

Optimising Tire Selection

Milan Hareshkumar Vaghani
BE Mechanical Engineering
DJ Sanghvi College of Engineering

Manav Nagda
BE Mechanical Engineering
DJ Sanghvi College of Engineering

Deep Shah
BE Mechanical Engineering
DJ Sanghvi College of Engineering

Aayush Birla
BE Mechanical Engineering
DJ Sanghvi College of Engineering

Abstract - This paper outlines efficient steps for selecting optimal tires for Formula Student race cars, leveraging insights from our experience with 'DJS Racing.' It guides teams through data collection, analysis, tire characteristics, testing, simulation, and cost considerations. These strategies empower Formula Student teams to make informed decisions, enhancing their competitiveness.

I. INTRODUCTION

The first patent of a pneumatic tire was lodged by a Scottish inventor Robert William Thomson in 1847 when he realized that “rubber could withstand the wear and tear of being a tire while also maintaining its resilience.” The major boom in pneumatic tires came in 1946 when Michelin developed the radial tire method of construction. Due to its superior handling properties and increased fuel economy, pneumatic tires are still used as the mainstream tires for almost all automobiles today. Tires are the only four parts that are in contact with the ground, hence each force acting on the car and the road is the product of interaction between the tire surface and the road. Realizing this key point Chevrolet made the first tire testing lab to test the forces and their reactions acting on the tire and the vehicle. In today’s world, there are numerous tires, and each tire has different properties due to the difference in compounds used, thus selecting optimum tires for a car is a lengthy, costly, and tiresome process. Thus, selecting optimum tires for a car must be done by computer to have a selection range from multiple tires. This project deals with this problem by using tire data provided by the tire testing consortium and using that data, to firstly write a code for segregation, filtering, and sorting of data, secondly to use the filtered data to build a model to predict lateral force exerted by the tire and lastly to find the most suitable parameters to compare the tire data from.

II. PAPER MOTIVATION

In the field of tire selection, it is observed that automotive companies set up separate tire testing labs or simply have a standard predefined set of tires on which they sometimes base or use their car on. This creates myriad issues for the cars at later stages, not only is this detrimental to the car's performance but it also decreases its fuel efficiency. Moreover, there have been cases of accidents due to cars slipping due to unsuitable tires. These cases are due to overloading, unsuitable tire pressure, and tire slip due to excessive tire forces. Also, the selection of unsuitable tires can be seen in many student competitions like Formula Students and Baja. In these competitions, it is observed that most of the teams have no justification whatsoever for the selection of tires. These teams simply copy the tires used by the previous year’s champion or buy multiple sets of tires that they test from. This method is unsafe and costly as each set of tires costs \$1400 and can only be used for a total of fifty kilometers total. Moreover, the tire data obtained has many advantages. It helps provide crucial values like slip angle, the moment acting on tires, and the peak lateral force which helps in designing a car suspension and tuning the car. Moreover, the Engine/ motor drivetrain can be set in such a way as to provide peak forces obtained by the tire. Therefore, the codes and algorithms to select tires will take into account the maximum force that a tire can exert under various load-bearing conditions and at different pressure conditions for the optimum selection of tires.

III. LITERATURE REVIEW

Objective

To make a selection procedure for tires using TTC tire data, filter the testing data to sort it using MATLAB and use that data for training algorithms on Python for prediction of lateral force experienced by the tire.

Tire Testing Consortium

The FSAE Tire Test Consortium (FSAE TTC) was established to provide high-quality tire data to the participating FSAE teams and was used in the design and setup of race cars. Currently, tire testing work has been carried out five times at Calspan’s Tire Research Facility, and the data has been distributed to all the consortium members. The fee charged by the TTC is \$500 for a lifetime subscription.

Tire Testing Bed

This machine at the Calspan Tire Research Facility (TIRF) is tested by TTC. The tire, wheel, force sensing balance, and hydraulic motor to drive or break the tire are mounted in the movable upper head. The head provides steer, camber, and vertical motions to the tire. These motions (as well as vertical loading) are servo-controlled and programmed to maximize the test efficiency. The 28-inch-wide roadway comprises a stainless-steel belt covered with a material that simulates the frictional properties of actual road surfaces. The roadway is driven by one of the two 67-inch diameter drums over which the tire is run. The road speed is servo-controlled, and it may be programmed to be constant or varied.



Fig 1.1– Tire testing bed

Tire testing bed results

The unfiltered results provided by the tire testing bed are as follows. The frequency of test data collection is 50Hz, and the data channels available are elapsed time for the test (ET), road speed (V), slip angle (α), inclination angle (γ), slip ratio (r), normal load (FZ), lateral force (FY), longitudinal force (FX), aligning moment (MZ), overturning moment (MX), loaded radius (RL), effective radius (RE), wheel rotational speed (N), road surface temperature (RST),

0 5000 10000 15000 20000 25000 30000 35000
 -15 -10 -5 0 5 10 15 sample number slip angle ($^{\circ}$)

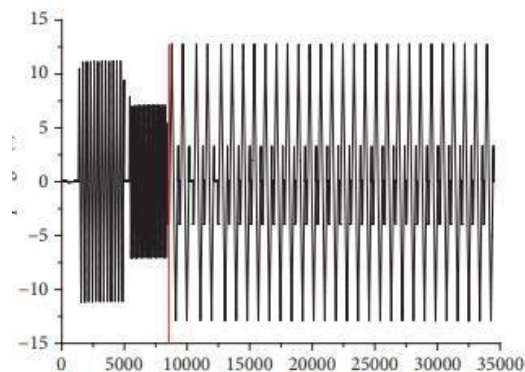


Fig 1.2 Unfiltered data provided by TTC

TTC now gives experimental data on tires which is crucial for the development of codes and algorithms to process and segregate tire data.

Issues faced due to improper tires

Due to improper tires and tire use, there are some issues faced, these issues decrease the tire life and increase the risk of accidents caused. The main cause of these incidents is the wrong tire data used, this can lead to many problems such as balding, graining, uneven tire tread wear, center wear, and flat spots.

Tire Failure due to uneven tread wear

Due to improper tire alignment, mainly insufficient or high camber, caster, and toe lead to uneven tire wear.



Fig 1.3 – Tire wear due to camber

In the above figure, we see that one side of the tire has a high degree of tire wear, this is due to the high camber set by the car’s suspension that tilted the tire at one end. The optimum camber range given by tire data can be used here to minimize tire degradation.

Tire wear due to balding and flat – spots

This type of tire wear takes place due to Abrasion. Abrasion is generally initiated by local stress concentrations at the contact between track asperities and rubber. Abrasion intensity depends on the shape rather than the size of the asperities. Experiments have shown that road surfaces exhibit wide variations in abrasion characteristics.

This abrasion in the car is caused due to excessive downforce, aerodynamics, due to more than the required grip produced by the car, and due to excessive torque produced by the drivetrain. The excessive vertical load when converted to high traction, often wears out the tires resulting in a bald spot. This can be avoided by using tire data to get the maximum amount of traction produced by the tire and designing the car accordingly.



Fig 1.4 – balding and flat spot

Tire wear due to graining

Graining is also caused due to abrasion of the rubber surface. Graining is usually an indicator of worse tire problems like balding, flat spots, and in the worst cases, complete removal of the tire thread.



Fig 1.5 –Graining



Fig 1.6 – Removal of tire thread

As seen in Figure 1.6 the tire thread has been completely removed and therefore the inner steel mesh can be seen. This is the worst consequence of graining.

Tire wear due to pressure

Due to excessive or insufficient pressure, the tire shows different properties and gives different traction forces and lateral forces to the same inputs. Due to overinflation or underinflation tire wear around the center or tire wear.



Fig 1.7 – tire wear due to pressure

As seen in the above figure, the tire on the left-hand side is an example of tire wear due to overinflation and the tire on the right-hand side is an example of tire wear due to underinflation. As we see the left tire has shown degradation in the middle due to overinflation and the tire on the right shows tire wear on the edges due to underinflation.

Optimum tire

This is what a tire that has used tire data and has selected the optimum tire looks like.



Fig 1.8 – Optimum tire

As seen in Figure 1.8, the above tire has even tire wear. There are no signs of graining, center wear, edge wear, bald spots, wear due to high traction, wear due to pressure, and wear due to misalignment. This is possible due to the optimum selection of tires and the use of tire data to design the optimum suspension setup. The tire has been selected by using and comparing it with multiple tires on the parameters of optimum camber, different loads, and different lateral forces.

Use of Tire Data in the Design of a Car

Every facet of the car's performance is reacted through the four contact patches. Fundamentally, brakes don't stop the car, the engine doesn't accelerate the car and the steering doesn't turn the car. All these tasks are achieved as a result of a reaction between the tire contact patch and the ground.

