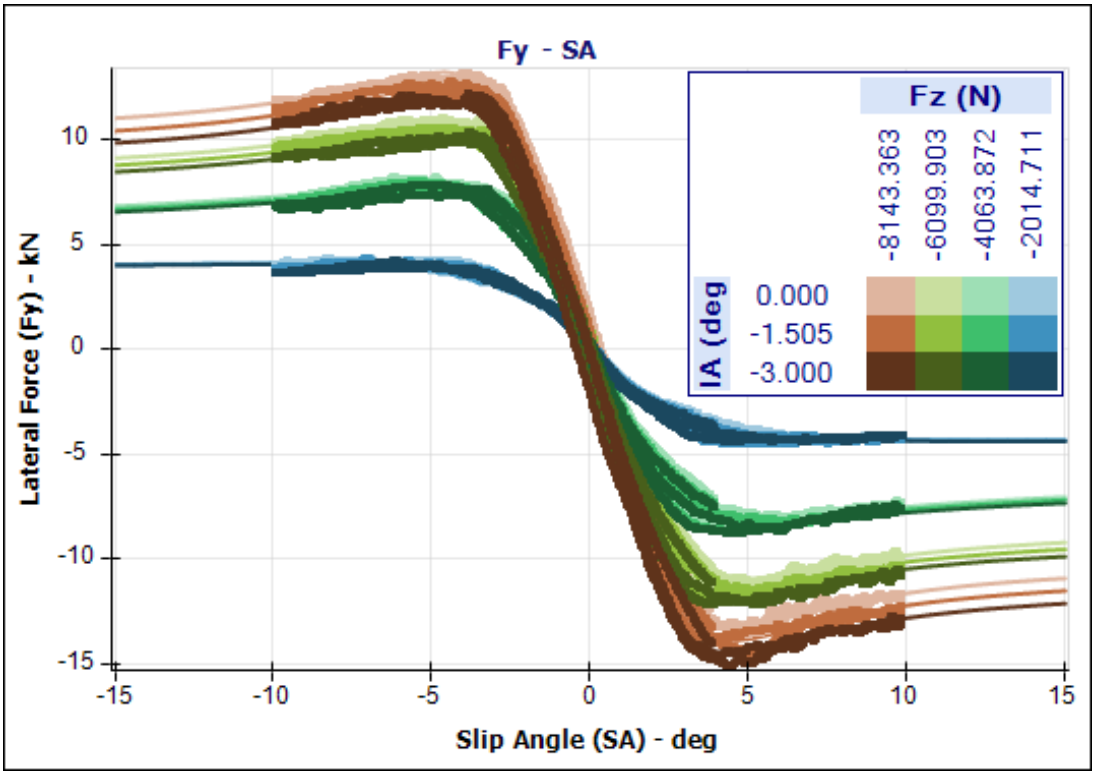
$$W = mg + F_d \tag{33}$$

where W is the vehicle total load, in newtons, m is the vehicle mass in kg and g is the gravity, in m/s^2. However, to determine the load transfer with more accuracy, the car's weight and downforce distributions are considered, shown in Equation (34):

$$W_F = m_F W + C_{\ell_{dist}} F_d \tag{34}$$

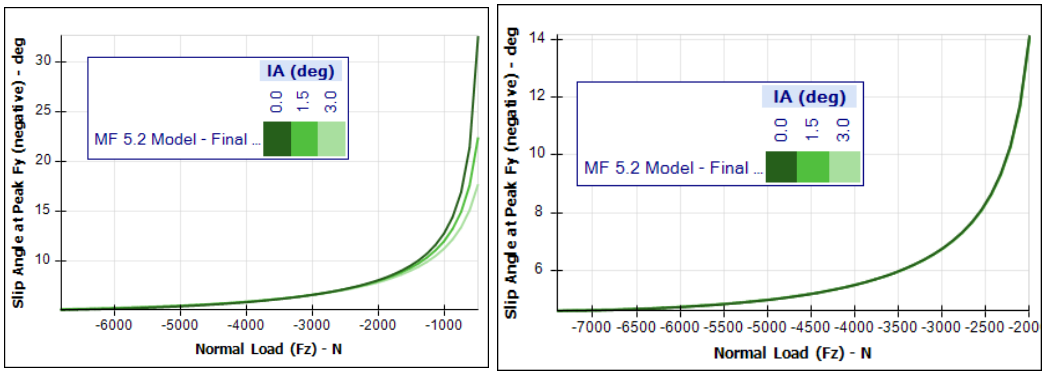
where W_F is the normal load on the front axle, in newtons, m_F and $C_{\ell_{dist}}$ are the weight and downforce distributions in percentage (1 is 100%), respectively. Similarly, for the rear axle:

Figure 63 – F_y vs α plot for the proposed GT3 rear tires (SAE coordinates).



Source: Author (2021)

Figure 64 – Peak slip angle α shift with relation to the normal load F_z.



(a) Front tire (b) Rear tire

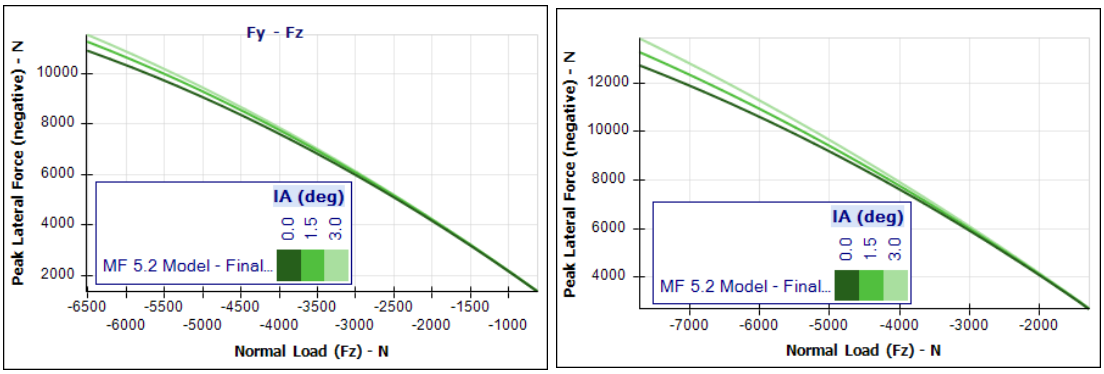
Source: Author (2021)

$$W_R = (1 - m_F)W + (1 - C_{\ell_{dist}})F_d \tag{35}$$

where W_R is the normal load on the rear axle. The total weight transfer can be simplified by collapsing the front and rear track widths into an average. The total weight transfer ΔW is given by:

$$\Delta W = \frac{W A_y h_{CG}}{\frac{t_F + t_R}{2}} \tag{36}$$

Figure 65 – Tire peak lateral force F_y with slip angle F_z variation for different values of inclination angle γ.



(a) Front tire (b) Rear tire

Source: Author (2021)

where ΔW is the total weight transfer, in newtons, A_y is the lateral acceleration, in m/s^2, h_{CG} is the CG height, in meters and t_F and t_R are the front and rear track widths, respectively. Moreover, the lateral weight transfer is not the same for the front and rear axles. According to William F. Milliken and Douglas L. Milliken (1995), many parameters can affect these values, such as anti-roll bars and tire pressure. To simplify this case study, a lateral weight transfer distribution (*LLTD*) value was considered, which already contains all the other terms. The *LLTD* is given by:

$$LLTD = \frac{\Delta W_F}{\Delta W} \tag{37}$$

where *LLTD* is the lateral load transfer distribution and ΔW_F is the weight transfer on the front axle, in newtons. By imposing a value to the *LLTD*, the load transfer on the front axle and rear axle, ΔW_F and ΔW_R respectively, are given by the equations:

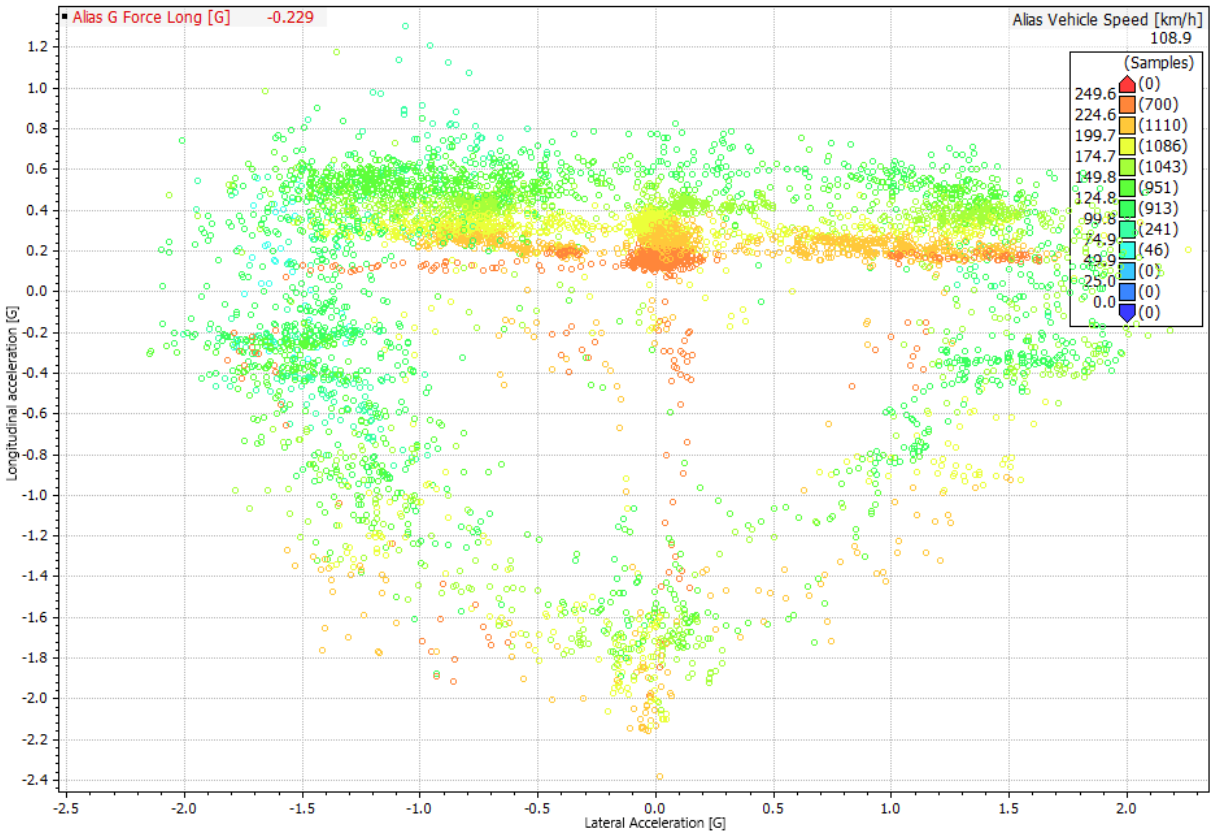
$$\Delta W_F = \Delta W LLTD \tag{38}$$

$$\Delta W_R = \Delta W(1 - LLTD) \tag{39}$$

An average speed and lateral load are assumed using historical data. The G-G diagram allows a more accurate estimation of speed and lateral acceleration to which the car will subject to. Figure 66 shows a G-G diagram of a GT3 race car during a flying lap in Sebring.

