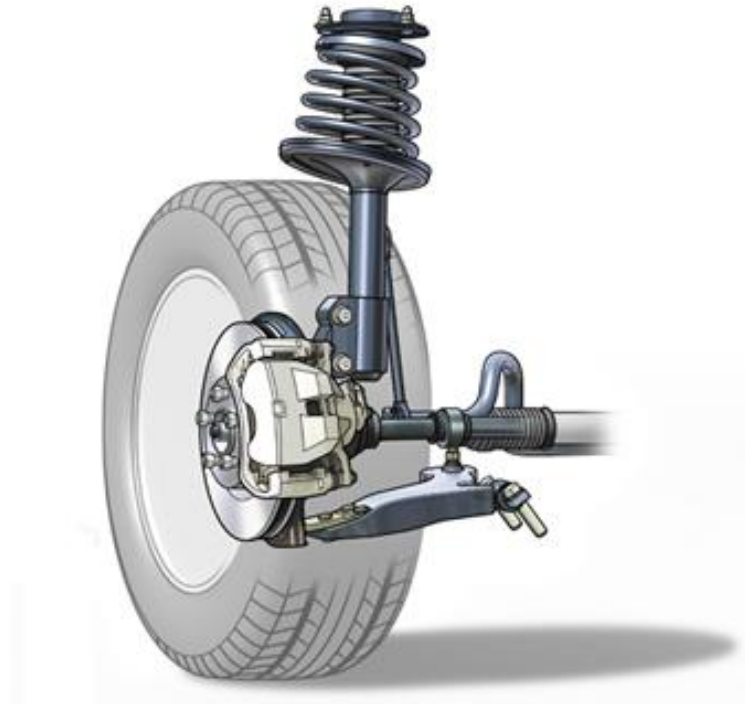




CHALMERS
UNIVERSITY OF TECHNOLOGY



Suspension Kinematics Programme

Master's thesis in Automotive Engineering

Xunbo Yang

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Suspension Kinematics Programme

XUNBO YANG



Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Suspension Kinematics Programme
Xunbo Yang

© Xunbo Yang, 2019.

Academic Supervisor
Ingemar Johansson
Professor of the Practice, Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Senior Advisor, Wheel Suspension, CEVT

Technical Supervisor
Andrew Dawkes
Design Engineer, Wheel Suspension, CEVT

Examiner
Mathias Lidberg
Associate Professor, Vehicle Engineering and Autonomous System
Chalmers University of Technology

Master's Thesis 2019:24
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: MacPherson Suspension
Source: <http://maybach300c.blogspot.com/2012/09/mcpherson-struts-and-strut-damper.html>
Department of Mechanics and Maritime Sciences
Gothenburg, Sweden 2019

Suspension Kinematics Programme
Xunbo Yang
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

In the automotive industry, during the early design phase for the vehicle suspension system, a design iteration between the suspension kinematics performance and packaging requirements need to be performed. Hence, to increase the working efficiency, a quick and robust design tool which can carry out basic kinematics simulation is required. Comparing to the existed commercial software on the market in this area (eg. MSC ADAMS/CAR), this tool will only covers selected suspension type and pure kinematics simulation. Delivering the required suspension kinematics parameters with easy and friendly user interface is the main goal.

This master thesis only focus on MacPherson suspension. First, it starts with finding a suitable method for the modelling of the suspension mechanism system. Different modelling methods are compared including planar modelling and spatial modelling, relative coordinates system modelling and global coordinate system modelling. Constraint equations are formulated based on the chosen modelling method. The nonlinear equation solving problem was converted into an optimization problem to solve. Slack variable method was applied here to provide more freedom to the solver. After the equation system was solved, the data was post processed then given as required output parameters.

After all the mathematical process are completed, the software implementation is performed to realize all these functions in chosen software platform. First, a simple MATLAB version is completed for early verification and back-up plan, then EXCEL is chosen to be the final software platform due to its user friendly interface and widely application. All functions are achieved by EXCEL Visual Basic Application (VBA, EXCEL coding language) with EXCEL add-in Solver. A comparison between the result from EXCEL and ADAMS/CAR was made at last to validate the accuracy of the EXCEL program.

Keywords: MacPherson suspension, Kinematics, Modelling, Excel VBA, Optimization, Simulation

Acknowledgements

I would like to express my gratitude to all the people in CEVT and Chalmers who have helped me during this whole master thesis work especially my supervisors and examiner. Ingemar Johansson was a patient and excellent supervisor who helped me a lot not only in this master thesis but much more beyond this range. Andrew Dawkes was a great design engineer who provided me with professional engineering views and experiences. Professor Mathias Lidberg conducted me with numerous academic suggestions including mathematical knowledge, technical report writing and presentation skills. I would like to say thank you for all your dedicated work through out this master thesis project.

Xunbo Yang, Gothenburg, June 2019

Notations

δ	Toe angle
σ	Kingpin inclination angle
τ	Caster angle
θ_i	Inner wheel steer angle
θ_o	Outer wheel steer angle
θ_{it}	Ideal inner wheel steer angle
φ	Camber angle
AD	Anti dive
AL	Anti lift
CoG	Center of gravity
DOF	Degree of freedom
GRG	Generalized reduced gradient
HP	Hard points
IC	Instantaneous center
IR	Installation ratio
K	Percentage Ackermann
L	Wheel base
RC	Roll center
t	Track width
VBA	Visual basic application
WC	Wheel center

Contents

List of Figures	xiii
------------------------	-------------

List of Tables	xv
-----------------------	-----------

1	Introduction	1
1.1	Background	1
1.2	Objective	2
1.3	Deliverables	2
1.4	Limitations	2
1.5	Vehicle suspension	2
1.5.1	Vehicle Axis System	3
1.5.2	Suspension Parameters	4
2	Suspension Modelling	15
2.1	MacPherson Suspension	15
2.2	Modelling	16
2.2.1	Spatial Modelling	16
2.2.2	Coordinate System	16
2.3	Degree of Freedom	16
2.4	Hardpoints	17
2.5	Constraint Equation Formulation	18
2.5.1	Constraints	18
2.5.2	Equation Formulation	18
2.6	Driven data	21
2.7	Equation Solving	22
2.7.1	General Structure	22
2.7.2	Algorithm	23
2.7.3	Slack Variable	24
2.7.4	Optimization Problem	25
2.7.5	Solving Process	26
2.7.5.1	Vertical Simulation	26
2.7.5.2	Steering Simulation	27
2.8	Delimitation	28
2.9	Output Parameter	30
2.9.1	Vertical Simulation	30
2.9.2	Steering Simulation	31

2.9.3	Parameter Calculation	31
3	Software Implementation	39
3.1	Excel Introduction	39
3.2	Overall structure	40
3.3	Worksheet Introduction	41
3.3.1	Front Cover	42
3.3.2	Input	42
3.3.3	Simulation	45
3.3.4	Intermediate	47
3.3.5	Vertical output & Steering output	47
3.3.6	Vertical figure & Steering figure	47
3.4	Module	48
4	Result	51
4.1	Simulation Set up	51
4.1.1	Input data	51
4.1.2	Simulation Setting Parameters	51
4.2	Simulation Result	52
4.2.1	Vertical simulation	52
4.2.1.1	Parameter table	52
4.2.1.2	Graph result	52
4.2.2	Steering simulation	55
4.3	Validation	56
4.3.1	Vertical Simulation	56
4.3.2	Steering Simulation	59
5	Conclusion	61
5.1	Conclusion	61
5.2	Discussion	61
5.2.1	Modelling Method	61
5.2.2	Software	62
5.3	Future work	62
5.3.1	Other suspension	62
5.3.2	Steering simulation	62
5.3.3	User interface	62
	Bibliography	63
A	Programme User Instructions	I
A.1	Preparation	I
A.2	CEVT Template	I
A.3	Simulation Process	II
A.3.1	Running Simulation	II
A.3.2	Post Processing	II
B	Excel Module, <i>Read_data</i>	V

C	Excel Module, <i>Vertical</i>	VII
D	Excel Module, <i>Steering</i>	XXI
E	Excel Module, <i>Vertical figure</i>	XXXIII
F	Excel Module, <i>Steering figure</i>	XLIII
G	Excel Module, <i>Export</i>	XLIX

List of Figures

1.1	Design iteration, early design phase	1
1.2	Vehicle suspension system[9]	3
1.3	Vehicle axis system[1]	4
1.4	Vehicle axis system, design purpose[7]	4
1.5	Camber angle[8]	5
1.6	Camber gain	6
1.7	Toe angle[8]	6
1.8	Zero bumpsteer geometry	7
1.9	Wheel center longitudinal displacement	8
1.10	Trackrod to steering arm angle	8
1.11	Different Ackermann geometry	9
1.12	Roll center	9
1.13	Roll motion	10
1.14	Kingpin inclination angle[8]	11
1.15	Caster angle[8]	11
1.16	Installation ratio	13
1.17	Steady state longitudinal load transfer	13
1.18	Anti dive	14
1.19	Anti lift	14
2.1	Schematic, MacPherson suspension[3]	15
2.2	Hardpoints position	17
2.3	Vector loops in macpherson suspension	18
2.4	Distance constraints	19
2.5	Spindle axis and tire contacting hardpoint	20
2.6	Driven data, vertical and steering simulation	21
2.7	GRG Nonlinear algorithm	23
2.8	Solving iteration	25
2.9	Steering geometry, symmetry	28
2.10	Kingpin inclination angle	29
2.11	Caster angle	29
2.12	Camber angle, calculation	31
2.13	Toe angle, calculation	32
2.14	Trackrod to steering arm angle calculation	33
2.15	Ideal Ackermann steering geometry	33
2.16	Roll center calculation	34

2.17	Kingpin inclination angle calculation	35
2.18	Caster angle calculation	35
2.19	Scrub radius calculation	36
2.20	Caster trail calculation	36
2.21	Installation ratio calculation	37
2.22	Anti-dive calculation	37
2.23	Anti-lift calculation	38
3.1	Excel Visual Basic interface	40
3.2	Flow chart, vertical simulation	41
3.3	Flow chart, steering simulation	41
3.4	Control button, input worksheet	42
3.5	Simulation setting parameter, example	43
3.6	<i>input</i> worksheet, hardpoints area	44
3.7	Distance constraint, <i>simulation</i> worksheet	45
3.8	Variables, <i>simulation</i> worksheet,	46
3.9	Equations, <i>simulation</i> worksheet	46
3.10	Control area, <i>figure</i> worksheet	47
3.11	Figure area, <i>figure</i> worksheet, example	48
3.12	Parameter table, <i>figure</i> worksheet, example	48
3.13	Module	49
4.1	Input data	51
4.2	Graph result, vertical simulation, part 1	53
4.3	Graph result, vertical simulation, part 2	54
4.4	Graph result, steering simulation	55
4.5	Camber angle, validation	56
4.6	Toe angle, validation	57
4.7	Kingpin inclination angle, validation	57
4.8	Caster angle, validation	58
4.9	Roll center height, validation	58
4.10	Right steer angle, validation	59
4.11	Left steer angle, validation	59
4.12	Percent Ackermann, validation	60
A.1	CEVT template	II

List of Tables

2.1	Hardpoints list	17
2.2	Vertical Simulation Output Parameter, graph	30
2.3	Vertical Simulation Output Parameter, value	30
2.4	Steering Simulation Output Parameter	31
3.1	Worksheets list	40
3.2	Module list	49
4.1	Simulation setting parameters	52
4.2	Output parameter table	52

1

Introduction

1.1 Background

In automotive industry, during the initial engineering phase to develop concept designs and packaging solutions for the suspension systems, the suspension kinematics geometry need to be checked. Therefore, a small, quick and robust programme which is able to carry out simple suspension kinematic simulation (parallel vertical bump and steering) is needed. The output from this programme should cover all the main design parameters which interested in suspension design and the result should given by figure for better understanding. This suspension kinematics program will be used only in the initial design phase, the detailed kinematics and compliance analysis will be carried out later in specialized software, e.g. ADAMS/Car

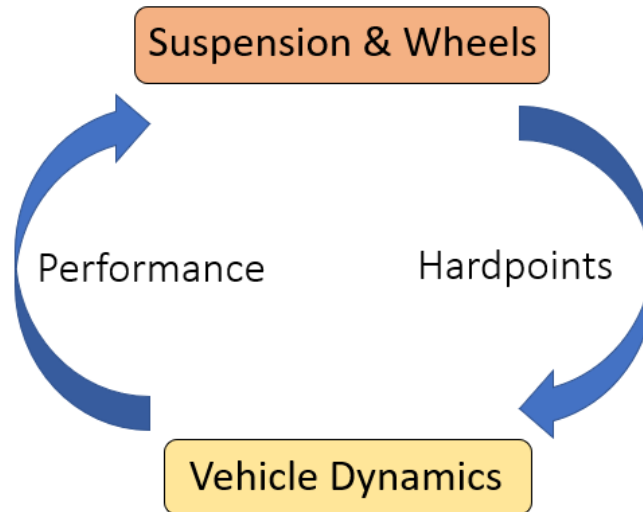


Figure 1.1: Design iteration, early design phase

In this thesis, the kinematics of a 3-dimensional Macpherson mechanism are investigated. Nonlinear equation solving is performed for position analysis. The results of these analysis are stated in terms familiar to suspension engineers, eg. camber angle, toe angle and roll center height.

Excel has been chosen to be the platform for the program due to its a widely application and simple manipulation. For the verification purpose, a MATLAB

version has been developed first which has all the function required. The EXCEL solver add-in has been used for the equation solving and all the function was realized through Excel Visual Basic Application language.

1.2 Objective

The objective of this master thesis project is to develop a fast and robust suspension kinematics analysis program based on EXCEL to be used by the design engineers to assess the effect of the changes on hard point coordinates when developing concept and system design solutions for the suspension systems during concept design phase.

1.3 Deliverables

The following deliverables are expected from this Master Thesis project:

- Literature study
- Software selection, select the software for the simulation and analysis
- Software implementation, develop a analysis programme based on the selected software
- Input data format to suit the selected suspension systems
- Output format to suit the expected parameters to be presented
- Instructions on how to use the analysis programme
- Master Thesis report

1.4 Limitations

- The analysis programme will only carry out pure kinematics simulation, compliance analysis is not included.
- The analysis programme only covers the selected suspension concepts
- The geometries for the spring, damper and the anti-roll bar is not included, the loads from these components are not included in the analysis.
- The steering simulation was performed based on fixing the outer ball joint rather than fixing the height of the wheel.
- All the output parameters are calculated on plane (2-D).

1.5 Vehicle suspension

Vehicle suspension system includes all the force transfer components attached between vehicle frame and axle. The function of it is to transfer all the vertical, lateral and longitudinal counter-acting force and resulting moment from tires to frame to ensure the normal riding of the vehicle.

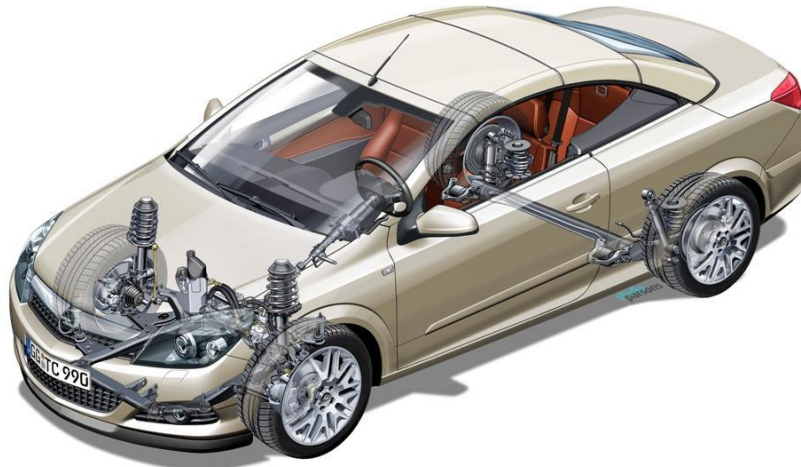


Figure 1.2: Vehicle suspension system[9]

The main components of suspension system can be classified as elastic components, absorbers and linkages. Elastic components are used to buffer the impact from the road then convert it into kinetic energy and stores in itself. In most of the passenger cars the elastic components are coil springs. In certain heavy vehicles (trucks) they might be plate springs, torsion springs or air springs. Absorbers are used to absorb the kinetic energy stored by the elastic component. The most widely used absorbers are dampers, it utilizes the friction force inside oil to absorb the kinetic energy and convert it into heat then transfers into air. Linkages are the connection parts between the sprung mass and the unsprung mass. It decides the direction of the load and moment.

1.5.1 Vehicle Axis System

For different design purposes, there exist different definitions for the vehicle axis system. The ISO 8855 is the international standard vehicle axis system. In ISO 8855, the X axis is pointing the front, Y axis points to the left and Z axis points up.

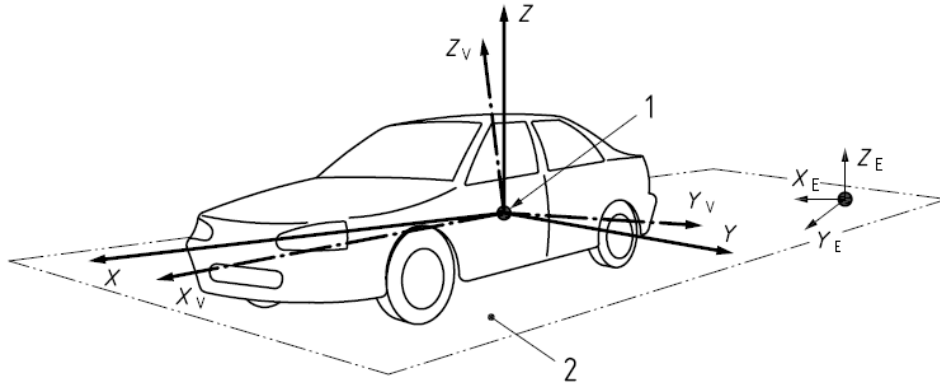


Figure 1.3: Vehicle axis system[1]

It needs to be mentioned that, ISO 8855 focus on vehicle dynamics analysis, this is the reason why the X axis points to the front since that is the driving direction. For vehicle design purpose, in vehicle manufacture company, a vehicle axis system which has a X axis points to the rear is always used.

The reason for using this vehicle axis system can be explained as, back in the early time in automotive design industry, design engineers make technical drawing by hand. For convenience, the axis system is always be put in the bottom left of the paper. As industry develops, this design standard was has been kept.

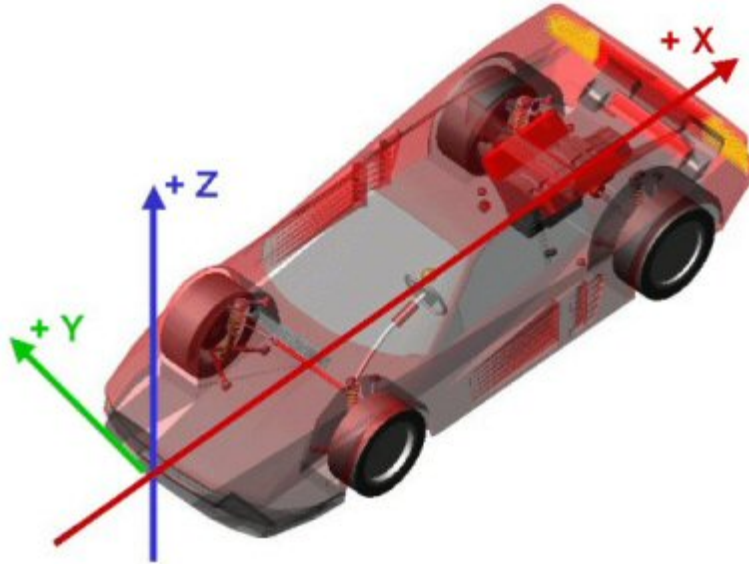


Figure 1.4: Vehicle axis system, design purpose[7]

1.5.2 Suspension Parameters

The following parameters are the most important ones in suspension kinematics design. The value of these parameters have a major affect on the vehicle dynamics performance.

- **Camber Angle**

Camber angle is the angle between the wheel plane and the X-Z plane measured in front view. According to ISO 8855:2011, it is considered positive when the wheel leans outward at the top, relative to the vehicle body, and negative when it leans inward.

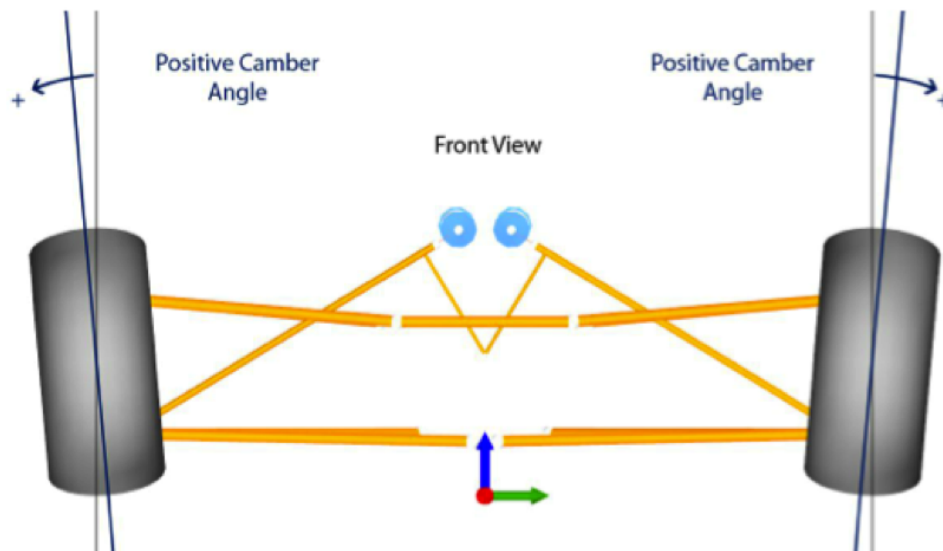


Figure 1.5: Camber angle[8]

For a passenger car, usually the camber angle is a small negative value. For a racing car, the camber angle is usually bigger in negative. A negative static camber with a negative camber gain design can help the outer wheel closer to vertical. For a car designed to cooperate with high centrifugal force, this can help the car during cornering. Apart from this, a negative camber can cause a force called "camber-thrust" which will can improve the straight stability (two tires on the same axle tend to get closer during rolling forward). However, too-much camber (no matter negative or positive) will cause faster tire wear, reducing the tire life.

- **Camber gain**

The changing rate of camber with respect to the wheel vertical traveled is called camber gain. Usually this value is taken at the design position (gradient of the camber to wheel travel curve).

Camber gain only depends on the instantaneous center (IC) position. An IC inside the suspension will bring a negative camber gain (camber angle reduces during bump) and an IC outside the suspension indicates a positive camber gain.

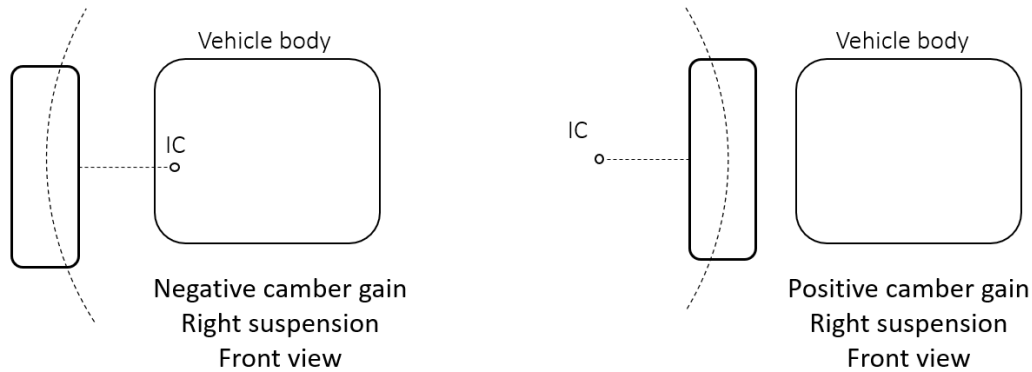


Figure 1.6: Camber gain

- **Toe Angle**

Toe angle is the angle between the wheel plane and the X-Y plane measure in top view. According to ISO 8855:2011, it is considered "toe-in" if the forward portion of the wheel is closer to the vehicle centerline than the wheel center and "toe-out" if it is farther away.

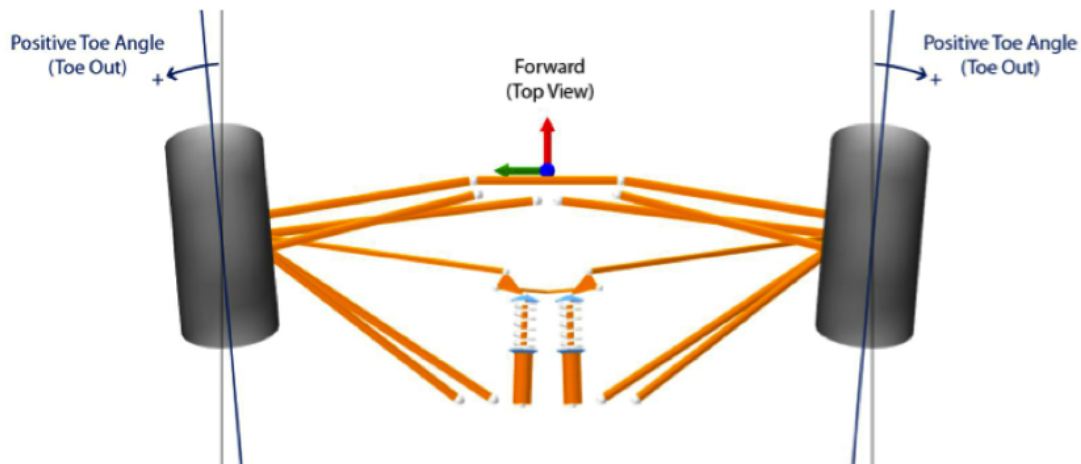


Figure 1.7: Toe angle[8]

Starting the longest tire life, a zero toe angle will be the best. But usually to improve the car performance, designer will choose to add a certain amount of toe angle. In general, for front axle, toe in can improve the straight line stability and toe out can improve the steering response. Street cars often are set up with toe-in; For good straight-line stability cornering is sacrificed. Race cars are often set up with toe-out; Straight-line stability is sacrificed for good cornering.

- **Bumpsteer (toe gain)**

Bumpsteer refers the toe angle changes during the wheel vertical travel. It represents the amount of the affect on the toe angle from the vertical movement.

Bumpsteer is the pure kinematics parameter, it results from the position relationship between the IC of the suspension linkages and the IC of the steering arm (tie rod). If the tie rod IC is on the line to the suspension IC, then this suspension will have zero bumpsteer.

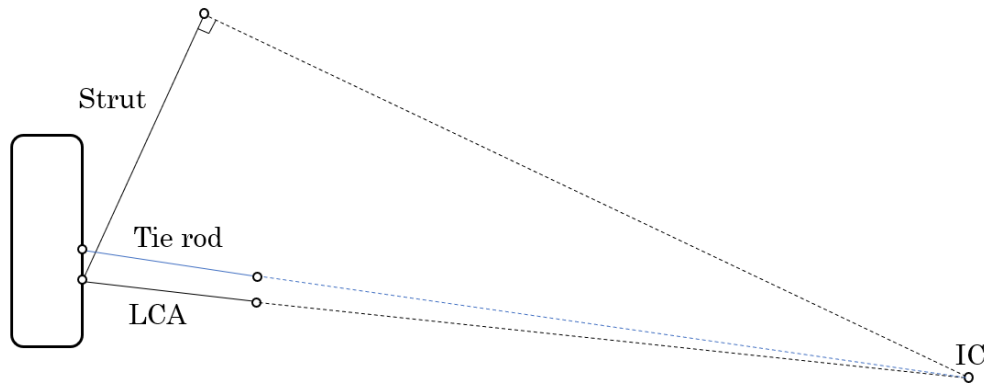


Figure 1.8: Zero bumpsteer geometry

A certain amount of bumpsteer can help improve some certain performance of the vehicle. For example, negative bumpsteer (toe angle reduces during bump travel) in front axle can help improve the stability during braking, but too much bumpsteer will also speed up the wearing of the tire.

- **Trackwidth variation**

Trackwidth is the distance between the right side wheel center and left side wheel center in Y coordinates. Trackwidth variation is a trade-off with camber gain design since they both depend on the front view instantaneous center position. A instantaneous center which locates infinity far from the suspension indicates a zero camber gain but will have a huge trackwidth variation which will accelerate the wear up of the tire. With the instantaneous center move closer to the wheel center, the camber gain will increase and the trackwidth variation will decrease.

- **Wheel center longitudinal displacement**

The longitudinal displacement of the wheel center, measured in X coordinates. For MapPherson and double wishbone suspension, this value is usually small, but for a trailing arm suspension, due to the geometry of the linkages, this parameter will be much bigger.

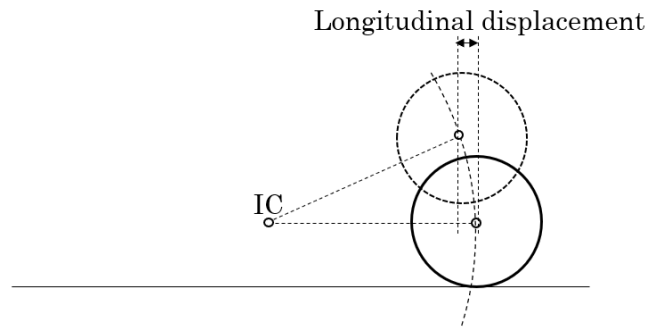


Figure 1.9: Wheel center longitudinal displacement

Wheel center longitudinal displacement can affect the ride comfort of the vehicle. For example, for a suspension with positive this parameter (wheel moves forward during bump), when the vehicle driving over a bump, the "strike" feeling will be exaggerate leading to uncomfortable feeling for the passengers.

- **Track rod to steering arm angle**

The angle between the steering arm (tie rod outer to kingpin axis) and track rod (tie rod). In order to steer the wheel, this angle needs to be between 0 to 180 degrees. When it get closer to 0 or 180, the steering will become difficult since most part of the force is used to "drag" the wheel rather than steer it therefore this situation needs to be prevented.