Tire Size

The first choice that must be made when choosing tires is what size to use and this dictates the wheel size both diameter and width. There are some advantages to choosing a 10-inch rim diameter, such as lower rotational inertia. 13-inch rim wheels usually offer a better choice of tires, so one advantage may be canceled by another. 10" wheels have less internal space to house brake and suspension components, and the suspension geometry desired may have some compromises at the chassis end. The tire size also determines many factors, such as the inertia produced by tires, as larger tires produce more inertia, they are harder to turn and control. The size also determines the acceleration as smaller tires have less radius for the same drivetrain torque, they produce more force and hence produce more acceleration. The top speed is also affected by tire size as a larger tire has a larger radius, thus a larger perimeter, and hence the distance covered by the tires is more.

Ackermann angle

The real importance of Ackermann is the difference in front wheel toe angles as the steering is turned. This can be affected by things other than the Ackermann angle such as the steering rack position. Excessive Ackermann angle will wear the tire by causing the inner tire to be dragged. Uneven Ackermann across the car (usually caused by failing to centralize the rack

before doing a wheel alignment or a bent steering arm) can cause problems that may be difficult to analyze. The Ackermann angle also dictates the toe condition of the car, the toe-in and toe-out conditions have different tire wear and produce different slips, by looking at the graph plotted by lateral force vs. the slip angle we can find out the wear produced by the toe angle of the car

Tire angles

The codes and algorithm on tire data gives us an optimum range of camber for a given tire. The suspension system can be designed for the camber angle range given by the tire data. This can significantly improve the tire performance range. The code does this by plotting the camber vs. longitudinal force for the given tire, since the graph is a polynomial function, it has a range for minima and maxima. This range gives us the optimum camber range for a given car. We can then design the car to run in this camber range by giving suspension some geometric constraints for heave and setting up the anti-roll bar for roll.

Engine/motor and drivetrain input

The codes and algorithm on tire data will give us the maximum traction force that the car. This can be used to calculate exactly how much torque the engine and motor will provide and what will be the on-wheel torque of the car. This will provide us with the maximum G-forces that the tire will be able to withstand. The code does this by plotting the maximum force for the vehicle vs. the static load acting on the tire and this condition gives us the maximum tractive force produced by the given tire. This force when divided by the radius of the tire gives the maximum tire torque which can be exerted by the vehicle without slipping and this can be used to design and select the optimum drivetrain and engine/motor.

Tractive, lateral, and vertical forces

The codes and algorithm on tire data give us the maximum lateral forces, traction forces, and peak longitudinal forces. These forces can help optimize the suspension design, the brakes design, and the aerodynamic design. The peak forces give us the maximum forces that a car could theoretically withstand, this gives us the G-force exerted on the car. These G-forces can then be used to optimize the suspension design and the vertical forces give us the maximum friction that a car could exert, hence optimizing the braking distance. The load acting in a vertical direction gives us the extent to which down forces may act to improve the car's performance, hence optimizing the design of the aero package of the car.

Graphs Used in Tire Data Sorting

To make an easy basis and bias for selection we used some graphs that helped us in the visual representation of different tire properties. Since tires are made of viscoelastic materials they behave differently at different temperatures, pressures and when different forces and moments are applied to them. The graphs are non-linear and hence there are quite some properties to look at in the graphs. Mainly the parameters are that the optimum force values should be reached quickly, and the values should remain in the maximum range in a large

number of situations. Given below are the list of graphs, their significance, and their roles in reading and understanding the tire data.

Fy vs S.A (lateral force vs Slip angle) graph

Lateral force varies with slip angle. The lateral force increases with increasing slip angle until, at some given slip angle, it reaches its maximum value. After this maximum value of lateral force has been reached any further increase in slip angle will result in a decrease in lateral force, and a corresponding decrease in lateral force the tire "breaks loose" or loses traction. This graph basically shows us, when the tire will slip, and this also gives us the value of maximum lateral force that a tire can generate and the optimum slip angle range.

nFy vs S.A (Normalized lateral force vs Slip angle) graph

The normalized lateral force is the ratio of lateral force acting on the tire divided by the vertical force acting on the tire. This can also be defined as one other thing called the coefficient of friction. Therefore, this graph gives us the frictional force acting on the car in a sideways direction, it also gives us one other thing that is the measure of the sensitivity of the vertical force acting on the tire (tire load sensitivity) which basically measures that, how will a car respond when a slight input in the lateral direction is given to the car. This gives us how much effort will be required to steer the car.

Mz vs. S.A (Self-aligning torque vs. slip angle)

Due to static angles like camber, caster, and Toe. The tire contact patch of the tire shifts from the center of the tire to some other place on the y-axis. Due to this shift, a moment is generated and this moment is called self-aligning torque. This torque has two main functions, firstly it provides steering feedback to the driver and secondly, it self-aligns the car in the straight direction. This graph gives us the measure of exactly how the change in static angles changes the steering capabilities of the car.

Fy vs. SA at different pressures

This graph gives us a change in forces due to pressure.

Fy vs. SA at different camber angles

This graph gives us a change in forces due to slip at different camber angles. This gives us the optimum camber range for a given vehicle.

IV. TIRE SELECTION

Using the Calspan Testing data, we were able to utilize the data for the tire and use the graphs and the other values to enable us to better design our car. Tire being the most important part of the car, the designing phase starts with the selection of tires. There are different rounds of testing, and each round has a set of tires that are tested.

The following constructions were tested, each on two different rim widths:

Round 3: -

- MICHELIN SIZE (16/53-13) RADIAL 'B'COMPOUND
Round 4: -
- MICHELIN (16/53-13) RADIAL 'X' S6B
Round 7: -
- Avon 7.0/16.0-10
Round 8: -
- Hoosier 43070, 16.0 x 6.0 - 10, LCO
- Hoosier 43070, 16.0 x 6.0 - 10, R25B
- Hoosier 43075, 16.0 x 7.5 - 10, LCO
- Hoosier 43075, 16.0 x 7.5 - 10, R25B

We have used R25B 10 x 6.0 Hoosier Tires. We have considered Round 8 Run 9 data for our calculations.

Run data was preferable because we had more vivid data to work out as compared to Raw data.

Each set of data has 3 sweeps wherein it has certain specifications.

The one we have used is Round 8 Run 9 for our calculations because it has uniform velocity.

Summary Table

Table 1: - Summary table round 8

Test Schedule												
Calspan												
FSAE												
Project Number: 1965, July 2018												
Run	Tire	Road Condition	P (psi)	V (mph)	Slip Ratio	SA (deg)	IA (deg)	Load (lb)	Run Comments	Run File	Tags	Wheel
1965-1	1965ire11	120, 3-Mile, Stoned	P1 r	V1	Free Rolling	S1	I1	L1	W1	1965 SA1 10in	DATE CODE IN	10x7
1965-2	1965ire11	120, 3-Mile, Stoned	P1 r	Z5	Free Rolling	S1	I1	L1	W2	1965 SA2 10in	DATE CODE IN	10x7
1965-3	1965ire11	120, 3-Mile, Stoned	P2 r	V3	Free Rolling	S2	I2	L2	W2,C1	1965 SA3 10in	DATE CODE IN	10x7
1965-4	1965ire11	120, 3-Mile, Stoned	P2 r	V3	Free Rolling	S2	I2	L2	W2,C2	1965 SA3 10in	DATE CODE IN	10x7
1965-5	1965ire12	120, 3-Mile, Stoned	P1 r	V1	Free Rolling	S1	I1	L1	W1	1965 SA1 10in	DATE CODE IN	10x8
1965-6	1965ire12	120, 3-Mile, Stoned	P1 r	Z5	Free Rolling	S1	I1	L1	W2	1965 SA2 10in	DATE CODE IN	10x8
1965-7	1965ire12	120, 3-Mile, Stoned	P2 r	V3	Free Rolling	S2	I2	L2	W2	1965 SA3 10in	DATE CODE IN	10x8
1965-8	1965ire20	120, 3-Mile, Stoned	P1 r	V1	Free Rolling	S1	I1	L1	W1	1965 SA1 10in	DATE CODE IN	10x6
1965-9	1965ire20	120, 3-Mile, Stoned	P1 r	Z5	Free Rolling	S1	I1	L1	W2	1965 SA2 10in	DATE CODE IN	10x6
1965-10	1965ire20	120, 3-Mile, Stoned	P2 r	V3	Free Rolling	S2	I2	L2	W2	1965 SA3 10in	DATE CODE IN	10x6

For the highlighted area we had the following conditions

Camber Schedule	I1: 0, 2, 4
Camber Schedule	I2: 0->-4->4->0
Normal Load Schedule	L1: 50, 100, 150, 200, 250
Normal Load Schedule	L2: 50, 100, 150, 200, 250, 350
Normal Load Schedule	L3: 50, 100, 150, 250, 350
Slip Angle Schedule	S1: -1, 1, 6
Slip Angle Schedule	S2: -4->12->-12->4
Velocity Schedule	V1: 0, 25, 2
Velocity Schedule	V3: 25, 15, 45
Warmup Schedule	W1: None
Warmup Schedule	W2: +/-12 SA
Run Comment	C1: Test aborted due to machine error
Run Comment	C2: Re-Run of test 3
Run Comment	C3: No Data presented
Run Comment	C4: Test aborted due to reaching machine limits
Run Comment	C5: Test aborted due to tire debanding
Run Comment	C6: No Video

We have used Avon 7.0/16.0-10. We have considered Round 7 Run 2 data for our calculations.