The plot shows a dense concentration on the lateral acceleration $A_y \approx 1.65(G)$, with a speed varying between 90 and 150 km/h. Therefore, this case study will assume a vehicle that is pulling 1.65G of lateral force at 120 km/h. The input data shown in Table 13 contains some specifications of the average GT3 race car. Applying the values

Figure 66 – G-G diagram of a GT3 race car for a full lap in Sebring.



Source: Author (2021)

from Table 13 to Equation (32) through (39), yields on the normal tire loads shown in Table 14.

Table 13 – Common parameters of a GT3 car

Input parameter	Variable	Input value	Unit
Total mass	m	1420	kg
Total Mass distribution (% Front)	m_F	46	%
Total mass CG height	h_{CG}	388	mm
Downforce coefficient multiplied by frontal area	$C_\ell A$	2.27	–
Downforce distribution (% Front)	$C_{\ell_{dist}}$	32.80	%
Air density	ρ_{air}	1.12	kg/m^3
Speed	V	120	km/h
Target lateral acceleration	A_y	1.65	G
Gravity	g	9.81	m/s^2
Lateral load transfer distribution (% Front)	$LLTD$	53.67	%

Source: Author (2021)

Table 14 – Weight transfer estimation for the proposed conditions.

	Tire vertical load (*N*)		
Axle	Left	Right	ΔW **(*N*)**
Front	649.5	6220.5	2785.4
Rear	1829.7	6639.6	2404.9

Source: Author (2021)

5.4 DESIGN SPACE

The design space is defined in terms of 3-dimensional geometric regions. As presented in subsection 3.6.2, box and sphere elements are used to define the regions inside which the resulting system's pickup points will be found. All the boundaries were defined as hard so that the search does not deviate from the intended search space.

The boundaries are presented in two groups for each axle: the inboard and outboard points. The system is considered symmetric through the optimization process, so only the left pickup points are shown in the next several tables, once the right equivalent has the same coordinates with a negative y-coordinate.

The inboard points are defined inside a box boundaries, which can be contemplated in Table 15 and Table 17 for the front and rear axles respectively. On the other hand, the outboard points are defined inside a sphere, given the wheel geometry. The parameters that describe the outboard points of the front and rear axles are shown in Table 16 and Table 17 respectively. All coordinates are given in millimeters.

The range defined by these bounds were assumed mostly by experience and considering the coordinates of the initial position, since there is no access to the CAD model nor packaging measures of the car being optimized. This can be seen as an advantage for the study once this empowers even more the possibilities that the GA can explore.

A special attention is required for the steering rack. As shown previously, the optimal position of the tie rod outer ball joint for a positive Ackerman value is behind the wheel and towards the center of the car. For packaging and load path purposes, the steering rack was positioned behind the king pin, providing a better use of the given design space.

5.5 GA CONFIGURATION

The chosen GA configuration is shown in Table 19. The choice of such parameters took advantage from the parametric analysis run on chapter 4. Given the large number of design variables and objective functions, a bigger population is needed, which allows the algorithm to explore more possibilities on the population initialization.

Table 15 – Front suspension box boundaries definition for case study.

Point name		**Limits** **Lower**	**Upper**
CHAS_tie_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -215.0 \\ 472.4 \\ 110.0 \end{bmatrix}$	$\begin{bmatrix} -105.0 \\ 472.4 \\ 310.0 \end{bmatrix}$
CHAS_lower_fore_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} 137.4 \\ 216.9 \\ 80.0 \end{bmatrix}$	$\begin{bmatrix} 157.4 \\ 516.9 \\ 210.0 \end{bmatrix}$
CHAS_lower_aft_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -269.3 \\ 215.5 \\ 86.6 \end{bmatrix}$	$\begin{bmatrix} -249.3 \\ 515.5 \\ 246.6 \end{bmatrix}$
CHAS_upper_fore_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} 82.2 \\ 373.2 \\ 152.5 \end{bmatrix}$	$\begin{bmatrix} 42.2 \\ 653.2 \\ 552.5 \end{bmatrix}$
CHAS_upper_aft_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -140.3 \\ 360.3 \\ 115.1 \end{bmatrix}$	$\begin{bmatrix} -120.3 \\ 660.3 \\ 465.1 \end{bmatrix}$

Source: Author (2021)

Moreover, since the study requires a single final solution, the weighting functions and scaling factors are applied to the objective functions. This allows the use of a steady state selection for replacement operator, which pulls the search in a given direction. The selection size of 200 individuals is arbitrary. In addition, the crossover and mutation operators were maintained the same as in chapter 4.

The selected settings allow the algorithm to perform a more diverse — and sparse — search through the generations, ultimately avoiding local minima traps. However, the ranked selection is computationally more expensive, because it compares all the objective functions of all the individuals in the population.

5.6 CONVERGENCE

This case study optimization process was run on a AWS EC2[1] c5a.8xlarge Ubuntu instance. The instance has a 2nd generation AMD EPYC 7002 processor with running frequencies up to 3.3GHz, 32 virtual CPUs and 64GB of available RAM. The setup used 32 working threads to evaluate the individuals. The total time to run the 4000 generations was 14 hours, 10 minutes and 20 seconds.

[1] `https://aws.amazon.com/ec2/`

Table 16 – Front suspension spherical boundaries definition for case study.

Point name		**Center**	**Radius**
UPRI_upper_I	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} 0.0 \\ 650.0 \\ 350.0 \end{bmatrix}$	150.0
UPRI_lower_I	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} 0.0 \\ 700.0 \\ 275.0 \end{bmatrix}$	150.
UPRI_tie_I	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -200.0 \\ 700.0 \\ 300.0 \end{bmatrix}$	100.

Source: Author (2021)

The evolutionary process took around 1000 generations to start showing convergence signals. However, the only stopping criteria imposed to the algorithm was the number of generations. Even though this case study used a higher number of generations, the algorithm kept finding better a individual every — roughly — 70 generations when the optimization was about to reach is termination condition. This is a good indication that the optimization should have been carried on for a longer period of time, which could result on an even better solution.

Figure 67 shows the convergence KPIs of this optimization. The selection pressure assumed values greater than 0.95 only after the 1000th generation, which indicates a very slow convergence rate. The slow convergence rate is a good parameter for this type of optimization, because it helps the GA to keep a higher diversity and avoid local minima trapping. However, it is computationally more costly, since it takes much longer to find an optimized configuration.

Interestingly, Figure 67 shows a drop of almost two orders of magnitude on the best individual. This proves the capability of the algorithm to explore the whole solution space. On the other hand, this drop can indicate a bad choice of scaling factors and weighting functions.

The main indicators of the optimization for the last generation are displayed in Table 20. The results show that the evolutionary process increases the solution quality by 92.59%, with relation to the initial population.

5.7 EVALUATION AND RESULTS

The suspension system which is optimized in this case study is evaluated in all possible movements, independently. This section separates the objective functions per

Table 17 – Rear suspension box boundaries definition for case study.

Point name		Limits Lower	Upper
CHAS_tie_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2770. \\ 235.0 \\ 100.0 \end{bmatrix}$	$\begin{bmatrix} -2920.0 \\ 485.0 \\ 300.0 \end{bmatrix}$
CHAS_lower_fore_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2474.3 \\ 237.8 \\ 94.0 \end{bmatrix}$	$\begin{bmatrix} -2454.3 \\ 537.8 \\ 294.5 \end{bmatrix}$
CHAS_lower_aft_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2850.0 \\ 280.0 \\ 75.0 \end{bmatrix}$	$\begin{bmatrix} -2830.0 \\ 480 \\ 175.0 \end{bmatrix}$
CHAS_upper_fore_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2450.0 \\ 345.0 \\ 240.0 \end{bmatrix}$	$\begin{bmatrix} -2400.0 \\ 495.0 \\ 440.0 \end{bmatrix}$
CHAS_upper_aft_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2810.0 \\ 340.0 \\ 250.0 \end{bmatrix}$	$\begin{bmatrix} -2760.0 \\ 590.50 \\ 500.0 \end{bmatrix}$

Source: Author (2021)

evaluation motion. They are: heave, roll, pitch and steering. The results are compared with the objectives as they are described. An overview of the objective functions of the resulting individual is shown in Figure 68.

The slices' size of the inner circle in Figure 68 represent the contribution of each objective on the overall fitness of the resulting individual, while the outer circle represents the evaluation function — motion — at which they are evaluated. This chart highlights the variance of the objectives, for example the front roll center variation in row and mechanical trail in steering have the biggest shares between all the objectives. Table 21 shows the chart in numbers and the scaling factors attributed to each objective.

The table clearly shows that some objectives are favored in detriment to others. Moreover, it shows the immense difference between some objective fitness values, as the one seen between the camber angle variation in roll on the rear and the roll center height variation in heave on the front. This indicates either that some objectives are unfeasible for a given design space or the algorithm preferred one over another.

A possible solution for this behavior is to change the weighting functions and scaling factors targeting a normalization between all the objectives. However, the sensitivity of each objective to the design variables variance could not be normalized in this work.

The normalization of the objective functions would be beneficial to the definition

Table 18 – Rear suspension spherical boundaries definition for case study.

Point name		Center	Radius
UPRI_upper_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2630.0 \\ 750.0 \\ 400.0 \end{bmatrix}$	150.0
UPRI_lower_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2630.0 \\ 800.0 \\ 230.0 \end{bmatrix}$	100.
UPRI_tie_l	$\begin{bmatrix} x^P \\ y^P \\ z^P \end{bmatrix}$	$\begin{bmatrix} -2780.0 \\ 850.0 \\ 250.0 \end{bmatrix}$	100.

Source: Author (2021)

Table 19 – GA setup used in case study.