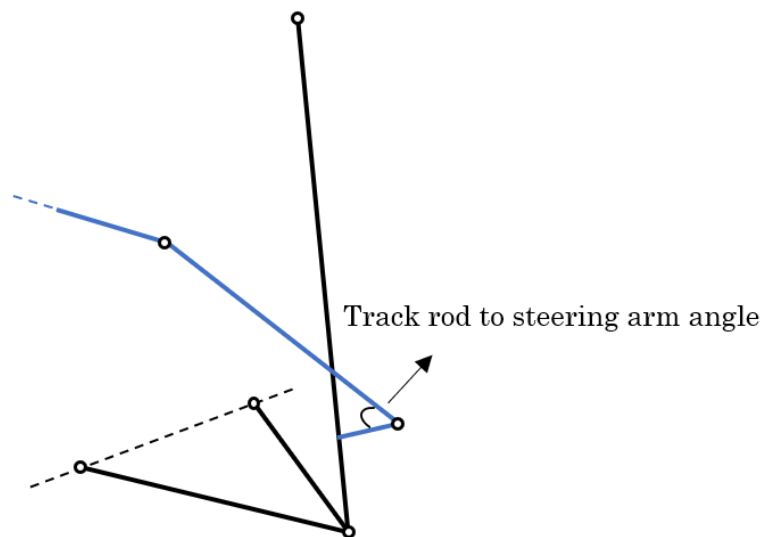


Figure 1.10: Trackrod to steering arm angle

- **Percent Ackermann**

Percent ackermann is a quantified parameter widely used to describe the steering geometry [1]. It shows the relationship between the outer wheel steer angle and inner wheel steer angle.

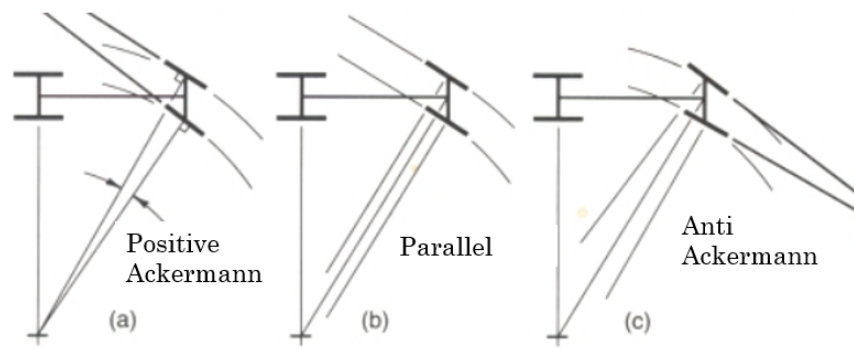


Figure 1.11: Different Ackermann geometry

For a 100% Percent Ackermann, the steering geometry matches perfectly with the ideal Ackermann geometry. Positive Ackermann indicates the inner wheel steers more than the outer wheel and negative Ackermann indicates the outer wheel steers more than the inner wheel.

The reason that Percent Ackermann is used in design phase is because, first, the ideal Ackermann sometimes is difficult to reach in kinematics design due to the packaging reason, the steering rack position has to consider other issue as well. Second, the steering geometry is not pure kinematic, it will be affected by the slip angle from the tire as well especially in high speed situation [6].

- **Roll center height**

For the roll center in suspension design, there are kinematics roll center and force based roll center. Force based roll center usually requires K&C testing machine after the full chassis manufacture is finished. In this thesis, the roll center will only focus on kinematics roll center. For a symmetric suspension, the position of it can be found through the figure below.

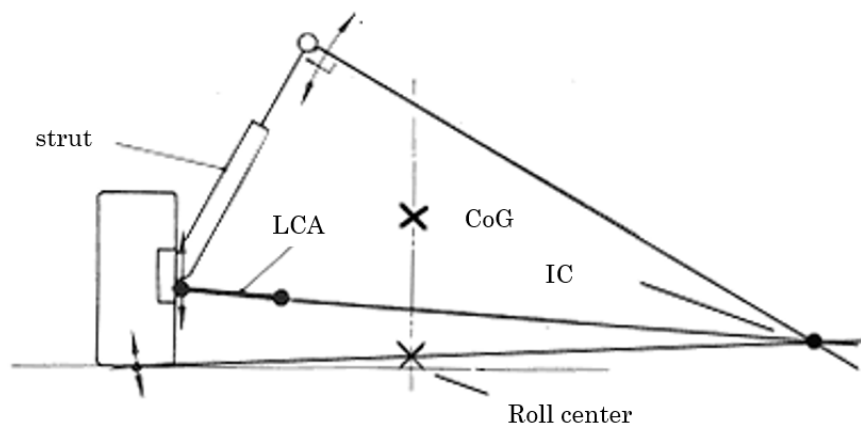


Figure 1.12: Roll center

For both front and rear axle, there exist a corresponding roll center. By connecting these two points with a straight line, the roll axis can be defined.

During the roll motion, the vehicle is defined to roll around roll axis.

The major function of roll center is it controls the lateral load transfer distribution. It decides how much lateral load transfer goes through the actuation system (makes vehicle roll) and how much goes through the linkages (won't makes vehicle roll) by the height relationship between the roll center and center of gravity point. For a passenger cars, the center of gravity point is usually much higher than the roll center. Hence, for a higher roll center (closer to COG height), the vehicle will roll less comparing to a lower roll center design.

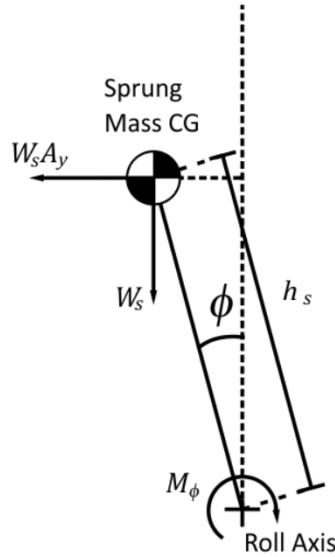


Figure 1.13: Roll motion

where W_s stands for sprung weight, A_y stands for lateral acceleration, ϕ stands for roll angle, M_ϕ stands for the counter roll moment from the actuation system and h_s stands for the distance between sprung mass center of gravity to the roll axis (roll center on front view).

- **Kingpin inclination angle**

The angle between the kingpin axis and X-Z plane measured in front view. It is considered positive when the top of kingpin axis is inclined inward.

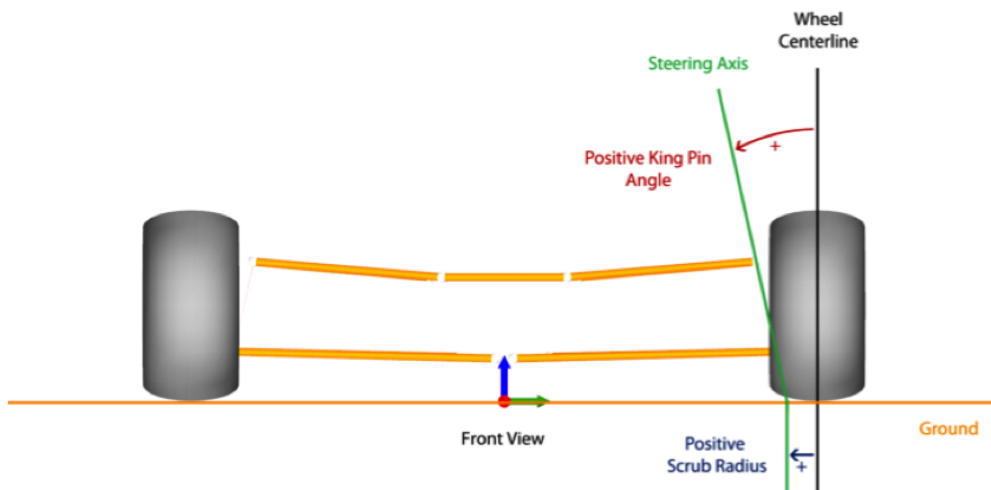


Figure 1.14: Kingpin inclination angle[8]

The kingpin inclination angle provides the alignment torque for the steering system from gravity. With a kingpin inclination angle, the wheel will be "lifted" when it is steered, then the gravity force from the wheel will tend to steer the wheel back hence create the alignment torque. However, a too big kingpin axis will also make the steering wheel heavy to turn.

- **Caster angle**

The angle between the kingpin axis and the Y-Z plane measured in side view. It is considered positive when the top of the kingpin axis is inclined rearward.

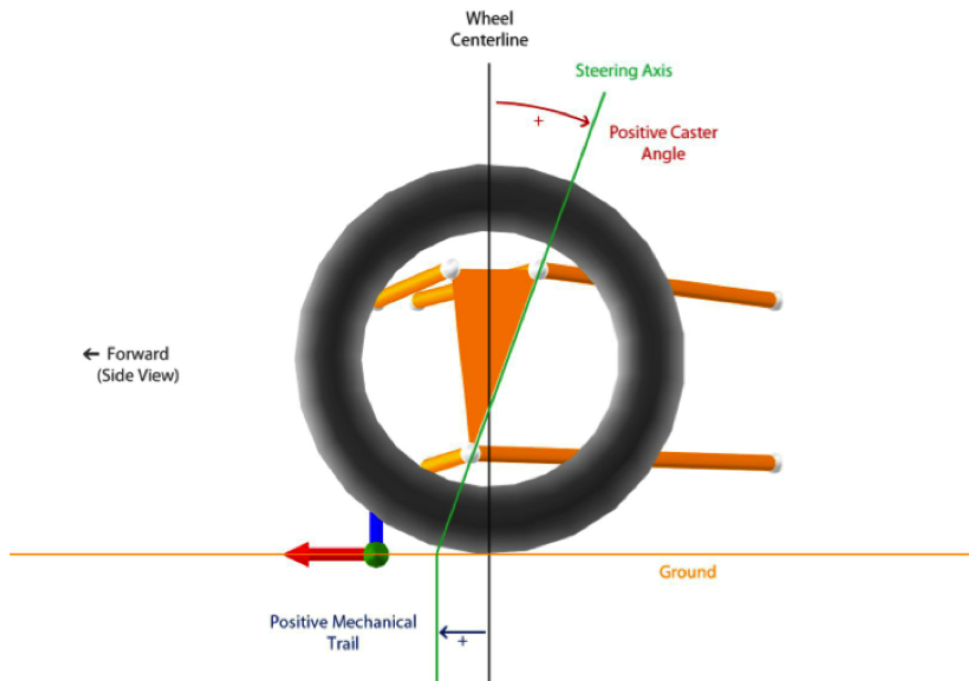


Figure 1.15: Caster angle[8]

The main function of caster angle is providing alignment torque as well but from different theory comparing to kingpin inclination angle. The alignment torque from caster angle is based on the lateral force from the contacting patch during cornering. To be more precisely, the parameter which provide alignment torque is caster trail which will be explained later, caster angle is the parameter to control caster trail.

- **Scrub Radius**

See figure 1.14. Scrub radius is the distance from the contact center to the point where the kingpin axis intersect with ground measure in front view. When the intersection point is inside the contact center, then it is a positive scrub radius.

A bigger scrub radius will make the steering heavier since the lever arm is longer. Apart from that, for positive and negative scrub radius, the front wheel will show different behaviour during braking and acceleration. For example, during braking, a front wheel with positive scrub radius will tend to have more "toe out" and negative scrub radius will tend to have more "toe in".

- **Caster trail**

See figure 1.15. It is also called "castor offset at wheel center" in ISO 8855:2011. It is the distance between the contact center to the point where the kingpin axis intersect with ground measured in side view. When the intersection point is in front of the wheel center, then it is a positive caster trail.

As mentioned above, the main function of caster trail (also called "mechanical trail") is to provide alignment torque during cornering. One thing need to mention here is: during cornering the total alignment torque lever arm actually comes from the sum of caster trail and pneumatic trail. Pneumatic trail is a parameter for the tire property, it beyond the range of this paper so it will be skipped here. More detail explanation can be found in [6].

- **Installation ratio**

The ration between the wheel vertical movement and damper movement along it axis. In this master thesis, the installation ratio parameter only calculates the installation ratio damper (there are installation ratio spring as well) since first, the axis of the damper and spring is very close (in most case they actually aligned with each other), second, hp44 - damper bottom is chosen to be the end point of the strut in the modelling, by changing this point to some point on spring axis, then this output parameter will represent installation ratio spring.

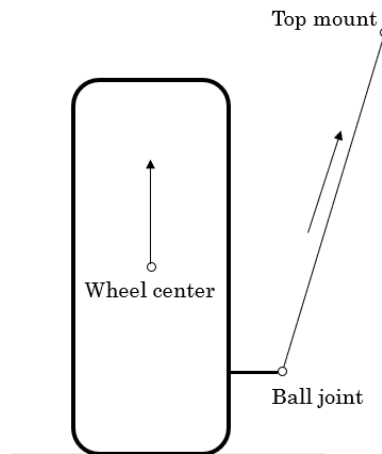


Figure 1.16: Installation ratio

- **Anti-dive**

The "anti" effect in suspension is a term that describes the longitudinal to vertical force coupling between the sprung and unsprung masses. The "anti" effect does not change the steady-state load transfer. The word "anti" indicates the pitch motion of the vehicle is reduced by part of the longitudinal load transfer is taken by the resulting force from longitudinal force (driving or braking). Hence the force is going through the linkages rather than the spring. It only depends on the angle (slope) of the side view suspension geometry.[x]

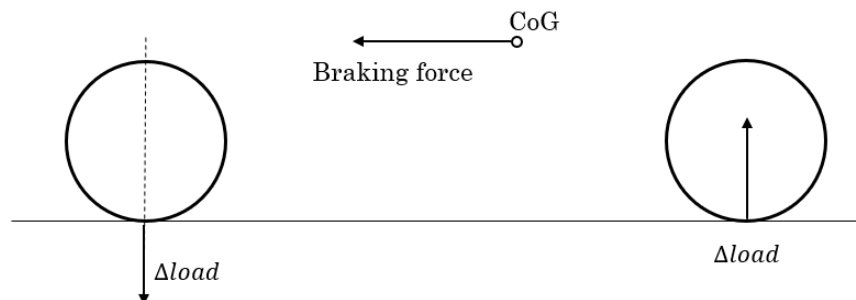


Figure 1.17: Steady state longitudinal load transfer

It can be noticed that one very important prerequisite for "anti" effect to exist is whether to have longitudinal force in this suspension. Therefore, for a front suspension, anti-dive will always exist (also depends on IC position) but anti-lift only occurs on front driven car. For a rear drive car front suspension, there is no anti-lift.

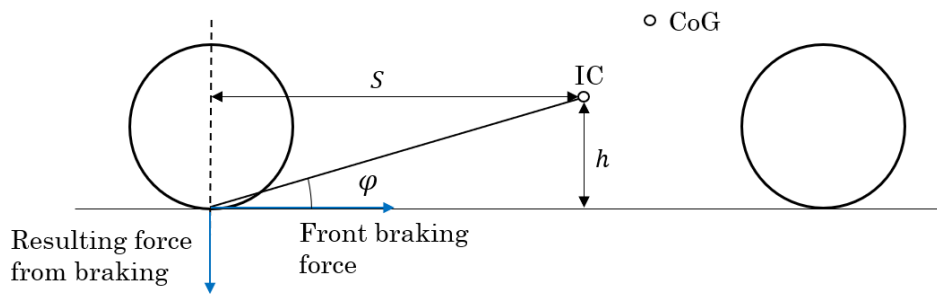


Figure 1.18: Anti dive

- **Anti-lift**

The only difference between anti-dive and anti-lift is the generating of longitudinal force. For anti-dive, the longitudinal force is braking force, generated by the contact patch between the tire and the ground, therefore it starts on the ground. For anti-lift, the longitudinal force is driven force which comes from the drive shaft, hence it starts on the center of the wheel.

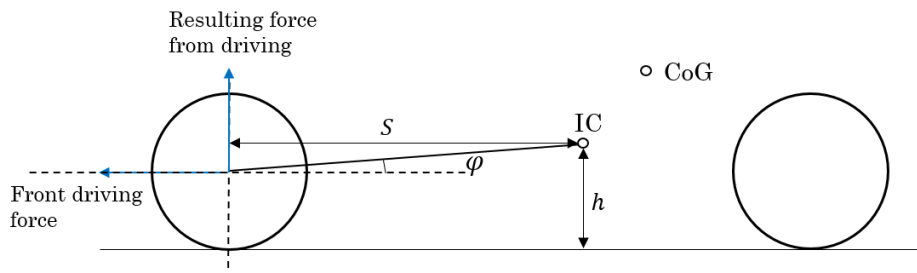


Figure 1.19: Anti lift

2

Suspension Modelling

This chapter illustrates the modelling methods for the MacPherson suspension in this master thesis, starting from the mathematical model building, equation formulation, problem definition and optimization solving process.

2.1 MacPherson Suspension

The following figure presents a schematic diagram for a typical Macpherson suspension. The major components are A-arm, knuckle, strut and tie rod.

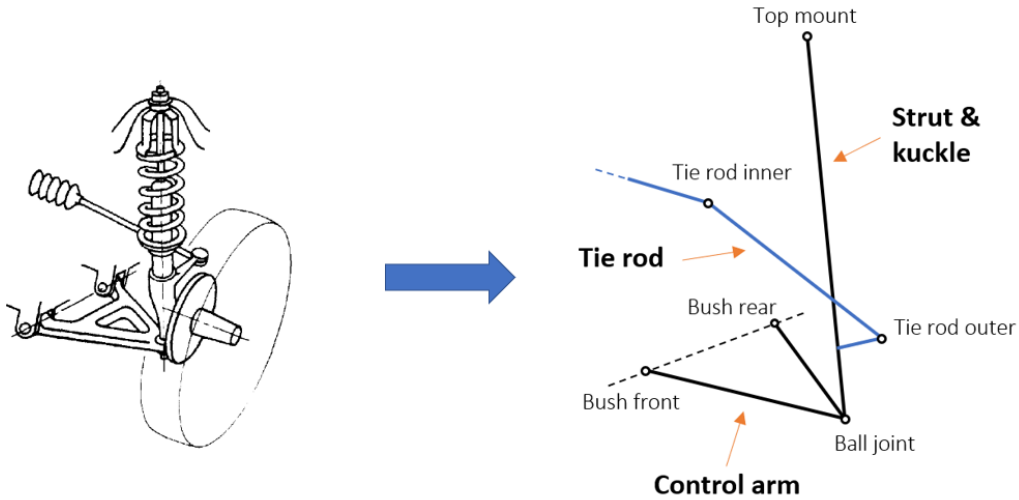


Figure 2.1: Schematic, MacPherson suspension[3]

There are a lot of different naming standards for the hardpoints in MacPherson suspension. For better understanding and also avoiding unnecessary confusion, in this master thesis, all the hardpoints names follow the CEVT template.

According to CEVT template, the three hard points on the lower control arm are called ball joint, bush front and bush rear. The top mounting point of the strut is called top mount and the two connecting points on the tie rod are called tie rod outer and tie rod inner.

2.2 Modelling

Building the mathematical model to represent the existing problem and convert it to a mathematical form to solve is called modelling. To be more specific, in this master thesis, modelling is to formulate the constraint equation system for the MacPherson suspension system.

2.2.1 Spatial Modelling

Even though planar modelling simplifies the suspension mechanism into a 2-D version and can provide relative accurate result (depends on the specific structure), due to the requirements from CEVT, in this master thesis all the modelling has been required to be spatial which is 3-D modelling to capture the hardpoints motion in all three direction.

2.2.2 Coordinate System

To model a mechanism system, in most cases, the constraint equations come from physical relationships which means the coordinates are the fundamental information. Different coordinate systems used categorize the modelling into different types.

In vehicle suspension modelling, depends on the coordinate system used, the most common modelling methods are global coordinate system method and relative coordinate system method. It is easy to understand literally that the meaning of these two different methods. The major difference between them is in relative coordinate system modelling, all the hardpoints which attached to the knuckle and wheel are based on another individual coordinate system. Hence during equation formulation, certain coordinate conversion need to be perform before calculation.

2.3 Degree of Freedom

Theoretically, a free part in 3 dimensional space has 6 degrees of freedom, three for translation motion along three axis and three for revolution motion around them.

In Macpherson suspension mechanism, the lower control arm, the knuckle and the strut each has 6 degrees of freedom. Hence, in total the system has 18 degrees of freedom. The two spherical joint on two side of the strut, each will have 3 constraints. The lower control arm can be replaced by a revolute joint on the frame which will bring 5 constraints. Between the knuckle and the strut there will be a translation joint which ensure these two part always colinear with each other and it will bring 5 constraints. Thus, the degrees of freedom for the Macpherson suspension mechanism can be calculated as:

$$DOF = 3 \times 6 - 3 - 3 - 5 - 5 = 2 \quad (2.1)$$

These two degrees of freedoms are the steering and vertical motion of the wheel.

2.4 Hardpoints

Hardpoints are the geometric points locate on the joint in suspension system, them control the kinematics characteristics of it. The following table lists the name and the code for all the hardpoints in this MacPherson system, the format follows the CEVT template.

Table 2.1: Hardpoints list

Code	Name
hp3	Bush Front
hp4	Bush Rear
hp6	Ball Joint
hp7	Top Mount
hp9	Wheel Center
hp20	Spindle Axis
hp21	Tire Contacting
hp12	Tie rod outer
hp14	Tie rod inner
hp36	Strut alinging point
hp44	Damper bottom
hp46	Auxiliary

The following figure shows all the hardpoints and where they locate in the suspension system

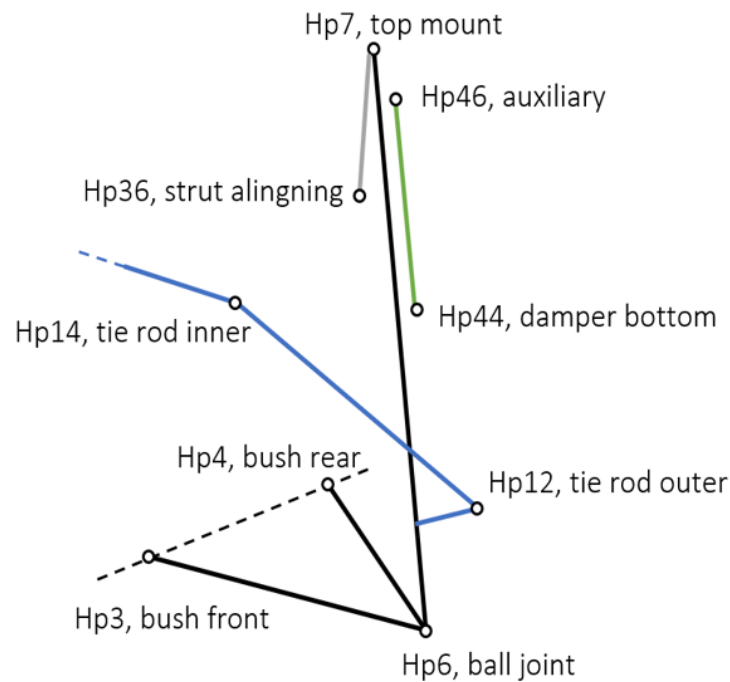


Figure 2.2: Hardpoints position

Apart from the original hardpoints from CEVT template, 3 extra hardpoints are added in this master thesis for the mathematical reason, they are *hp20*-spindle axis, *hp21*-tire contacting and *hp46*-auxiliary.

2.5 Constraint Equation Formulation

2.5.1 Constraints

Constraints are the physical relationships that the system needs to keep and obey during the movement. These relationships can come from different places. The most common ones are loop-closure and physical relationship.

Loop-closure is based on vectore calculation methods. It indicates the existed closed vector loops. The constraint equation is build according to the sum of the all the vectors inside this loop equals to zero. For MacPherson suspension, there are two major different closed vector loops. The first one consists of the chassis, lower control arm, the knuckle and the strut. The second one is composed of chassis, lower control arm, the knuckle and the tie rod. For the rigid link and joint constraint, all the length of the links stays constant and the knuckle and strut have to move together when an excitation comes from the wheel.

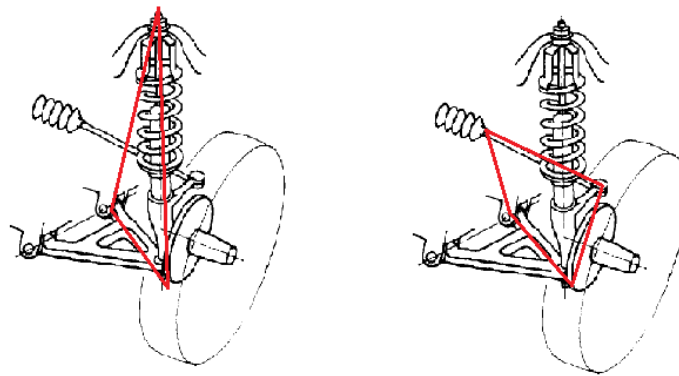


Figure 2.3: Vector loops in macpherson suspension

Physical relationship is more straight forward. It includes distance, parallel and all the other types of physical relationship. The most common one is the distance, which is the distance of the linkages always stay constant during the motion.

2.5.2 Equation Formulation

In master thesis, all the constraint equation formulation are based on physical relationships includes distance constraint and parallel constraint.

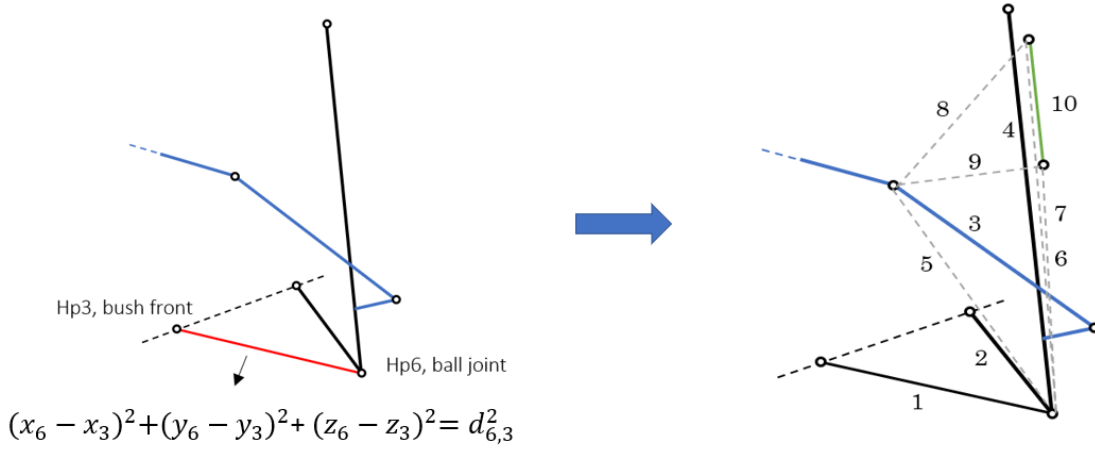


Figure 2.4: Distance constraints

The right half of the above figure shows an example of distance constraint. It took the linkage between hp3 and hp6, and the equation can ensure the distance value always stays constant (equal to $d_{6,3}$). Applying the same method to all the linkages exist (including the virtual ones), 10 constraint equations are derived.

$$(x_6 - x_3)^2 + (y_6 - y_3)^2 + (z_6 - z_3)^2 - d_{6,3}^2 = 0 \quad (2.2)$$

$$(x_6 - x_4)^2 + (y_6 - y_4)^2 + (z_6 - z_4)^2 - d_{6,4}^2 = 0 \quad (2.3)$$

$$(x_{12} - x_{14})^2 + (y_{12} - y_{14})^2 + (z_{12} - z_{14})^2 - d_{12,14}^2 = 0 \quad (2.4)$$

$$(x_7 - x_{36})^2 + (y_7 - y_{36})^2 + (z_7 - z_{36})^2 - d_{7,36}^2 = 0 \quad (2.5)$$

$$(x_6 - x_{14})^2 + (y_6 - y_{14})^2 + (z_6 - z_{14})^2 - d_{6,14}^2 = 0 \quad (2.6)$$

$$(x_6 - x_{44})^2 + (y_6 - y_{44})^2 + (z_6 - z_{44})^2 - d_{6,44}^2 = 0 \quad (2.7)$$

$$(x_6 - x_{46})^2 + (y_6 - y_{46})^2 + (z_6 - z_{46})^2 - d_{6,46}^2 = 0 \quad (2.8)$$

$$(x_{46} - x_{14})^2 + (y_{46} - y_{14})^2 + (z_{46} - z_{14})^2 - d_{46,14}^2 = 0 \quad (2.9)$$

$$(x_{44} - x_{14})^2 + (y_{44} - y_{14})^2 + (z_{44} - z_{14})^2 - d_{44,14}^2 = 0 \quad (2.10)$$

$$(x_{46} - x_{44})^2 + (y_{46} - y_{44})^2 + (z_{46} - z_{44})^2 - d_{44,46}^2 = 0 \quad (2.11)$$

Where $d_{i,j}$ is the distance between point i and point j which locate on the two sides of one particular linkage. x_i , y_i and z_i are the Cartesian coordinates of point i , respectively.

Due to the translational joint between the strut and the knuckle, the two points on each body are located along the axis of the joint and the constraint formulation should ensure that the four points(7,36,44,46) always remain colinear otherwise the knuckle and strut might separate with each other during the simulation. According to this feature, four constarint equations can be derived.

$$(y_{44} - y_{46})(z_{36} - z_7) - (z_{44} - z_{46})(y_{36} - y_7) = 0 \quad (2.12)$$

$$(x_{44} - x_{46})(z_{36} - z_7) - (z_{44} - z_{46})(x_{36} - x_7) = 0 \quad (2.13)$$

$$(y_{36} - y_{44})(z_{36} - z_7) - (z_{36} - z_{44})(y_{36} - y_7) = 0 \quad (2.14)$$

$$(x_{36} - x_{44})(z_{36} - z_7) - (z_{36} - z_{44})(x_{36} - x_7) = 0 \quad (2.15)$$

Equations 2.12 and 2.13 ensure that the knuckle and strut remains parallel to each other. Equations 2.14 and 2.15 ensure that they always attached together and don't go separate.

All the equations listed above are sufficient enough to decide the position and orientation of the suspension mechanism. However, in order to define the output parameter eg. camber angle, the coordinates of 3 more hard points (*hp9* - wheel center, *hp20* -tire contacting and *hp21* -spindle axis) need to be solved.