Run data was preferable because we had more vivid data to work out as compared to Raw data.

Each set of data has 3 sweeps wherein it has certain specifications.

The one we have used is Round 7 Run 2 for our calculations because it has uniform velocity.

Table 2: - summary table round 7

Test Schedule											
Calspan											
Cooper											
Project Number: 1706, March 2016											
Run	Tire	Road Condition	P* (psi)	V (mph)	Slip Ratio	SA (deg)	IA (deg)	Load (lb)	Run Comments	Run File	Tags
1706-1	1706ire0	120, 3-Mile, Stone (1x)	P1 r	V1	B1	S1	0	L3	W2	1654 SA3 10in	DATE CODE IN
1706-2	1706ire0	120, 3-Mile, Stone (1x)	P1 r	V2	B1	S2	12	L4	W3	1654 SA2 10in	DATE CODE IN
1706-3	1706ire0	120, 3-Mile, Stone (1x)	P2 r	V3	B1	S2	12	L4	W4	1654 SA3 10in	DATE CODE IN
1706-4	1706ire0	120, 3-Mile, Stone (1x)	P1 r	V1	B1	S1	0	L3	W2	1654 SA3 10in	DATE CODE IN
1706-5	1706ire0	120, 3-Mile, Stone (1x)	P1 r	V2	B1	S2	12	L4	W3	1654 SA2 10in	DATE CODE IN
1706-6	1706ire0	120, 3-Mile, Stone (1x)	P2 r	V3	B1	S2	12	L4	W4	1654 SA3 10in	DATE CODE IN

For the highlighted area we had the following conditions

Slip Ratio Schedule	B1:	Free Rolling
Camber Schedule	I2:	0, 2, 4
Normal Load Schedule	L1:	150, 250, 350
Normal Load Schedule	L2:	350, 150, 50, 250, 100
Normal Load Schedule	L3:	50, 150, 250
Normal Load Schedule	L4:	200, 150, 50, 250, 100
Slip Angle Schedule	S1:	-1, 1, 6
Slip Angle Schedule	S2:	-4->+12->-12->+4
Velocity Schedule	V1:	0, 25, 2
Velocity Schedule	V2:	25
Velocity Schedule	V3:	25, 15, 45
Velocity Schedule	V4:	25, 45
Warmup Schedule	W2:	Spring Rates at 2x Vel
Warmup Schedule	W3:	+/- 12 deg SA, +/- 8 deg SA
Warmup Schedule	W4:	None

We have used MICHELIN (16/53-13) RADIAL 'X' S6B. We have considered Round 4 Run 24,25,26 data for our calculations.

We used raw data as it was the only data available for this particular tire configuration. To combine all pressure conditions(10psi,12psi,14psi) we had to consider data from runs 24,25,26

Table 3: - Summary Table round 4

Cornering Test Schedule														
Calspan														
FSAE														
Project Number: 1320, October 2009														
Run	Tire	Tire Code	Manufacturer	Rim Width (inches)	Tire Size	P* (psi)	V (mph)	Slip Ratio	SA (deg)	IA (deg)	Load (lb)	Run Comments	Command	Tags
1320-20	1320in20	Goodyear	8	20.0x7.0-13	10.0 r	25	free-roll	S1	11	L1	W2	1320 FY2	IN	
1320-21	1320in20	Goodyear	8	20.0x7.0-13	14.0 r	25	free-roll	S1	11	L1	W2	1320 FY2	IN	
1320-22	1320in20	Goodyear	8	20.0x7.0-13	8.0 r	25	free-roll	S1	11	L1	W2,C1,C5	1320 FY2	IN	
1320-23	1320in20	Goodyear	8	20.0x7.0-13	12.0 r	25	free-roll	S1	11	L1	W2	1320 FY2	IN	
1320-24	1320in32	Michelin	7	16x3-13	12.0 r	25	free-roll	S1	11	L1	W1	1320 FY1,A	IN	
1320-25	1320in32	Michelin	7	16x3-13	10.0 r	25	free-roll	S1	11	L1	W2	1320 FY2	IN	
1320-26	1320in32	Michelin	7	16x3-13	14.0 r	25	free-roll	S1	11	L1	W2	1320 FY2	IN	

For the highlighted area we had the following conditions

Camber Schedule	I1:	0, 2, 4, 1, 3
Camber Schedule	I3:	Various : FSAE Procedure
Normal Load Schedule	L1:	350, 150, 50, 250, 100
Normal Load Schedule	L4:	Various : FSAE Procedure
Slip Angle Schedule	S1:	-4 -> +12 -> -12 -> +3 @ 4 deg/sec
Slip Angle Schedule	S4:	Various : FSAE Procedure
Warmup Schedule	W1:	3 spring rate, cold to hot, warmup, 2 conditioning
Warmup Schedule	W2:	2 conditioning sweeps
Warmup Schedule	W3:	None
Run Comment	C1:	Tire debeaded
Run Comment	C10:	Run out of sequence (Before the 12psi test)
Run Comment	C11:	Run out of sequence (after the 10psi test)
Run Comment	C2:	Incorrect tire Psi
Run Comment	C3:	No Data Presented
Run Comment	C4:	Check Test
Run Comment	C5:	Only Raw Data presented
Run Comment	C6:	TIRF Error, tested as Run 61 on new tire
Run Comment	C12:	Run at 12psi instead of 10psi, Re-tested as Run 36

We have used MICHELIN SIZE (16/53-13) RADIAL 'B' COMPOUND. We have considered Round 3 Run 18,19,20 data for our calculations.

We used raw data as it was the only data available for this particular tire configuration. To combine all pressure conditions(10psi,12psi,14psi) we had to consider data from runs 18,19,20

Table 4: - Summary Table round 3

Test Schedule											
Calspan											
FSAE											
Project Number: 1175, July 2007											
Run	Tire	Belt Code	Surface	P* (psi)	V (mph)	Slip Ratio	SA (deg)	IA (deg)	Load (lb)	Run Comments	Run File
1175-16	1175ire0	120, 3-Mile	P2 r	25	Free Roll	S2	0	L1	W1	W1.C3	
1175-17	1175ire10	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C2	
1175-18	1175ire19	120, 3-Mile	P2 r	25	Free Roll	S2	0	L1	W1	W1.C2	
1175-19	1175ire19	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C3	
1175-20	1175ire19	120, 3-Mile	P2 r	25	Free Roll	S2	0	L1	W1	W1.C3	
1175-21	1175ire7	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C2	
1175-22	1175ire7	120, 3-Mile	P2 r	25	Free Roll	S2	0	L1	W1	W1.C2	
1175-53	1175ire1	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C2, C7	
1175-54	1175ire2	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C3, C7	
1175-55	1175ire3	120, 3-Mile	12 r	25	Free Roll	S2	11	L1	W1	W1.C4, C7	