Parameter	Value
Population size	400 individuals
Number of generations	4000
Selection for reproduction	Ranked
Selection Size	1 rank
Selection for replacement	Steady State
Selection size	200 individuals
Crossover type	Voluminal
α_{BLX}	2
Mutation rate	5%
Mutation type	Gaussian
σ_g	2

Source: Author (2021)

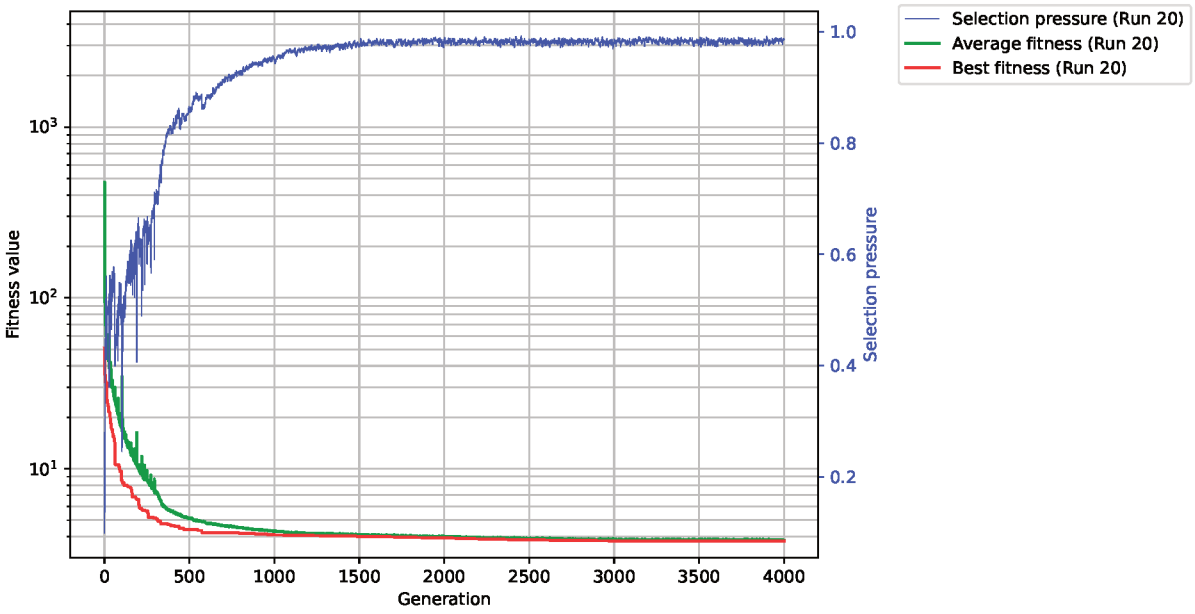
of weights and scaling factors. Still, the numbers are not easily defined, once the optimization compares angles to lengths in different orders of magnitude and importance. The normalization is a complex subject that should be handled in future versions of the GA.

5.7.1 Heave

The heave range is determined from the baseline range. The range used was -50 mm to +50mm from the static position, determined by the initial coordinates of the system. The objective functions for this motion are shown in Figure 69 and commented below.

(a) Kinematic Roll Center Z at Ground [Front]: shown in Figure 69a, the baseline has

Figure 67 – Convergence graphs of case study.



Source: Author (2021)

Table 20 – Final indicators of the optimization run for the case study.

	Generation	Best fitness	Average fitness	Fitness standard deviation	Selection pressure
0	0.000	50.849	413.053	2196.530	0.123
4000	4000.000	3.766	3.820	0.219	0.986
Difference		**47.083**	**409.233**	**2196.311**	**-0.863**
Drop (%)		**92.593**	**99.075**	**99.990**	**-700.858**

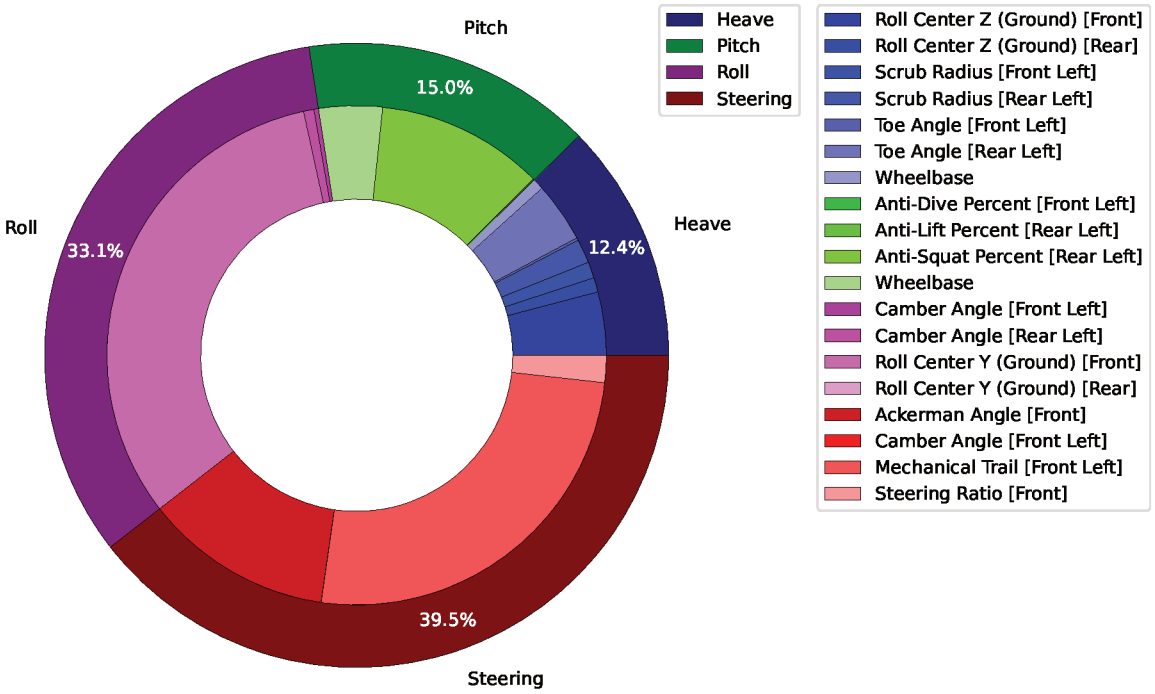
Source: Author (2021)

a static value of about 35mm. The objective is set to 55mm, which would yield on a more aggressive car on tight tracks. Ideally, the roll center height should not become negative, but as the plot shows, it happens for the baseline, while the optimized system keeps this value always positive.

(b) Kinematic Roll Center Z at Ground [Rear] : the baseline has a static roll center height of 55mm and the optimized should reach 85mm. Figure 69b shows that the optimized curve comes really close to the target. It was also desired that for every motion the front axle roll center height is never higher than the rear. This later is satisfied if Figure 69a and Figure 69b are compared.

(c) Scrub radius [Front]: the scrub radius is responsible for generating moments around the steering axis when longitudinal forces — such as braking or throttling — are applied at the tires. In reality, having a scrub radius equal to zero in the front

Figure 68 – Final result comparison of objectives.



Source: Author (2021)

axle could be detrimental to the driver's perception in case that the longitudinal force of the left and right tires are different, for example in a case of a puncture. This parameter cannot be determined precisely without a dynamic analysis. Thus, the scrub radius of 10mm was chosen arbitrarily in comparison to the initial value of more than 15mm and its variation in heave is also minimized. Figure 69c show that the algorithm found an excellent compromise for these objectives. The weight on both functions are decreasing as they move away from the initial position because the vehicle is operated most commonly around the initial position.

(d) Scrub radius [Rear]: on the same manner as the front scrub radius, the rear scrub radius should be minimized to reduce toe change due to acceleration or braking torques. Since the rear system does not have a steering mechanism that provides force feedback to the driver, this parameter is set to be minimized. Figure 69d shows that a good compromise was achieved for the rear scrub radius too.

(e) Toe angle [Front]: known as bump steer, this parameter should be minimized in most cases. Even though they can improve the vehicle's performance if chosen wisely, this effect is usually undesired. If a vehicle has too much bump steer, it could lead to tire overheating and stability issues on wavy roads. Figure 69e shows

Table 21 – Final result objectives fitness

Evaluation: Objective	Scaling factor	Fitness value	Share in overall fitness
Heave: Roll Center Z (Ground) [Front]	0.1	0.002	0.062 %
Heave: Roll Center Z (Ground) [Rear]	0.1	0.007	0.189 %
Heave: Scrub Radius [Front Left]	0.2	0.015	0.405 %
Heave: Scrub Radius [Rear Left]	0.1	0.001	0.029 %
Heave: Toe Angle [Front Left]	200.0	0.040	1.070 %
Heave: Toe Angle [Rear Left]	200.0	0.007	0.179 %
Heave: Wheelbase	0.1	0.089	2.357 %
Pitch: Anti-Dive Percent [Front Left]	0.1	0.059	1.573 %
Pitch: Anti-Lift Percent [Rear Left]	0.1	0.030	0.785 %
Pitch: Anti-Squat Percent [Rear Left]	0.1	0.123	3.266 %
Pitch: Wheelbase	0.1	0.011	0.287 %
Roll: Camber Angle [Front Left]	100.0	0.062	1.637 %
Roll: Camber Angle [Rear Left]	100.0	1.675	44.483 %
Roll: Roll Center Y (Ground) [Front]	0.5	0.000	0.001 %
Roll: Roll Center Y (Ground) [Rear]	0.5	0.000	0.001 %
Steering: Ackerman Angle [Front]	100.0	0.635	16.871 %
Steering: Camber Angle [Front Left]	10.0	0.033	0.876 %
Steering: Mechanical Trail [Front Left]	0.1	0.878	23.321 %
Steering: Steering Ratio [Front]	1.0	0.098	2.610 %

Source: Author (2021)

that this objective was successfully minimized when compared to the original system.