Both wheel center and spindle axis hard point coordinates can be calculated by three distant constraint equation (constrained to three other hard points).

$$(x_6 - x_9)^2 + (y_6 - y_9)^2 + (z_6 - z_9)^2 - d_{6,9}^2 = 0 \quad (2.16)$$

$$(x_{46} - x_9)^2 + (y_{46} - y_9)^2 + (z_{46} - z_9)^2 - d_{46,9}^2 = 0 \quad (2.17)$$

$$(x_{44} - x_9)^2 + (y_{44} - y_9)^2 + (z_{44} - z_9)^2 - d_{44,9}^2 = 0 \quad (2.18)$$

$$(x_{20} - x_9)^2 + (y_{20} - y_9)^2 + (z_{20} - z_9)^2 - d_{20,9}^2 = 0 \quad (2.19)$$

$$(x_{20} - x_6)^2 + (y_{20} - y_6)^2 + (z_{20} - z_6)^2 - d_{20,6}^2 = 0 \quad (2.20)$$

$$(x_{20} - x_{12})^2 + (y_{20} - y_{12})^2 + (z_{20} - z_{12})^2 - d_{20,12}^2 = 0 \quad (2.21)$$

The tire contacting hard point is defined to be the center of the contact patch. The Z coordinate of it also indicate the height of the ground and it will be used when calculating the output parameter which relates to the ground height such as roll center height.

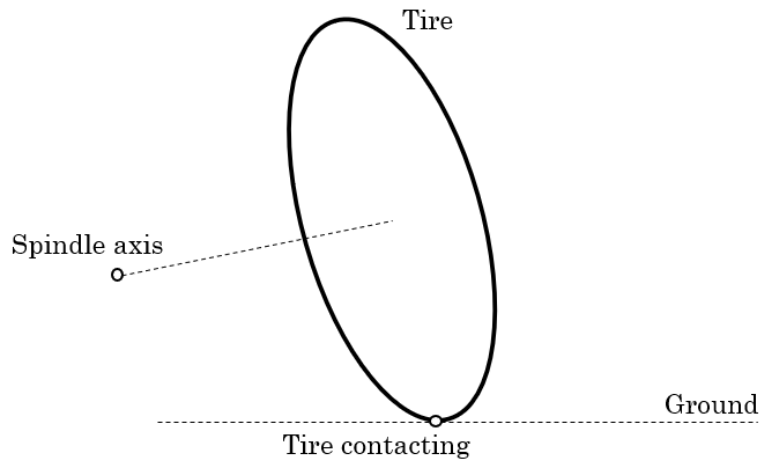


Figure 2.5: Spindle axis and tire contacting hardpoint

$$(x_{21} - x_9)^2 + (y_{21} - y_9)^2 + (z_{21} - z_9)^2 - d_{20,9}^2 = 0 \quad (2.22)$$

$$(x_{21} - x_9)(x_{20} - x_9) + (y_{21} - y_9)(y_{20} - y_9) + (z_{21} - z_9)(z_{20} - z_9) = 0 \quad (2.23)$$

$$(x_{21} - x_9)(y_{20} - y_9) - (x_{20} - x_9)(y_{21} - y_9) = 0 \quad (2.24)$$

The first constraint equation is the distance between the tire contacting point and spindle axis point. The second equation is formulated by the vector formed by the tire contacting point and spindle axis which is perpendicular to the vector formed by the wheel center and spindle axis. The third equation is the projection on the ground made by the vector between wheel center and spindle axis which is always co-linear with the projection made by the vector between wheel center and tire contacting.

2.6 Driven data

Due to the MacPherson suspension has 2 degrees of freedom, in order to solve the equation system to get an exclusive solution, one of the degree of freedom needs to be fixed and the other one will become the driven data. The two variables to control these two degrees of freedoms are the Z coordinates of hp6-outer ball joint and the Y coordinates of hp14-tie rod inner. Changing the Z value of outer ball joints represents the vertical movement of the suspension, theoretically speaking, changing the Z value of any hardpoints on the kuckle will have the same effect since they all connected together. Changing the Y value of tie rod inner represents the lateral motion of the steering rack when the steering wheel is turned. By fixing one variable and only control another one, the Macpherson suspension system will have a specific spatial position which means the equation system will have an exclusive solution.

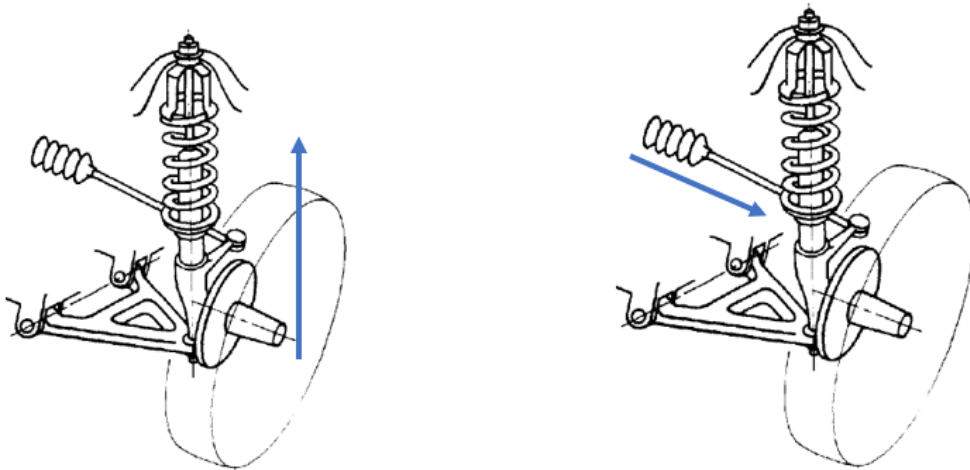


Figure 2.6: Driven data, vertical and steering simulation

The hard points which connect the suspension to the frame will be fixed during the simulation, including *hp3*, *hp4*, *hp7* and *hp14* (except for Y value in steering

simulation). These coordinates will be input data and will be considered as known variable (parameters) during the equation solving iteration.

For the equation solving, there are 23 equations to solve for 23 unknown variables, they are the Cartesian coordinates for 8 different hard points (one is given as driven data). These 8 hard points are *hp6*, *hp12*, *hp36*, *hp44*, *hp46*, *hp9*, *hp20* and *hp21*.

2.7 Equation Solving

In order to find the hardpoints coordinates values in different position, the problem is converted to a nonlinear equation system solving. The goal is to find a solution which can fulfill all the equation (equal to zero). Mathematically, the realization of the nonlinear equation solving in computer programming is based on optimization method.

2.7.1 General Structure

Theoretically, the 23 variables can be solved at the same time since there are 23 equations. However, since all the constraint equations are non-linear here, more equations means longer time and also more difficult for the computer to handle. It is easy to see that the first 14 equations can be a independent equations system to only solve 14 variables which is the coordinates of *hp6*, *hp12*, *hp36*, *hp44* and *hp46*. Hence, the solving process can be divided into two part. First, solve the first 14 equations, then by using the result to solve the last 9 equations to get the coordinates of *hp9*, *hp20* and *hp21*.

This solving process can reduce the time significantly but however will also bring a drawback because it means that the first 14 equations together has to be an independent equation system which is solvable. Apart from this, since the MacPherson suspension system has two degrees of freedom, when simulating on one degrees of freedom, the other one needs to be fixed to be able to solve the equations.

In vertical simulation, tie rod inner point is fixed to cancel steering movement then Z coordinate of outer ball joint is chosen to be the driven data. This match the real situation perfectly and will not bring any error. However, in steering simulation, the outer ball joint is fixed and Y coordinate of tie rod inner is chosen to be the driven data. This method doesn't works the same as the suspension system on the real car when front axle is steered which become a delimitation.

To get the "correct" steering simulation, the hardpoint which should be fixed should be *hp21*, tire contacting point but it exists in the second sub equation system. Fixing it means the 23 equations needs to be solve at the same time which will increase the solving time significantly as mentioned above.

2.7.2 Algorithm

In EXCEL solver add-in, there are three built-in solving algorithm, GRG Nonlinear, Simplex LP and Evolutionary.

The Simplex LP Solving Method for linear programming uses the Simplex and dual Simplex method with bounds on the variables, and problems with integer constraints use the branch and bound method, as implemented by John Watson and Daniel Fylstra. Since it can only be used for linear problems, it is fast and very robust. Also, for a linear problem, the solution will always be a globally optimum one.

GRG stands for “Generalized Reduced Gradient”. The GRG Nonlinear Solving Method for nonlinear optimization uses the Generalized Reduced Gradient (GRG2) code, which was developed by Leon Lasdon, University of Texas at Austin, and Alan Waren, Cleveland State University. In its most basic form, this solver method looks at the gradient or slope of the objective function as the input values (or decision variables) change and determines that it has reached an optimum solution when the partial derivatives equal zero. Comparing to another nonlinear solving algorithm (Evolutionary), the GRG is the fastest but it also comes with a compromise. The solution of it highly depends on the initial guess therefore it might not be the global optimum one. This algorithm will find the solution which is nearest to the initial guess then stops there. This means in order to make this algorithm precise enough, the initial guess is quite important and needs to be choose carefully, which is relatively close the the global optimum one.

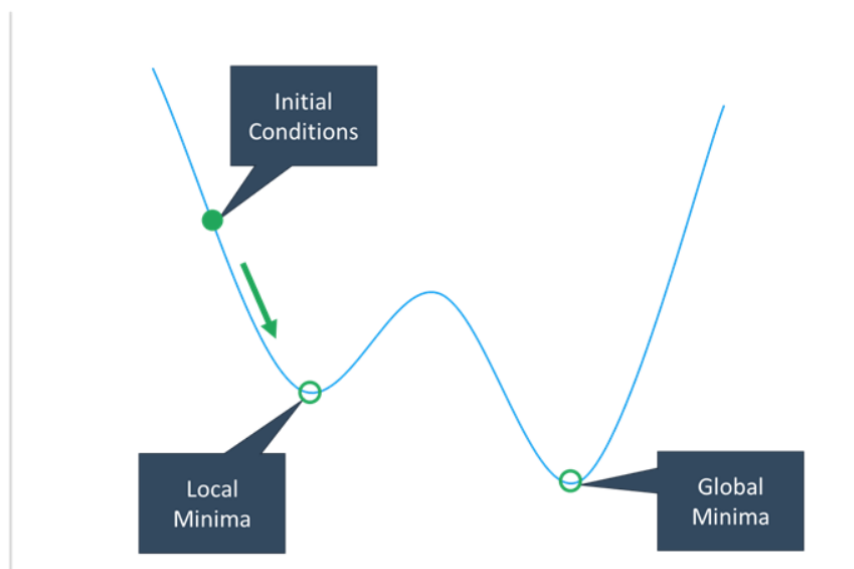


Figure 2.7: GRG Nonlinear algorithm

The Evolutionary solving method can also works for nonlinear problem, comparing to GRG Nonlinear, it is more robust but also much slower. This method was based on the Theory of Natural Selction. It works well in this case because the optimum outcome ahs been defined beforehand. In simple terms, the solver starts with a

random “population” of sets of input values. These sets of input values are plugged into the model and the results are evaluated relative to the target value. The sets of input values that result in a solution that’s closest to the target value are selected to create a second population of “offspring”. The offspring are a “mutation” of that best set of input values from the first population.

Due to this different structure, this algorithm can handle non-smooth problem as well and able to find the global optimum solution. But the time will be significantly longer than GRG Nonlinear.

In this MacPherson suspension kinematic simulation case, all the equations are nonlinear but smooth, and the initial guess can be chosen to be relatively close to the global optimum since all the hardpoint coordinates at design position is already given, the GRG Nonlinear will be the most suitable algorithm.

2.7.3 Slack Variable

In an optimization problem, a slack variable is a variable that is added to an inequality constraint to transform it into an equality constraint. In this master thesis, all the slack variables are also set to be the parameters which the solver can change itself to fulfill the optimization goal. By adding slack variable, the solver get to have more degree of freedom to perform during the optimization. The optimization goal is set to be the sum of all the slack variable. During the optimization, all the constraint equations are also set to keep (equal to zero).

- Original position

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d) - s = 0 \quad (2.25)$$

while

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d) = 0 \quad (2.26)$$

$$s = 0 \quad (2.27)$$

where x_1, y_1, z_1 represent the coordinate of *hp1* which needs to be solved. d represent the driven data, s stands for the slack variable. When the suspension is at design position, this equation will be satisfied (equal to zero) and slack variable will be zero as well.

- New position

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d') - s = 0 \quad (2.28)$$

while

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d') \neq 0 \quad (2.29)$$

$$s \neq 0 \quad (2.30)$$

When driven data d moves to a new position d' , the constraint equation lost the equality therefore the value move away from zero. Slack variable here was set to be same value as the equation value to force the overall equation back

to zero.

- Solving

Solving is to change the value of unknown variable (x,y,z) while reducing the value of slack variable to zero. When both of them are succeeded, the solution has been found.

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d') - s = 0 \quad (2.31)$$

while

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d') = 0 \quad (2.32)$$

$$s = 0 \quad (2.33)$$

The following figure shows how the iteration works.

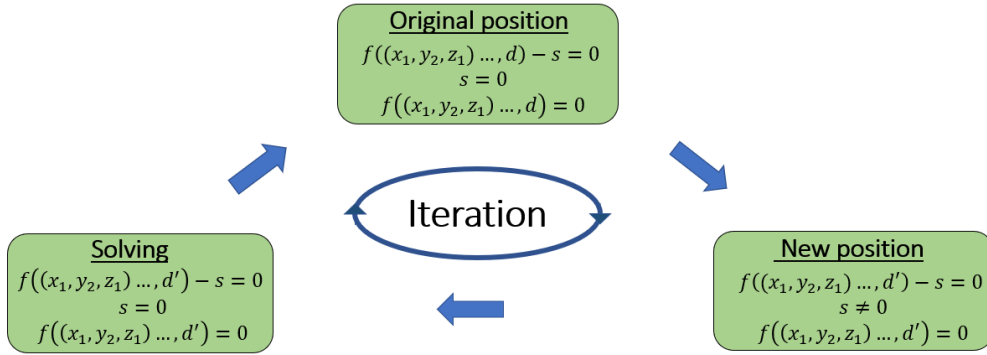


Figure 2.8: Solving iteration

2.7.4 Optimization Problem

In this master thesis, the non-linear equation was not solved directly but was converted to an optimization problem to solve due to EXCEL cannot handle complex non-linear equation system. The settings for defining the optimization problem are shown below.

- Optimization goal

$$\sum_{n=1}^{n=14} s_n^2 = 0 \quad (2.34)$$

- Variables to change

$$(x_1, y_1, z_1), (x_2, y_2, z_2) \dots (x_{14}, y_{14}, z_{14}) \quad (2.35)$$

where all variables here are assumed to be positive (only search for solution in one direction).

$$s_1, s_2, s_3 \dots s_{14} \quad (2.36)$$

- Constraints

$$f((x_1, y_1, z_1), (x_2, y_2, z_2) \dots, d') = 0 \quad (2.37)$$

2.7.5 Solving Process

This subsection explains the detail of the structure for the optimization problem.

2.7.5.1 Vertical Simulation

In the first equation system, there are 15 unknown variables(x, y, z coordinates for 5 different hardpoints) to solve. In vertical simulation, the z coordinates of outer ball joint will be driven data therefore the unknown variable will be 14. The value of this driven data will change after every solving iteration to make the suspension mechanism move to another position. Apart from this, the result of each iteration will be inserted to the next iteration as the initial guess to help the system find the global optimum solution.

With slack variable method, the slack variable will also be considered as a unknown variable which the algorithm can change to meet the optimization objective. As mentioned in last subsection, the optimization objective is the sum of the square of all the slack variable. In general, the first equation system has 28 unknown variables and 14 constraint equations to suit one optimization object. The structure of it is shown below.

First, set the initial value of the slack variable to be the equation value.

$$s_1 = (x_6 - x_3)^2 + (y_6 - y_3)^2 + (z_6 - z_3)^2 - d_{6,3}^2 \quad (2.38)$$

$$s_2 = (x_6 - x_4)^2 + (y_6 - y_4)^2 + (z_6 - z_4)^2 - d_{6,4}^2 \quad (2.39)$$

$$s_3 = (x_{12} - x_{14})^2 + (y_{12} - y_{14})^2 + (z_{12} - z_{14})^2 - d_{12,14}^2 \quad (2.40)$$

$$s_4 = (x_7 - x_{36})^2 + (y_7 - y_{36})^2 + (z_7 - z_{36})^2 - d_{7,36}^2 \quad (2.41)$$

$$s_5 = (x_6 - x_{14})^2 + (y_6 - y_{14})^2 + (z_6 - z_{14})^2 - d_{6,14}^2 \quad (2.42)$$

$$s_6 = (x_6 - x_{44})^2 + (y_6 - y_{44})^2 + (z_6 - z_{44})^2 - d_{6,44}^2 \quad (2.43)$$

$$s_7 = (x_6 - x_{46})^2 + (y_6 - y_{46})^2 + (z_6 - z_{46})^2 - d_{6,46}^2 \quad (2.44)$$

$$s_8 = (x_{46} - x_{14})^2 + (y_{46} - y_{14})^2 + (z_{46} - z_{14})^2 - d_{46,14}^2 \quad (2.45)$$

$$s_9 = (x_{44} - x_{14})^2 + (y_{44} - y_{14})^2 + (z_{44} - z_{14})^2 - d_{44,14}^2 \quad (2.46)$$

$$s_{10} = (x_{46} - x_{44})^2 + (y_{46} - y_{44})^2 + (z_{46} - z_{44})^2 - d_{44,46}^2 \quad (2.47)$$

$$s_{11} = (y_{44} - y_{46})(z_{36} - z_7) - (z_{44} - z_{46})(y_{36} - y_7) \quad (2.48)$$

$$s_{12} = (x_{44} - x_{46})(z_{36} - z_7) - (z_{44} - z_{46})(x_{36} - x_7) \quad (2.49)$$

$$s_{13} = (y_{36} - y_{44})(z_{36} - z_7) - (z_{36} - z_{44})(y_{36} - y_7) \quad (2.50)$$

$$s_{14} = (x_{36} - x_{44})(z_{36} - z_7) - (z_{36} - z_{44})(x_{36} - x_7) \quad (2.51)$$

The goal is to find a solution that all the slack variable equals to zero(or close enough to zero). The optimization goal will be to minimize the sum of all the slack variable

square.

$$h = \sum_{n=1}^{n=14} s_n^2 \quad (2.52)$$

When solving for the solution, the original constraint equation needs to be satisfied as well. This tells the algorithm try to make all the slack variable as close to zero as possible. The constraint equation is exactly the same as the original ones.

$$(x_6 - x_3)^2 + (y_6 - y_3)^2 + (z_6 - z_3)^2 - d_{6,3}^2 = 0 \quad (2.53)$$

$$(x_6 - x_4)^2 + (y_6 - y_4)^2 + (z_6 - z_4)^2 - d_{6,4}^2 = 0 \quad (2.54)$$

$$(x_{12} - x_{14})^2 + (y_{12} - y_{14})^2 + (z_{12} - z_{14})^2 - d_{12,14}^2 = 0 \quad (2.55)$$

$$(x_7 - x_{36})^2 + (y_7 - y_{36})^2 + (z_7 - z_{36})^2 - d_{7,36}^2 = 0 \quad (2.56)$$

$$(x_6 - x_{14})^2 + (y_6 - y_{14})^2 + (z_6 - z_{14})^2 - d_{6,14}^2 = 0 \quad (2.57)$$

$$(x_6 - x_{44})^2 + (y_6 - y_{44})^2 + (z_6 - z_{44})^2 - d_{6,44}^2 = 0 \quad (2.58)$$

$$(x_6 - x_{46})^2 + (y_6 - y_{46})^2 + (z_6 - z_{46})^2 - d_{6,46}^2 = 0 \quad (2.59)$$

$$(x_{46} - x_{14})^2 + (y_{46} - y_{14})^2 + (z_{46} - z_{14})^2 - d_{46,14}^2 = 0 \quad (2.60)$$

$$(x_{44} - x_{14})^2 + (y_{44} - y_{14})^2 + (z_{44} - z_{14})^2 - d_{44,14}^2 = 0 \quad (2.61)$$

$$(x_{46} - x_{44})^2 + (y_{46} - y_{44})^2 + (z_{46} - z_{44})^2 - d_{44,46}^2 = 0 \quad (2.62)$$

$$(y_{44} - y_{46})(z_{36} - z_7) - (z_{44} - z_{46})(y_{36} - y_7) = 0 \quad (2.63)$$

$$(x_{44} - x_{46})(z_{36} - z_7) - (z_{44} - z_{46})(x_{36} - x_7) = 0 \quad (2.64)$$

$$(y_{36} - y_{44})(z_{36} - z_7) - (z_{36} - z_{44})(y_{36} - y_7) = 0 \quad (2.65)$$

$$(x_{36} - x_{44})(z_{36} - z_7) - (z_{36} - z_{44})(x_{36} - x_7) = 0 \quad (2.66)$$

This algorithm and solving process requires all the variables need to be solved have to be non-negative. Due to the ISO coordinate system of the vehicle, it can only be applied on the right suspension (left suspension y coordinates will be negative).

2.7.5.2 Steering Simulation

The solving process for steering simulation is very similar to the vertical one except for two point. First, the driven data is set to be the Y coordinate of tie rod inner, which stands for the lateral movement of the steering rack. Second, since the some certain output of steering simulation needs the result for both side of the wheel (eg.akcermann) and this algorithm can only be applied to the right side, the right suspension simulation result was projected to left side to get the result.

Since in most case for a passenger cars, the two side of front suspension will be totally symmetric, this means steering the same angle to left or right will give the same result but just in different direction. When steering to right, the right wheel will be the inner wheel and left wheel will be outer wheel. Since there is no data for left wheel, instead of run the simulation for the left side, one can use the data when the

right wheel steering to left since at this time the right wheel will become outer wheel.

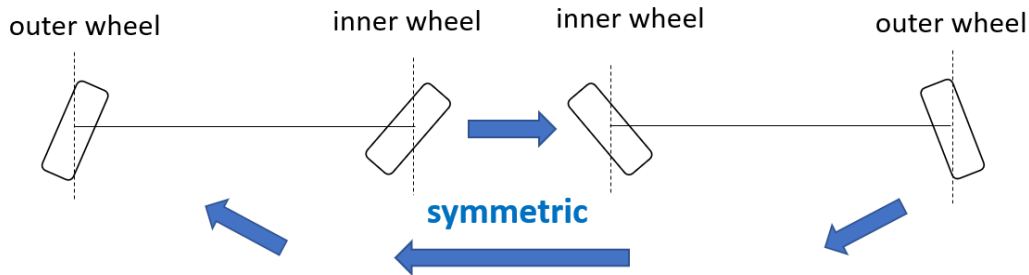


Figure 2.9: Steering geometry, symmetry

2.8 Delimitation

In the steering simulation, due to the solving process of the equation system (see next subsection), a compromise was made which lead to a delimitation.

For a real car, when it is steered, the hardpoint value which is not changing is the height of the tire contacting point (Z coordinate). However, in this master thesis, the actual fixed value is the Z coordinate of *hp6* ball joint. Since there are already two constraint equation to control the position of ball joint, by fixing the Z value, this hard point is fixed completely, cannot move at all.

This bring a difference between the simulation here and the real situation. In order to verify this difference is within the acceptable range, another simulation is made in MATLAB by fixing the Z value of *hp21* tire contacting. By observing the movement of *hp6*-ball joint in this condition, one can conclude that how much difference it is between these two simulation set up. For better understanding, the kingpin inclination angle and caster angle are chosen to represent the movement of ball joint since they are widely used suspension design parameter.

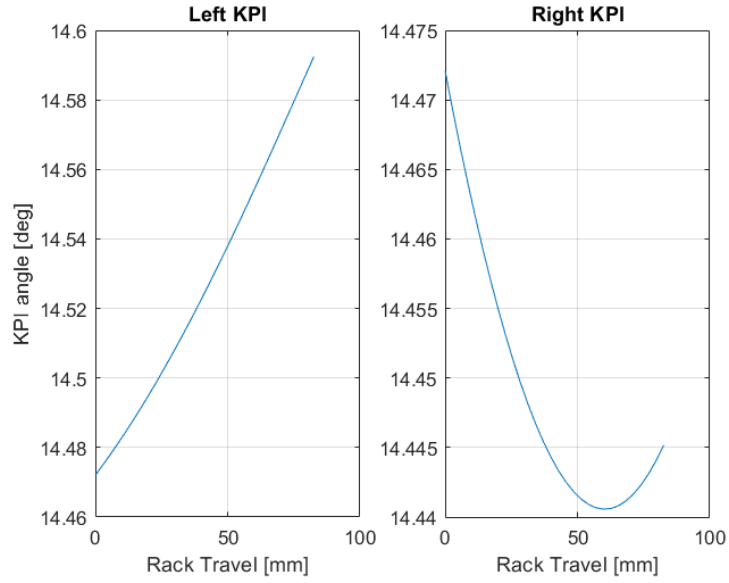


Figure 2.10: Kingpin inclination angle

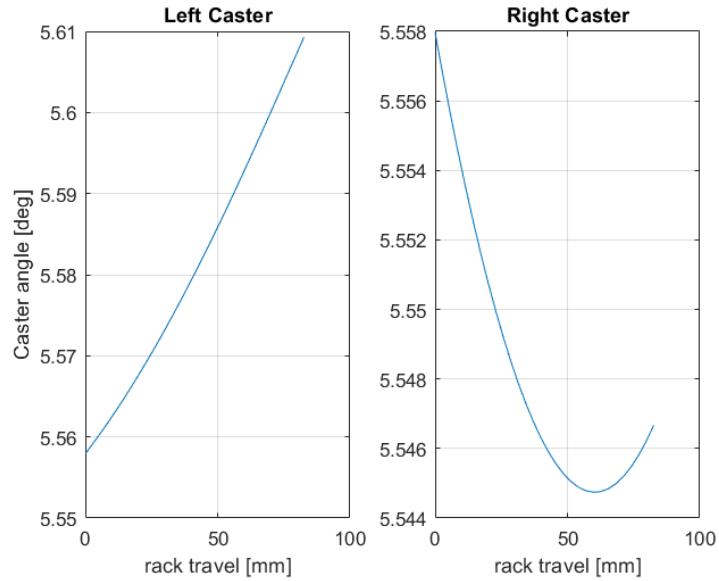


Figure 2.11: Caster angle

The simulation condition is full steering simulation from right to left (+82.6mm rack travel). From the result it can be observed that, both the kingpin inclination angle and caster angle variation is small. To quantify the variation, percentage value is calculated.

$$L_{KPI} = (14.59 - 14.47)/14.59 \times 100\% = 0.14\% \quad (2.67)$$

$$R_{KPI} = (14.47 - 14.44)/14.47 \times 100\% = 0.21\% \quad (2.68)$$

$$L_{Caster} = (5.609 - 5.558)/5.609 \times 100\% = 0.91\% \quad (2.69)$$

$$R_{Caster} = (5.558 - 5.545)/5.558 \times 100\% = 0.23\% \quad (2.70)$$

Since all the variation is below 1%, this limitation is within acceptable range.

2.9 Output Parameter

The interested output parameters required from CEVT company are listed below. For vertical and steering simulation, due to the different focus, the output parameters are a bit different as well

2.9.1 Vertical Simulation

The following table lists the output parameters which require graph as result.

Table 2.2: Vertical Simulation Output Parameter, graph

Output Parameter	Unit
Camber Angle	Degrees
Toe Angle	Degrees
Track width	mm
Wheel center longitudinal displacement	mm
Roll center height	mm
Kingpin Inclination Angle	Degrees
Caster Angle	Degrees
Scrub Radius	mm
Caster Trail	mm
Installation Ratio	-

The following table lists the output parameters which requires value as result.

Table 2.3: Vertical Simulation Output Parameter, value

Output Parameter	Unit
Camber Angle gain	degrees/m
Toe Angle gain	Degrees/m
Track width variation	mm/m
Wheel center longitudinal displacement gain	mm/m
Roll center height gain	mm/m
Kingpin Inclination Angle gain	Degrees/m
Caster Angle gain	Degrees/m
Scrub Radius gain	mm/m
Caster Trail	mm/m
Anti lift	%
Anti dive	%

For all the "gain" value, +10mm and -10mm vertical wheel travel are the calculating position.

2.9.2 Steering Simulation

The following table lists the output parameters which requires graph as result.

Table 2.4: Steering Simulation Output Parameter

Output Parameter	Unit
Camber Angle	Degrees
Toe Angle	Degrees
Track width	mm
Wheel center longitudinal displacement	mm
Trackrod to steering arm angle	Degrees
Ackermann	-

For steering simulation, there is no specific required output parameter as value.

2.9.3 Parameter Calculation

In this subsection, the equation used for calculating all the output parameter are introduced. All the nomenclature here follows international standard ISO 8855:2011[1].

- Camber Angle

Camber angle is calculated on front view, based on the Y and Z coordinate of hardpoint wheel center and spindle axis. In ISO 8855:2011[1], it is denoted by φ .

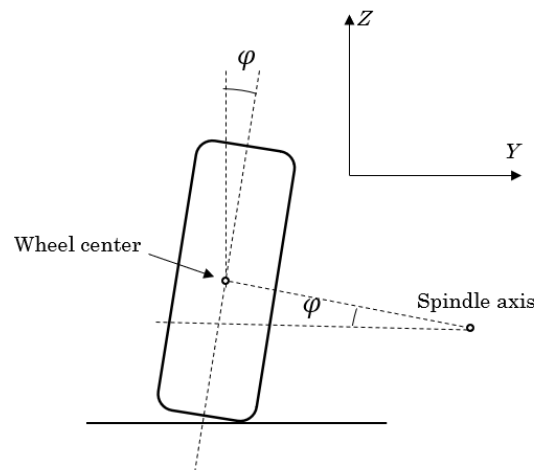


Figure 2.12: Camber angle, calculation

According to the figure, camber angle can be calculated as:

$$\arctan \varphi \quad (2.71)$$

- Toe Angle

Toe angle is calculated on the top view, based on the X and Y coordinate of hardpoint wheel center and spindle axis. In ISO 8855:2011[1], it is denoted by δ .