For the highlighted area we had the following conditions

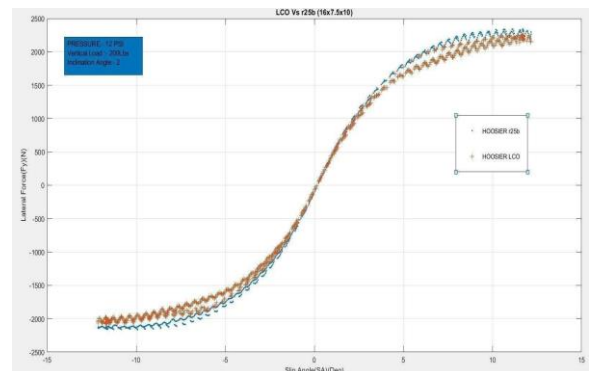
Graph Comparisons

We can find graphs using the data values from above:-

1) Comparison between Hoosier LCO Vs Hoosier R25B (16x7.5x10): -

a) Fy (Lateral Force) Vs SA (Slip Angle): -

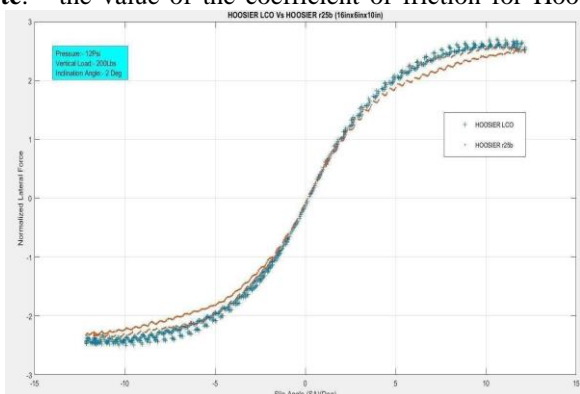
Note: - From the above graph we see that in the lateral force of Hoosier R25B, the peak is achieved much earlier than



Hoosier LCO in a steeper way

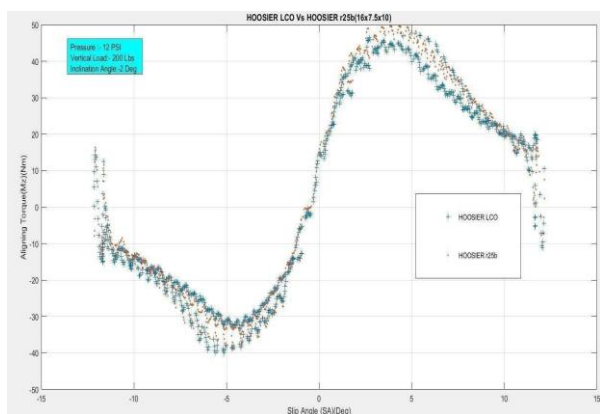
Pressure Schedule	P1	16, 10, 14, 8, 12
Pressure Schedule	P2	16, 10, 14, 12
Camber Schedule	I1:	0, 2, 4, 1, 3
Normal Load Schedule	L1:	350, 150, 50, 250, 100
Slip Angle Schedule	S2:	-4 -> +12 -> -12 -> +3 (4 deg/sec)
Warmup Schedule	W1:	+/- 2 SA, IA 2 min, 2 cond sweeps @ 250 lbs
Run Comment	C1:	Test abort, tire debeaded
Run Comment	C2:	8" Rim
Run Comment	C3:	7" Rim
Run Comment	C4:	6" Rim
Run Comment	C5:	Run with letters out
Run Comment	C7:	Run with Letters IN, Used Tire

b) nFy (Normalised Lateral Force) Vs SA (Slip Angle): -
Note: - the value of the coefficient of friction for Hoosier



LCO is a bit greater than the R25B

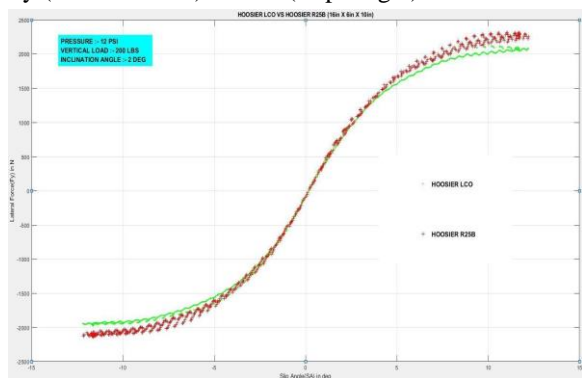
c) Mz (Aligning Torque) Vs SA (Slip Angle): -



Note: - The peak value of Hoosier R25B for the aligning torque is greater than that of Hoosier LCO, the peak is achieved near 3°, Hence overall sense Hoosier R25B is qualified to be compared to the next set.

2) Comparison between Hoosier LCO vs. Hoosier R25B (16x6x10): -

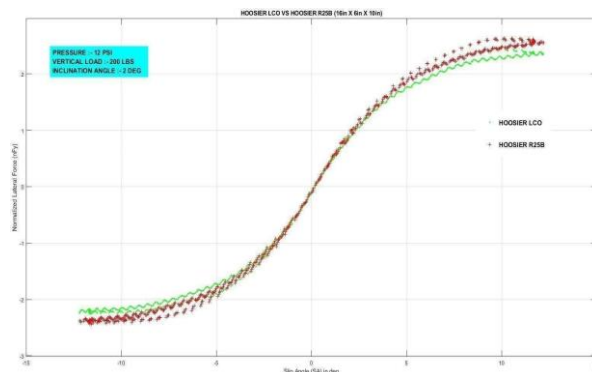
a) Fy (Lateral Force) Vs SA (Slip Angle): -



Note: -From the above graph we see that in the lateral force of Hoosier R25B, the peak is achieved much earlier than

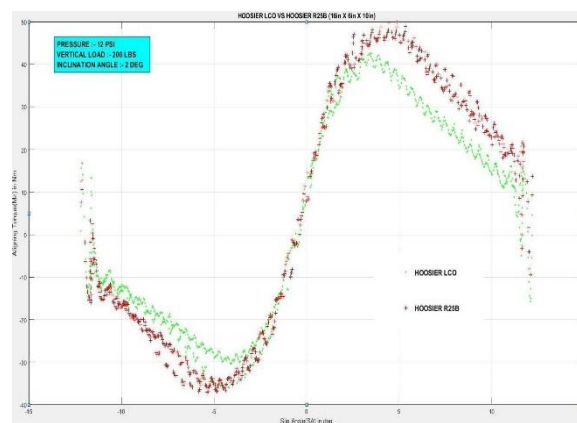
Hoosier LCO in a steeper way.

b) nFy (Normalised Lateral Force) Vs SA (Slip Angle): -



Note: - the value of the coefficient of friction for Hoosier R25B is a bit greater than the LCO.

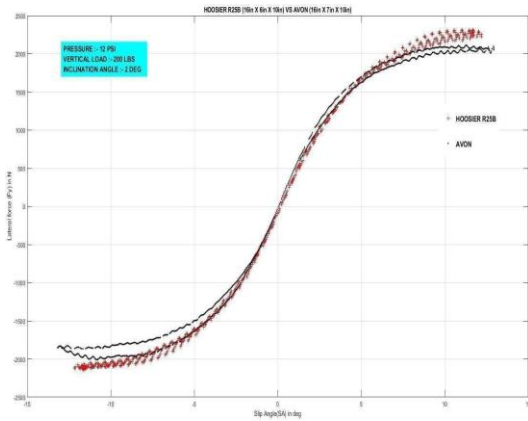
c) Comparison between Mz (Aligning Torque) Vs SA (Slip Angle): -



Note: - The peak value of Hoosier R25B for the aligning torque is greater than that of Hoosier LCO also it's delayed by a degree, the peak is achieved near 4.5°, overall sense Hoosier R25B is qualified to be compared to the next set.

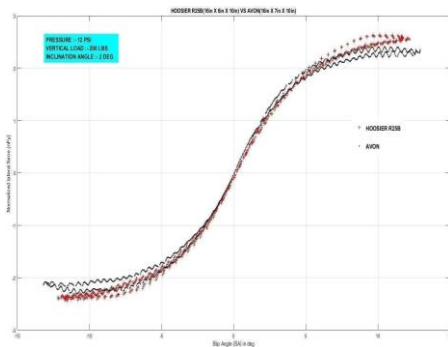
3) Comparison between Avon 7.0/16.0-10 Vs Hoosier R25B (16x6x10): -

a) Fy (Lateral Force) Vs SA (Slip Angle): -



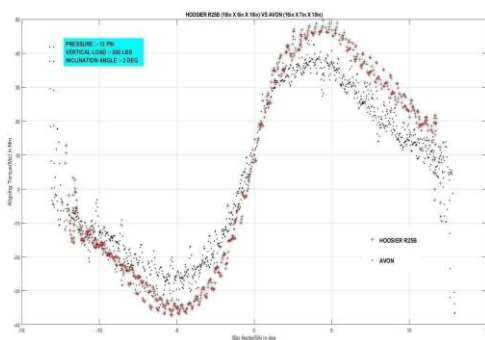
Note: -From the above comparisons we observe that the peak value for the lateral force of Hoosier R25B (16x6x10) is much greater than Avon 7.0/16.0-10

b) nFy (Normalised Lateral Force) Vs SA (Slip Angle): -



Note: - coefficient of friction of Hoosier R25B (16x6x10) is much greater than Avon 7.0/16.0-10.