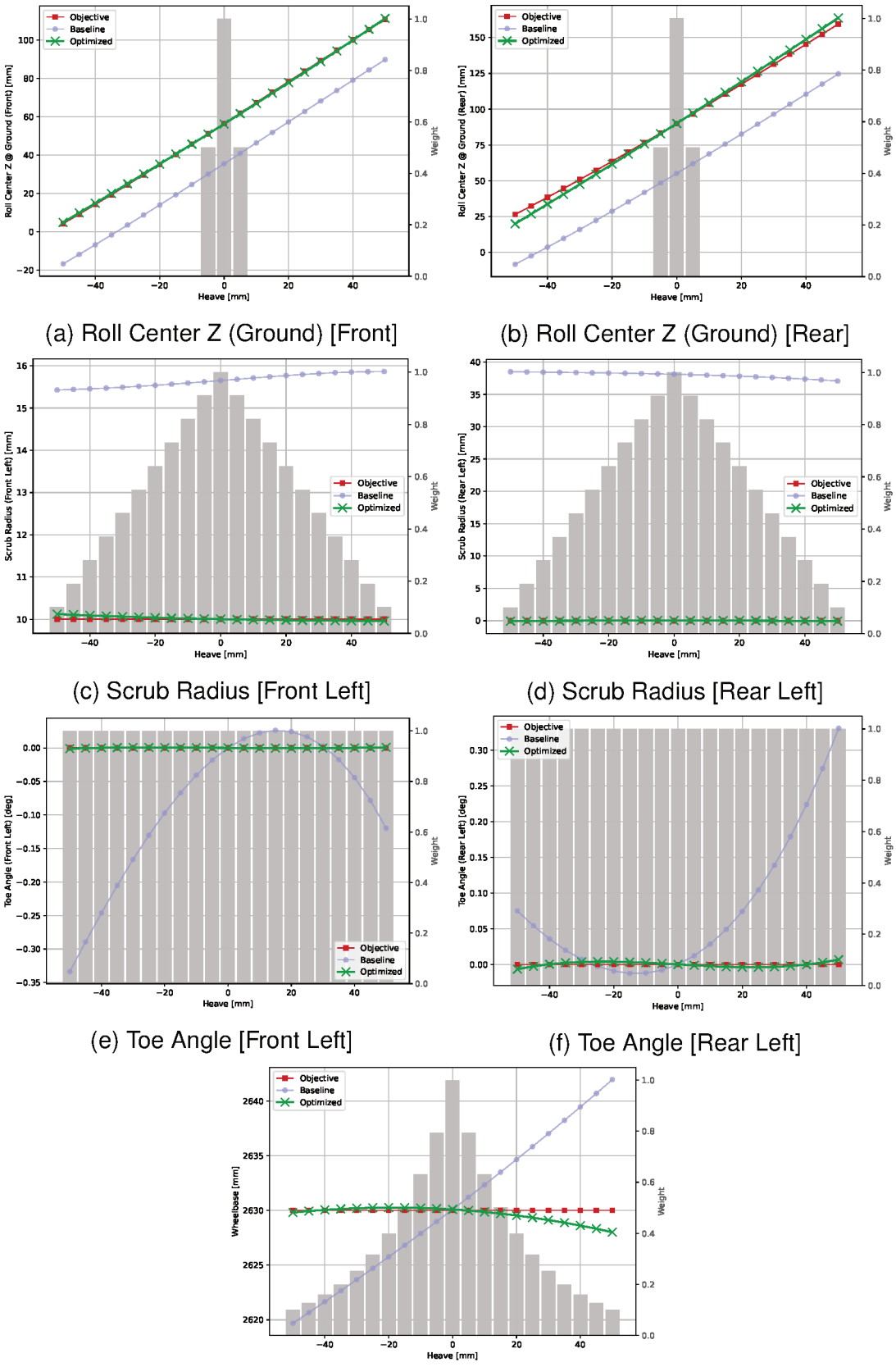
(f) Toe angle [Rear]: the bump steer outputs on the rear axle is illustrated by Figure 69f. The plot shows that the optimized function satisfied the objective on a better manner than in the front axle. This is due to the broader space that the vehicle has on the rear search space.

(g) Wheelbase: the wheelbase change was chosen to be minimized to avoid too much change on the pitch moments when the car is subject to longitudinal accelerations. This parameter is not as sensitive as anti-dive, for example. The optimized suspension system achieved a function which is much closer to the objective, as shown in Figure 69g.

5.7.2 Roll

The roll range is determined by the roll gradient and an an arbitrary lateral acceleration that is used as an upper bound. A constant roll gradient of $0.58^{\circ}/G$ is assumed — taken as an average of the historical roll gradient data for front and rear

Figure 69 – Heave motion evaluation objectives.



(a) Roll Center Z (Ground) [Front]

(b) Roll Center Z (Ground) [Rear]

(c) Scrub Radius [Front Left]

(d) Scrub Radius [Rear Left]

(e) Toe Angle [Front Left]

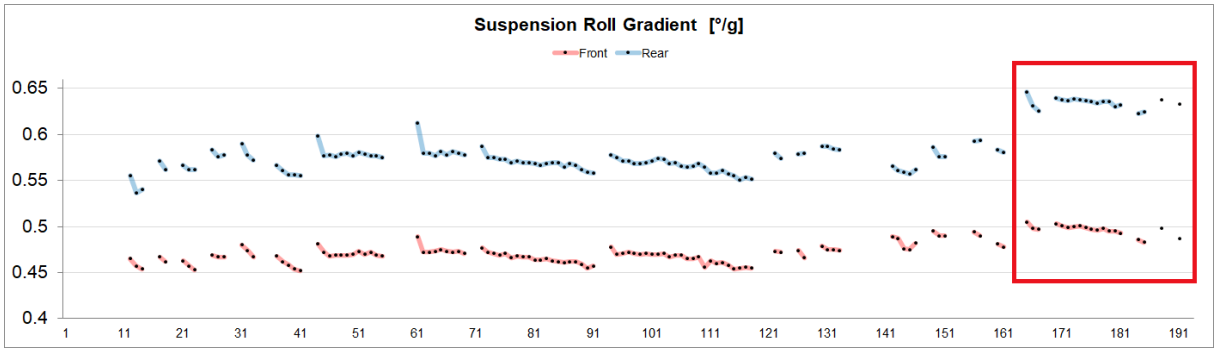
(f) Toe Angle [Rear Left]

(g) Wheelbase

Source: Author (2021)

axles shown in Figure 70, from the outing that is highlighted by the red box — and a supposed maximum lateral acceleration of 2.3G. The resulting roll angle is 1.33°, which is rounded to 1.5°. Thus, the roll motion ranges from ±1.5°, with its axis defined at the ground level and at $Y = 0$.

Figure 70 – Vehicle suspension roll gradient for several laps.



Source: Author (2021)

The data displayed in Figure 70 is an average roll gradient for each lap, which is obtained by the potentiometers of the suspension and translated as a roll movement assuming constant motion ratios and track widths. The objective functions for the roll motion are shown in Figure 71 and commented below.

(a) Camber angle [Front Left]: the objective for the camber variation in roll is to keep it where it generates the most lateral force. However, if the camber variation in roll is positive — that is, a negative camber on the left wheel for negative values of roll angle — it can generate too much scrub and uneven tire temperature in heave. Furthermore it could be detrimental for longitudinal tire force, when braking or accelerating. The tire model being used in this case study points that a camber of –3° is the one that generates the most lateral force. On the other hand, the car has quite an amount of downforce, thus, in fast corners, the vertical position of the car will be lower, inducing even more camber, resultant from the heave motion. In addition, the steering mechanism also produces a camber variation to the car when steered, which is shown later in this section. Therefore, for simplification purposes, this value is set to zero, as shown in Figure 71a. The optimized system is able to generate negative camber for the outside wheel when the car is under roll. This behavior is sub-optimal given the objective function, but definitely acceptable given the tire model presented previously.

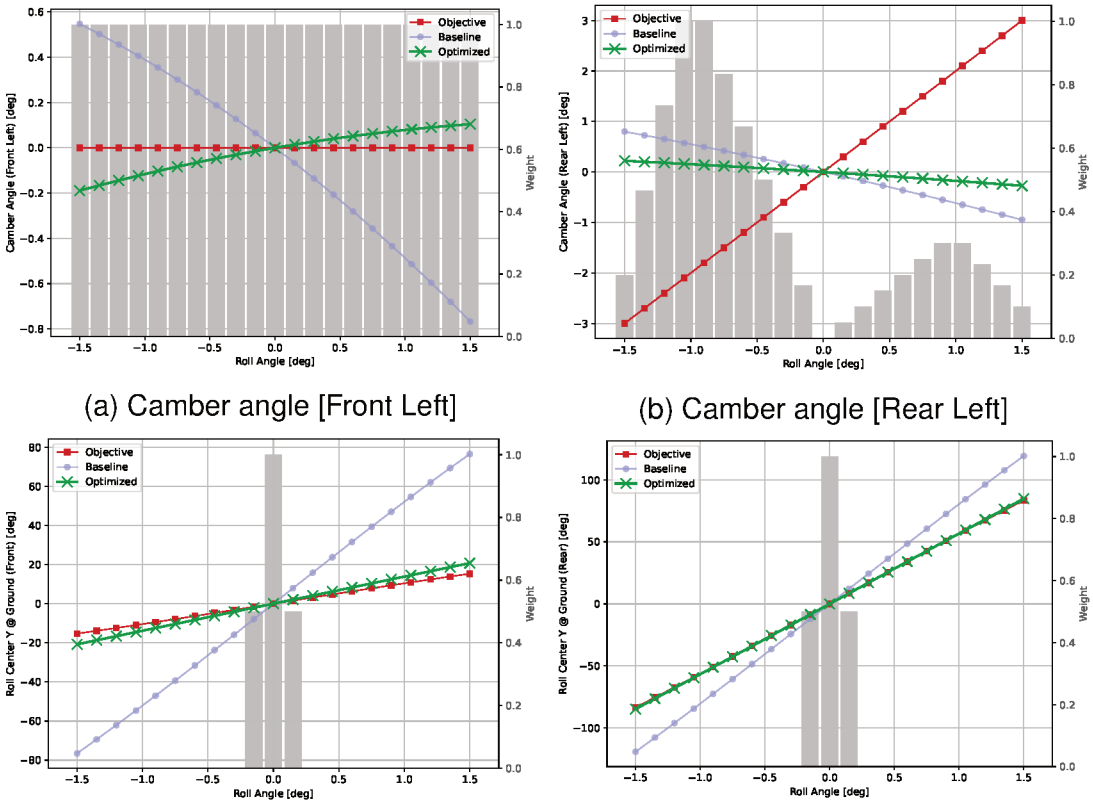
(b) Camber angle [Rear Left]: the rear axle does not rely on a steering mechanism, thus, the camber variation in roll is wished to induce a negative camber on the outside wheel. On the other hand, the rear axle is more sensitive to downforce, once its distribution is biased towards the rear. The objective and optimized functions

are shown in Figure 71b.The weight function was built in a way that it prioritizes the mostly used roll angles, obtained from Figure 70 and Figure 66. The result does not come close to the objective, but it is still better than the baseline.