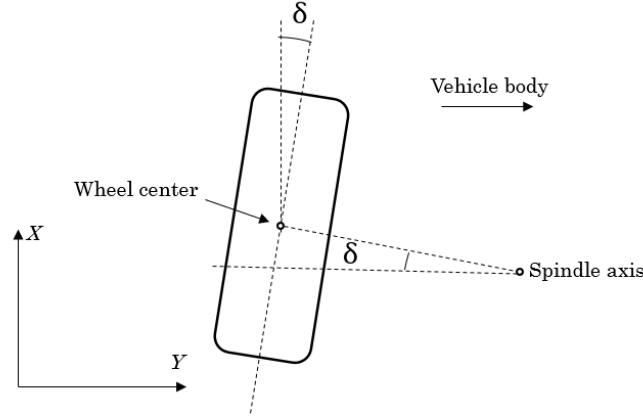


Figure 2.13: Toe angle, calculation

According to the figure above, toe angle can be calculated as:

$$\arctan \delta \quad (2.72)$$

- Track width

Track width is the distance between the wheel center in two side of the suspension measured in Y direction.

$$t = |Y_{LWC} - Y_{RWC}| \quad (2.73)$$

- Wheel center longitudinal displacement

This parameter is the distance between the current wheel center position and the original one (at design position) measure in X direction. It can be calculated based on the following equation.

$$X_{WC} - X'_{WC} \quad (2.74)$$

where X'_{WC} stands for the X coordinate for wheel center at design position.

- Trackrod to steering arm angle

This parameter requires to find the perpendicular point on the kingpin axis first. Then the angle can be calculated based on spatial vector method.

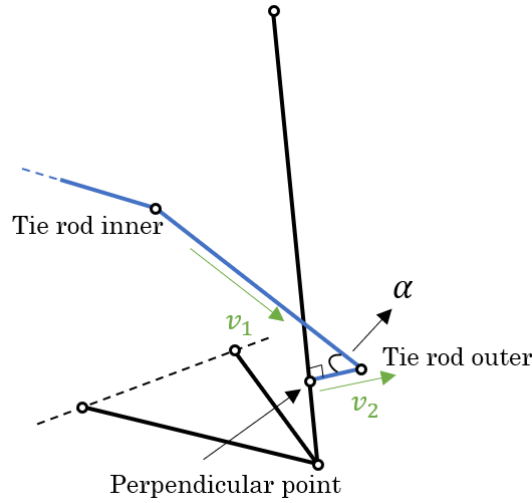


Figure 2.14: Trackrod to steering arm angle calculation

$$\alpha = \arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|}\right) \quad (2.75)$$

- Percent Ackermann

Percent Ackermann describes the difference between the steer angle of the inner wheel and the outer wheel between the real situation and ideal one. The ideal Ackermann only depends on the trackwidth and the wheel base.

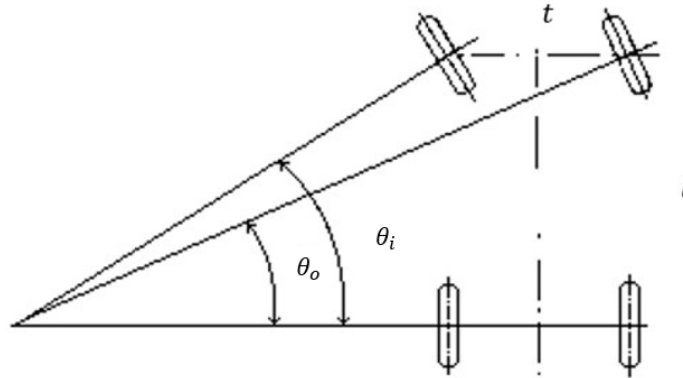


Figure 2.15: Ideal Ackermann steering geometry

For ideal Ackermann steering geometry, the inner and outer wheel follow the relationship as the figure above. θ_i and θ_o indicate the steer angle of the wheel, B is the track width and L is the wheel base. The following equation can be derived from the figure.

$$\cot\theta_i = \cot\theta_o - t/L \quad (2.76)$$

By moving θ_i to the left side:

$$\theta_{it} = \operatorname{arccot}(L - t \cdot \theta_o / \tan\theta_o \cdot L) \quad (2.77)$$

This equations indicates that for ideal Ackermann geometry, for a certain outer wheel steer angle, there will always be a correspond inner wheel steer angle. Their relationship will follows the equation above (θ_{it} indicates the ideal inner wheel steer angle). In this paper, percent Ackermann will be a percentage value and indicates as K .

$$K = \left(\frac{\theta_i - \theta_o}{\theta_{it} - \theta_o} \right) \times 100\% \quad (2.78)$$

Here, θ_i and $\theta - o$ indicate the "real" steer angle from inner and outer wheel. If $K = 100\%$, then the steering geometry is ideal, if $K = 0\%$, then it is a parallel steering geometry, if K is a negative value, then it is a anti-Ackermann geometry.

- Roll Center Height

The roll center height in this master thesis is the kinematics roll center height which calculated on the front view. Instantaneous center position needs to be find before calculating roll center height. Vector method can be applied here.

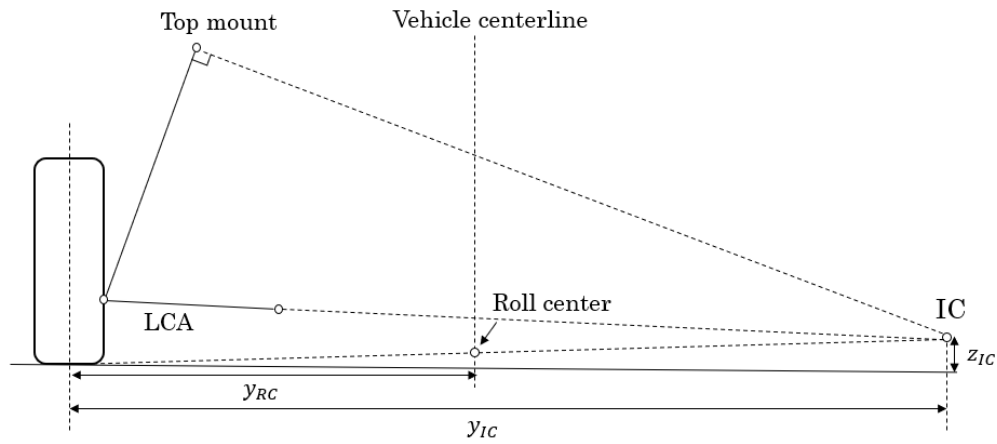


Figure 2.16: Roll center calculation

$$RCH = \frac{z_{IC} \cdot y_{RC}}{y_{IC}} \quad (2.79)$$

- Kingpin Inclination Angle

Kingpin inclination angle is calculated on front view based on the Y and Z coordinates of top mount and ball joint hardpoint. In ISO 8855:2011[1], it is denoted by σ .

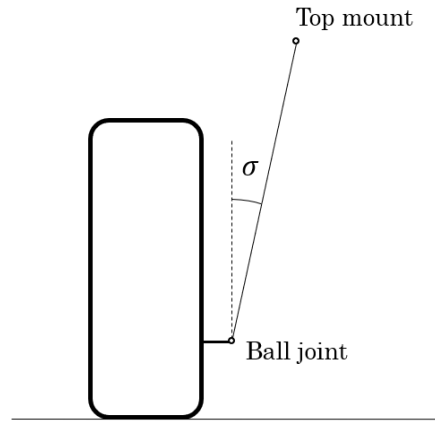


Figure 2.17: Kingpin inclination angle calculation

According to the figure above, kingpin inclination angel can be calculated as:

$$\arctan \sigma \quad (2.80)$$

- Caster Angle

Caster angle is calculated on side view based on the X and Z coordinates of top mount and ball joint hardpoints. In ISO 8855:2011[1], it is denoted by τ .

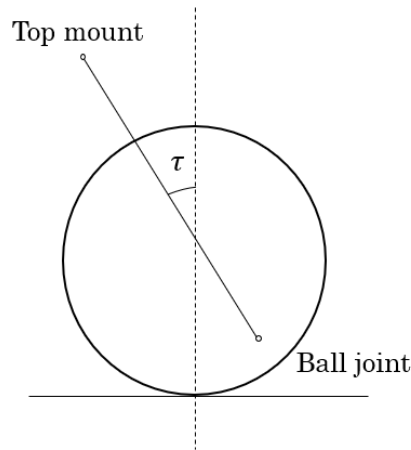


Figure 2.18: Caster angle calculation

According to the figure above, caster angel can be calculated as:

$$\arctan \tau \quad (2.81)$$

- Scrub Radius

Scrub radius is calculated on the front view based on the Y and Z coordinate of the hardpoint top mount and ball joint. In ISO 8855:2011[1], it is denoted by r_k .

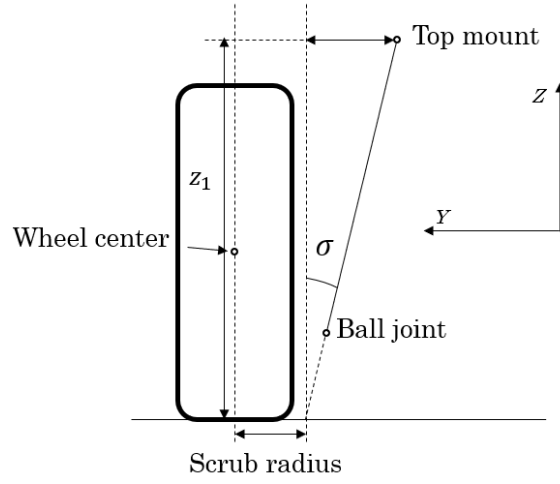


Figure 2.19: Scrub radius calculation

$$r_k = y_{wc} - (\tan \sigma \cdot z_1 + y_{tm}) \quad (2.82)$$

where y_{wc} represents the y coordinates for hardpoint wheel center and y_{tm} represents the y coordinate for hardpoint top mount.

- Caster trail

Caster trail is calculated on the side view based on the Y and Z coordinate of the hardpoint top mount and ball joint. In ISO 8855:2011[1], it is denoted by n_k .

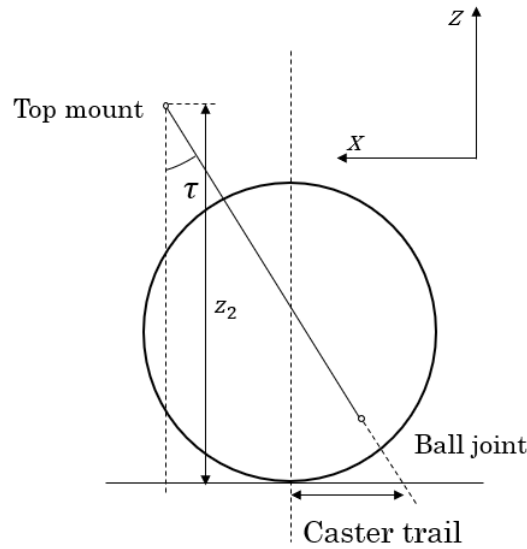


Figure 2.20: Caster trail calculation

$$n_k = x_{wc} - (x_{tm} - z_2 \cdot \tan \tau) \quad (2.83)$$

- Installation ratio

Installation ratio is calculated by using the wheel vertical movement divided by the damper displacement.

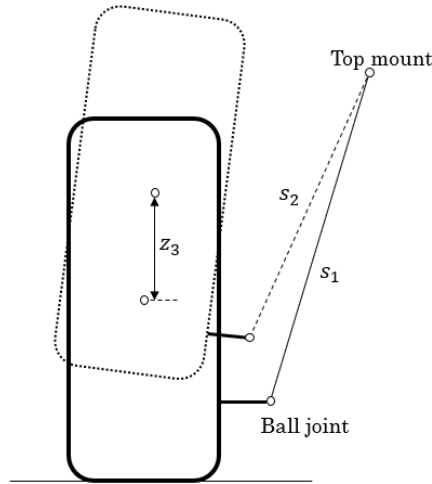


Figure 2.21: Installation ratio calculation

According to the figure above, installation ratio can be calculated as:

$$IR = \frac{z_3}{s_1 - s_2} \quad (2.84)$$

- Anti-dive & Anti-lift

Anti-dive and Anti-lift are calculated based on the tangent value of the line between the instantaneous center to the corresponding force applying point. The only difference between them is where the line was connected. For anti-dive, since the braking force comes from ground therefore the line was connected to the contacting point. However, for anti-lift since the driving force comes from the drive shaft, so the line was connected to the wheel center point.

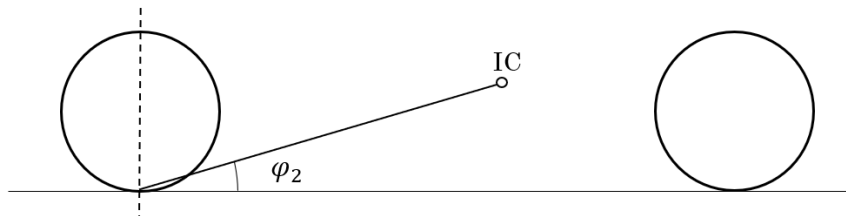


Figure 2.22: Anti-dive calculation

$$AD = \tan \varphi_2 \quad (2.85)$$

The only difference between them is where the line was connected. For anti-dive, since the braking force comes from ground therefore the line was connected to the contacting point. However, for anti-lift since the driving force comes from the drive shaft, so the line was connected to the wheel center point.

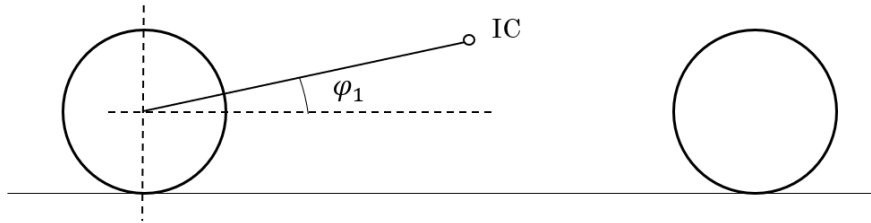


Figure 2.23: Anti-lift calculation

$$AL = \tan\varphi_1 \quad (2.86)$$

3

Software Implementation

Due to its widely application and as a common used office tool, EXCEL was chosen to be the software platform for this master thesis. With the help from the SOLVER add-in from itself, EXCEL is able to convert complicate nonlinear equation solving problem to an optimization problem then find the solution for it.

The coding in EXCEL was based on Visual Basic Application (VBA, EXCEL coding language) and developed in module which is the EXCEL official script.

3.1 Excel Introduction

Microsoft EXCEL is a spreadsheet office tool developed by Microsoft for different operating system and platform. Since version 5 which released in 1993, EXCEL now has become the most widely used office tool. For this master thesis, all the EXCEL programming was based on EXCEL 2016 version.

Visual Basic for Applications (VBA) is an implementation of Microsoft's event-driven programming language Visual Basic 6. It is a powerful tool to enable EXCEL to have more functions than the original spreadsheet. In this master thesis, all the code was implemented in "module", one module is an individual script which has its own functions.

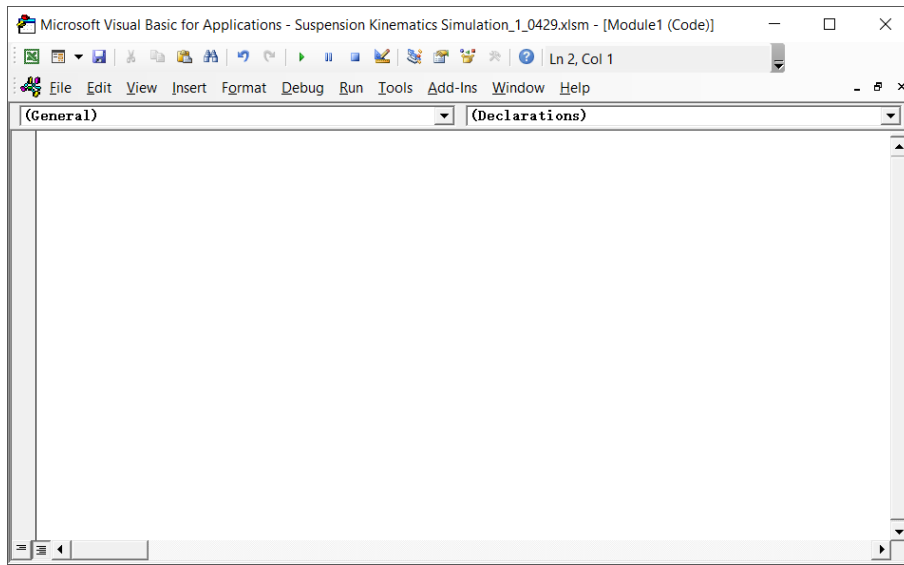


Figure 3.1: Excel Visual Basic interface

3.2 Overall structure

In general, this suspension kinematics programme is a Microsoft Excel Macro-Enabled Worksheet (.xlsm) file. It consists of 8 basic different worksheets, each worksheets has its own function and goal (there will be new worksheets after each simulation).

Table 3.1: Worksheets list

Worksheets Name	Function
Front cover	Cover page, instructions
Input	Input data, button for running simulation
Simulation	Equation solving
Intermediate	Save indermediate data
Vertical Output	Result for last vertical simulation
Steering Output	Result for last steering simulation
Vertical figure	Output parameter figure for vertical simulation
Steering figure	Output parameter figure for steering simulation

Import, export and simulation are controlled by the buttons on the *input* worksheet. After each of the simulation, the hardpoints coordinates for different steps will be stored in *intermediate* worksheet and the final output parameters will be calculated and saved in *vertical output* or *steering output* depends on the simulation itself. Apart from this, when the simulation is finished, a new worksheet will be created to store all the output parameters together with the input data. The results are given as figures and values which can be found in *vertical figure* and *steering figure* worksheets. Multiple simulations result can be plotted at the same time for comparison as well.

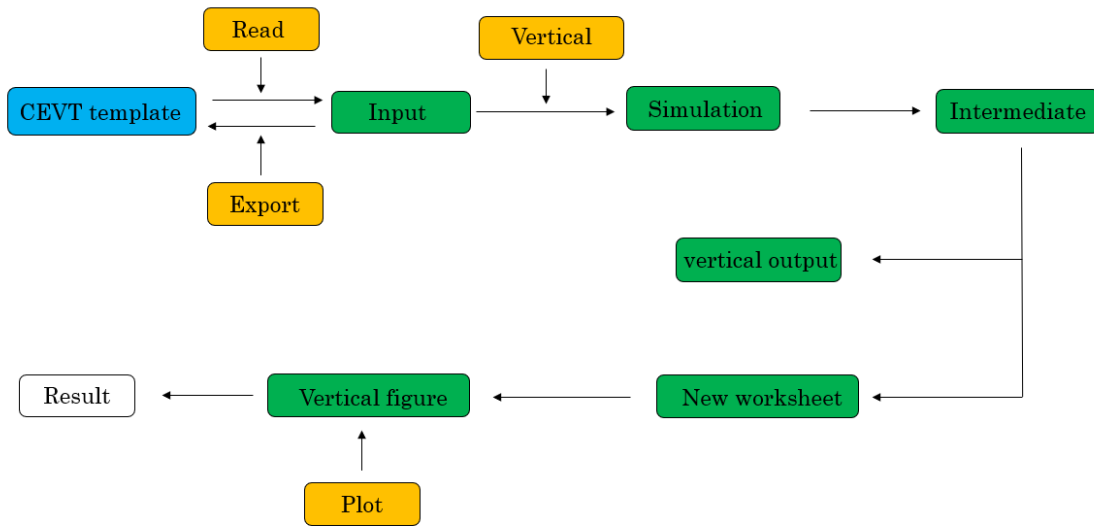


Figure 3.2: Flow chart, vertical simulation

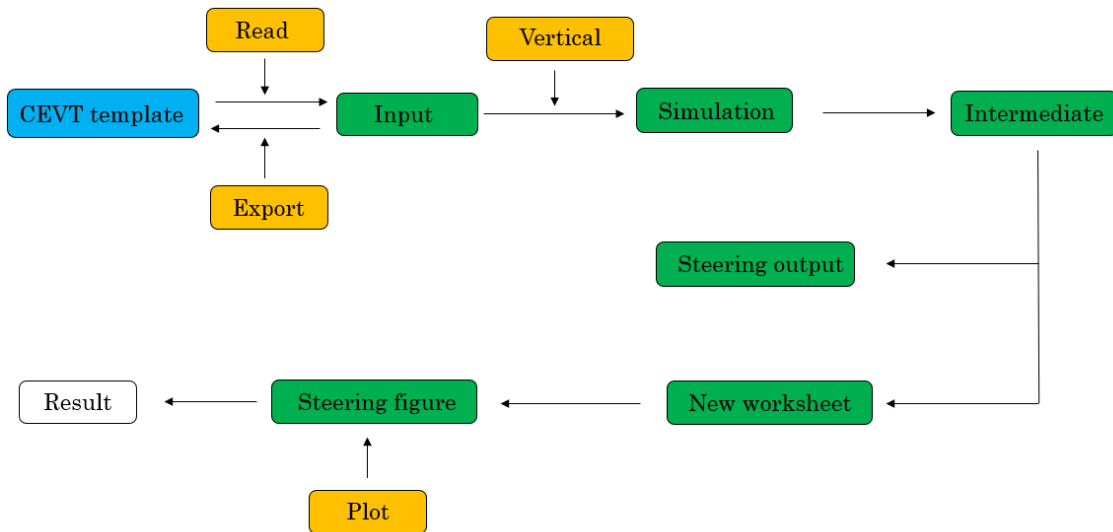


Figure 3.3: Flow chart, steering simulation

In the above flow chart figure, the green block represent the worksheet, orange block represent the control button. The control buttons only exist in *input*, *vertical figure* and *steering figure* worksheets. Since for a normal simulation, there are the only three worksheets which users need to touch. Hence, to avoid unnecessary confusion, in the default settings, all the other worksheets are hidden.

3.3 Worksheet Introduction

This section briefly introduced the main purpose and function for each of the worksheet.

3.3.1 Front Cover

This worksheets includes all the user instruction. It is necessary for the new users to read before using this software.

3.3.2 Input

The beginning of the simulation is the *input* worksheet. It consists of three main area, control button area, simulation setting parameter table area and input hardpoints coordinates area.

All the buttons for running the simulation locates in the *input* worksheet, there are four buttons in total.

Control Button Area	
Load data from template	READ
Start vertical simulation	VERTICAL
Start steering simulation	STEERING
Export output back to	EXPORT

Figure 3.4: Control button, input worksheet

- Read
Read button reads the input data (the hardpoints coordinates at the design position) from CEVT template (see Appendix A) EXCEL file. After reading is finished, CEVT template will close automatically itself.
- Vertical
Run vertical simulation.
- Steering
Run steering simulation
- Export
Export the current hardpoints coordinates in *input* worksheet back to the CEVT template file and saves there. It will automatically find a spare space to save so the original information will no be covered

Before running the simulation, the simulation setting parameter table need to be filled. Each of the parameter here controls a certain property of the simulation. Leaving blank here will potentially cause the simulation fail.

Simulation setting parameter	
Bump [mm]	80
Rebound [mm]	-110
Step [-]	80
Rack travel [mm]	82.6
Tire Loaded Radius [mm]	331.8
tolerance	0.00000001
wheel base [mm]	2730
static camber [deg]	-0.9
static toe (in +, out -) [deg]	0.04
max time [s]	100
max iteration [-]	1000000

Figure 3.5: Simulation setting parameter, example

The definition for these parameters are shown below.

- **Bump [mm]**
The bump travel for the vertical simulation. Measured at wheel center hard-point, starting from design position to the highest position in Z direction.
- **Rebound [mm]**
The rebound travel for the vertical simulation. Measured at wheel center hard-point, starting from design position to the lowest position in Z direction.
- **Step [-]**
The total steps for the simulation. It defines the number of different positions which are solved in the simulation. Larger steps will give smoother curves but will also increase simulation time. Too small steps may cause failure of the solving process due to the initial guess is too far from the solution.
- **Rack travel [mm]**
The total steering rack travel, measured from the middle position to one end side position.
- **Tire loaded radius [mm]**
The radius of the tire under standard design load.
- **Tolerance [mm]**
A number between 0 (zero) and 1 that specifies the degree of precision with which constraints (including integer constraints) must be satisfied. The default precision is 0.000001. A smaller number of decimal places (for example, 0.0001) indicates a lower degree of precision. In general, the higher the degree of precision you specify (the smaller the number), the more time Solver will take to reach solutions.

3. Software Implementation

- Wheel base [mm]
The wheel base of the vehicle.
- Static camber angle [deg]
The camber angle at design position
- Static toe angle [deg]
The toe angle at design position. Positive value indicates toe in while negative value indicates toe out.
- Max time [s]
The maximum time for the whole simulation. This parameter is to prevent the solving from infinitive searching. When the simulation reaches this time it will stops itself and saves the current information.
- Max iteration [-]
The max iteration number at one solving step. This parameter is to prevent the solving from infinitive searching. When the solver reaches this value it will stops itself and saves the current information then continue next step.

After the simulation setting parameters are filled, the simulation is ready to start. By clicking the READ button, the hardpoints coordinates area will be filled with input data from template file.

Wheel Suspension Hard Point	HP no		X	Y	Z
Bush Front	3		1731.2	-373	361
Bush Rear	4		2015	-408	368
Ball Joint, control arm	6		1700.4	-766	332.5
Top Mount	7		1766.86	-589.724	1015.47
Wheel Center ETXX	9, ET		1713.32	-793.8	447.047
Interface wheel	11		1713.32	-845.793	447.864
Interface driveshaft	10		1713.32	-775.802	446.764
Ball Joint, tie rod outer	12		1864.5	-732.518	410.6
Ball Joint, tie rod inner	14		1908	-370	423.5
Strut aligning point	36		1710.8	-649	485.717
Bumpstop, upper end	41		1763.2	-593.595	980.879
Bumpstop, lower end	42		1756.26	-600.938	915.253
Bumpstop, interface	43		1753.46	-603.891	888.864
Damper, bottom	44		1715.15	-644.405	526.783
Interface damper	45		1719.54	-639.76	568.293
Coil spring, upper end	58		1765.92	-590.719	1006.58
Coil spring, lower end	59		1753.6	-643.5	834.5
Antirollbar mounting, LHS	61		2043	-300.2	420
Antirollbar mounting, RHS	62		2043	300.2	420
Ball joint, drop link ARB	63		1786.51	-580	461.02
Ball joint, drop link suspension	65		1778.99	-591.086	710.2

Figure 3.6: *input* worksheet, hardpoints area

By clicking the Vertical or Steering button, corresponding simulation will start.

3.3.3 Simulation

simulation worksheets is the place where this software prepares and actually solves all the equations. It includes all the mathematical parts for this software.

Before the specific solving process, some preparation work needs to be finished. The most important one is calculating all the distance constraint value then save them. They can be found in the distance constraint table in this worksheet.

Distance constraint	
Distance square value	156209.89
	228397.41
	133477.9603
	244480.8025
	52748.51618
	75664.36029
	34149.46432
	2444.808025
	43568.99118
	56788.06257
	14060.78897
	28679.83287
	40570.49801
	10000
	110091.2319
	18143.50863
	25548.80629
	9054.836523
	20004.81393
	3634.128347

Figure 3.7: Distance constraint, *simulation* worksheet

On the top left of this worksheets locates the variable area. This area shows the initial guess for each solving and the solution result. The information here updates between each steps.

3. Software Implementation

hard point name		to solve	initial guess
hp6, ball joint	x	1698.733	1700.4
	y	763.51701	766
	z	414.55128	332.5
hp44, damper bottom	x	1715.686	1715.145754
	y	640.0105	644.405025
	z	605.39182	526.7826317
hp2, auxiliary	x	1721.7641	1720.317278
	y	634.03796	638.9369225
	z	654.09699	575.6514685
hp12, tie rod outer	x	1863.8883	1864.5
	y	727.07712	732.518
	z	486.9669	410.6
hp36, strut aligning point	x	1706.0804	1715.145754
	y	649.44936	644.405025
	z	528.41926	526.7826317
hp9, wheel center	x	1714.3465	1713.32
	y	790.11774	793.7996299
	z	526.99632	447.047122
hp20, spindle axis	x	1712.9843	1713.389813
	y	690.15657	693.811991
	z	524.56541	445.4763902
hp21, tire contacting	x	1714.4564	1713.32
	y	798.18275	793.7996299
	z	195.29438	115.2471342

Figure 3.8: Variables, *simulation* worksheet,

All the equations are written in this worksheets as well. They are not implemented in module but directly inserted in the worksheets. There are 23 equations in total.

f(x)
$= (x_{1-rp1_x})^2 + (x_{2-rp1_y})^2 + (x_{3-rp1_z})^2 - d_{51}$
$= (x_{1-rp2_x})^2 + (x_{2-rp2_y})^2 + (x_{3-rp2_z})^2 - d_{52}$
$= (x_{10-rp3_x})^2 + (x_{11-rp3_y})^2 + (x_{12-rp3_z})^2 - d_{83}$
$= (x_{13-rp4_x})^2 + (x_{14-rp4_y})^2 + (x_{15-rp4_z})^2 - d_{94}$
$= (x_{4-x_1})^2 + (x_{5-x_2})^2 + (x_{6-x_3})^2 - d_{65}$
$= (x_{7-x_1})^2 + (x_{8-x_2})^2 + (x_{9-x_3})^2 - d_{75}$
$= (x_{10-x_1})^2 + (x_{11-x_2})^2 + (x_{12-x_3})^2 - d_{85}$
$= (x_{7-x_4})^2 + (x_{8-x_5})^2 + (x_{9-x_6})^2 - d_{76}$
$= (x_{10-x_4})^2 + (x_{11-x_5})^2 + (x_{12-x_6})^2 - d_{86}$
$= (x_{10-x_7})^2 + (x_{11-x_8})^2 + (x_{12-x_9})^2 - d_{87}$
$= (x_{8-x_5}) * (x_{15-rp4_z}) - (x_{9-x_6}) * (x_{14-rp4_y})$
$= (x_{7-x_4}) * (x_{15-rp4_z}) - (x_{9-x_6}) * (x_{13-rp4_x})$
$= (x_{14-x_8}) * (x_{15-rp4_z}) - (x_{15-x_9}) * (x_{14-rp4_y})$
$= (x_{13-x_7}) * (x_{15-rp4_z}) - (x_{15-x_9}) * (x_{13-rp4_x})$
$= (x_{1-x_16})^2 + (x_{2-x_17})^2 + (x_{3-x_18})^2 - d_{510}$
$= (x_{4-x_16})^2 + (x_{5-x_17})^2 + (x_{6-x_18})^2 - d_{610}$
$= (x_{7-x_16})^2 + (x_{8-x_17})^2 + (x_{9-x_18})^2 - d_{710}$
$= (x_{1-x_19})^2 + (x_{2-x_20})^2 + (x_{3-x_21})^2 - d_{511}$
$= (x_{16-x_19})^2 + (x_{17-x_20})^2 + (x_{18-x_21})^2 - d_{1011}$
$= (x_{10-x_19})^2 + (x_{11-x_20})^2 + (x_{12-x_21})^2 - d_{811}$
$= (x_{22-x_16})^2 + (x_{23-x_17})^2 + (x_{24-x_18})^2 - d_{1012}$
$= (x_{22-x_16}) * (x_{19-x_16}) + (x_{23-x_17}) * (x_{20-x_17}) + (x_{24-x_18}) * (x_{21-x_18})$
$= (x_{22-x_16}) * (x_{20-x_17}) - (x_{19-x_16}) * (x_{23-x_17})$

Figure 3.9: Equations, *simulation* worksheet

3.3.4 Intermediate

intermediate worksheets temporarily stores the all the hardpoints coordinates in each position. The data here will be used later for calculating the actual output parameters.