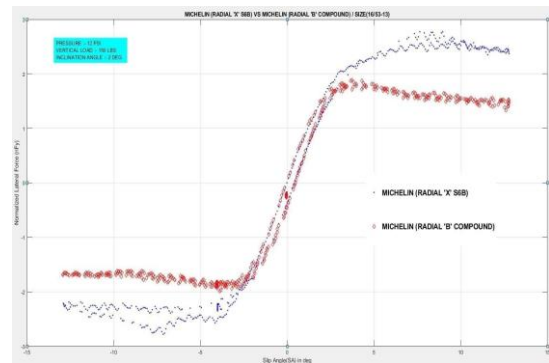
c) Mz (Aligning Torque) Vs SA (Slip Angle): -



Note: - the value of aligning torque is also greater; their peaks are approximately achieved at similar slip angles. Hence Hoosier R25B qualifies further into the comparison.

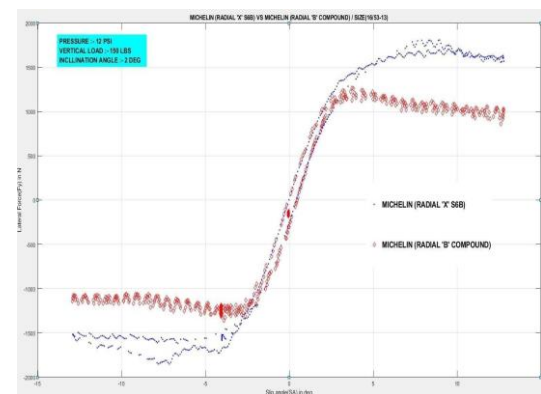
4) Comparison MICHELIN (RADIAL 'X' S6B) VS MICHELIN (RADIAL 'B' COMPOUND) / SIZE (16/53-13):

a) Fy (Lateral Force) Vs SA



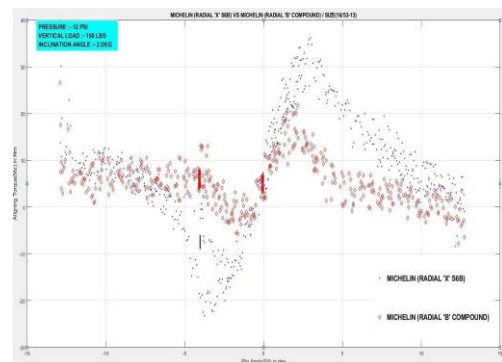
Note: - From the above graph we can see that the peak value of "lateral force" for MICHELIN (RADIAL 'X' S6B) is greater than that of MICHELIN (RADIAL 'B' COMPOUND) / SIZE (16/53-13).

b) Comparison between nFy (Normalised Lateral Force) Vs SA (Slip Angle): -



Note: - From the above graph we can see that the peak value of "coefficient of friction" for MICHELIN (RADIAL 'X' S6B) is greater than that of MICHELIN (RADIAL 'B' COMPOUND) / SIZE (16/53-13)

c) Comparison between Mz (Aligning Torque) Vs SA (Slip Angle): -

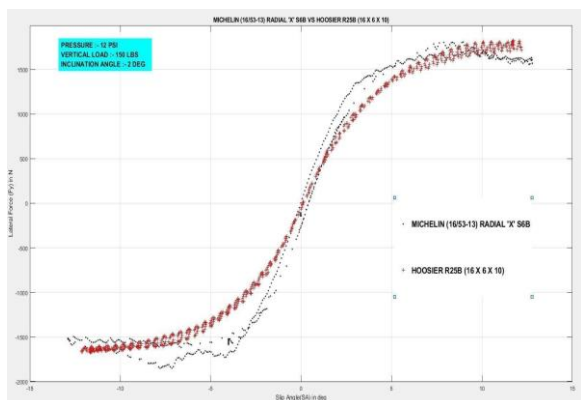


Note: - The peak value for aligning torque is also higher as well as leading to of MICHELIN (RADIAL 'B'

COMPOUND) by a 1°. Hence MICHELIN (RADIAL 'X' S6B) is qualified to move into the final comparison.

5) MICHELIN (16/53-13) RADIAL 'X' S6B VS HOOSIER R25B (16 X 6 X 10): -

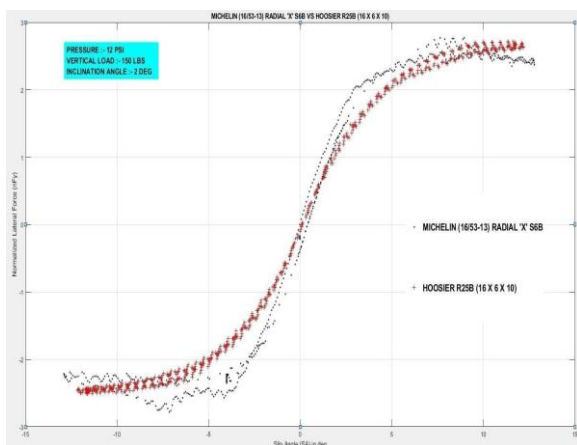
a) Fy (Lateral Force) Vs SA



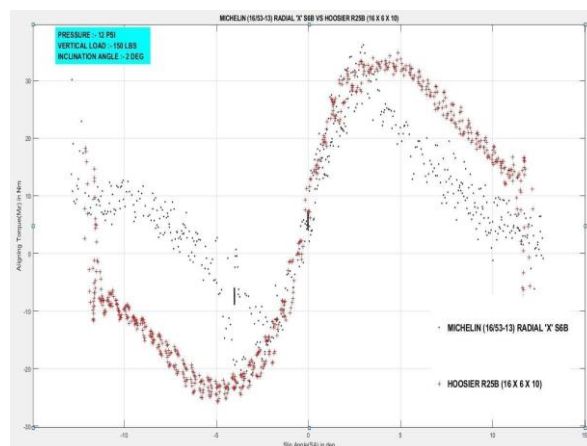
Note: - The peak value for lateral force is achieved at quicker rate for MICHELIN (16/53-13) RADIAL 'X' S6B as compared to HOOSIER R25B (16 X 6 X 10) but at higher slip angle the lateral force is steady in case of HOOSIER R25B whereas in case of MICHELIN, it starts to fall.

b) Comparison between NFy (Normalised Lateral Force) Vs SA (Slip Angle): -

Note: - Similar trend is observed with the lateral coefficient of friction



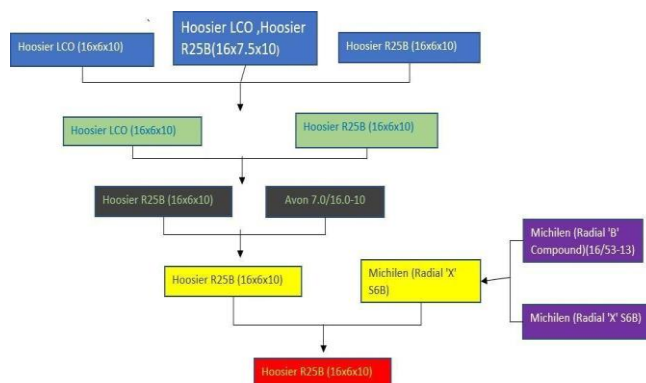
c) Comparison between Mz (Aligning Torque) Vs SA (Slip Angle): -



Note: - There is a considerable difference in the aligning torques HOOSIER R25B has a higher peak value, and it leads by a degree w.r.t MICHELIN.

Hence, from all the above comparisons we can see that HOOSIER R25B (16 X 6 X 10) is clearly the best choice.

Comparison by elimination



Once our Tire is selected, we need to find optimum working conditions by internal comparison, i.e.

We have used R25B 10 x 6.0 Hoosier Tires. We have considered Round 8 Run 9 data for our calculations.

Run data was preferable because we had more vivid data to work out as compared to Raw data.

Each set of data has 3 sweeps wherein it has certain specifications. The one we have used is Round 8 Run 9 for our calculations because it has uniform velocity.