(c) Roll center Y (Ground) [Front]: from experience, the roll center variation in the Y direction is a parameter that is desired to be kept as low as possible. However, it is a really difficult objective to achieve. Thus, the desired target is that the variation of the roll center Y coordinate in the front axle should be less than in the rear axle. This objective was improved well by the optimization, as shown in Figure 71c.

(d) Roll center Y (Ground) [Rear]: similarly to the front axle, the rear roll center Y variation in roll of the optimized system is very close to the objective, as demonstrated in Figure 71d.

Figure 71 – Roll motion evaluation objectives.



(a) Camber angle [Front Left]

(b) Camber angle [Rear Left]

(c) Roll Center Y (Ground) [Front]

(d) Roll Center Y (Ground) [Rear]

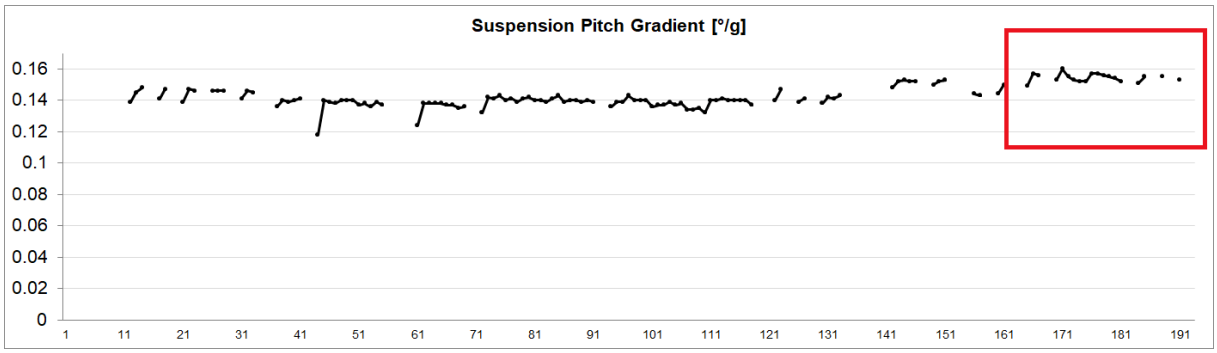
Source: Author (2021)

5.7.3 Pitch

Similarly to the roll motion, the pitch motion is defined with an assumed pitch gradient, taken from the same outing as the data used for roll, indicated by the red box

in Figure 72. Even assuming a 2.2G of deceleration, as Figure 66 indicates, the result pitch is less than 1°. Thus the pitch range is arbitrarily assumed to be ±1°.

Figure 72 – Vehicle suspension pitch gradient for several laps.



Source: Author (2021)

The pitch motion is most strongly influenced by longitudinal accelerations, as mentioned previously. Thus, the objectives for this motion are primarily the anti-features geometries. Figure 73 shows the objectives of this motion in comparison with the baseline and optimized suspension systems. The objectives are commented below.

(a) Anti-Dive % [Front]: despite that the initial system has an anti-dive greater than 100%, this parameters is set to a lower value. The value of 30% for the static anti-dive objective was picked from track experience, while its variation is reduced through the pitch motion. As Figure 73a shows, the optimized system reaches the objective accordingly, except for the variation.

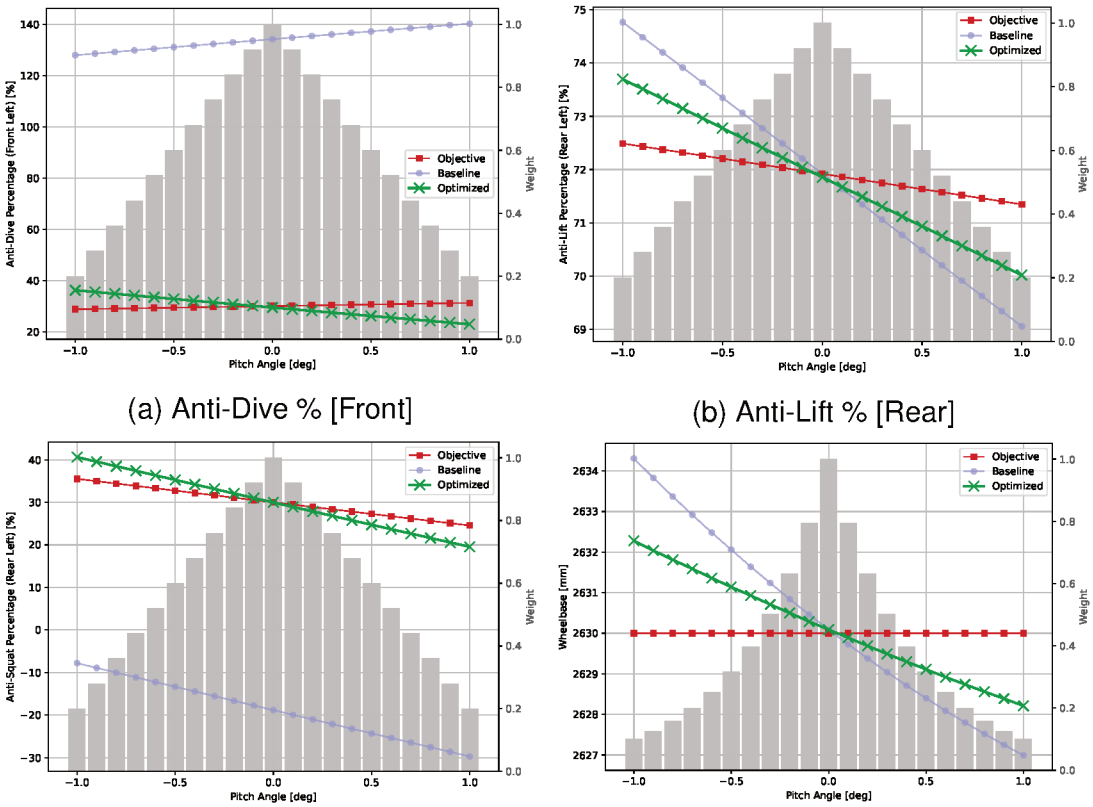
(b) Anti-Lift % [Rear]: Similarly to the anti-dive objective, the anti-lift on the rear axle was targeted to be around the same value as the baseline system for static conditions, while reducing its variation in pitch motion. Figure 73b demonstrates that the optimized curve reached the target for static position in detriment of its variation. Still, the variation is better than the baseline.

(c) Anti-Squat % [Rear]: on contrary to the initial system, the anti-squat percentage was changed by a great amount. Since the initial system had a negative value for anti-squat — thus, loading the springs even more in acceleration — the objective was set to be around 30% in static condition and its variation to be reduced in pitch. Figure 73c shows that the optimization was able to overcome such difference, providing an improved individual with relation to this objective function.

(d) Wheelbase: following the same thought presented for the heave motion, the wheelbase variation is wished to be as minimum as possible, mainly on longitudinal weight transfer situations. Thus, this parameter is set to a minimum variation

in pitch. Figure 73d illustrates that the optimized curve is not too close to the objective, but the variation is yet lower than the baseline system

Figure 73 – Pitch motion evaluation objectives.



(a) Anti-Dive % [Front]

(b) Anti-Lift % [Rear]

(c) Anti-Squat % [Rear]

(d) Wheelbase

Source: Author (2021)

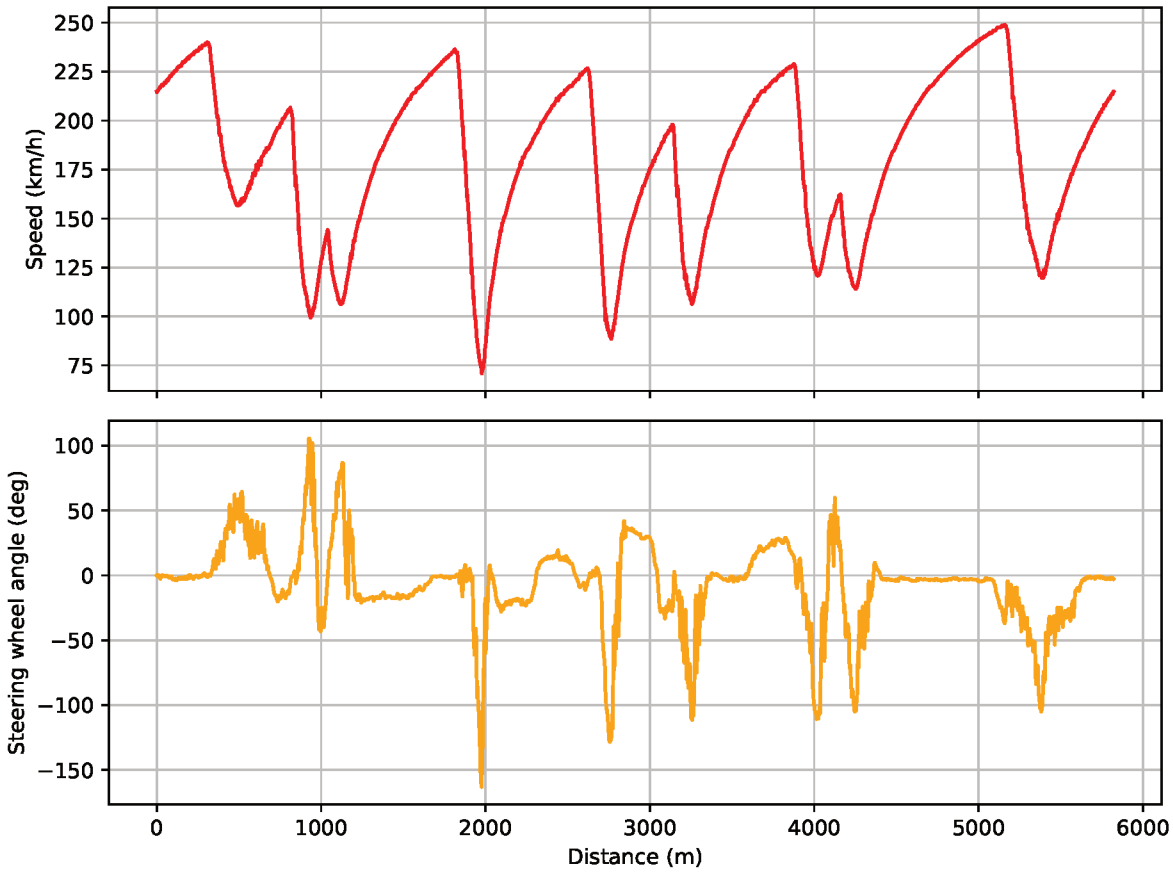
5.7.4 Steering

The steering motion was defined as the limit imposed by the steering rack of the car. Figure 74 shows the steering angle channel of a GT3 car on a flying lap in Sebring. Even though the graph shows that the steering wheel angle does not exceed 160°, the maximum steering wheel angle of 270° is assumed. However, Figure 74 illustrates the range of steering motion that is more relevant to the kinematics of the system.