3.3.5 Vertical output & Steering output

These two worksheets both calculates the output parameters based on the data from *intermediate* worksheet and stores them. Depends on the simulation type (vertical or steering), the cooredsponding worksheet will be used. This worksheet will only save the result from the last simulation since the output parameters will be copied to a new worksheet with specific name after each of teh simulation. All the data in these two worksheets will be cleared in the beginning of new simulation to avoid confusion between different simulation.

3.3.6 Vertical figure & Steering figure

These two worksheets are figure control sheets. This is the place where the users to compare and observe the simulation result. It has three main areas: control area, figure area and parameter table area.

- Control area

In this area, the users can choose from different simulation results and decide which are the ones they are interested in.

Simulation Name	User Comment	Type "1" to plot	plot	delete
15 16 54v	compare	1		
15 22 48v	compare	1		

Figure 3.10: Control area, *figure* worksheet

The name of the simulation result will be generated automatically by the software itself follows the *hour minute second* format. The letter *v* indicates vertical simulation and letter *s* indicates steering simulation respectively. The comment will be decided by the users, a message window box will appears after the simulation finished and users can type some words which represent the key words for this simulation in order to distinguish between each other later. The logical value 1 means plot this simulation result. If more than one simulation results are chosen to plot, the will be plot together with cooresponding legend.

After deciding which simulation result to plot, by clicking *plot* button, chosen simulation result will be shown. By clicking *delete* button, all the figures will be deleted.

- Figure area

This is the area where all the figures will be plotted. Due to different re-

3. Software Implementation

quirement for the vertical and steering simulation, the number and type of the figures for two different simulations are a bit different as well.

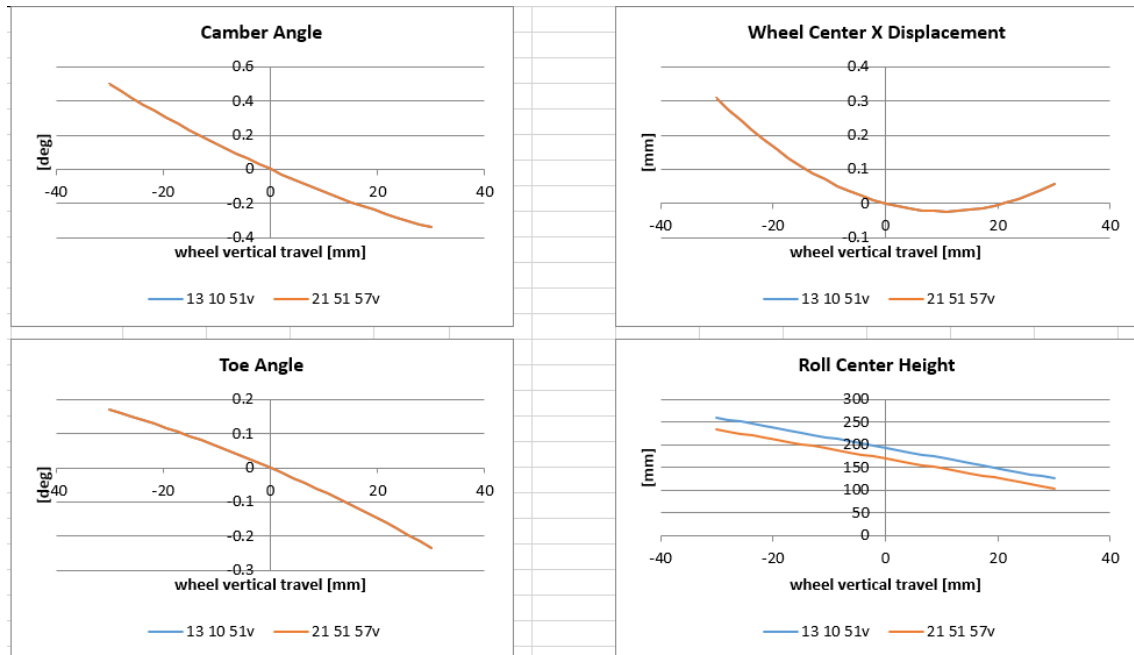


Figure 3.11: Figure area, *figure* worksheet, example

- Parameter table

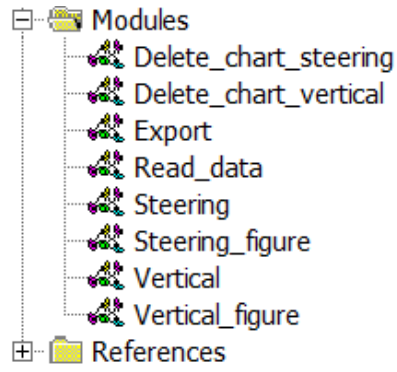
This table is to show all the parameters required as value which in most case is the gain value. It locates on the right side of the figure area and only exist in *vertical figure* worksheet.

	15 16 54v	15 22 48v
camber gain [deg/m]	-14.2656	-14.2656
toe gain [deg/m]	-6.75019	-6.75019
track width gain [mm/m]	87.1321	87.1321
wheel center X gain [mm/m]	-4.89908	-4.89908
rch gain [mm/m]	-2286.36	-2286.36
kpi angle gain [deg/m]	25.8873	25.8873
caster angle gain [deg/m]	10.8017	10.8017
caster trail gain [mm/m]	69.5576	69.5576
scrub radius gain [mm/m]	-133.246	-133.246
anti-dive (tan)	0.06999	0.06999
anti-lift (tan)	0.00868	0.00868

Figure 3.12: Parameter table, *figure* worksheet, example

3.4 Module

A module is an individual script in EXCEL VBA which can carry out certain function. It is connected directly to one button.

**Figure 3.13:** Module

In this suspension kinematics program, there are 8 modules in total which indicates 8 buttons. They are listed below.

Table 3.2: Module list

Module	Connected button	Location (worksheet)
Delete_chart_steering	Delete	<i>steering figure</i>
Delete_chart_vertical	Delete	<i>vertical figure</i>
Export	Export	<i>input</i>
Read_data	Read	<i>input</i>
Steering	Steering	<i>input</i>
Steering_figure	Plot	<i>steering figure</i>
Vertical	Vertical	<i>input</i>
Vertical_figure	Plot	<i>vertical figure</i>

4

Result

This chapter shows the result from the suspension kinematics programme for all the output parameter. Then validation work is done based on the comparison between this programme and ADAMS/CAR.

4.1 Simulation Set up

4.1.1 Input data

Input data indicates the hard points coordinates at design position. The input data used in this chapter is provided by CEVT company.

Wheel Suspension Hard Point	HP no		X	Y	Z
Bush Front	3		1731.2	-373	361
Bush Rear	4		2015	-408	368
Ball Joint, control arm	6		1700.4	-766	332.5
Top Mount	7		1766.861	-589.724	1015.471
Wheel Center ETXX	9, ET		1713.32	-793.8	447.0471
Interface wheel	11		1713.32	-845.793	447.8639
Interface driveshaft	10		1713.32	-775.802	446.764
Ball Joint, tie rod outer	12		1864.5	-732.518	410.6
Ball Joint, tie rod inner	14		1908	-370	423.5
Strut aligning point	36		1713.648	-645.989	512.629
Bumpstop, upper end	41		1763.2	-593.595	980.8788
Bumpstop, lower end	42		1756.255	-600.938	915.2526
Bumpstop, interface	43		1753.463	-603.891	888.8637
Damper, bottom	44		1713.648	-645.989	512.629
Interface damper	45		1719.539	-639.76	568.2932
Coil spring, upper end	58		1765.92	-590.719	1006.575
Coil spring, lower end	59		1753.6	-643.5	834.5
Antirollbar mounting, LHS	61		2043	-300.2	420
Antirollbar mounting, RHS	62		2043	300.2	420
Ball joint, drop link ARB	63		1786.506	-580	461.02
Ball joint, drop link suspension	65		1778.986	-591.086	710.2

Figure 4.1: Input data

4.1.2 Simulation Setting Parameters

The following table lists all the required simulation parameters and their value.

Table 4.1: Simulation setting parameters

Parameter Name	Value	Unit
Bump travel	80	mm
Rebound travel	-110	mm
Step	50	-
Rack travel	82.6	mm
Tire loaded radius	331.8	mm
Wheel base	2730	mm
Static camber angle	-0.9	deg
Static toe angle	0.04	deg

4.2 Simulation Result

4.2.1 Vertical simulation

The following table and figures shown the graph results from vertical simulation, all simulation is based on the front right suspension.

4.2.1.1 Parameter table

The output parameter which are required as value, here it has been converted and shown in a table format for better understanding.

Table 4.2: Output parameter table

Parameter Name	Value	Unit
Camber gain	-14.2656	deg/m
Toe gain	-6.7502	deg/m
Trackwidth gain	87.1321	mm/m
Wheel center longitudinal displacement gain	-4.8991	mm/m
Roll center height gain	-2286.36	mm/m
Kingpin inclination angle gain	25.8873	deg/m
Caster angle gain	10.8017	deg/m
Caster trail gain	69.5576	mm/m
Scrub radius gain	-133.246	mm/m
Anti-dive (tan)	0.0699	-
Anti-lift (tan)	0.0087	-

4.2.1.2 Graph result

This subsection shows the result as graph for vertical and steering simulation. One more figure "Simulation Error" was added here which is not required by CEVT. It is the sum of all the equation value which represents the quality of the solving. Design engineers can use it to check the convergence of the equation solving.

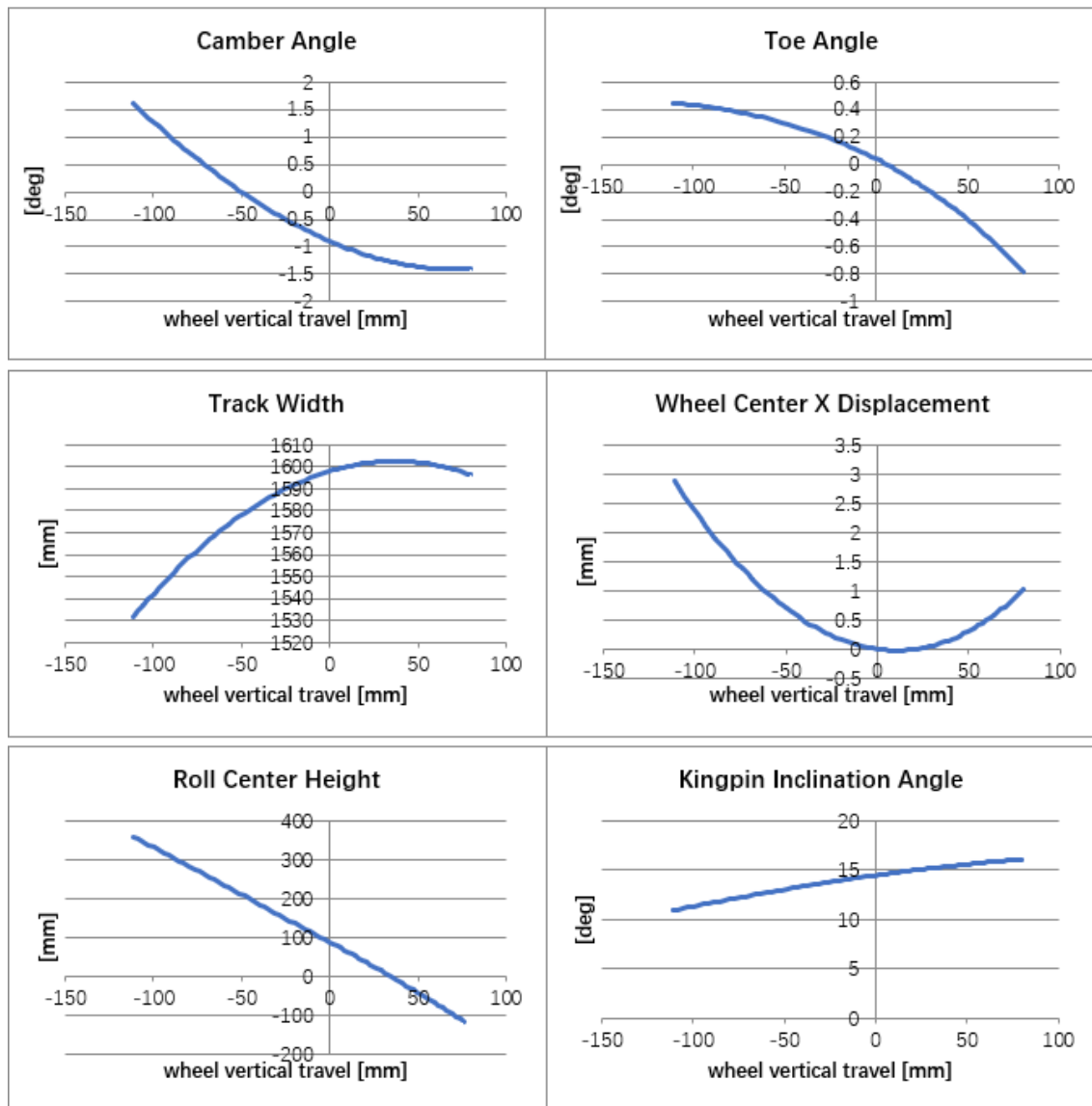


Figure 4.2: Graph result, vertical simulation, part 1

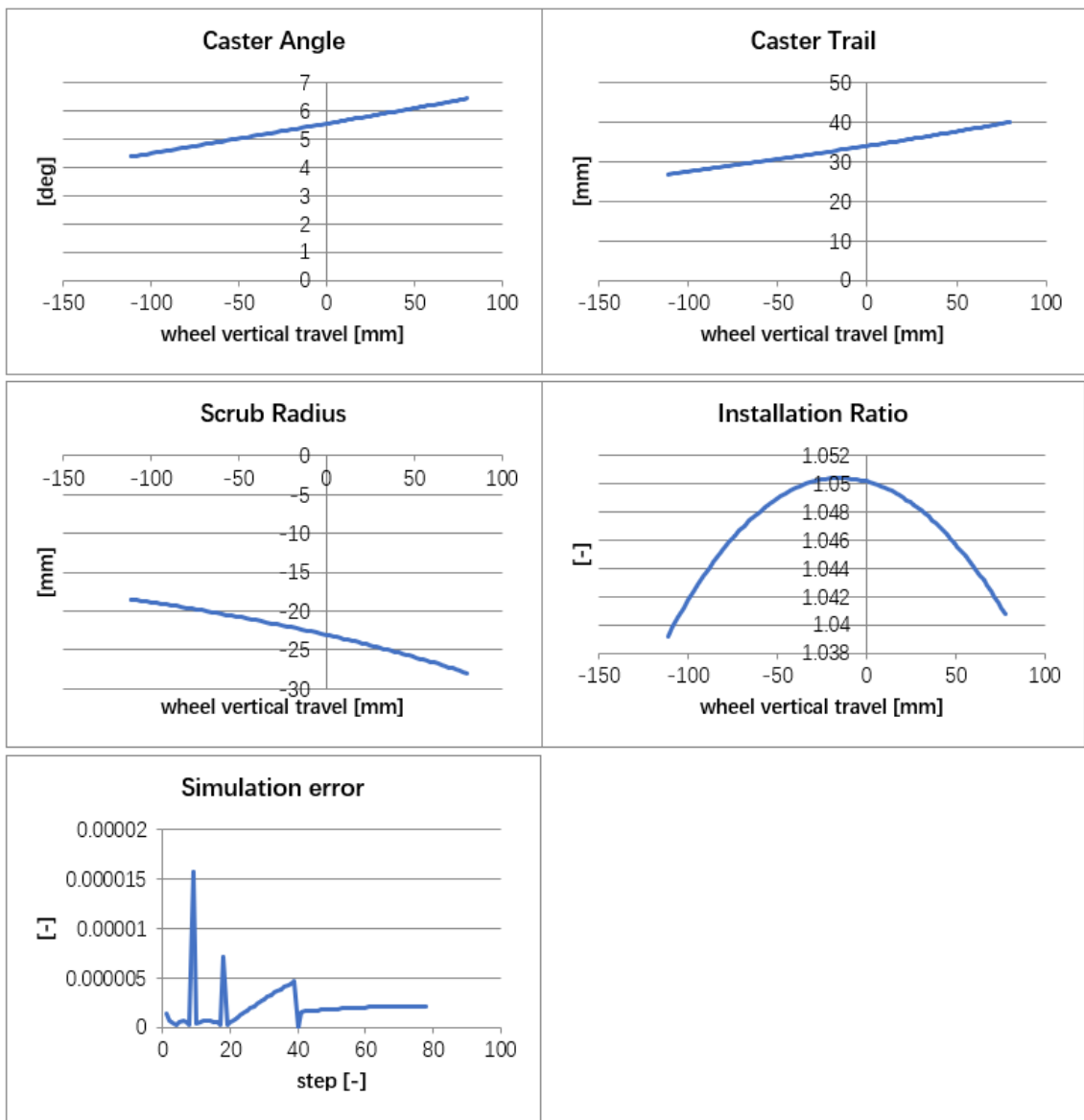


Figure 4.3: Graph result, vertical simulation, part 2

4.2.2 Steering simulation

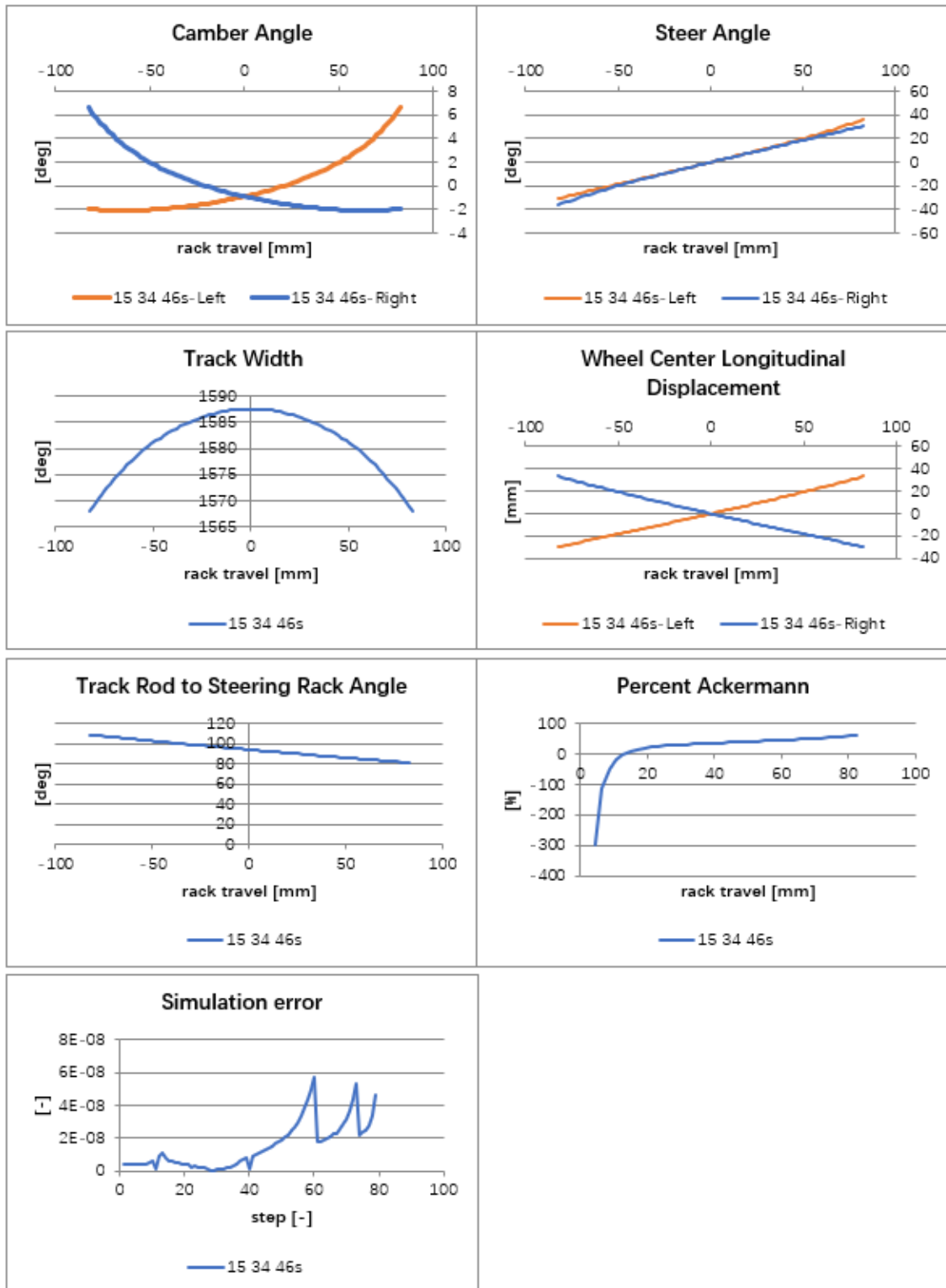


Figure 4.4: Graph result, steering simulation

4.3 Validation

In order to validate the this programme, another simulation is made by the vehicle dynamics group from CEVT company based on the same simulation set up. Several critical output parameters are picked out for the comparison.

4.3.1 Vertical Simulation

- Camber angle

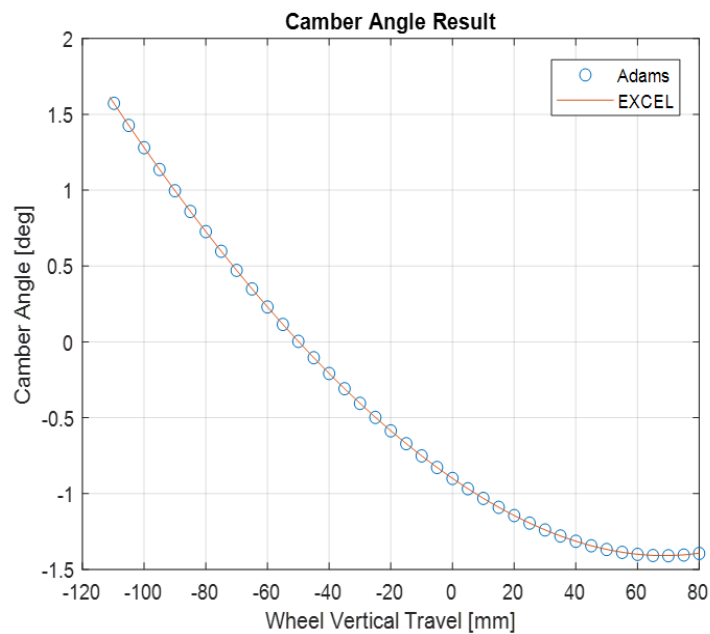


Figure 4.5: Camber angle, validation

The camber angle matches the ADAMS result perfectly.

- Toe angle

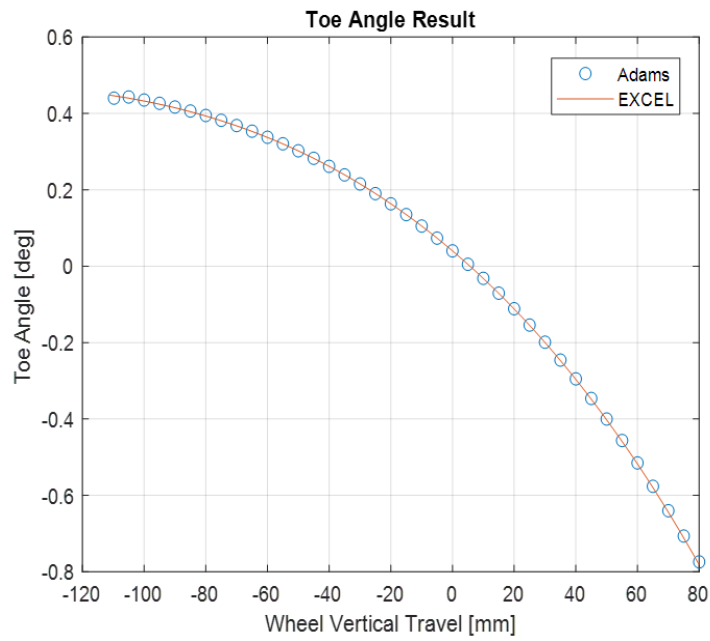


Figure 4.6: Toe angle, validation

The toe angle matches the ADAMS result perfectly.

- Kingpin inclination angle

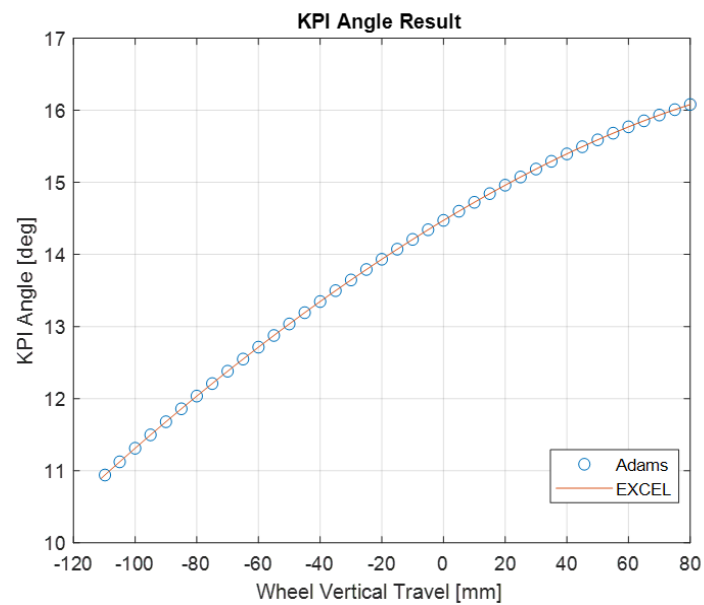


Figure 4.7: Kingpin inclination angle, validation

The kingpin inclination angle matches the ADAMS result perfectly.

- Caster Angle

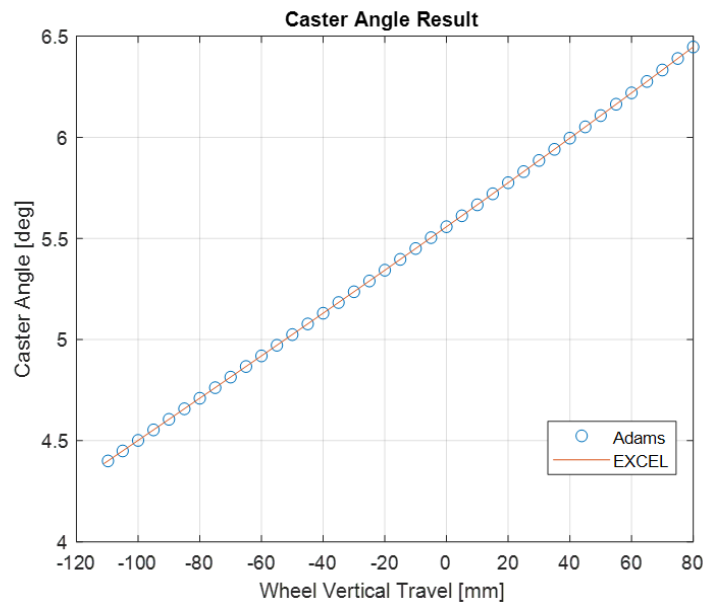


Figure 4.8: Caster angle, validation

The caster angle matches the ADAMS result perfectly.

- Roll Center Height

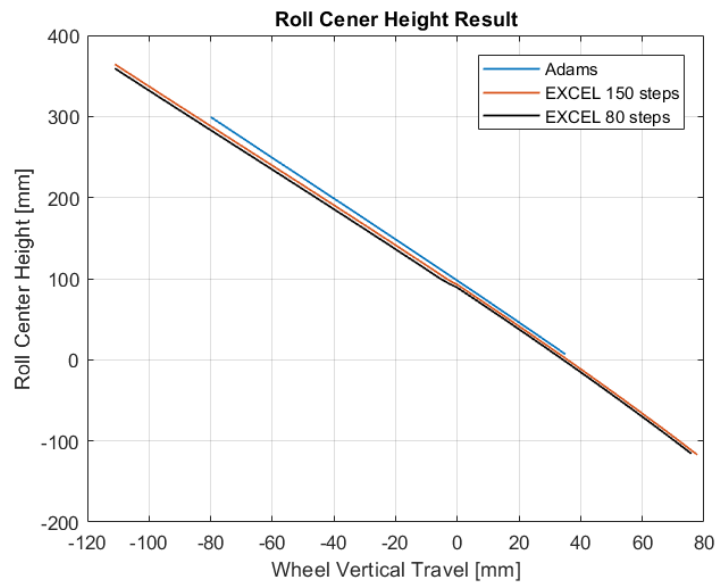


Figure 4.9: Roll center height, validation

As can be noticed from the above figure, the roll center height has a small difference compared to the ADAMS result. By increasing the steps the results gets closer. Therefore it can be concluded that the calculation is correct, the difference comes from mathematical calculation and can be eliminated when the step increase to infinity.