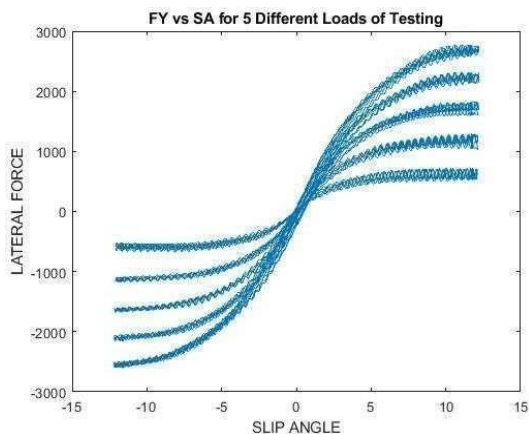
Table 5: -summary table round 9

Calspan												
Test Schedule												
FSAE												
Project Number: 1965, July 2018												
Run	Tire	Road Condition	SP (psi)	V (mph)	Slip Ratio	SA (deg)	Load (lb)	Run Comments	Run File	Tags	Wheel	
1965-1	1965tire11	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W1	1965 SA1 10in	DATE CODE IN	10x7
1965-2	1965tire11	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W2	1965 SA2 10in	DATE CODE IN	10x7
1965-3	1965tire11	120, 3Mts, Stoned	P2.1	V3	Free Rolling	S2	12	L2	W2,C1	1965 SA3 10in	DATE CODE IN	10x7
1965-4	1965tire12	120, 3Mts, Stoned	P2.1	V3	Free Rolling	S2	12	L2	W2,C2	1965 SA3 10in	DATE CODE IN	10x7
1965-5	1965tire12	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W1	1965 SA1 10in	DATE CODE IN	10x8
1965-6	1965tire12	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W2	1965 SA2 10in	DATE CODE IN	10x8
1965-7	1965tire12	120, 3Mts, Stoned	P2.1	V3	Free Rolling	S2	12	L2	W2	1965 SA3 10in	DATE CODE IN	10x8
1965-8	1965tire20	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W1	1965 SA1 10in	DATE CODE IN	10x8
1965-9	1965tire20	120, 3Mts, Stoned	P1.1	V1	Free Rolling	S1	11	L1	W2	1965 SA2 10in	DATE CODE IN	10x8
1965-10	1965tire20	120, 3Mts, Stoned	P2.1	V3	Free Rolling	S2	12	L2	W2	1965 SA3 10in	DATE CODE IN	10x8

For the highlighted area we had the following conditions

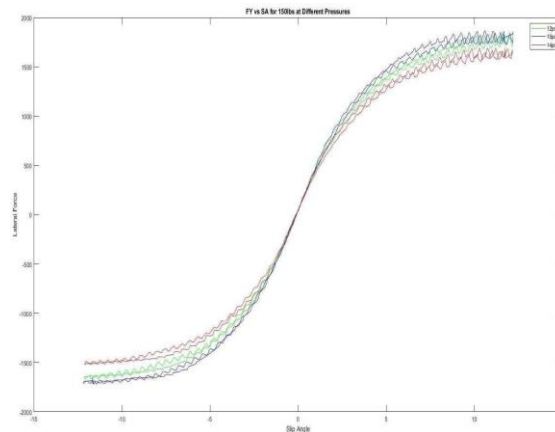
Camber Schedule	I1: 0, 2, 4
Camber Schedule	I2: 0->-4->4->0
Normal Load Schedule	L1: 50, 100, 150, 200, 250
Normal Load Schedule	L2: 50, 100, 150, 200, 250, 350
Normal Load Schedule	L3: 50, 100, 150, 250, 350
Slip Angle Schedule	S1: -1, 1, 6
Slip Angle Schedule	S2: -4->12->-12->4
Velocity Schedule	V1: 0, 25, 2
Velocity Schedule	V3: 25, 15, 45
Warmup Schedule	W1: None
Warmup Schedule	W2: +/-12 SA
Run Comment	C1: Test aborted due to machine error
Run Comment	C2: Re-Run of test 3
Run Comment	C3: No Data presented
Run Comment	C4: Test aborted due to reaching machine limits
Run Comment	C5: Test aborted due to tire debeading
Run Comment	C6: No Video

We can find graphs using these data values. Finding the FY vs SA graphs for 5 different loads: 250 lbs 200lbs 150 lbs 100 lbs 50lbs



Thus, we see from the graph that as we increase the Fz load, we get more Lateral Force values

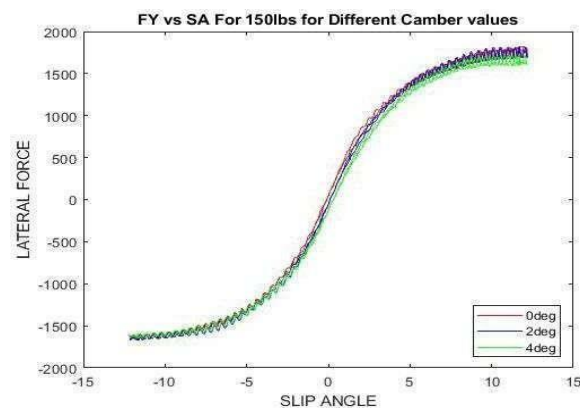
We have also found out other graphs wherein we can see the lateral force for the same load but with different Tire Pressure.



Thus, we see that Lateral Force is more when it comes to lesser Tire pressure because of more surface area.

A major aim was to analyze the difference in lateral force (Fy) when the camber values were changed.

So, for a particular Fz load (150 lbs) we plotted the graph for different Camber angles Camber schedule: 0 deg, 2 deg, 4 deg.



V. LATERAL FORCE PREDICTION

Dataset

The dataset that we are currently using is the data from the Tire Testing Consortium which tests the tires at varying tire pressure, tire camber, normal load, and slip angles to get the lateral force produced by the tire as the output.

Aim

Our aim is to create a prediction model that outputs the lateral force of the tire by giving the values above as inputs.

The prediction models that we used are:

- 1) Neural Networks
- 2) Random Forest Regressor

Models like Linear Regression and Logistic Regression were considered but eliminated as they were not fit for predicting on our dataset.

1) Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms.

Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

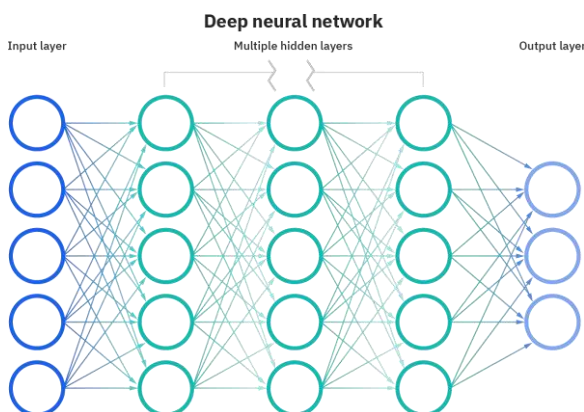
Artificial neural networks (ANNs) are composed of a node layer, containing an input layer, one or more hidden layers, and an output layer.

Each node, or artificial neuron, connects to another and has an associated weight and threshold.

If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

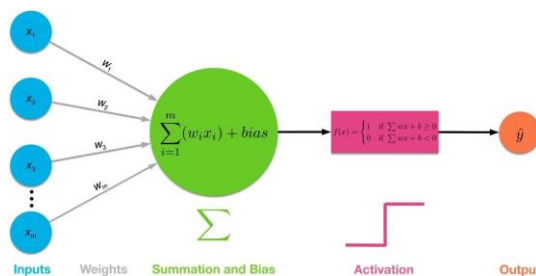
Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity.

Tasks in speech recognition or image recognition can take minutes versus hours when compared to manual identification by human experts. One of the most well-known neural networks is Google’s search algorithm.



Neuron

A single neuron is basically a logistic regression. Neuron is a mathematical function that computes the weighted average of its inputs and then this sum is passed through a nonlinear function (often called an activation function) together with the bias. A few activation functions to mention are sigmoid, ReLU, Tanh, etc. Neurons can be arranged in multiple layers to form a neural network. Each neuron serves as an input to the neuron of the next layer.



In a Neural network, some inputs are provided to an artificial neuron, and with each input a weight is associated. Weight increases the steepness of the activation function. This means weight decides how fast the activation function will trigger whereas bias is used to delay the triggering of the activation function. The values of weights are biased and are chosen in a way that the predicted target value is as close to the actual target value.

Generally, this is achieved after several iterations.

Linear functions are those whose graph is a straight line. A linear function has the following form

$$y = f(x) = a + bx.$$

A linear function has one independent variable and one dependent variable. The independent variable is x, and the dependent variable is y.

Often a linear function cannot explain the relationship between variables. In such cases, a nonlinear function must be used. These could be power, quadratic, polynomial, rational, exponential, logarithmic, and sinusoidal. Non-linear means the graph is not a straight line. The graph of a nonlinear function is a curved line.

Neural Networks Working

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias \quad (Eq.1)$$

$$output = f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0 & \text{if } \sum w_1 x_1 + b \leq 0 \end{cases} \quad (Eq.2)$$

Once an input layer is determined, weights are assigned. These weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs. All inputs are then multiplied by their respective weights and then summed. Afterward, the output is passed through an activation function, which determines the output. If that output exceeds a given threshold, it “fires” (or activates) the node, passing data to the next layer in the network. This results in the output of one node becoming the input of the next node.