The steering objectives and optimized curves are shown in Figure 75. The choice of objectives are introduced below.

(a) Ackerman Angle [Front]: the Ackerman angle is chosen based on the tires behavior. As introduced in section 5.2, the slip angle α in peak F_y shift is used to determine this objective. As estimated in Table 14, the loads of the inner and outer tires on the front axle are approximately 650N and 6220N. In accordance with Figure 64a, the optimal Ackerman should be greater than 10 degrees for the

Figure 74 – Vehicle speed and steering wheel angle channels of a GT3 car on a flying lap in Sebring.



Source: Author (2021)

average steering angle on a lap, at a given circuit. However, such number could be physically infeasible. The feasibility would not be achieved because the tie rod outer ball joint position would need to be carried over towards the outside of the car with relation to the virtual king pin. Thus, a pro-Ackerman angle δ_{Ack} of +6° at the maximum and minimum rack travel is chosen as the objective. The curve of this output along the steering wheel angle is shown in Figure 75a. As the graph suggests, the optimized curve is not perfectly drawn over the objective because the later was defined by hand.

(b) Camber angle [Front Left]: the camber angle in steering can potentially cover the losses imposed by the roll movement limitations. It is desired that the outside wheel of the corner produces a negative camber while the inside wheel induces a positive camber, so to maximize the tires' F_y. In this case study, the baseline already had a good amount of camber gain in steering, thus, a very subtle adjustment was proposed. The optimized system was not improved with regard to the baseline for the outside wheel (negative steering wheel angle), as shown in

Figure 75b. However, since the camber gain in roll was improved, this optimized parameter is plausible.

(c) Mechanical trail [Front Left]: the mechanical trail is mostly associated with the load transferred to the tie rods and furthermore to the steering rack, altering the steering feedback to the driver. This value is not easily determined without a force analysis on the system, once it influences not only on the steering torque but also on the driver's perception. Hence, this parameter was kept the same in static position, but its variation was set to be reduced. Figure 75c shows that the resulting system has a worse behavior with relation to the initial system. This discrepancy is related to the compromise between this and other objective functions.

(d) Steering ratio [Front]: in the same manner as the previous item, the steering ratio was set to be maintained from the baseline, however, this objective is obviously worse than the initial system's, once the initial system is the target. The result for this objective is shown in Figure 75d. Moreover, the optimized system indicates that the result yields a progressive steering ratio with relation to the steering wheel angle, which can be advantageous on some cases where the tires' M_Z are inverted for high slip angle values.

5.8 RESULTING SYSTEM

Despite that the GA does not need an initial guess solution, the optimized system is compared to the baseline in this work. The optimization yielded an interesting solution with non-intuitive peculiarities, demonstrated in this section. The resulting set of pickup points and their absolute variation between the optimized and the initial systems is shown in Table 22.

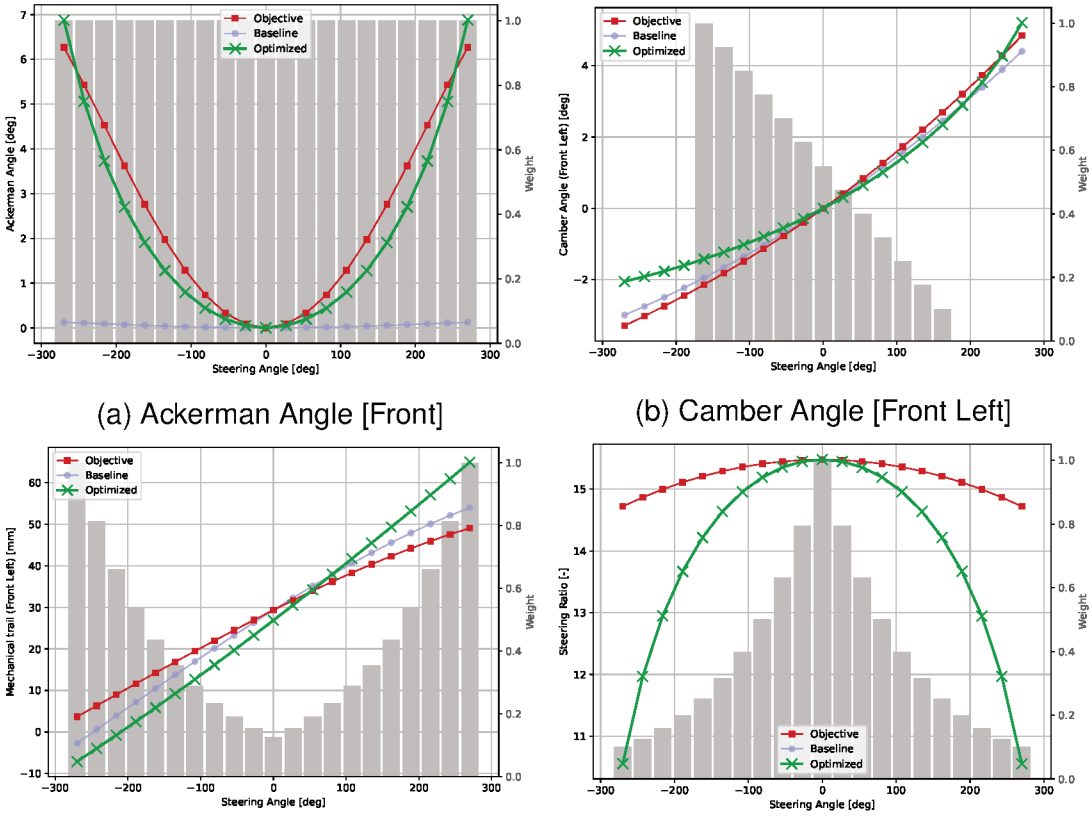
Moreover, the systems are visually compared in Figures 76, 77 and 78. As illustrated, the optimization algorithm was able to explore different regions of the design space, usually ignored by a human. The resulting system is also feasible in terms of manufacturing. The packaging restrictions need to be further investigated.

It is shown by Figure 77b that the outer ball joints of the upper and lower wishbones are really close to the rim limits, if not exceeding them. Thus, another optimization could be run with a narrower design space, once the main variables are known.

Figure 78 shows that the optimized system has crooked wishbone axes on the top view. This is a great proof that the optimization can search for different topologies which are very unlikely to be designed by humans. Also, this case study does not comprise the actuation system optimization.

In addition, a force analysis should be run in both systems to investigate the load paths. The steering system position on the optimized system is concerning, once

Figure 75 – Steering motion evaluation objectives.



(a) Ackerman Angle [Front]

(b) Camber Angle [Front Left]

(c) Mechanical Trail [Front Left]

(d) Steering Ratio [Front]

Source: Author (2021)

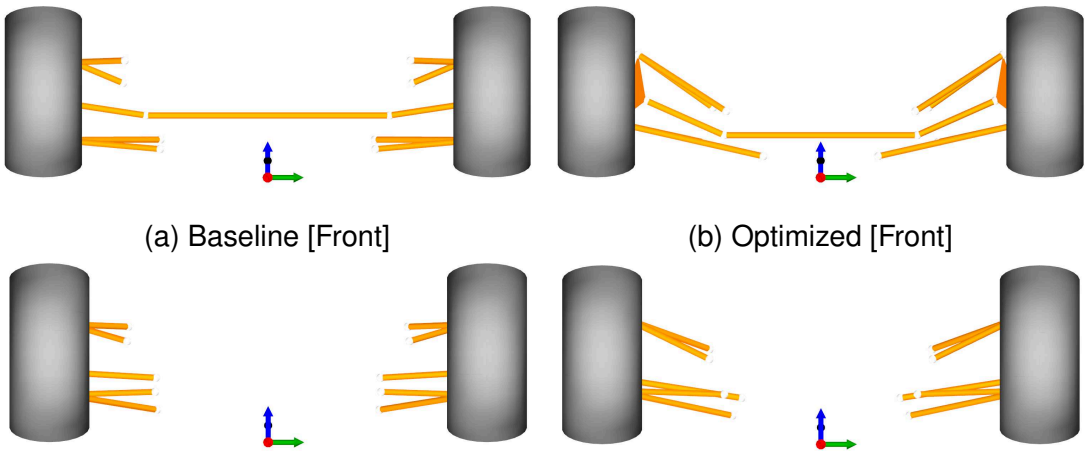
it could generate bending moments on the steering rack and furthermore generate undesired mechanical issues.

Table 22 – Optimized system pickup points.

Point name	**X**	**Y**	**Z**	**ΔX**	**ΔY**	**ΔZ**	Units
Front suspension							
CHAS_lower_aft_l	-250.9	455.3	136.9	-1.6	39.8	-9.8	mm
CHAS_lower_fore_l	143.3	222.9	82.3	5.8	-194.0	-27.7	mm
CHAS_tie_l	-195.9	374.2	163.6	-260.9	-98.2	-76.4	mm
CHAS_upper_aft_l	-120.3	417.2	270.4	-0.0	-143.1	-94.7	mm
CHAS_upper_fore_l	100.5	366.3	256.3	18.3	-186.9	-196.3	mm
UPRI_lower_l	-11.8	790.9	210.5	-9.8	-29.2	65.1	mm
UPRI_tie_l	-170.5	676.5	297.6	-281.4	-110.4	9.7	mm
UPRI_upper_l	-60.4	705.9	475.4	6.2	-46.6	29.7	mm
Rear suspension							
CHAS_lower_aft_l	-2837.6	343.1	110.8	-10.3	-88.3	-14.8	mm
CHAS_lower_fore_l	-2443.7	374.6	193.0	10.6	-63.2	-1.6	mm
CHAS_tie_l	-2720.0	308.0	178.9	96.5	-127.0	-70.0	mm
CHAS_upper_aft_l	-2809.8	431.2	367.1	-48.3	-110.3	-78.5	mm
CHAS_upper_fore_l	-2427.8	432.5	328.0	23.9	-114.0	-61.4	mm
UPRI_lower_l	-2598.0	803.9	207.6	-24.9	11.3	24.3	mm
UPRI_tie_l	-2688.0	714.1	242.3	92.0	-96.0	-28.6	mm
UPRI_upper_l	-2556.7	747.2	480.9	35.7	-21.4	26.8	mm

Source: Author (2021)

Figure 76 – Suspension optimization comparison (Front View).