4.3.2 Steering Simulation

- Steer angle

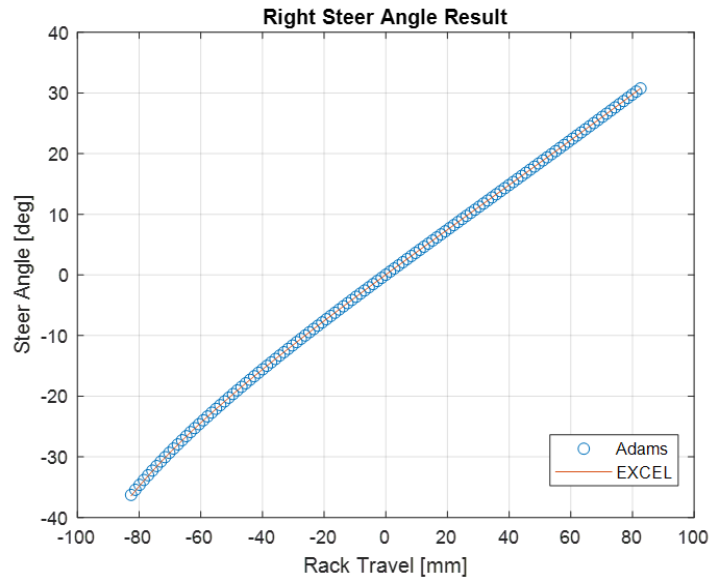


Figure 4.10: Right steer angle, validation

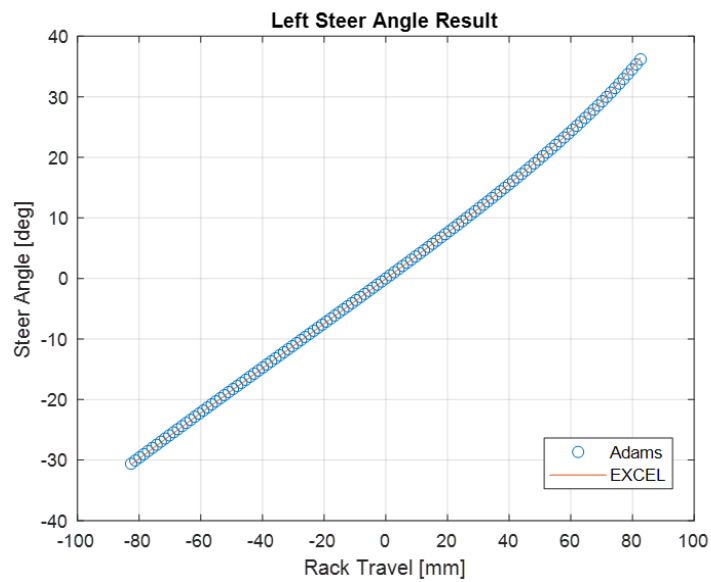


Figure 4.11: Left steer angle, validation

Both the left and right steer angle matches the ADAMS result perfectly.

- Percent Ackermann

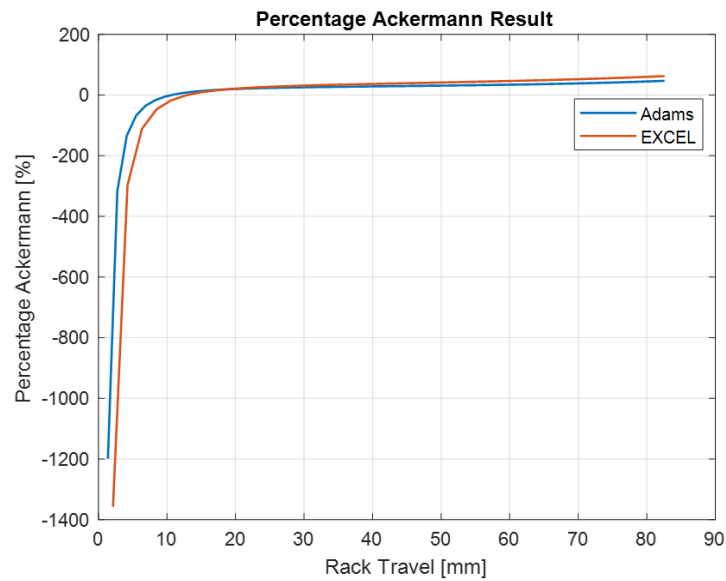


Figure 4.12: Percent Ackermann, validation

For the percent Ackermann there is a small difference between the ADAMS and EXCEL. The software succeed to captures the main trend of the curve.

From the steer angle figure, it can be noticed that the result from EXCEL was quite accurate since the curve matches perfectly with ADAMS. Since the percent Ackermann only depends on the steer angle, the reason which cause a small difference could be in the ideal outer steer angle calculation, the trackwidth was as a parameter. The vehicle model which used for analysis in this master thesis has different front and rear trackwidth. The EXCEL software only used the front trackwidth for calculation but ADAMS took both of them into consideration.

5

Conclusion

This chapter summarized the thesis work, presented the conclusion, discussion and the future work which can be developed later.

5.1 Conclusion

This master thesis work successfully developed a software based on EXCEL which satisfied all the requirements raised by the CEVT company. It still has some limitations make it not a mathematical precise simulation tool. However, by the comparison between this tool and ADAMS, the limitation is still within acceptable range therefore it can be used as a quick and robust tool in concept design phase for suspension development.

During this master thesis development, Excel has proved the possibility for itself to become the software platform for the suspension kinematics analysis simulation. With the help from the SOLVER add-in, it can convert the non-linear equation solving problem to an optimization problem. The slack variable played an important role in the optimization , it gave the system more freedom to find the solution.

5.2 Discussion

5.2.1 Modelling Method

As mentioned in the beginning of this report, there exist two (or even more) modelling method for a spatial mechanism system. The modelling method (global coordinate) used in this master thesis is the most straight forward one since all the constraint equations are formulated based on the physical relations eg. the length of the linkages. It is easy to understand and formulate but also will bring a huge amount of variables.

Relative coordinate is another modelling method, by using relative coordinates the number of equations (same as variables) can be reduced to the minimal value but however will bring more complicated equations such as complex sinusoidal equations. For a more complicate type suspension (more linkages and hardpoints), this modelling method is worth to try.

5.2.2 Software

The main reason for this master thesis project to choose EXCEL to be the platform is the wide application of it. During the development phase, EXCEL also shows a lot of unforeseen advantages. With the help from the EXCEL coding language (VBA) and macros, EXCEL cooperate with other software perfectly. It shows its potential to be developed into other mathematical simulation tool as well as long as the problem is defined into a solvable form inside it (eg. optimization problem).

5.3 Future work

This section lists the work in this master thesis which could be improved in the future.

5.3.1 Other suspension

The same development approached can applied on other suspension types as well. The most easy one will be the double wishbone suspension since the structure is very similar to Macpherson suspension. For other widely used passenger cars suspension (eg. trailing arm and multi-link), more complicate structure will bring more equations to solve therefore the solving time will increase, but the starting from a mathematical point of view the problem itself doesn't change so it will still be possible.

5.3.2 Steering simulation

The steering simulation is not "precise" due to the limitation that not fixing the correct hard point. This will lead to a small error in the steering simulation result. It is acceptable in the early design phase but it can also be improved in the future by rearrange the structure of the equation system.

5.3.3 User interface

Due to the time limitation for this master thesis, the user interface in this suspension kinematics programme remains in a most straight forward type. To improve the user coveniency especially for the new users, graphic user interface can be added in the future.

Bibliography

- [1] International Standards Organization.(2011): *ISO 8855:2011, Road Vehicles - Vehicle Dynamics and Road Holding Ability - Vocabulary*. International Standards Organization, Switzerland, 2011.
- [2] William F.Milliken, Douglas L.Milliken.(1994): Race Car Vehicle Dynamics. *SAE International*, 1994, ISBN: 978-1-56091-526-3
- [3] D.L.Cronin.(1981): MacPherson Strut Kinematics. *Mechanism and Machine Theory*, Vol. 16 No. 6, 1981, pp. 631-644
- [4] Hazem Ali Attia.(2003): Numerical Kinematic Analysis of the Standard Macpherson Motor-vehicle Suspension System. *KSME International Journal*, Vol. 17 No. 12, 2003, pp. 1961-1968
- [5] K.Vikranth Reddy, Madhu Kodati, Kishen Chatra, Sandipan Bandyopadhyay.(2016): A comprehensive kinematic analysis of the double wishbone and MacPherson strut suspension systems. *Mechanism and Machine Theory*, Vol. 105, 2016, pp. 441-470
- [6] Hans B. Pacejka.(2012): Tire and Vehicle Dynamics, 3rd edition, 2012. ISBN: 978-0-08-097016-5
- [7] Karl Niklass.(2010): Conventions and vehicle axis coordinate systems. *WHITE-SMOKE*, 20 Sep, 2010, <http://white-smoke.wikifoundry.com/page/Conventions+and+vehicle+axis+coordinate+systems>
- [8] Charlie Constant.(2013): Suspension design: definitions and effects on vehicle behavior. *Car engineer*. 28 Jan, 2013. <http://www.car-engineer.com/suspension-design-definitions-and-effects-on-vehicle-behavior/>
- [9] Tanmay Halaye.(2019): Automotive Suspension Systems Market Size, Status, Global outlook 2019 To 2025. Industry Reports. 18 Apr, 2019. <https://industryreports24.com/137141/automotive-suspension-systems-market-size-status-global-outlook-2019-to-2025/>

A

Programme User Instructions

A.1 Preparation

Before using this EXCEL file to run the simulation, several preparation works need to be done. In default EXCEL settings, the solver add-ins and macro are not enabled. Before running the simulation, they need to be turned on.

- Turn on the developer tab
This process is to turn on the developer tab then enable the users to enter visual basic editor and use it

.

file - options - customizeribbon - click on developer - ok

- Turn on the solver add-in
This process is to enable EXCEL to use the in-built add-ins SOLVER which is necessary to the simulation.

file - options - Go - click on solver - ok

After this, EXCEL still needs to add this add-in as its reference.

developer - visualbasic - tools - references - click on solver - ok

A.2 CEVT Template

The start of the simulation is the CEVT template. This software is built based on the CEVT template and can only work with its format. It is very important to make sure any changes in the template does not change the overall position of the cells which stores the coordinate information.

CEVT

China Euro Vehicle Technology AB

HARD POINT LIST

Hard Points List - Front Suspension MacP

Information from First Page

Document name

Document no

Revision

Release date

General information

Load condition

Tyre size

Tyre Static Loaded Radius (SLR)

Bump travel

Rebound travel

Over travel bump

Steering rack travel

Revision

Status Hard Points list

e.g. changes from previous

Wheel Suspension Hard Point

HP no

Bush Front

Bush Rear

Ball Joint, control arm

Top Mount

Wheel Center ETXX

Interface wheel

Interface driveshaft

Ball Joint, tie rod outer

Ball Joint, tie rod inner

Strut aligning point

Bumpstop, upper end

Bumpstop, lower end

Bumpstop, interface

Damper, bottom

Interface damper

Coil spring, upper end

Coil spring, lower end

Antirollbar mounting, LHS

General instruction 1: All areas in the document that are high lighted in blue must be filled in for all documents that are issued

General instruction 2: Fill in coordinates for the Left Hand Side

1WIP

0

0

0

Curb+2

80mm

106mm

3mm

202.6mm

27

J1

X

Y

Z

1731.200

-373.000

361.000

2015.000

-408.000

368.000

1700.400

-766.000

332.500

1766.891

-589.724

1015.471

1713.320

-793.800

447.047

1713.320

-845.793

447.864

1713.320

-775.802

446.764

1884.500

-732.518

410.600

1908.000

-370.000

423.500

1710.800

-649.000

485.717

1763.200

-593.595

980.879

1756.255

-600.938

915.253

1753.463

-603.891

888.864

1715.146

-644.405

526.783

1719.539

-639.760

568.293

1765.820

-590.719

1006.575

1753.600

-643.500

834.500

2043.000

-300.200

420.000

X

Y

Z

X

Y

Z

X

Y

Z

X

Y

Z

HP List Front MacP First Page

HP list Front MacP Current

HP Definitions Front MacP

CAD pictures

HP Change Log

Figure A.1: CEVT template

A.3 Simulation Process

This section detailed describes how to run the simulation and the post processing in this software.

A.3.1 Running Simulation

First, by clicking the read button, a message window will appears to let the user select the template file. Then the software will read the hardpoints data from the selected file into itself automatically.

After the hardpoints data has been imported to the software, next step is to set all the simulation set up parameters. Detail description for all the parameters can be found in the comment in the corresponding cell.

Then, bu clicking the VERTICAL or STEERING button, the corresponding simu-
lation will run itself.

A.3.2 Post Processing

After each simulation finishes, a message window will appears to let the users insert the comment for it. This comment will be shown later in figure control sheet to help the engineers distinguish between different simulation result. Apart from the comment, the software will name the simulation itself automatically according to the time and simulation type (eg. 13 05 01 s stands for a steering simulation finishes

at 13:05:01). A new worksheet will be created with the same name as the simulation name which stores all the output parameters data together with the input data.

By typing "1" on the right to the simulation, the corresponding simulation result will be plotted. The two buttons on the worksheet controls the figure plot and figure delete.

At last, if engineer is satisfied with the simulation result, by clicking the EXPORT button, the current input data will be send back the the template file.

B

Excel Module, *Read_data*

```
Sub readdata()

'-----define variable type-----

Dim src As Workbook
Dim R As Integer
Dim i As Integer
Dim fd As Office.FileDialog

Set fd = Application.FileDialog(msoFileDialogFilePicker)

'-----message window-----

With fd

    .AllowMultiSelect = False
    .Title = "Please select the template file"
    .Filters.Add "Excel", "*.xls"
    .Filters.Add "All", "*.*"

    If .Show = True Then
        txtfilename = .SelectedItems(1)
    ,
    End If

End With

Set src = Workbooks.Open(txtfilename)

'-----copy data-----
ThisWorkbook.Activate

Range("A1:F22").Value = src.Worksheets("HP list Front MacP Current").Range("B30:G51").Value

src.Close

'-----naming-----

Range("A23").Value = "Auxiliary"
Range("B23").Value = 2
Range("A24").Value = "Spindle axis"
Range("A25").Value = "Tire contacting"
Range("B24").Value = 20
Range("B25").Value = 21

End Sub
```


C

Excel Module, *Vertical*

```
Sub vertical()  
  
'-----define variable type-----  
Dim e1 As Integer  
Dim i As Integer  
Dim j As Integer  
  
'----- Clear the simulation result from last simulation-----  
  
'-----clear simulation worksheet-----  
  
With Worksheets("simulation")  
  
.Range("C2:C48").ClearContents  
.Range("D2:D25").ClearContents  
.Range("F29:F48").ClearContents  
.Range("J29:J40").ClearContents  
e1 = .Cells(1, Columns.Count).End(xlToLeft).Column  
  
For i = 1 To 25  
For j = 12 To e1  
.Cells(i, j).ClearContents  
Next j  
Next i  
  
End With  
  
'-----clear intermediate worksheet-----  
  
With Worksheets("intermediate")  
  
.Range("C56:C90").ClearContents  
  
e1 = .Cells(1, Columns.Count).End(xlToLeft).Column  
For i = 1 To 53  
For j = 3 To e1  
.Cells(i, j).ClearContents  
Next j  
Next i  
  
e1 = .Cells(65, Columns.Count).End(xlToLeft).Column  
For i = 65 To 66  
For j = 7 To e1  
.Cells(i, j).ClearContents  
Next j  
Next i  
  
End With  
  
'-----clear vertical output worksheet-----  
  
With Worksheets("vertical output")  
  
e1 = .Cells(Rows.Count, 1).End(xlUp).Row
```

```
For i = 1 To 14
For j = 2 To e1
.Cells(j, i).ClearContents
Next j
Next i

.Range("P1:P13").ClearContents

For i = 17 To 24
For j = 2 To 21
.Cells(j, i).ClearContents
Next j
Next i

End With

'-----preparation-----
' calculate spindle axis and tire contacting

Worksheets("input").Activate

'Right side
Range("A27:F27").Value = Range("A1:F1").Value ' head line
Range("A28:B51").Value = Range("A2:B25").Value ' name and code
Range("D28:D48").Value = Range("D2:D22").Value 'x
Range("E28:E48").Value = Range("E2:E22").Value 'y
Range("F28:F48").Value = Range("F2:F22").Value 'z

' Y to opposite value
For i = 28 To 51
Cells(i, 5).Value = -1 * Cells(i, 5).Value
Next i

Dim R As Single
Dim camber As Double
Dim toe As Double

R = Range("I7").Value

Pi = Application.Pi()
camber = Range("I10").Value * (Pi / 180)
toe = -Range("I11").Value * (Pi / 180)

'auxiliary
Range("D23").Value = 0.1 * (Range("D5").Value - Range("D15").Value) + Range("D15").Value
Range("E23").Value = 0.1 * (Range("E5").Value - Range("E15").Value) + Range("E15").Value
Range("F23").Value = 0.1 * (Range("F5").Value - Range("F15").Value) + Range("F15").Value

'spindle axis
Range("D24").Value = Range("D6").Value - 100 * Sin(toe)
Range("E24").Value = Range("E6").Value + 100 - 100 * (1 - Cos(camber)) - 100 * (1 - Cos(toe))
Range("F24").Value = Range("F6").Value + 100 * Sin(camber)

'tire contacting
Range("D25").Value = Range("D6").Value
Range("E25").Value = Range("E6").Value
```



```

Range("F25").Value = Range("F6").Value - R

'-----

Range("D49").Value = Range("D23").Value
Range("E49").Value = -Range("E23").Value
Range("F49").Value = Range("F23").Value

Range("D50").Value = Range("D24").Value
Range("E50").Value = -Range("E24").Value
Range("F50").Value = Range("F24").Value

Range("D51").Value = Range("D25").Value
Range("E51").Value = -Range("E25").Value
Range("F51").Value = Range("F25").Value

'-----calculation preparation-----
With Worksheets("input")

Worksheets("simulation").Range("F29").Value = (.Range("D4").Value - .Range("D2").
    Value) ^ 2 + (.Range("E4").Value - .Range("E2").Value) ^ 2 + (.Range("F4").
    Value - .Range("F2").Value) ^ 2
Worksheets("simulation").Range("F30").Value = (.Range("D4").Value - .Range("D3").
    Value) ^ 2 + (.Range("E4").Value - .Range("E3").Value) ^ 2 + (.Range("F4").
    Value - .Range("F3").Value) ^ 2
Worksheets("simulation").Range("F31").Value = (.Range("D10").Value - .Range("D9").
    Value) ^ 2 + (.Range("E10").Value - .Range("E9").Value) ^ 2 + (.Range("F10").
    Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F32").Value = (.Range("D15").Value - .Range("D5").
    Value) ^ 2 + (.Range("E15").Value - .Range("E5").Value) ^ 2 + (.Range("F15").
    Value - .Range("F5").Value) ^ 2

Worksheets("simulation").Range("F33").Value = (.Range("D15").Value - .Range("D4").
    Value) ^ 2 + (.Range("E15").Value - .Range("E4").Value) ^ 2 + (.Range("F15").
    Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F34").Value = (.Range("D23").Value - .Range("D4").
    Value) ^ 2 + (.Range("E23").Value - .Range("E4").Value) ^ 2 + (.Range("F23").
    Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F35").Value = (.Range("D9").Value - .Range("D4").
    Value) ^ 2 + (.Range("E9").Value - .Range("E4").Value) ^ 2 + (.Range("F9").
    Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F36").Value = (.Range("D23").Value - .Range("D15").
    Value) ^ 2 + (.Range("E23").Value - .Range("E15").Value) ^ 2 + (.Range("F23").
    Value - .Range("F15").Value) ^ 2
Worksheets("simulation").Range("F37").Value = (.Range("D15").Value - .Range("D9").
    Value) ^ 2 + (.Range("E15").Value - .Range("E9").Value) ^ 2 + (.Range("F15").
    Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F38").Value = (.Range("D23").Value - .Range("D9").
    Value) ^ 2 + (.Range("E23").Value - .Range("E9").Value) ^ 2 + (.Range("F23").
    Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F39").Value = (.Range("D6").Value - .Range("D4").
    Value) ^ 2 + (.Range("E6").Value - .Range("E4").Value) ^ 2 + (.Range("F6").
    Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F40").Value = (.Range("D15").Value - .Range("D6").
    Value) ^ 2 + (.Range("E15").Value - .Range("E6").Value) ^ 2 + (.Range("F15").
    Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F41").Value = (.Range("D23").Value - .Range("D6").
    Value) ^ 2 + (.Range("E23").Value - .Range("E6").Value) ^ 2 + (.Range("F23").
    Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F42").Value = (.Range("D24").Value - .Range("D6").
    Value) ^ 2 + (.Range("E24").Value - .Range("E6").Value) ^ 2 + (.Range("F24").
    Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F43").Value = (.Range("D25").Value - .Range("D6").
    Value) ^ 2 + (.Range("E25").Value - .Range("E6").Value) ^ 2 + (.Range("F25").
    Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F44").Value = (.Range("D24").Value - .Range("D4").
    Value) ^ 2 + (.Range("E24").Value - .Range("E4").Value) ^ 2 + (.Range("F24").
    Value - .Range("F4").Value) ^ 2

```

```

Worksheets("simulation").Range("F45").Value = (.Range("D24").Value - .Range("D9").
    Value) ^ 2 + (.Range("E24").Value - .Range("E9").Value) ^ 2 + (.Range("F24").
    Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F46").Value = (.Range("D24").Value - .Range("D15")
    .Value) ^ 2 + (.Range("E24").Value - .Range("E15").Value) ^ 2 + (.Range("F24")
    .Value - .Range("F15").Value) ^ 2
Worksheets("simulation").Range("F47").Value = (.Range("D24").Value - .Range("D23")
    .Value) ^ 2 + (.Range("E24").Value - .Range("E23").Value) ^ 2 + (.Range("F24")
    .Value - .Range("F23").Value) ^ 2
Worksheets("simulation").Range("F48").Value = (.Range("D24").Value - .Range("D11")
    .Value) ^ 2 + (.Range("E24").Value - .Range("E11").Value) ^ 2 + (.Range("F24")
    .Value - .Range("F11").Value) ^ 2

End With

With Worksheets("simulation")

    .Range("J29").Value = Range("D28").Value
    .Range("J30").Value = Range("E28").Value
    .Range("J31").Value = Range("F28").Value

    .Range("J32").Value = Range("D29").Value
    .Range("J33").Value = Range("E29").Value
    .Range("J34").Value = Range("F29").Value

    .Range("J35").Value = Range("D36").Value
    .Range("J36").Value = Range("E36").Value
    .Range("J37").Value = Range("F36").Value

    .Range("J38").Value = Range("D31").Value
    .Range("J39").Value = Range("E31").Value
    .Range("J40").Value = Range("F31").Value

End With

'----- defining simulation set up-----
Dim bump As Integer
Dim rebound As Integer
Dim step As Integer
Dim stepsize As Single
Dim tol As Double
Dim maxtime As Double
Dim maxiteration As Double

bump = Range("I3").Value

rebound = Range("I4").Value

step = Range("I5").Value

tol = Range("I8").Value

maxtime = Range("I12")

maxiteration = Range("I13")

stepsize_bump = bump / (step / 2 - 1)

stepsize_rebound = Abs(rebound) / (step / 2 - 1)

Worksheets("simulation").Range("rp3_y").Value = Range("E36").Value

Worksheets("simulation").Activate

'-----set initial guess-----
Range("D2").Value = Worksheets("input").Range("D30").Value

```

```

Range("D3").Value = Worksheets("input").Range("E30").Value
Range("D4").Value = Worksheets("input").Range("F30").Value

Range("D5").Value = Worksheets("input").Range("D41").Value
Range("D6").Value = Worksheets("input").Range("E41").Value
Range("D7").Value = Worksheets("input").Range("F41").Value

Range("D8").Value = Worksheets("input").Range("D49").Value
Range("D9").Value = Worksheets("input").Range("E49").Value
Range("D10").Value = Worksheets("input").Range("F49").Value

Range("D11").Value = Worksheets("input").Range("D35").Value
Range("D12").Value = Worksheets("input").Range("E35").Value
Range("D13").Value = Worksheets("input").Range("F35").Value

Range("D14").Value = Worksheets("input").Range("D41").Value
Range("D15").Value = Worksheets("input").Range("E41").Value
Range("D16").Value = Worksheets("input").Range("F41").Value

Range("D17").Value = Worksheets("input").Range("D32").Value
Range("D18").Value = Worksheets("input").Range("E32").Value
Range("D19").Value = Worksheets("input").Range("F32").Value

Range("D20").Value = Worksheets("input").Range("D50").Value
Range("D21").Value = Worksheets("input").Range("E50").Value
Range("D22").Value = Worksheets("input").Range("F50").Value

Range("D23").Value = Worksheets("input").Range("D51").Value
Range("D24").Value = Worksheets("input").Range("E51").Value
Range("D25").Value = Worksheets("input").Range("F51").Value

'-----reset coordinates-----
Range("C2:C25").Value = Range("D2:D25").Value
'Range("C20:C22").Value = 0

'-----reset slack variable-----
Range("C26:C48").Value = Range("F2:F24").Value

For j = 1 To step - 1
Cells(1, 11 + j) = j
Next j

'-----
'          SOLVE FOR REBOUND
'-----

For i = 1 To step / 2

'----- solver -----

SolverReset

Solveroptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$2", MaxMinVal:=2, ByChange:= _
        "$C$2:$C$3,$C$5:$C$19,$C$26:$C$42", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$2:$F$18", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False
'SolverOptions Iterations:=1E+16, Precision:=0.000001, Convergence:=0.000000001,
    AssumeNonNeg:=True
'-----Equation value -----

```

```

Cells(2, step / 2 + 12 - i).Value = Range("F2").Value
Cells(3, step / 2 + 12 - i).Value = Range("F3").Value
Cells(4, step / 2 + 12 - i).Value = Range("F4").Value
Cells(5, step / 2 + 12 - i).Value = Range("F5").Value
Cells(6, step / 2 + 12 - i).Value = Range("F6").Value
Cells(7, step / 2 + 12 - i).Value = Range("F7").Value
Cells(8, step / 2 + 12 - i).Value = Range("F8").Value
Cells(9, step / 2 + 12 - i).Value = Range("F9").Value
Cells(10, step / 2 + 12 - i).Value = Range("F10").Value
Cells(11, step / 2 + 12 - i).Value = Range("F11").Value
Cells(12, step / 2 + 12 - i).Value = Range("F12").Value
Cells(13, step / 2 + 12 - i).Value = Range("F13").Value
Cells(14, step / 2 + 12 - i).Value = Range("F14").Value
Cells(15, step / 2 + 12 - i).Value = Range("F15").Value
Cells(16, step / 2 + 12 - i).Value = Range("F16").Value
Cells(17, step / 2 + 12 - i).Value = Range("F17").Value
Cells(18, step / 2 + 12 - i).Value = Range("F18").Value

'----- solver2 -----

SolverReset
    SolverOptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$3", MaxMinVal:=2, ByChange:= _
        "$C$20:$C$25,$C$43:$C$48", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$19:$F$24", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False
'----- Equation value -----
Cells(19, step / 2 + 12 - i).Value = Range("F19").Value
Cells(20, step / 2 + 12 - i).Value = Range("F20").Value
Cells(21, step / 2 + 12 - i).Value = Range("F21").Value
Cells(22, step / 2 + 12 - i).Value = Range("F22").Value
Cells(23, step / 2 + 12 - i).Value = Range("F23").Value
Cells(24, step / 2 + 12 - i).Value = Range("F24").Value

'-----save data to result-----

Worksheets("intermediate").Cells(2, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(3, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_2").Value
Worksheets("intermediate").Cells(4, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(5, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(6, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_5").Value
Worksheets("intermediate").Cells(7, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(8, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(9, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_8").Value
Worksheets("intermediate").Cells(10, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(11, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_10").Value
Worksheets("intermediate").Cells(12, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_11").Value
Worksheets("intermediate").Cells(13, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_12").Value
Worksheets("intermediate").Cells(14, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_13").Value
Worksheets("intermediate").Cells(15, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_14").Value
Worksheets("intermediate").Cells(16, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_15").Value

```

```

Worksheets("intermediate").Cells(17, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_16").Value
Worksheets("intermediate").Cells(18, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_17").Value
Worksheets("intermediate").Cells(19, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_18").Value
Worksheets("intermediate").Cells(20, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_19").Value
Worksheets("intermediate").Cells(21, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_20").Value
Worksheets("intermediate").Cells(22, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_21").Value
Worksheets("intermediate").Cells(23, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_22").Value
Worksheets("intermediate").Cells(24, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_23").Value
Worksheets("intermediate").Cells(25, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_24").Value

Range("x_3").Value = Range("x_3").Value - stepsize_rebound
Next i

'reset coordinates
Range("C2:C25").Value = Range("D2:D25").Value
'Range("C20:C22").Value = 0

'reset slack variable
Range("C26:C48").Value = Range("F2:F24").Value

'-----
'               SOLVE FOR BUMP
'-----
For i = 1 To step / 2

'----- solver -----
SolverReset

SolverOptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration
    SolverOk SetCell:="$I$2", MaxMinVal:=2, ByChange:= _
        "$C$2:$C$3,$C$5:$C$19,$C$26:$C$42", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$2:$F$18", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False
'SolverOptions Iterations:=1E+16, Precision:=0.000001, Convergence:=0.000000001,
    AssumeNonNeg:=True
'-----Equation value -----

Cells(2, step / 2 + 10 + i).Value = Range("F2").Value
Cells(3, step / 2 + 10 + i).Value = Range("F3").Value
Cells(4, step / 2 + 10 + i).Value = Range("F4").Value
Cells(5, step / 2 + 10 + i).Value = Range("F5").Value
Cells(6, step / 2 + 10 + i).Value = Range("F6").Value
Cells(7, step / 2 + 10 + i).Value = Range("F7").Value
Cells(8, step / 2 + 10 + i).Value = Range("F8").Value
Cells(9, step / 2 + 10 + i).Value = Range("F9").Value
Cells(10, step / 2 + 10 + i).Value = Range("F10").Value
Cells(11, step / 2 + 10 + i).Value = Range("F11").Value
Cells(12, step / 2 + 10 + i).Value = Range("F12").Value
Cells(13, step / 2 + 10 + i).Value = Range("F13").Value
Cells(14, step / 2 + 10 + i).Value = Range("F14").Value
Cells(15, step / 2 + 10 + i).Value = Range("F15").Value
Cells(16, step / 2 + 10 + i).Value = Range("F16").Value
Cells(17, step / 2 + 10 + i).Value = Range("F17").Value
Cells(18, step / 2 + 10 + i).Value = Range("F18").Value

'----- solver2 -----

SolverReset