This process of passing data from one layer to the next layer defines this neural network as a feedforward network.

To give an example, should we go surfing? (Yes: 1, No: 0). The decision to go or not to go is our predicted outcome or \hat{y} . Let's assume that there are three factors influencing our decision-making:

- Are the waves good? (Yes: 1, No: 0)
- Is the line-up empty? (Yes: 1, No: 0)
- Has there been a recent shark attack? (Yes: 0, No: 1)

Then, let's assume the following, giving us the following inputs:

- $X_1 = 1$, the waves are good.
- $X_2 = 0$, there is a long line.
- $X_3 = 1$, there hasn't been a recent shark attack.

Now, we need to assign some weights to determine the importance of each input. Larger weights signify that the particular variable is of greater importance to the decision or outcome.

- $W_1 = 5$, since large swells don't come around often
- $W_2 = 2$, since you're used to the crowds
- $W_3 = 4$, since you have a fear of sharks

Finally, we'll also assume a threshold value of 3, which would translate to a bias value of -3. With all the various inputs, we can start to plug in values into the formula to get the desired output.

$$\hat{Y} = (1*5) + (0*2) + (1*4) - 3 = 6$$

If we use the activation function from the beginning of this section, we can determine that the output of this node would be 1 since 6 is greater than 0. In this instance, you would go surfing; but if we adjust the weights or the threshold, we can achieve different outcomes from the model.

When we observe one decision, like in the above example, we can see how a neural network could make increasingly complex decisions depending on the output of previous decisions or layers.

In the example above, we used perceptron to illustrate some of the mathematics at play here, but neural networks leverage sigmoid neurons, which are distinguished by having values between 0 and 1. Since neural networks behave similarly to decision trees, cascading data from one node to another, having x values between 0 and 1 will reduce the impact of any given change of a single variable on the output of any given node, and subsequently, the output of the neural network.

As we train the model, we'll want to evaluate its accuracy using a cost (or loss) function. This is also commonly referred to as the mean squared error (MSE). In the equation below,

- i represents the index of the sample,
- \hat{y} is the predicted outcome,
- y is the actual value, and
- m is the number of samples.

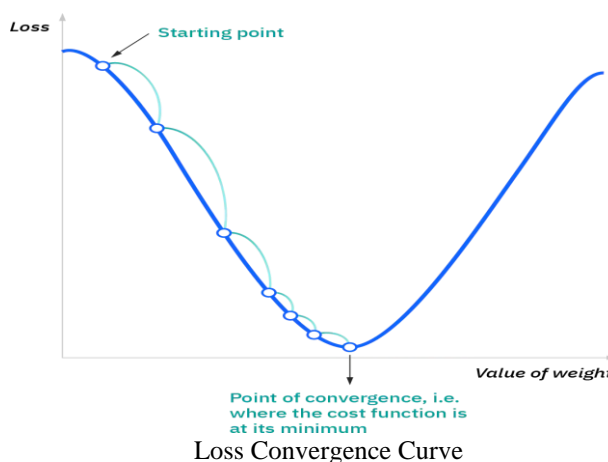
$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y} - y)^2 \quad (\text{Eq.3})$$

Ultimately, the goal is to minimize our cost function to ensure the correctness of fit for any given observation.

As the model adjusts its weights and biases, it uses the cost function and reinforcement learning to reach the point of convergence or the local minimum.

The process in which the algorithm adjusts its weights is through gradient descent, allowing the model to determine the direction to take to reduce errors (or minimize the cost function).

With each training example, the parameters of the model adjust to gradually converge at the minimum.



Our Neural Network

Our dataset contains 60,000 samples with the features (predictors) being:

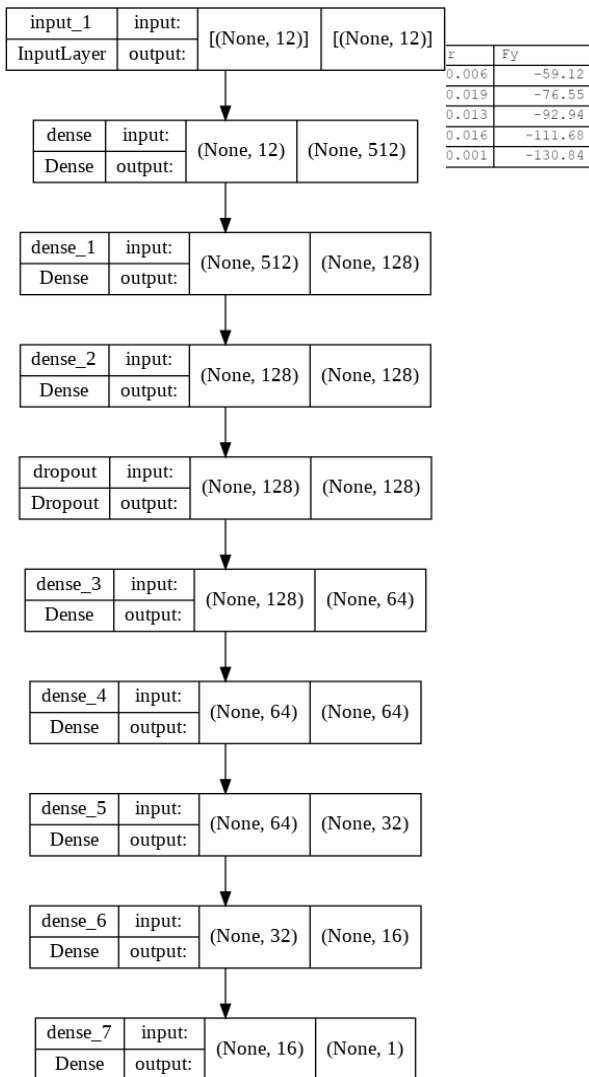
```
df2.columns>>>Index(['SA ', 'Fz', 'P', 'Camber', 'Fy'], dtype='object')
```

- SA – Slip angle (Predictor)
- Fz – Normal Load in lbs. (Predictor)
- P – Tire pressure in kpa. (Predictor)
- Camber – Tire Camber (Predictor)
- Fy – Lateral Force (Response)

We converted the tire pressure and normal load to psi and lbs respectively.

Our data frame table looked like this:
 (Table – 7) – Raw Dataset

```
df2.head()>>>
After applying data transformation techniques like one-hot encoding and standardizing the data, we split the predictors
```



Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 512)	6656
dense_9 (Dense)	(None, 128)	65664
dense_10 (Dense)	(None, 128)	16512
dropout_1 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 64)	8256
dense_12 (Dense)	(None, 64)	4160
dense_13 (Dense)	(None, 32)	2080
dense_14 (Dense)	(None, 16)	528
dense_15 (Dense)	(None, 1)	17

Total params: 103,873
 Trainable params: 103,873
 Non-trainable params: 0

deep-learning model.

Network Architecture:

NN = Sequential(name='NN')

```

NN.add(Input(shape =X_train.shape[1]))
NN.add(Dense(512,
activation=tf.keras.layers.LeakyReLU(alpha=0.3)))
NN.add(Dense(128, activation='tanh'))
NN.add(Dense(128, activation='tanh'))
NN.add(Dropout(0.3))
NN.add(Dense(64, activation='tanh'))
NN.add(Dense(64, activation='tanh'))
NN.add(Dense(32, activation='tanh'))
NN.add(Dense(16, activation='tanh'))
NN.add(Dense(1, activation='tanh'))
  
```

```

rlrop = ReduceLRonPlateau(monitor='val_loss', factor=0.8,
patience=20)
NN.compile(loss =
tf.keras.losses.MeanSquaredError(), metrics = ['mse']
, optimizer = tf.keras.optimizers.Adam(learning_rate =
0.001))
NN.summary()
>>>
Model: "NN"
  
```

and response variables.
 The predictors' dataset:
 (Table – 8) – Transformed Dataset

index	SA	Fz_-150	Fz_-100	Fz_-50	P_10.0	P_12.0	P_14.0	Camber_	Camber_	Camber_4
40228	10.164	0	0	0	0	1	1	0	0	0
51513	11.634	0	0	0	0	0	1	0	1	0
50579	0.644	0	0	1	0	0	1	0	1	0
4931	-2.663	0	0	1	0	1	0	1	0	0
53680	-1.495	0	1	0	0	0	1	0	1	0

We then split the dataset to train and test sets.
 Train – The data on which the model will be trained. (75% of the dataset)
 Test – The data we will use for testing the predictions of our model. (25% of the dataset)