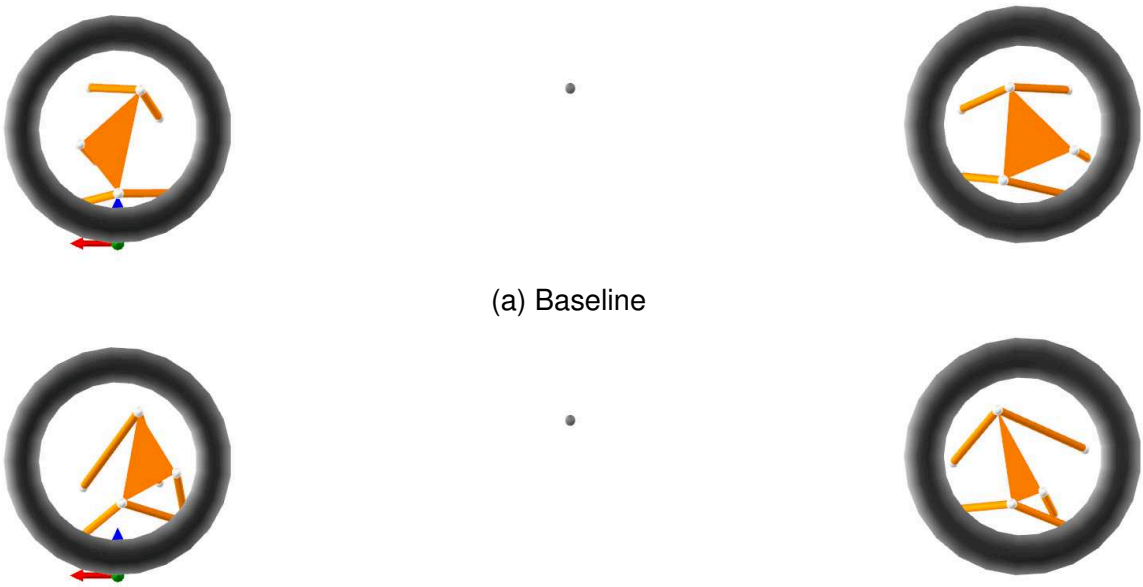


(a) Baseline [Front] (b) Optimized [Front]

(c) Baseline [Rear] (d) Optimized [Rear]

Source: Author (2021)

Figure 77 – Suspension optimization comparison (Side View).

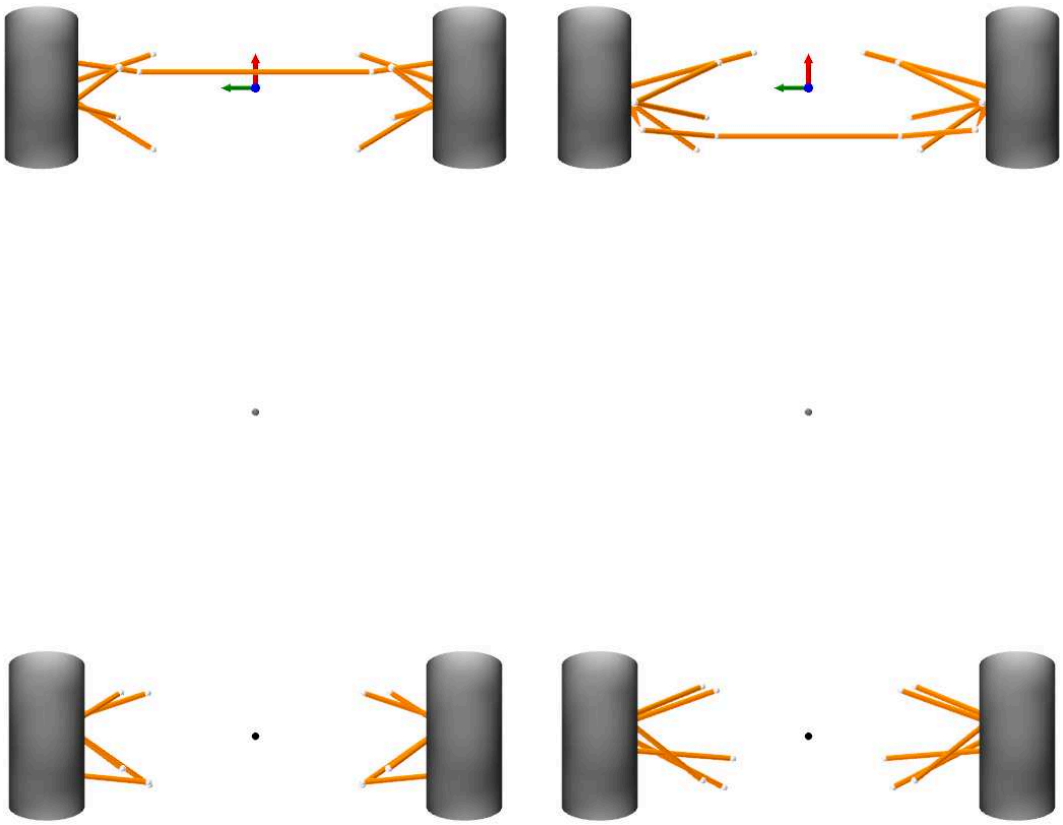


(a) Baseline

(b) Optimized

Source: Author (2021)

Figure 78 – Suspension optimization comparison (Top View).



(a) Baseline (b) Optimized

Source: Author (2021)

6 CONCLUSIONS

This work shows that suspension kinematics design and optimization is an exceptionally complex task and therefore demands too much time and resources. Furthermore, the multi-objective nature of this type of problem can lead to an overwhelming design process if done manually, as the objectives are sometimes controversial.

The optimization process using GA proposed in this work is an exceptional shortcut for the design process of a new suspension mechanism or an improved version of an existing one. Even though the optimization can be computationally demanding depending on the number of objectives and how broad is the design space, it can reduce weeks of workload to a few hours of computational time.

Despite that the GA proved to accelerate the iterative design process, its setup can be staggering for novice suspension engineers who need to deal with multiple objectives that have different scaling and weighting functions. Moreover, the user needs to know not only the kinematics weights and scaling factors, but also its limitations and the project's packaging restrictions in order to perform a successful optimization. The automatic determination of weight functions and scaling factors is a suggestion for future work. Removing this responsibility from the user can after all provide a more efficient optimization setup and lead to better results.

The stochastic nature of the EMOO makes it a powerful tool to overcome local minima that may happen along the evolution process. On the other hand, it can yield different results for the same set of objectives and design variables. Thus, the user can run the same setup optimization several times and possibly gather different results.

In addition, the GA has a quite complicated setup process that influences heavily on the results quality and search efficiency. An automated setup process based on the design variables and objectives is suggested for future work. Likewise, different design space geometries, such as cylinder, ellipse and 3D CAD models are also suggested to define the design space along with exclusion regions.

A penalty system can also be implemented in future versions of the GA, allowing the objectives and design variables to be described in more detail. Different crossover and mutation operators could be exploited too as a solution to the computational time.

Ultimately, the same process can be extended to dynamic analyses, once it requires very few adaptations. In order to achieve the dynamic analysis — as for example a virtual 7-post rig — a multi-body solver implementation is suggested.

REFERENCES

ABRAHAM, Ajith; GOLDBERG, Robert. **Evolutionary Multiobjective Optimization: Theoretical Advances and Applications**. London, United Kingdom: Springer London, 2006. (Advanced Information and Knowledge Processing). ISBN 9781846281372.

AFKAR, Amir; MAHMOODI-KALEIBAR, Mehdi; PAYKANI, Amin. Geometry optimization of double wishbone suspension system via genetic algorithm for handling improvement. **Journal of Vibroengineering**, v. 14, p. 827–837, June 2012.

ARIKERE, Aditya; SARAVANA KUMAR, Gurunathan; BANDYOPADHYAY, Sandipan. Optimisation of Double Wishbone Suspension System Using Multi-Objective Genetic Algorithm. In: DEB, Kalyanmoy et al. (Eds.). **Simulated Evolution and Learning**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. P. 445–454.

BOYD, Stephen P.; VANDENBERGHE, Lieven. **Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares**. [S.l.]: Cambridge University Press, 2018. DOI: 10.1017/9781108583664.

CHENG, Xianfu; LIN, Yuqun. Multiobjective Robust Design of the Double Wishbone Suspension System Based on Particle Swarm Optimization. **TheScientificWorldJournal**, v. 2014, p. 354857, Feb. 2014. DOI: 10.1155/2014/354857.

DEBIAN. **The Computer Language Benchmarks Game**. 2021. Available from: https://benchmarksgame-team.pages.debian.net/benchmarksgame/index.html. Visited on: 19 Mar. 2021.

DIXON, John C. **Suspension Geometry and Computation**. Chichester, United Kingdom: Wiley, 2009. ISBN 9780470682890.

DRÉO, Johann; PÉTROWSKI, Alain; SIARRY, Patrick; TAILLARD, Eric. **Metaheuristics for Hard Optimization: Methods and Case Studies**. Berlin, Germany: Springer Berlin Heidelberg, 2006. ISBN 9783540309666.

FLORES, Paulo. **Concepts and Formulations for Spatial Multibody Dynamics**. [S.l.]: Springer International Publishing AG Switzerland, 2015. ISBN 9783319161907. DOI: 0.1007/978-3-319-16190-7.

GENTA, Giancarlo; MORELLO, Lorenzo. **The Automotive Chassis: Volume 1: Components Design**. [S.l.]: Springer Netherlands, 2008. (Mechanical Engineering Series). ISBN 9781402086762.

GUPTA, Rajesh K. **Numerical Methods: Fundamentals and Applications**. [S.l.]: Cambridge University Press, 2019. DOI: 10.1017/9781108685306.

ISO/IEC 14882:2020: Programming languages — C++. Standard. Geneva, CH, Dec. 2020.

JAZAR, Reza N. **Vehicle Dynamics: Theory and Application**. Bundoora, Australia: Springer New York, 2013. (SpringerLink : Bücher). ISBN 9781461485445.

MASTINU, Giampiero; GOBBI, Massimiliano; MIANO, Carlo. **Optimal Design of Complex Mechanical Systems: With Applications to Vehicle Engineering**. Heidelberg, Germany: Springer Berlin Heidelberg, 2007. ISBN 9783540343554.