```

```

SolverOptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration
    SolverOk SetCell:="$I$3", MaxMinVal:=2, ByChange:= _
        "$C$20:$C$25,$C$43:$C$48", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$19:$F$24", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False
'----- Equation value -----
Cells(19, step / 2 + 10 + i).Value = Range("F19").Value
Cells(20, step / 2 + 10 + i).Value = Range("F20").Value
Cells(21, step / 2 + 10 + i).Value = Range("F21").Value
Cells(22, step / 2 + 10 + i).Value = Range("F22").Value
Cells(23, step / 2 + 10 + i).Value = Range("F23").Value
Cells(24, step / 2 + 10 + i).Value = Range("F24").Value

'-----save data to result-----

Worksheets("intermediate").Cells(2, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(3, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_2").Value
Worksheets("intermediate").Cells(4, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(5, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(6, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_5").Value
Worksheets("intermediate").Cells(7, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(8, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(9, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_8").Value
Worksheets("intermediate").Cells(10, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(11, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_10").Value
Worksheets("intermediate").Cells(12, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_11").Value
Worksheets("intermediate").Cells(13, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_12").Value
Worksheets("intermediate").Cells(14, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_13").Value
Worksheets("intermediate").Cells(15, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_14").Value
Worksheets("intermediate").Cells(16, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_15").Value
Worksheets("intermediate").Cells(17, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_16").Value
Worksheets("intermediate").Cells(18, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_17").Value
Worksheets("intermediate").Cells(19, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_18").Value
Worksheets("intermediate").Cells(20, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_19").Value
Worksheets("intermediate").Cells(21, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_20").Value
Worksheets("intermediate").Cells(22, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_21").Value
Worksheets("intermediate").Cells(23, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_22").Value
Worksheets("intermediate").Cells(24, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_23").Value
Worksheets("intermediate").Cells(25, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_24").Value

Range("x_3").Value = Range("x_3").Value + stepsize_bump

Next i

```

```

For i = 1 To (step - 1)

Cells(25, 11 + i).Value = Abs(Cells(2, 11 + i)) + Abs(Cells(3, 11 + i)) + Abs(
    Cells(4, 11 + i)) + Abs(Cells(5, 11 + i)) + Abs(Cells(6, 11 + i)) + Abs(Cells
        (7, 11 + i)) _
        + Abs(Cells(8, 11 + i)) + Abs(Cells(9, 11 + i)) + Abs(Cells
            (10, 11 + i)) + Abs(Cells(11, 11 + i)) + Abs(Cells(12,
                11 + i)) + Abs(Cells(13, 11 + i)) _
        + Abs(Cells(14, 11 + i)) + Abs(Cells(15, 11 + i)) + Abs(
            Cells(16, 11 + i)) + Abs(Cells(17, 11 + i)) + Abs(Cells
                (18, 11 + i)) + Abs(Cells(19, 11 + i)) _
        + Abs(Cells(20, 11 + i)) + Abs(Cells(21, 11 + i)) + Abs(
            Cells(22, 11 + i)) + Abs(Cells(23, 11 + i)) + Abs(Cells
                (24, 11 + i))

'Cells(25, 11 + i).Value = Application.Sum(Range(Cells(2, 11 + i), Cells(24, 11 +
    i)))
Worksheets("intermediate").Cells(1, i + 2).Value = i
Worksheets("intermediate").Cells(26, i + 2).Value = 0

Next i

'-----calculating output-----

For i = 1 To step
Worksheets("vertical output").Cells(i + 1, 1).Value = Worksheets("intermediate").
    Cells(19, i + 2).Value
Next i

Worksheets("intermediate").Activate

Pi = Application.Pi()

Range("p34x").Value = 0.5 * (Worksheets("input").Range("D28").Value + Worksheets("
    input").Range("D28").Value)
Range("p34y").Value = 0.5 * (Worksheets("input").Range("F28").Value + Worksheets("
    input").Range("F29").Value)
Range("p34z").Value = 0.5 * (Worksheets("input").Range("E28").Value + Worksheets("
    input").Range("E29").Value)

Worksheets("intermediate").Range("icy").Value = -500
Worksheets("intermediate").Range("icz").Value = 0

Range("p7x").Value = Worksheets("input").Range("D31").Value
Range("p7y").Value = Worksheets("input").Range("E31").Value
Range("p7z").Value = Worksheets("input").Range("F31").Value

Range("p3x").Value = Worksheets("input").Range("D28").Value
Range("p3y").Value = Worksheets("input").Range("E28").Value
Range("p3z").Value = Worksheets("input").Range("F28").Value

Range("p4x").Value = Worksheets("input").Range("D29").Value
Range("p4y").Value = Worksheets("input").Range("E29").Value
Range("p4z").Value = Worksheets("input").Range("F29").Value

Range("sp6x").Value = Worksheets("input").Range("D4").Value
Range("sp6y").Value = -Worksheets("input").Range("E4").Value
Range("sp6z").Value = Worksheets("input").Range("F4").Value

SolverReset
Solveroptions AssumeNonNeg:=False

    SolverOk SetCell:="$E$74", MaxMinVal:=3, valueof:=0, ByChange:="$C$80:$C$81", _
        Engine:=2
        'EngineDesc:="Simplex LP"

    SolverAdd CellRef:="$E$71:$E$72", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

```



```

Worksheets("vertical output").Cells(10, 16).Value = Worksheets("intermediate").
    Range("sicx").Value
Worksheets("vertical output").Cells(11, 16).Value = Worksheets("intermediate").
    Range("sicz").Value

For i = 1 To step - 1

With Worksheets("vertical output")

'wheel travel
.Cells(i + 1, 2).Value = Worksheets("vertical output").Cells(i + 1, 1) -
    Worksheets("input").Range("F32").Value

' camber angle
.Cells(i + 1, 3).Value = (180 / Pi) * Atn((Worksheets("intermediate").Cells(22, i
+ 2).Value - Worksheets("intermediate").Cells(19, i + 2).Value) / (Worksheets(
"intermediate").Cells(18, i + 2).Value - Worksheets("intermediate").Cells(21,
i + 2).Value))

'toe angle
.Cells(i + 1, 4).Value = (180 / Pi) * Atn((Worksheets("intermediate").Cells(17, i
+ 2).Value - Worksheets("intermediate").Cells(20, i + 2).Value) / (Worksheets(
"intermediate").Cells(21, i + 2).Value - Worksheets("intermediate").Cells(18,
i + 2).Value))

'track width
.Cells(i + 1, 5).Value = 2 * Worksheets("intermediate").Cells(18, i + 2).Value

'wheel center X
.Cells(i + 1, 6).Value = Worksheets("intermediate").Cells(17, i + 2) - Worksheets(
"input").Range("D32").Value

'roll center height
Range("p6x").Value = Cells(2, i + 2).Value
Range("p6y").Value = Cells(3, i + 2).Value
Range("p6z").Value = Cells(4, i + 2).Value

Range("p21x").Value = Cells(23, i + 2).Value
Range("p21y").Value = Cells(24, i + 2).Value
Range("p21z").Value = Cells(25, i + 2).Value

SolverReset
SolverOptions AssumeNonNeg:=False

    SolverOk SetCell:="$E$68", MaxMinVal:=3, valueof:=0, ByChange:="$C$77:$C$78", _
        Engine:=2
        'EngineDesc:="Simplex LP"

    SolverAdd CellRef:="$E$65:$E$66", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Worksheets("intermediate").Cells(65, i + 5).Value = Worksheets("intermediate").
    Range("E65").Value
Worksheets("intermediate").Cells(66, i + 5).Value = Worksheets("intermediate").
    Range("E66").Value

Range("rch").Value = (0 - Range("p21y")) * (Range("icz") - Range("p21z")) / (Range
("icy") - Range("p21y")) + Range("p21z")

.Cells(i + 1, 7).Value = Worksheets("intermediate").Range("icy").Value
.Cells(i + 1, 8).Value = Worksheets("intermediate").Range("icz").Value
.Cells(i + 1, 9).Value = Worksheets("intermediate").Range("rch").Value -
    Worksheets("intermediate").Range("p21z").Value

'kpi angle
.Cells(i + 1, 10).Value = (180 / Pi) * Atn((Cells(3, i + 2) - Range("p7y")) / (
    Range("p7z") - Cells(4, i + 2)))

'caster angle

```



```

.Cells(i + 1, 11).Value = (180 / Pi) * Atn((Range("p7x") - Cells(2, i + 2)) / (
    Range("p7z") - Cells(4, i + 2)))

'caster trail
.Cells(i + 1, 12).Value = Cells(17, i + 2) - Range("p7x") + ((Range("p7x") - Cells
    (2, i + 2)) / (Range("p7z") - Cells(4, i + 2))) * (Range("p7z") - Cells(25, i
    + 2))

'scrub radius
.Cells(i + 1, 13).Value = Cells(18, i + 2) - ((Cells(3, i + 2) - Range("p7y")) / (
    Range("p7z") - Cells(4, i + 2))) * (Range("p7z") - Cells(25, i + 2)) - Range("
    p7y")

'simulation error (abs value)
.Cells(i + 1, 18).Value = Worksheets("simulation").Cells(25, 11 + i).Value
.Cells(i + 1, 17).Value = Worksheets("simulation").Cells(1, 11 + i).Value

End With
Next i

'ratio
For i = 1 To step - 2
Worksheets("vertical output").Cells(i + 1, 14).Value = (Cells(25, i + 3) - Cells
    (25, 3)) / Sqr((Cells(5, i + 3) - Cells(5, 3)) ^ 2 + (Cells(6, i + 3) - Cells
    (6, 3)) ^ 2 + (Cells(7, i + 3) - Cells(7, 3)) ^ 2)
Next i

'-----calculating output required as "value" (gain)
-----
Dim nega10 As Double
Dim posi10 As Double
Dim numnega As Integer
Dim numposi As Integer
Dim travel As Double

step = Worksheets("input").Range("I5").Value

Worksheets("vertical output").Activate

nega10 = Application.VLookup(-10, Range(Cells(2, 2), Cells(step, 2)), 1, True)
posi10 = Application.VLookup(10, Range(Cells(2, 2), Cells(step, 2)), 1, True)

For i = 1 To step
If Cells(i, 2).Value = nega10 Then
numnega = i
End If
Next i

For i = 1 To step
If Cells(i, 2).Value = posi10 Then
numposi = i
End If
Next i

travel = (posi10 - nega10) / 1000

'-----camber gain-----
Cells(1, 16) = (Cells(numposi, 3) - Cells(numnega, 3)) / travel

'-----toe gain-----
Cells(2, 16) = (Cells(numposi, 4) - Cells(numnega, 4)) / travel

'-----track width gain-----

```

```

Cells(3, 16) = (Cells(numposi, 5) - Cells(numnega, 5)) / travel
'-----wheel center X gain-----
Cells(4, 16) = (Cells(numposi, 6) - Cells(numnega, 6)) / travel
'-----roll center gain-----
Cells(5, 16) = (Cells(numposi, 9) - Cells(numnega, 9)) / travel
'-----kpi angle gain-----
Cells(6, 16) = (Cells(numposi, 10) - Cells(numnega, 10)) / travel
'-----caster angle gain-----
Cells(7, 16) = (Cells(numposi, 11) - Cells(numnega, 11)) / travel
'-----caster trail gain-----
Cells(8, 16) = (Cells(numposi, 12) - Cells(numnega, 12)) / travel
'-----scrub radius gain-----
Cells(9, 16) = (Cells(numposi, 13) - Cells(numnega, 13)) / travel
'-----anti dive-----
Cells(12, 16).Value = (Cells(11, 16).Value - Worksheets("input").Range("F25").
    Value) / Abs(Cells(10, 16).Value - Worksheets("input").Range("D6").Value)
'-----anti lift-----
Cells(13, 16).Value = (Cells(11, 16).Value - Worksheets("input").Range("F6").Value
    ) / Abs(Cells(10, 16).Value - Worksheets("input").Range("D6").Value)
'-----copy hard point-----
Worksheets("input").Range("A1:F22").Copy Destination:=Worksheets("vertical output"
    ).Range("S1")

'-----creating new sheet for storage-----
Dim currTime As Date
Dim sheetname As String

currTime = Now()
sheetname = Format(Now(), "hh nn ss")
Sheets.Add(After:=Sheets(Sheets.Count)).name = sheetname & "v"

Worksheets("vertical figure").Range("A" & Rows.Count).End(xlUp).Offset(1) =
    sheetname & "v"

Dim comment As String
'Dim xSht As Object
'On Error Resume Next
comment = InputBox("Please enter a comment for this simulation result ", "
    Kutools for Excel")
'If xName = "" Then Exit Sub
'Set xSht = Sheets(xName)
'If Not xSht Is Nothing Then
    MsgBox "Sheet cannot be created as there is already a worksheet with
        the same name in this workbook"
'Exit Sub
'End If

```

```
Worksheets("vertical figure").Range("B" & Rows.Count).End(xlUp).Offset  
    (1).Value = comment  
  
Worksheets("vertical output").Cells.Copy Destination:=Worksheets(sheetname & "v").  
    Range("A1")  
  
'-----Count-----  
  
Worksheets("vertical figure").Activate  
  
Dim myRange As Range  
Dim NumRows As Integer  
  
Set myRange = Range("A:A")  
NumRows = Application.WorksheetFunction.CountA(myRange)  
  
End Sub
```


D

Excel Module, *Steering*

```
Sub steering1()  
  
Dim e1 As Integer  
Dim i As Integer  
Dim j As Integer  
  
'-----Clear the simulation result from last simulation-----  
  
'-----clear simulation worksheet-----  
With Worksheets("simulation")  
  
    .Range("C2:C48").ClearContents  
    .Range("D2:D25").ClearContents  
    .Range("F29:F48").ClearContents  
    .Range("J29:J40").ClearContents  
    e1 = .Cells(1, Columns.Count).End(xlToLeft).Column  
  
    For i = 1 To 25  
        For j = 12 To e1  
            .Cells(i, j).ClearContents  
        Next j  
    Next i  
  
End With  
  
'-----clear intermediate worksheet-----  
  
With Worksheets("intermediate")  
  
    .Range("C56:C79").ClearContents  
  
    e1 = .Cells(1, Columns.Count).End(xlToLeft).Column  
    For i = 1 To 53  
        For j = 3 To e1  
            .Cells(i, j).ClearContents  
        Next j  
    Next i  
  
    e1 = .Cells(65, Columns.Count).End(xlToLeft).Column  
    For i = 65 To 66  
        For j = 7 To e1  
            .Cells(i, j).ClearContents  
        Next j  
    Next i  
  
End With  
  
'-----clear steering output worksheet-----  
  
With Worksheets("steering output")  
  
    e1 = .Cells(Rows.Count, 1).End(xlUp).Row  
  
    For i = 1 To 20  
        For j = 2 To e1
```

```

.Cells(j, i).ClearContents
Next j
Next i

End With

Worksheets("input").Activate

Dim R As Single
Dim camber As Double
Dim toe As Double

R = Range("I7").Value

Pi = Application.Pi()
camber = Range("I10").Value * (Pi / 180)
toe = -Range("I11").Value * (Pi / 180)

'auxiliary, vector top mount 7 - damper bottom 44
Range("D23").Value = 0.1 * (Range("D5").Value - Range("D15").Value) + Range("D15").Value
Range("E23").Value = 0.1 * (Range("E5").Value - Range("E15").Value) + Range("E15").Value
Range("F23").Value = 0.1 * (Range("F5").Value - Range("F15").Value) + Range("F15").Value

'spindle axis
Range("D24").Value = Range("D6").Value - 100 * Sin(toe)
Range("E24").Value = Range("E6").Value + 100 - 100 * (1 - Cos(camber)) - 100 * (1 - Cos(toe))
Range("F24").Value = Range("F6").Value + 100 * Sin(camber)

'tire contacting
Range("D25").Value = Range("D6").Value
Range("E25").Value = Range("E6").Value
Range("F25").Value = Range("F6").Value - R

'-----

Range("D49").Value = Range("D23").Value
Range("E49").Value = -Range("E23").Value
Range("F49").Value = Range("F23").Value

Range("D50").Value = Range("D24").Value
Range("E50").Value = -Range("E24").Value
Range("F50").Value = Range("F24").Value

Range("D51").Value = Range("D25").Value
Range("E51").Value = -Range("E25").Value
Range("F51").Value = Range("F25").Value
'-----calculation preparation-----
With Worksheets("input")

Worksheets("simulation").Range("F29").Value = (.Range("D4").Value - .Range("D2").Value) ^ 2 + (.Range("E4").Value - .Range("E2").Value) ^ 2 + (.Range("F4").Value - .Range("F2").Value) ^ 2
Worksheets("simulation").Range("F30").Value = (.Range("D4").Value - .Range("D3").Value) ^ 2 + (.Range("E4").Value - .Range("E3").Value) ^ 2 + (.Range("F4").Value - .Range("F3").Value) ^ 2
Worksheets("simulation").Range("F31").Value = (.Range("D10").Value - .Range("D9").Value) ^ 2 + (.Range("E10").Value - .Range("E9").Value) ^ 2 + (.Range("F10").Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F32").Value = (.Range("D15").Value - .Range("D5").Value) ^ 2 + (.Range("E15").Value - .Range("E5").Value) ^ 2 + (.Range("F15").Value - .Range("F5").Value) ^ 2

```

```

Worksheets("simulation").Range("F33").Value = (.Range("D15").Value - .Range("D4").
Value) ^ 2 + (.Range("E15").Value - .Range("E4").Value) ^ 2 + (.Range("F15").
Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F34").Value = (.Range("D23").Value - .Range("D4").
Value) ^ 2 + (.Range("E23").Value - .Range("E4").Value) ^ 2 + (.Range("F23").
Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F35").Value = (.Range("D9").Value - .Range("D4").
Value) ^ 2 + (.Range("E9").Value - .Range("E4").Value) ^ 2 + (.Range("F9").
Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F36").Value = (.Range("D23").Value - .Range("D15").
Value) ^ 2 + (.Range("E23").Value - .Range("E15").Value) ^ 2 + (.Range("F23").
Value - .Range("F15").Value) ^ 2
Worksheets("simulation").Range("F37").Value = (.Range("D15").Value - .Range("D9").
Value) ^ 2 + (.Range("E15").Value - .Range("E9").Value) ^ 2 + (.Range("F15").
Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F38").Value = (.Range("D23").Value - .Range("D9").
Value) ^ 2 + (.Range("E23").Value - .Range("E9").Value) ^ 2 + (.Range("F23").
Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F39").Value = (.Range("D6").Value - .Range("D4").
Value) ^ 2 + (.Range("E6").Value - .Range("E4").Value) ^ 2 + (.Range("F6").
Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F40").Value = (.Range("D15").Value - .Range("D6").
Value) ^ 2 + (.Range("E15").Value - .Range("E6").Value) ^ 2 + (.Range("F15").
Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F41").Value = (.Range("D23").Value - .Range("D6").
Value) ^ 2 + (.Range("E23").Value - .Range("E6").Value) ^ 2 + (.Range("F23").
Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F42").Value = (.Range("D24").Value - .Range("D6").
Value) ^ 2 + (.Range("E24").Value - .Range("E6").Value) ^ 2 + (.Range("F24").
Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F43").Value = (.Range("D25").Value - .Range("D6").
Value) ^ 2 + (.Range("E25").Value - .Range("E6").Value) ^ 2 + (.Range("F25").
Value - .Range("F6").Value) ^ 2
Worksheets("simulation").Range("F44").Value = (.Range("D24").Value - .Range("D4").
Value) ^ 2 + (.Range("E24").Value - .Range("E4").Value) ^ 2 + (.Range("F24").
Value - .Range("F4").Value) ^ 2
Worksheets("simulation").Range("F45").Value = (.Range("D24").Value - .Range("D9").
Value) ^ 2 + (.Range("E24").Value - .Range("E9").Value) ^ 2 + (.Range("F24").
Value - .Range("F9").Value) ^ 2
Worksheets("simulation").Range("F46").Value = (.Range("D24").Value - .Range("D15").
Value) ^ 2 + (.Range("E24").Value - .Range("E15").Value) ^ 2 + (.Range("F24").
Value - .Range("F15").Value) ^ 2
Worksheets("simulation").Range("F47").Value = (.Range("D24").Value - .Range("D23").
Value) ^ 2 + (.Range("E24").Value - .Range("E23").Value) ^ 2 + (.Range("F24").
Value - .Range("F23").Value) ^ 2
Worksheets("simulation").Range("F48").Value = (.Range("D24").Value - .Range("D11").
Value) ^ 2 + (.Range("E24").Value - .Range("E11").Value) ^ 2 + (.Range("F24").
Value - .Range("F11").Value) ^ 2

End With

With Worksheets("simulation")

.Range("J29").Value = Range("D28").Value
.Range("J30").Value = Range("E28").Value
.Range("J31").Value = Range("F28").Value

.Range("J32").Value = Range("D29").Value
.Range("J33").Value = Range("E29").Value
.Range("J34").Value = Range("F29").Value

.Range("J35").Value = Range("D36").Value
.Range("J36").Value = Range("E36").Value
.Range("J37").Value = Range("F36").Value

.Range("J38").Value = Range("D31").Value
.Range("J39").Value = Range("E31").Value
.Range("J40").Value = Range("F31").Value

```

```

End With

Dim rack As Single
Dim step As Integer
Dim stepsize As Single
Dim k As Integer
Dim tol As Double
Dim L As Double
Dim t As Double
Dim maxtime As Double
Dim maxiteration As Double

'-----simulation set up-----

step = Range("I5").Value
rack = Range("I6").Value
tol = Range("I8").Value
L = Range("I9").Value
t = 2 * Range("E32").Value
stepsize = rack / (step / 2 - 1)
Worksheets("simulation").Range("rp3_y").Value = Range("E36").Value

'-----back to simulation sheet-----

Worksheets("simulation").Activate

'-----set initial guess-----

Range("D2").Value = Worksheets("input").Range("D30").Value
Range("D3").Value = Worksheets("input").Range("E30").Value
Range("D4").Value = Worksheets("input").Range("F30").Value

Range("D5").Value = Worksheets("input").Range("D41").Value
Range("D6").Value = Worksheets("input").Range("E41").Value
Range("D7").Value = Worksheets("input").Range("F41").Value

Range("D8").Value = Worksheets("input").Range("D49").Value
Range("D9").Value = Worksheets("input").Range("E49").Value
Range("D10").Value = Worksheets("input").Range("F49").Value

Range("D11").Value = Worksheets("input").Range("D35").Value
Range("D12").Value = Worksheets("input").Range("E35").Value
Range("D13").Value = Worksheets("input").Range("F35").Value

Range("D14").Value = Worksheets("input").Range("D41").Value
Range("D15").Value = Worksheets("input").Range("E41").Value
Range("D16").Value = Worksheets("input").Range("F41").Value

Range("D17").Value = Worksheets("input").Range("D32").Value
Range("D18").Value = Worksheets("input").Range("E32").Value
Range("D19").Value = Worksheets("input").Range("F32").Value

Range("D20").Value = Worksheets("input").Range("D50").Value
Range("D21").Value = Worksheets("input").Range("E50").Value
Range("D22").Value = Worksheets("input").Range("F50").Value

Range("D23").Value = Worksheets("input").Range("D51").Value
Range("D24").Value = Worksheets("input").Range("E51").Value
Range("D25").Value = Worksheets("input").Range("F51").Value

'-----reset coordinates and slack variable-----

Range("C2:C25").Value = Range("D2:D25").Value

```



```

'Range("C20:C22").Value = 0

Range("C26:C48").Value = Range("F2:F24").Value

'-----step counter-----
For j = 1 To step
Cells(1, 11 + j) = j
Next j

'----- first half, rp3_y+ -----

For i = 1 To step / 2

'----- solver -----

SolverReset
Solveroptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$2", MaxMinVal:=3, valueof:=0, ByChange:= _
        "$C$2:$C$3,$C$5:$C$19,$C$26:$C$42", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$2:$F$18", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = True
'SolverOptions Iterations:=1E+16, Precision:=0.000001, Convergence:=0.000000001,
    AssumeNonNeg:=True
'-----Equation value -----
Cells(2, step / 2 + 12 - i).Value = Range("F2").Value
Cells(3, step / 2 + 12 - i).Value = Range("F3").Value
Cells(4, step / 2 + 12 - i).Value = Range("F4").Value
Cells(5, step / 2 + 12 - i).Value = Range("F5").Value
Cells(6, step / 2 + 12 - i).Value = Range("F6").Value
Cells(7, step / 2 + 12 - i).Value = Range("F7").Value
Cells(8, step / 2 + 12 - i).Value = Range("F8").Value
Cells(9, step / 2 + 12 - i).Value = Range("F9").Value
Cells(10, step / 2 + 12 - i).Value = Range("F10").Value
Cells(11, step / 2 + 12 - i).Value = Range("F11").Value
Cells(12, step / 2 + 12 - i).Value = Range("F12").Value
Cells(13, step / 2 + 12 - i).Value = Range("F13").Value
Cells(14, step / 2 + 12 - i).Value = Range("F14").Value
Cells(15, step / 2 + 12 - i).Value = Range("F15").Value
Cells(16, step / 2 + 12 - i).Value = Range("F16").Value
Cells(17, step / 2 + 12 - i).Value = Range("F17").Value
Cells(18, step / 2 + 12 - i).Value = Range("F18").Value
'----- solver2 -----

SolverReset
Solveroptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$3", MaxMinVal:=3, valueof:=0, ByChange:= _
        "$C$20:$C$25,$C$43:$C$48", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$19:$F$24", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = True
'----- Equation value -----
Cells(19, step / 2 + 12 - i).Value = Range("F19").Value
Cells(20, step / 2 + 12 - i).Value = Range("F20").Value
Cells(21, step / 2 + 12 - i).Value = Range("F21").Value
Cells(22, step / 2 + 12 - i).Value = Range("F22").Value
Cells(23, step / 2 + 12 - i).Value = Range("F23").Value
Cells(24, step / 2 + 12 - i).Value = Range("F24").Value

'-----save data to result, right side-----

Worksheets("intermediate").Cells(2, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(3, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_2").Value

```

```

Worksheets("intermediate").Cells(4, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(5, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(6, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_5").Value
Worksheets("intermediate").Cells(7, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(8, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(9, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_8").Value
Worksheets("intermediate").Cells(10, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(11, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_10").Value
Worksheets("intermediate").Cells(12, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_11").Value
Worksheets("intermediate").Cells(13, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_12").Value
Worksheets("intermediate").Cells(14, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_13").Value
Worksheets("intermediate").Cells(15, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_14").Value
Worksheets("intermediate").Cells(16, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_15").Value
Worksheets("intermediate").Cells(17, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_16").Value
Worksheets("intermediate").Cells(18, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_17").Value
Worksheets("intermediate").Cells(19, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_18").Value
Worksheets("intermediate").Cells(20, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_19").Value
Worksheets("intermediate").Cells(21, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_20").Value
Worksheets("intermediate").Cells(22, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_21").Value
Worksheets("intermediate").Cells(23, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_22").Value
Worksheets("intermediate").Cells(24, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_23").Value
Worksheets("intermediate").Cells(25, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_24").Value
Worksheets("intermediate").Cells(26, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("rp3_y").Value

'-----save data to result, left side-----

Worksheets("intermediate").Cells(29, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(30, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_2").Value * (-1)
Worksheets("intermediate").Cells(31, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(32, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(33, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_5").Value * (-1)
Worksheets("intermediate").Cells(34, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(35, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(36, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_8").Value * (-1)
Worksheets("intermediate").Cells(37, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(38, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_10").Value

```

```

Worksheets("intermediate").Cells(39, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_11").Value * (-1)
Worksheets("intermediate").Cells(40, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_12").Value
Worksheets("intermediate").Cells(41, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_13").Value
Worksheets("intermediate").Cells(42, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_14").Value * (-1)
Worksheets("intermediate").Cells(43, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_15").Value
Worksheets("intermediate").Cells(44, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_16").Value
Worksheets("intermediate").Cells(45, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_17").Value * (-1)
Worksheets("intermediate").Cells(46, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_18").Value
Worksheets("intermediate").Cells(47, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_19").Value
Worksheets("intermediate").Cells(48, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_20").Value * (-1)
Worksheets("intermediate").Cells(49, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_21").Value
Worksheets("intermediate").Cells(50, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_22").Value
Worksheets("intermediate").Cells(51, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_23").Value * (-1)
Worksheets("intermediate").Cells(52, step / 2 + 1 + i).Value = Worksheets("simulation").Range("x_24").Value
Worksheets("intermediate").Cells(53, step / 2 + 1 + i).Value = Worksheets("simulation").Range("rp3_y").Value * (-1)

Worksheets("simulation").Range("rp3_y").Value = Worksheets("simulation").Range("rp3_y").Value + stepsize
Next i

'-----reset coordinates and slack variable-----

Range("C2:C25").Value = Range("D2:D25").Value
'Range("C20:C22").Value = 0

Range("C26:C48").Value = Range("F2:F24").Value
Worksheets("simulation").Range("rp3_y").Value = Worksheets("input").Range("E36").Value

'-----second half, rp3_y-----
For i = 1 To step / 2

'----- solver -----
SolverReset
    SolverOptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$2", MaxMinVal:=3, valueof:=0, ByChange:= _
        "$C$2:$C$3,$C$5:$C$19,$C$26:$C$42", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$2:$F$18", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False
'SolverOptions Iterations:=1E+16, Precision:=0.000001, Convergence:=0.000000001,
    AssumeNonNeg:=True
'-----Equation value -----
Cells(2, step / 2 + 10 + i).Value = Range("F2").Value
Cells(3, step / 2 + 10 + i).Value = Range("F3").Value
Cells(4, step / 2 + 10 + i).Value = Range("F4").Value
Cells(5, step / 2 + 10 + i).Value = Range("F5").Value
Cells(6, step / 2 + 10 + i).Value = Range("F6").Value
Cells(7, step / 2 + 10 + i).Value = Range("F7").Value