MCDOUGALL, Trevor J.; WOTHERSPOON, Simon J. A simple modification of Newton's method to achieve convergence of order 1+2. **Applied Mathematics Letters**, v. 29, p. 20–25, 2014. ISSN 0893-9659. DOI: https://doi.org/10.1016/j.aml.2013.10.008.

MIETTINEN, Kaisa. Some Methods for Nonlinear Multi-objective Optimization. In: EVOLUTIONARY Multi-Criterion Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. P. 1–20.

MILLIKEN, William F.; MILLIKEN, Douglas L. **Race Car Vehicle Dynamics**. [S.l.]: SAE International, 1995. (Premiere Series). ISBN 9781560915263.

MOHAN, Ganesh; ROUELLE, Claude; HUGON, Emmanuel. A New Method to Evaluate Bump Steer and Steering Influence on Kinematic Roll and Pitch Axes for All Independent Suspension Types. In: SAE Technical Paper. [S.l.]: SAE International, Dec. 2008. DOI: 10.4271/2008-01-2949.

NIKRAVESH, Parviz E. **Computer-Aided Analysis of Mechanical Systems**. USA: Prentice-Hall, Inc., 1988. ISBN 0131642200.

OPTIMUMG. **Advanced Vehicle Dynamics Seminar**. [S.l.: s.n.], 2020.

OPTIMUMG. **OptimumKinematics Manual**. Denver, CO, USA, 2019.

PACEJKA, Hans B. **Tyre and Vehicle Dynamics**. Burlington, United Kingdom: Butterworth-Heinemann, 2006. (Automotive engineering). ISBN 9780750669184.

PAPAIOANNOU, Georgios; GAUCI, Christian; VELENIS, Efstathios; KOULOCHERIS, Dimitrios. Sensitivity Analysis of Vehicle Handling and Ride Comfort with Respect to Roll Centers Height. In: ADVANCES in Dynamics of Vehicles on Roads and Tracks. Cham: Springer International Publishing, 2020. P. 1730–1739.

RAGHAVAN, Madhusudan. Suspension Synthesis for N:1 Roll Center Motion. **Journal of Mechanical Design**, v. 127, n. 4, p. 673–678, July 2004. ISSN 1050-0472. DOI: 10.1115/1.1867500. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/127/4/673/5807162/673_1.pdf. Available from: https://doi.org/10.1115/1.1867500.

RODRIGUEZ, Nibaldo; GUPTA, Abhishek; ZABALA, Paula; CABRERA-GUERRERO, Guillermo. Optimization Algorithms Combining (Meta)heuristics and Mathematical Programming and Its Application in Engineering. **Mathematical Problems in Engineering**, v. 2018, p. 1–3, Sept. 2018. DOI: 10.1155/2018/3967457.

RUMPE, Bernhard. **Modeling with UML: Language, Concepts, Methods**. [S.l.]: Springer International Publishing, 2016. ISBN 9783319339337.

SANCIBRIAN, Ramon; GARCIA, Pablo; VIADERO, Fernando; FERNANDEZ, Alfonso; DE-JUAN, Ana. Kinematic design of double-wishbone suspension systems using a multiobjective optimisation approach. **Vehicle System Dynamics**, Taylor & Francis, v. 48, n. 7, p. 793–813, 2010. DOI: 10.1080/00423110903156574.

SEWARD, Derek. **Race Car Design**. [S.l.]: Palgrave Macmillan, 2014. ISBN 9781137030153.

STROUSTRUP, Bjorn. **A Tour of C++**. [S.l.]: Pearson Education, 2018. (C++ In-Depth Series). ISBN 9780134998039.

ZÄPFEL, Günther; BRAUNE, Roland; BÖGL, Michael. **Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics**. [S.l.]: Springer Berlin Heidelberg, 2010. ISBN 9783642113437.

APPENDIX A – KINEMATIC OUTPUTS CALCULATIONS

The wheel axis vector $\vec{a} = [a_x, a_y, a_z]^T$ is defined by:

$$\vec{a} = \mathbf{T_i} - \mathbf{T_o} \tag{40}$$

where $\mathbf{T_i}$ and $\mathbf{T_o}$ are the inner and outer wheel axis points, denoted in this work as TIRE_axis_inner_<side> and TIRE_axis_outer_<side> respectively, where <side> is substituted by "l" and "r", identifying left or right side of the system, respectively. Through the definition of the wheel axis $\vec{a}$, the camber angles on the left and right wheels, C_{Left} and C_{Right}, are given by Equation (41) and Equation (42), respectively.

$$C_{Left} = \frac{\pi}{2} - \arccos\frac{a_z}{\|\vec{a}\|} \tag{41}$$

$$C_{Right} = -\left(\frac{\pi}{2} - \arccos\frac{a_z}{\|\vec{a}\|}\right) \tag{42}$$

where the vector $\vec{a}$ is the wheel axis vector corresponding to the given side, left or right. Moreover, the left and right toe angles δ_L and δ_R are defined by Equation (43) and Equation (44), respectively.

$$\delta_L = \frac{\pi}{2} - \arccos\frac{a_x}{\|\vec{a}\|} \tag{43}$$

$$\delta_R = -\left(\frac{\pi}{2} - \arccos\frac{a_x}{\|\vec{a}\|}\right) \tag{44}$$

Thus, the average steering at a given axle is:

$$\delta_{Avg} = \frac{-\delta_L + \delta_R}{2} \tag{45}$$

The Ackerman angle δ_{Ack} is given by:

$$\delta_{Ack} = \delta_L - \delta_R \tag{46}$$

The steering ratio value is linearized by a secant method. The steering system is displaced by $\Delta S = \pm 0.1^\circ$. The steering ratio $\frac{\partial S}{\partial \delta_{Avg}}$ is given by:

$$\frac{\partial S}{\partial \delta_{Avg}} \approx \frac{\Delta S}{\Delta \delta_{Avg}} \tag{47}$$

where $\Delta\delta_{Avg}$ is given by:

$$\Delta\delta_{Avg} = \delta^-_{Avg} - \delta^+_{Avg} \tag{48}$$

where δ^-_{Avg} and δ^+_{Avg} are the δ_{Avg} calculated when the system is at the steering position displaced by –0.1° and +0.1° from the reference position, respectively.

APPENDIX B – PARAMETRIC ANALYSIS DATA

Table 23 – System points used in parametric analysis data.

Point name	$\mathbf{X}(mm)$	$\mathbf{Y}(mm)$	$\mathbf{Z}(mm)$
TIRE_axis_inner_l	0.0	743.0	290.0
TIRE_axis_outer_l	0.0	997.0	290.0
TIRE_virtual_contact_patch_l	0.0	870.0	0.0
TIRE_contact_patch_l	0.0	870.0	0.0
TIRE_virtual_gnd_l	290.0	870.0	0.0
CHAS_coil_over_l	-250.0	450.0	700.0
UPRI_lower_l	47.0	780.0	150.0
CHAS_upper_aft_l	0.0	450.0	430.0
CHAS_lower_fore_l	30.0	390.0	160.0
UPRI_upper_l	-45.0	730.0	460.0
CHAS_lower_aft_l	-250.0	390.0	162.0
CHAS_upper_fore_l	-250.0	470.0	432.0
CHAS_tie_l	68.3	208.0	238.3
UPRI_tie_l	133.9	750.0	191.3
VIRT_coil_over_outer_l	-200.0	650.0	450.0
NSMA_attachment_l	-200.0	650.0	450.0

Source: Author (2021)

APPENDIX C – CASE STUDY BASELINE SYSTEM DATA

Figure 79 – Full set of parameters of the baseline suspension used in the case study extracted from OptimumKinematics Excel exporter.

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Double A-Arm	CHAS_LowFor	137.456	416.950	110.042	137.456	-416.950	110.042
	CHAS_LowAft	-249.314	415.522	146.651	-249.314	-415.522	146.651
	CHAS_UppFor	82.217	553.218	452.594	82.217	-553.218	452.594
	CHAS_UppAft	-120.324	560.309	365.183	-120.324	-560.309	365.183
	UPRI_LowPnt	-1.973	820.133	145.399	-1.973	-820.133	145.399
	UPRI_UppPnt	-66.624	752.534	445.662	-66.624	-752.534	445.662
	CHAS_TiePnt	64.998	472.384	240.045	64.998	-472.384	240.045
	UPRI_TiePnt	110.913	786.942	287.943	110.913	-786.942	287.943

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Direct CoilOver	NSMA_AttPnt_L	-23.454	716.320	207.556	-23.454	-716.320	207.556
	CHAS_AttPnt_L	-35.285	520.874	496.428	-35.285	-520.874	496.428

Rack Pinion	Steering Ratio	60.400

	Point Name	Left	Right
Wheels	Half Track	868.517	868.517
	Longitudinal Offset	0.000	0.000
	Lateral Offset	0.000	0.000
	Vertical Offset	0.000	0.000
	Static Camber	0.000	0.000
	Static Toe	0.000	0.000
	Rim Diameter	457.200	457.200
	Tire Diameter	663.000	663.000
	Tire Width	300.000	300.000

(a) Front Suspension

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Double A-Arm	CHAS_LowFor	-2454.305	437.806	194.573	-2454.305	-437.806	194.573
	CHAS_LowAft	-2827.300	431.400	125.552	-2827.300	-431.400	125.552
	CHAS_UppFor	-2451.692	546.461	389.424	-2451.692	-546.461	389.424
	CHAS_UppAft	-2761.554	541.538	445.610	-2761.554	-541.538	445.610
	UPRI_LowPnt	-2573.089	792.640	183.284	-2573.089	-792.640	183.284
	UPRI_UppPnt	-2592.375	768.562	454.090	-2592.375	-768.562	454.090
	CHAS_TiePnt	-2816.500	435.000	248.852	-2816.500	-435.000	248.852
	UPRI_TiePnt	-2780.000	810.131	270.854	-2780.000	-810.131	270.854
	Tierod Attachment	Chassis					

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Direct CoilOver	NSMA_AttPnt_L	-2562.000	707.000	222.352	-2562.000	-707.000	222.352
	CHAS_AttPnt_L	-2636.998	505.822	661.113	-2636.998	-505.822	661.113

	Point Name	Left	Right
Wheels	Half Track	847.043	847.043
	Longitudinal Offset	-2630.000	-2630.000
	Lateral Offset	0.000	0.000
	Vertical Offset	0.000	0.000
	Static Camber	0.000	0.000
	Static Toe	0.000	0.000
	Rim Diameter	457.200	457.200
	Tire Diameter	690.600	690.600
	Tire Width	310.000	310.000

(b) Rear Suspension

Source: Author (2021)