```

```

Cells(8, step / 2 + 10 + i).Value = Range("F8").Value
Cells(9, step / 2 + 10 + i).Value = Range("F9").Value
Cells(10, step / 2 + 10 + i).Value = Range("F10").Value
Cells(11, step / 2 + 10 + i).Value = Range("F11").Value
Cells(12, step / 2 + 10 + i).Value = Range("F12").Value
Cells(13, step / 2 + 10 + i).Value = Range("F13").Value
Cells(14, step / 2 + 10 + i).Value = Range("F14").Value
Cells(15, step / 2 + 10 + i).Value = Range("F15").Value
Cells(16, step / 2 + 10 + i).Value = Range("F16").Value
Cells(17, step / 2 + 10 + i).Value = Range("F17").Value
Cells(18, step / 2 + 10 + i).Value = Range("F18").Value

'----- solver2 -----

SolverReset

SolverOptions Precision:=tol, maxtime:=maxtime, Iterations:=maxiteration

    SolverOk SetCell:="$I$3", MaxMinVal:=3, valueof:=0, ByChange:= _
        "$C$20:$C$25,$C$43:$C$48", Engine:=1, EngineDesc:="GRG Nonlinear"
    SolverAdd CellRef:="$F$19:$F$24", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Application.ScreenUpdating = False

'----- Equation value -----
Cells(19, step / 2 + 10 + i).Value = Range("F19").Value
Cells(20, step / 2 + 10 + i).Value = Range("F20").Value
Cells(21, step / 2 + 10 + i).Value = Range("F21").Value
Cells(22, step / 2 + 10 + i).Value = Range("F22").Value
Cells(23, step / 2 + 10 + i).Value = Range("F23").Value
Cells(24, step / 2 + 10 + i).Value = Range("F24").Value

'-----save data to result, right side-----

Worksheets("intermediate").Cells(2, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(3, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_2").Value
Worksheets("intermediate").Cells(4, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(5, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(6, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_5").Value
Worksheets("intermediate").Cells(7, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(8, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(9, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_8").Value
Worksheets("intermediate").Cells(10, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(11, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_10").Value
Worksheets("intermediate").Cells(12, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_11").Value
Worksheets("intermediate").Cells(13, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_12").Value
Worksheets("intermediate").Cells(14, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_13").Value
Worksheets("intermediate").Cells(15, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_14").Value
Worksheets("intermediate").Cells(16, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_15").Value
Worksheets("intermediate").Cells(17, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_16").Value
Worksheets("intermediate").Cells(18, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_17").Value
Worksheets("intermediate").Cells(19, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_18").Value

```

```

Worksheets("intermediate").Cells(20, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_19").Value
Worksheets("intermediate").Cells(21, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_20").Value
Worksheets("intermediate").Cells(22, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_21").Value
Worksheets("intermediate").Cells(23, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_22").Value
Worksheets("intermediate").Cells(24, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_23").Value
Worksheets("intermediate").Cells(25, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("x_24").Value
Worksheets("intermediate").Cells(26, step / 2 + 1 + i).Value = Worksheets("
simulation").Range("rp3_y").Value

'-----save data to result, left side-----
Worksheets("intermediate").Cells(29, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_1").Value
Worksheets("intermediate").Cells(30, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_2").Value * (-1)
Worksheets("intermediate").Cells(31, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_3").Value
Worksheets("intermediate").Cells(32, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_4").Value
Worksheets("intermediate").Cells(33, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_5").Value * (-1)
Worksheets("intermediate").Cells(34, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_6").Value
Worksheets("intermediate").Cells(35, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_7").Value
Worksheets("intermediate").Cells(36, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_8").Value * (-1)
Worksheets("intermediate").Cells(37, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_9").Value
Worksheets("intermediate").Cells(38, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_10").Value
Worksheets("intermediate").Cells(39, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_11").Value * (-1)
Worksheets("intermediate").Cells(40, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_12").Value
Worksheets("intermediate").Cells(41, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_13").Value
Worksheets("intermediate").Cells(42, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_14").Value * (-1)
Worksheets("intermediate").Cells(43, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_15").Value
Worksheets("intermediate").Cells(44, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_16").Value
Worksheets("intermediate").Cells(45, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_17").Value * (-1)
Worksheets("intermediate").Cells(46, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_18").Value
Worksheets("intermediate").Cells(47, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_19").Value
Worksheets("intermediate").Cells(48, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_20").Value * (-1)
Worksheets("intermediate").Cells(49, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_21").Value
Worksheets("intermediate").Cells(50, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_22").Value
Worksheets("intermediate").Cells(51, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_23").Value * (-1)
Worksheets("intermediate").Cells(52, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("x_24").Value
Worksheets("intermediate").Cells(53, step / 2 + 3 - i).Value = Worksheets("
simulation").Range("rp3_y").Value * (-1)

Worksheets("simulation").Range("rp3_y").Value = Worksheets("simulation").Range("
rp3_y").Value - stepsize

```

```

Next i

For i = 1 To (step - 1)

Cells(25, 11 + i).Value = Abs(Cells(2, 11 + i)) + Abs(Cells(3, 11 + i)) + Abs(
    Cells(4, 11 + i)) + Abs(Cells(5, 11 + i)) + Abs(Cells(6, 11 + i)) + Abs(Cells
        (7, 11 + i)) _
        + Abs(Cells(8, 11 + i)) + Abs(Cells(9, 11 + i)) + Abs(Cells
            (10, 11 + i)) + Abs(Cells(11, 11 + i)) + Abs(Cells(12,
                11 + i)) + Abs(Cells(13, 11 + i)) _
        + Abs(Cells(14, 11 + i)) + Abs(Cells(15, 11 + i)) + Abs(
            Cells(16, 11 + i)) + Abs(Cells(17, 11 + i)) + Abs(Cells
                (18, 11 + i)) + Abs(Cells(19, 11 + i)) _
        + Abs(Cells(20, 11 + i)) + Abs(Cells(21, 11 + i)) + Abs(
            Cells(22, 11 + i)) + Abs(Cells(23, 11 + i)) + Abs(Cells
                (24, 11 + i))

'Cells(25, 11 + i).Value = Application.Sum(Range(Cells(2, 11 + i), Cells(24, 11 +
    i)))
Worksheets("intermediate").Cells(1, i + 2).Value = i

Next i

'-----calculating output-----

For i = 1 To step
Worksheets("steering output").Cells(i + 1, 1).Value = Worksheets("intermediate").
    Cells(26, i + 2).Value

Next i

Pi = Application.Pi()

Worksheets("intermediate").Activate

Range("p6x").Value = Cells(2, 3).Value
Range("p6y").Value = Cells(3, 3).Value
Range("p6z").Value = Cells(4, 3).Value

Range("p7x").Value = Worksheets("input").Range("D31").Value
Range("p7y").Value = Worksheets("input").Range("E31").Value
Range("p7z").Value = Worksheets("input").Range("F31").Value

Range("p12x").Value = Cells(11, 3).Value
Range("p12y").Value = Cells(12, 3).Value
Range("p12z").Value = Cells(13, 3).Value

Range("p21x").Value = Cells(23, 3).Value
Range("p21y").Value = Cells(24, 3).Value
Range("p21z").Value = Cells(25, 3).Value

Range("px").Value = Cells(17, 3).Value
Range("py").Value = Cells(18, 3).Value
Range("pz").Value = Cells(19, 3).Value

Range("p14x").Value = Worksheets("input").Range("D36").Value
Range("p14y").Value = Worksheets("input").Range("E36").Value
Range("p14z").Value = Worksheets("input").Range("F36").Value

SolverReset

    SolverOk SetCell:="$E$61", MaxMinVal:=3, valueof:=0, ByChange:="$C$68:$C$70", _
        Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:="$E$57:$E$59", Relation:=2, FormulaText:="0"
    SolverSolve userfinish = False

Dim a As Single
Dim b As Single

```

```

Dim c As Single
Dim d As Double
Dim e As Double
Dim f As Double

For i = 1 To step - 1

With Worksheets("steering output")
'rack travel
.Cells(i + 1, 2).Value = Worksheets("steering output").Cells(i + 1, 1) -
    Worksheets("input").Range("E36").Value

'right camber angle
.Cells(i + 1, 3).Value = (180 / Pi) * Atn((Cells(22, i + 2).Value - Cells(19, i +
    2).Value) / (Cells(18, i + 2).Value - Cells(21, i + 2).Value))
'left camber angle
.Cells(i + 1, 4).Value = (180 / Pi) * Atn((Cells(49, i + 2).Value - Cells(46, i +
    2).Value) / (Cells(48, i + 2).Value - Cells(45, i + 2).Value))

'right toe angle
.Cells(i + 1, 5).Value = (180 / Pi) * Atn((Cells(17, i + 2).Value - Cells(20, i +
    2).Value) / (Cells(21, i + 2).Value - Cells(18, i + 2).Value))
'left toe angle;
.Cells(i + 1, 6).Value = (180 / Pi) * Atn((Cells(44, i + 2).Value - Cells(47, i +
    2).Value) / (Cells(48, i + 2).Value - Cells(45, i + 2).Value))

'track width
.Cells(i + 1, 7).Value = Cells(18, i + 2).Value - Cells(45, i + 2).Value

'right wheel center X
.Cells(i + 1, 8).Value = Cells(17, i + 2) - Worksheets("input").Range("D32").Value
'left wheel center X
.Cells(i + 1, 9).Value = Cells(44, i + 2) - Worksheets("input").Range("D32").Value

'track rod angle
a = (Range("px") - Cells(11, i + 2)) * (Range("p14x") - Cells(11, i + 2)) + (Range
    ("py") - Cells(12, i + 2)) * (Range("p14y") - Cells(12, i + 2)) + (Range("pz")
    - Cells(13, i + 2)) * (Range("p14z") - Cells(13, i + 2))
b = Sqr((Range("px") - Cells(11, i + 2)) ^ 2 + (Range("py") - Cells(12, i + 2)) ^
    2 + (Range("pz") - Cells(13, i + 2)) ^ 2)
c = Sqr((Range("p14x") - Cells(11, i + 2)) ^ 2 + (Range("p14y") - Cells(12, i + 2)
    ) ^ 2 + (Range("p14z") - Cells(13, i + 2)) ^ 2)
Worksheets("steering output").Cells(i + 1, 10).Value = (180 / Pi) * Application.
    Acos(a / (b * c))

'ackermann
'e = t / L + 1 / d
'f = 1 / e
'.Cells(i + 1, 11).Value = (.Cells(i + 1, 5).Value - .Cells(i + 1, 6).Value) / (.
    Cells(i + 1, 5).Value - (180 / Pi) * Atn(f))
'acotd(cotd(theta_in)+t/L)

'error
'simulation error
.Cells(i + 1, 14).Value = Worksheets("simulation").Cells(25, 11 + i).Value 'error
.Cells(i + 1, 13).Value = Worksheets("simulation").Cells(1, 11 + i).Value 'step

End With

Next i

For i = 1 To step / 2 - 1

With Worksheets("steering output")

d = (Cells(44, i + 2).Value - Cells(47, i + 2).Value) / (Cells(48, i + 2).Value -
    Cells(45, i + 2).Value)
e = t / L + 1 / d
f = 1 / e

```



```

.Cells(i + 1, 11).Value = (180 / Pi) * Atn(f)
.Cells(i + 1, 12).Value = (.Cells(i + 1, 6).Value - .Cells(i + 1, 5).Value) / (.
    Cells(i + 1, 6).Value - (180 / Pi) * Atn(f)) * 100

End With

Next i

'-----copy hard point-----

Worksheets("input").Range("A1:F22").Copy Destination:=Worksheets("steering output"
).Range("O1")

'-----creating new sheet-----
Dim currTime As Date
Dim sheetname As String

currTime = Now()
sheetname = Format(Now(), "hh nn ss")
Sheets.Add(After:=Sheets(Sheets.Count)).name = sheetname & "s"

Worksheets("steering figure").Range("A" & Rows.Count).End(xlUp).Offset(1) =
    sheetname & "s"

Dim comment As String
'Dim xSht As Object
'On Error Resume Next
comment = InputBox("Please enter a comment for this simulation result ", "
    Kutools for Excel")
'If xName = "" Then Exit Sub
'Set xSht = Sheets(xName)
'If Not xSht Is Nothing Then
'    MsgBox "Sheet cannot be created as there is already a worksheet with
        the same name in this workbook"
'Exit Sub
'End If
Worksheets("steering figure").Range("B" & Rows.Count).End(xlUp).Offset
(1).Value = comment

Worksheets("steering output").Cells.Copy Destination:=Worksheets(sheetname & "s").
    Range("A1")

'-----Count-----

Worksheets("steering figure").Activate

Dim myRange As Range
Dim NumRows As Integer

Set myRange = Range("A:A")
NumRows = Application.WorksheetFunction.CountA(myRange)

End Sub

```


E

Excel Module, *Vertical figure*

```
Sub vertical_figure()  
  
Dim step As Integer  
  
step = Worksheets("input").Range("I5").Value  
  
Worksheets("vertical figure").Activate  
  
Dim myRange As Range  
Dim NumRows As Integer  
Dim name As String  
  
Set myRange = Range("A:A")  
NumRows = Application.WorksheetFunction.CountA(myRange)  
  
'-----define the plot needed-----  
'1.camber plot  
  
Dim ct1 As Chart  
Dim co1 As ChartObject  
Dim sc1 As SeriesCollection  
Dim ser1 As Series  
  
Set co1 = Worksheets("vertical figure").ChartObjects.Add(Range("D2").Left, Range("D2").Top, 300, 180)  
co1.name = "Camber"  
  
Set ct1 = co1.Chart  
With ct1  
  
    .HasLegend = True  
    .Legend.Position = xlLegendPositionBottom  
    .HasAxis(xlCategory, xlPrimary) = True  
    .Axes(xlCategory).HasTitle = True  
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"  
  
    .HasAxis(xlValue, xlPrimary) = True  
    .Axes(xlValue).HasTitle = True  
    .Axes(xlValue).AxisTitle.Text = "[deg]"  
  
    .HasTitle = True  
    .ChartTitle.Text = "Camber Angle"  
    .ChartTitle.Font.Size = 12  
End With  
  
'2.toe plot-----  
  
Dim ct2 As Chart  
Dim co2 As ChartObject  
Dim sc2 As SeriesCollection  
Dim ser2 As Series  
  
Set co2 = Worksheets("vertical figure").ChartObjects.Add(Range("D15").Left, Range("D15").Top, 300, 180)
```

```

co2.name = "Toe"

Set ct2 = co2.Chart
With ct2

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Toe Angle"
    .ChartTitle.Font.Size = 12
End With

'3.track width-----
Dim ct3 As Chart
Dim co3 As ChartObject
Dim sc3 As SeriesCollection
Dim ser3 As Series

Set co3 = Worksheets("vertical figure").ChartObjects.Add(Range("D28").Left, Range(
    "D28").Top, 300, 180)
co3.name = "track width"

Set ct3 = co3.Chart
With ct3

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[mm]"

    .HasTitle = True
    .ChartTitle.Text = "Track Width"
    .ChartTitle.Font.Size = 12
End With

'4.wheel center x-----
Dim ct4 As Chart
Dim co4 As ChartObject
Dim sc4 As SeriesCollection
Dim ser4 As Series

Set co4 = Worksheets("vertical figure").ChartObjects.Add(Range("K2").Left, Range("
    K2").Top, 300, 180)
co4.name = "wheel center x"

Set ct4 = co4.Chart
With ct4

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True

```

```

.Axes(xlValue).HasTitle = True
.Axes(xlValue).AxisTitle.Text = "[mm]"

.HasTitle = True
.ChartTitle.Text = "Wheel Center X Displacement"
.ChartTitle.Font.Size = 12

End With

'5.roll center height-----
Dim ct5 As Chart
Dim co5 As ChartObject
Dim sc5 As SeriesCollection
Dim ser5 As Series

Set co5 = Worksheets("vertical figure").ChartObjects.Add(Range("K15").Left, Range(
    "K15").Top, 300, 180)
co5.name = "rch"

Set ct5 = co5.Chart
With ct5

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[mm]"

    .HasTitle = True
    .ChartTitle.Text = "Roll Center Height"
    .ChartTitle.Font.Size = 12

End With

'6.kpi angle-----
Dim ct6 As Chart
Dim co6 As ChartObject
Dim sc6 As SeriesCollection
Dim ser6 As Series

Set co6 = Worksheets("vertical figure").ChartObjects.Add(Range("K28").Left, Range(
    "K28").Top, 300, 180)
co6.name = "kpi angle"

Set ct6 = co6.Chart
With ct6

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Kingpin Inclination Angle"
    .ChartTitle.Font.Size = 12

End With

'7.caster angle-----

```

```

Dim ct7 As Chart
Dim co7 As ChartObject
Dim sc7 As SeriesCollection
Dim ser7 As Series

Set co7 = Worksheets("vertical figure").ChartObjects.Add(Range("R2").Left, Range("
R2").Top, 300, 180)
co7.name = "caster angle"

Set ct7 = co7.Chart
With ct7

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Caster Angle"
    .ChartTitle.Font.Size = 12

End With

'8.caster trail-----
Dim ct8 As Chart
Dim co8 As ChartObject
Dim sc8 As SeriesCollection
Dim ser8 As Series

Set co8 = Worksheets("vertical figure").ChartObjects.Add(Range("R15").Left, Range(
"R15").Top, 300, 180)
co7.name = "caster trail"

Set ct8 = co8.Chart
With ct8

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[mm]"

    .HasTitle = True
    .ChartTitle.Text = "Caster Trail"
    .ChartTitle.Font.Size = 12

End With

'9.scrub radius-----
Dim ct9 As Chart
Dim co9 As ChartObject
Dim sc9 As SeriesCollection
Dim ser9 As Series

Set co9 = Worksheets("vertical figure").ChartObjects.Add(Range("R28").Left, Range(
"R28").Top, 300, 180)
co7.name = "scrub radius"

```

```

Set ct9 = co9.Chart
With ct9

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[mm]"

    .HasTitle = True
    .ChartTitle.Text = "Scrub Radius"
    .ChartTitle.Font.Size = 12

End With

'10. ratio-----
Dim ct10 As Chart
Dim co10 As ChartObject
Dim sc10 As SeriesCollection
Dim ser10 As Series

Set co10 = Worksheets("vertical figure").ChartObjects.Add(Range("Y2").Left, Range(
    "Y2").Top, 300, 180)
co10.name = "ratio"

Set ct10 = co10.Chart
With ct10

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "wheel vertical travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[-]"

    .HasTitle = True
    .ChartTitle.Text = "Installation Ratio"
    .ChartTitle.Font.Size = 12

End With

'11. simulation error-----
Dim ct11 As Chart
Dim co11 As ChartObject
Dim sc11 As SeriesCollection
Dim ser11 As Series

Set co11 = Worksheets("vertical figure").ChartObjects.Add(Range("Y15").Left, Range(
    "Y15").Top, 300, 180)
co11.name = "error"

Set ct11 = co11.Chart
With ct11

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "step [-]"

```

```
.HasAxis(xlValue, xlPrimary) = True
.Axes(xlValue).HasTitle = True
.Axes(xlValue).AxisTitle.Text = "[~]"

.HasTitle = True
.ChartTitle.Text = "Simulation error"
.ChartTitle.Font.Size = 12

End With

Range("G1").Value = "Vertical Simulation"

'-----assign curve series
'-----
Dim j As Integer

j = 1

For i = 1 To NumRows

If Cells(1 + i, 3).Value = 1 Then

name = Cells(1 + i, 1).Value

Cells(1, 8 + j).Value = name

j = j + 1

Cells(1, 31 + j).Value = name
'-----copy value parameter-----
Range(Cells(2, 31 + j), Cells(10, 31 + j)).Value = Worksheets(name).Range("P1:P9")
.Value
Cells(11, 31 + j).Value = Worksheets(name).Range("P12").Value
Cells(12, 31 + j).Value = Worksheets(name).Range("P13").Value

'-----camber-----
With ct1
Set sc1 = .SeriesCollection
Set ser1 = sc1.NewSeries

With ser1
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 3), Worksheets(name).
Cells(step, 3))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----toe-----
With ct2
Set sc2 = .SeriesCollection
Set ser2 = sc2.NewSeries

With ser2
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 4), Worksheets(name).
Cells(step, 4))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----track width-----
With ct3
```

```

Set sc3 = .SeriesCollection
Set ser3 = sc3.NewSeries

With ser3
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 5), Worksheets(name).
Cells(step, 5))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----wheel center x-----
With ct4
Set sc4 = .SeriesCollection
Set ser4 = sc4.NewSeries

With ser4
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 6), Worksheets(name).
Cells(step, 6))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----roll center height-----
With ct5
Set sc5 = .SeriesCollection
Set ser5 = sc5.NewSeries

With ser5
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 9), Worksheets(name).
Cells(step, 9))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----kpi angle-----
With ct6
Set sc6 = .SeriesCollection
Set ser6 = sc6.NewSeries

With ser6
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 10), Worksheets(name).
Cells(step, 10))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----caster angle-----
With ct7
Set sc7 = .SeriesCollection
Set ser7 = sc7.NewSeries

With ser7
.name = name

```

```
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 11), Worksheets(name).
Cells(step, 11))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----caster trail-----
With ct8
Set sc8 = .SeriesCollection
Set ser8 = sc8.NewSeries

With ser8
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 12), Worksheets(name).
Cells(step, 12))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----scrub radius-----
With ct9
Set sc9 = .SeriesCollection
Set ser9 = sc9.NewSeries

With ser9
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 13), Worksheets(name).
Cells(step, 13))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----ratio-----
With ct10
Set sc10 = .SeriesCollection
Set ser10 = sc10.NewSeries

With ser10
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step - 1, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 14), Worksheets(name).
Cells(step - 1, 14))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----error-----
With ct11
Set sc11 = .SeriesCollection
Set ser11 = sc11.NewSeries

With ser11
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 19), Worksheets(name).
Cells(step - 1, 17))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 18), Worksheets(name).
Cells(step - 1, 18))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
```



```
End With  
End With  
  
End If  
Next i  
  
End Sub
```


F

Excel Module, *Steering figure*

```
Sub steering_figure()  
  
Dim step As Integer  
  
step = Worksheets("input").Range("I5").Value  
  
Worksheets("steering figure").Activate  
  
Dim myRange As Range  
Dim NumRows As Integer  
Dim name As String  
  
Set myRange = Range("A:A")  
NumRows = Application.WorksheetFunction.CountA(myRange)  
  
'-----define the plot needed-----  
'1.camber-----  
Dim ct1 As Chart  
Dim co1 As ChartObject  
Dim sc1 As SeriesCollection  
Dim ser1 As Series  
  
Set co1 = Worksheets("steering figure").ChartObjects.Add(Range("D2").Left, Range("D2").Top, 300, 180)  
co1.name = "Camber"  
  
Set ct1 = co1.Chart  
With ct1  
  
    .HasLegend = True  
    .Legend.Position = xlLegendPositionBottom  
    .HasAxis(xlCategory, xlPrimary) = False  
    .Axes(xlCategory).HasTitle = True  
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"  
  
    .HasAxis(xlValue, xlPrimary) = False  
    .Axes(xlValue).HasTitle = True  
    .Axes(xlValue).AxisTitle.Text = "[deg]"  
  
    .HasTitle = True  
    .ChartTitle.Text = "Camber Angle"  
    .ChartTitle.Font.Size = 12  
  
End With  
  
'2.toe-----  
Dim ct2 As Chart  
Dim co2 As ChartObject  
Dim sc2 As SeriesCollection  
Dim ser2 As Series  
  
Set co2 = Worksheets("steering figure").ChartObjects.Add(Range("D15").Left, Range("D15").Top, 300, 180)
```

```

co2.name = "Toe"

Set ct2 = co2.Chart
With ct2

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = False
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"

    .HasAxis(xlValue, xlPrimary) = False
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Steer Angle"
    .ChartTitle.Font.Size = 12

End With

'3.track width-----
Dim ct3 As Chart
Dim co3 As ChartObject
Dim sc3 As SeriesCollection
Dim ser3 As Series

Set co3 = Worksheets("steering figure").ChartObjects.Add(Range("D28").Left, Range("D28").Top, 300, 180)
co3.name = "track width"

Set ct3 = co3.Chart
With ct3

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Track Width"
    .ChartTitle.Font.Size = 12

End With

'4.wheel center longitudinal-----
Dim ct4 As Chart
Dim co4 As ChartObject
Dim sc4 As SeriesCollection
Dim ser4 As Series

Set co4 = Worksheets("steering figure").ChartObjects.Add(Range("K2").Left, Range("K2").Top, 300, 180)
co4.name = "wheel center longitudinal"

Set ct4 = co4.Chart
With ct4

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = False
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"

```

```

.HasAxis(xlValue, xlPrimary) = False
.Axes(xlValue).HasTitle = True
.Axes(xlValue).AxisTitle.Text = "[mm]"

.HasTitle = True
.ChartTitle.Text = "Wheel Center Longitudinal Displacement"
.ChartTitle.Font.Size = 12

End With

'5.track rod angle-----
Dim ct5 As Chart
Dim co5 As ChartObject
Dim sc5 As SeriesCollection
Dim ser5 As Series

Set co5 = Worksheets("steering figure").ChartObjects.Add(Range("K15").Left, Range(
    "K15").Top, 300, 180)
co5.name = "track rod angle"

Set ct5 = co5.Chart
With ct5

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[deg]"

    .HasTitle = True
    .ChartTitle.Text = "Track Rod to Steering Rack Angle"
    .ChartTitle.Font.Size = 12

End With

'6.ackermann-----
Dim ct6 As Chart
Dim co6 As ChartObject
Dim sc6 As SeriesCollection
Dim ser6 As Series

Set co6 = Worksheets("steering figure").ChartObjects.Add(Range("K28").Left, Range(
    "K28").Top, 300, 180)
co6.name = "percent ackermann"

Set ct6 = co6.Chart
With ct6

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "rack travel [mm]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[%]"

    .HasTitle = True
    .ChartTitle.Text = "Percent Ackermann"
    .ChartTitle.Font.Size = 12

End With

```

```

'7.error-----
Dim ct7 As Chart
Dim co7 As ChartObject
Dim sc7 As SeriesCollection
Dim ser7 As Series

Set co7 = Worksheets("steering figure").ChartObjects.Add(Range("R2").Left, Range("
R2").Top, 300, 180)
co7.name = "simulation error"

Set ct7 = co7.Chart
With ct7

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .HasAxis(xlCategory, xlPrimary) = True
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "step [-]"

    .HasAxis(xlValue, xlPrimary) = True
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Text = "[-]"

    .HasTitle = True
    .ChartTitle.Text = "Simulation error"
    .ChartTitle.Font.Size = 12

End With

'-----assign curve series
-----
Dim j As Integer

j = 1

For i = 1 To NumRows

If Cells(1 + i, 3).Value = 1 Then

'Worksheets(Cells(1 + i, 1).Value).Range("J1").Value = 1
name = Cells(1 + i, 1).Value
Cells(1, 8 + j).Value = name

j = j + 1
'-----camber-----

With ct1
Set sc11 = .SeriesCollection
Set ser11 = sc11.NewSeries
Set sc12 = .SeriesCollection
Set ser12 = sc12.NewSeries

With ser11
.name = name & "-Right"
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 3), Worksheets(name).
Cells(step, 3))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 1
End With

With ser12
.name = name & "-Left"
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 4), Worksheets(name).
Cells(step, 4))

```

```

.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----toe-----
With ct2
Set sc21 = .SeriesCollection
Set ser21 = sc21.NewSeries
Set sc22 = .SeriesCollection
Set ser22 = sc22.NewSeries

With ser21
.name = name & "-Right"
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 5), Worksheets(name).Cells(step, 5))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With

With ser22
.name = name & "-Left"
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 6), Worksheets(name).Cells(step, 6))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----track width-----
With ct3
Set sc3 = .SeriesCollection
Set ser3 = sc3.NewSeries

With ser3
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 7), Worksheets(name).Cells(step, 7))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----wheel center longitudinal-----
With ct4
Set sc41 = .SeriesCollection
Set ser41 = sc41.NewSeries
Set sc42 = .SeriesCollection
Set ser42 = sc42.NewSeries

With ser41
.name = name & "-Right "
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 8), Worksheets(name).Cells(step, 8))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With

With ser42

```

```
.name = name & "-Left"
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 9), Worksheets(name).
Cells(step, 9))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----track rod angle-----
With ct5
Set sc5 = .SeriesCollection
Set ser5 = sc5.NewSeries

With ser5
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 10), Worksheets(name).
Cells(step, 10))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----ackermann-----
With ct6
Set sc6 = .SeriesCollection
Set ser6 = sc6.NewSeries

With ser6
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 2), Worksheets(name).
Cells(step / 2 - 1, 2))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 12), Worksheets(name).
Cells(step / 2 - 1, 12))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

'-----error-----
With ct7
Set sc7 = .SeriesCollection
Set ser7 = sc7.NewSeries

With ser7
.name = name
.XValues = Worksheets(name).Range(Worksheets(name).Cells(2, 13), Worksheets(name).
Cells(step, 15))
.Values = Worksheets(name).Range(Worksheets(name).Cells(2, 14), Worksheets(name).
Cells(step, 14))
.ChartType = xlXYScatterLinesNoMarkers
.Format.Line.Weight = 2
End With
End With

End If
Next i

End Sub
```


G

Excel Module, *Export*

```
Sub Export()  
  
Dim fd As Office.FileDialog  
Dim lcol As Integer  
Dim start As Integer  
  
Set fd = Application.FileDialog(msoFileDialogFilePicker)  
  
With fd  
    .AllowMultiSelect = False  
    .Title = "Please select the template file"  
    .Filters.Add "Excel", "*.xls"  
    .Filters.Add "All", "*.*"  
  
    If .Show = True Then  
        txtfilename = .SelectedItems(1)  
    End If  
End With  
  
Set src = Workbooks.Open(txtfilename)  
  
ThisWorkbook.Activate  
  
lcol = src.Worksheets("HP list Front MacP Current").Cells(31, Columns.Count).End(  
xlToLeft).Column  
  
Worksheets("input").Range("D2:F22").Copy src.Worksheets("HP list Front MacP  
Current").Cells(31, lcol + 2)  
  
End Sub
```