We used the Keras & Tensorflow library to create our Neural Network model.
 Keras was chosen since it allows the same code to run seamlessly on the CPU or GPU. It has a user-friendly API that makes it easy to quickly prototype deep-learning models. It has built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
 It supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, and so on. This means Keras is appropriate for building essentially any

Network Architecture

Notes on our model architecture:

As can be seen above, our model consists of 100,000 total trainable parameters. (weights & biases). We have used the 'tanh' activation function for most of the layers for its symmetrical output about the y-axis which would help in predicting the lateral force in either direction of the tire. We have used the Mean Squared Error as the loss function for its robustness and flexibility in regression problems. We have used the Adam optimizer to speed up the learning of the model by incorporating the benefits of momentum and stochastic gradient descent (SGD). We are also reducing the learning rate by a factor of '0.8' whenever the value of the loss function has been on a plateau for multiple training epochs. We have also used 'Dropout' to reduce the tendency

of overfitting in our model.

```

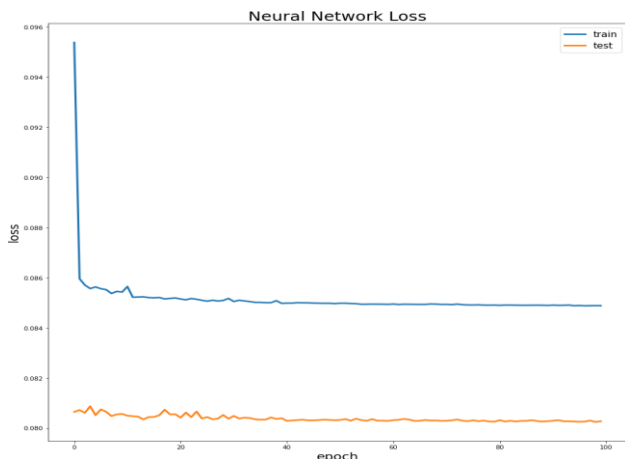
history = NN.fit(X_train, y_std, batch_size = 32, epochs = 100, verbose
= 2, validation_split = 0.3, callbacks = [rlrop], use_multiprocessing
= True, workers = 8)

>>>
Epoch 1/100
919/919 - 8s - loss: 0.0954 - mse: 0.0954 - val_loss: 0.0806 - val_mse:
0.0806 - lr: 0.0010 - 8s/epoch - 9ms/step
Epoch 2/100
919/919 - 4s - loss: 0.0860 - mse: 0.0860 - val_loss: 0.0807 - val_mse:
0.0807 - lr: 0.0010 - 4s/epoch - 4ms/step
Epoch 3/100
919/919 - 4s - loss: 0.0857 - mse: 0.0857 - val_loss: 0.0806 - val_mse:
0.0806 - lr: 0.0010 - 4s/epoch - 4ms/step
.
.
.
Epoch 97/100
919/919 - 3s - loss: 0.0849 - mse: 0.0849 - val_loss: 0.0803 - val_mse:
0.0803 - lr: 4.0960e-04 - 3s/epoch - 4ms/step
Epoch 98/100
919/919 - 3s - loss: 0.0849 - mse: 0.0849 - val_loss: 0.0803 - val_mse:
0.0803 - lr: 4.0960e-04 - 3s/epoch - 4ms/step
Epoch 99/100
919/919 - 4s - loss: 0.0849 - mse: 0.0849 - val_loss: 0.0802 - val_mse:
0.0802 - lr: 4.0960e-04 - 4s/epoch - 4ms/step
Epoch 100/100
919/919 - 4s - loss: 0.0849 - mse: 0.0849 - val_loss: 0.0803 - val_mse:
0.0803 - lr: 4.0960e-04 - 4s/epoch - 4ms/step
    
```

Model Training

Notes on model training:

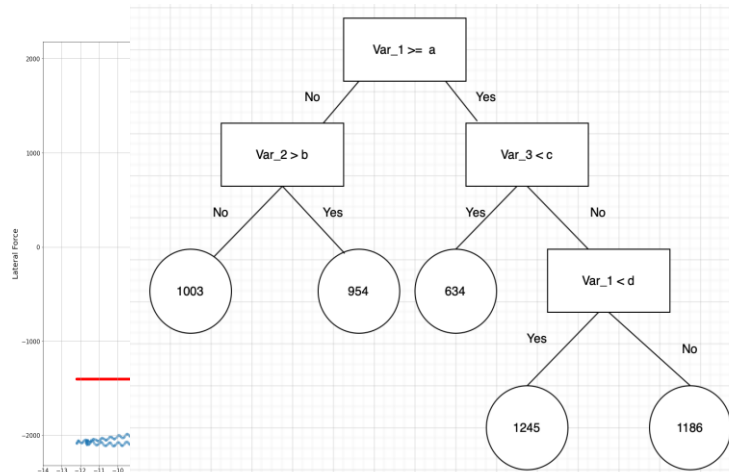
We have split the training set into training and validation datasets to check for model overfitting. We have set the batch size to 32 and trained the model for 100 epochs. As can be seen in the model training, over the 100 epochs of training, the learning rate has reduced from 0.001 to 0.0004. The validation set loss and training set loss have decreased during the training phase which shows that the model has reached convergence



Loss for each epoch

The graph above shows the network loss for each epoch. From the plot, we can see that the loss reduced drastically in the first few epochs and then remained reduced slightly for the rest of the training phase. This shows that the neural network has converted to a minimum.

Network Predictions:



Lateral Force vs Slip Angle

Notes on Network Predictions

From the graph above, we can see that the network produces high accuracy predictions when the slip angle is in the range of [-4°, +4°]. This indicates that the model has stopped learning for higher absolute values of slip angles.

This can be attributed to the nature of our ‘tanh’ activation function which flattens out as the absolute value of the input increases. Hence, as the value of the slip angle increases, the network is unable to learn the behavior of the features and outputs a horizontal line due to the presence of the ReLU function.

2) Random Forest Regressor

A Random forest is an ensemble of decision trees. This is to say that many trees, constructed in a certain “random” way form a Random Forest.

Decision Trees

Decision Trees are used for both regression and classification problems. They visually flow like trees, hence the name, and in the regression case, they start with the root of the tree and follow splits based on variable outcomes until a leaf node is reached and the result is given. An example of a decision tree is below

Decision Tree Diagram

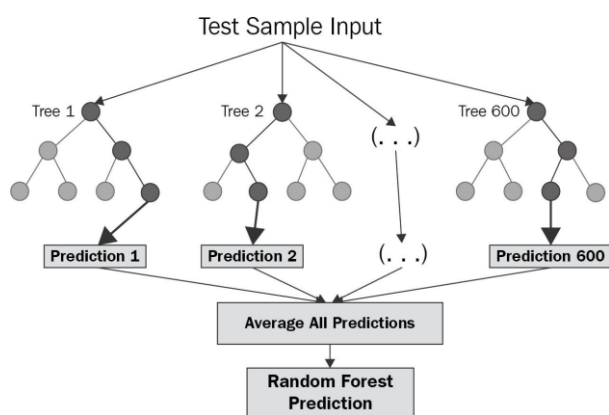
Here we see a basic decision tree diagram that starts with the Var_1 and splits based on specific criteria. When ‘yes’, the decision tree follows the represented path, when ‘no’, the decision tree goes down the other path. This process repeats until the decision tree reaches the leaf node and the resulting outcome is decided. For the example above, the values of a, b, c, or d could be representative of any numeric or categorical value.

While decision trees are common supervised learning algorithms, they can be prone to problems, such as bias and overfitting. However, when multiple decision trees form an ensemble in the random forest algorithm, they predict more accurate results, particularly when the individual trees are uncorrelated with each other.

Ensemble learning is the process of using multiple models, trained over the same data, averaging the results of each model ultimately finding a more powerful predictive/classification result. Our hope, and the requirement, for ensemble learning, is that the errors of each model (in this case decision tree) are independent and different from tree to tree.

Bootstrapping is the process of randomly sampling subsets of a dataset over a given number of iterations and a given number of variables. These results are then averaged together to obtain a more powerful result. Bootstrapping is an example of an applied ensemble model.

The bootstrapping Random Forest algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a new result that often leads to strong predictions.



Random Forest Schematic Diagram

Key Benefits of Random Forest Regressor

Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data. However, when there's a robust number of decision trees in a random forest, the classifier won't overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.

Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists. Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing. Easy to determine feature importance: Random Forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance.

Key Drawbacks of Random Forest Regressor

Since random forest algorithms can handle large data sets, they can provide more accurate predictions but can be slow to process data as they are computing data for each individual decision tree. Since random forests process large data sets, they'll require more resources to store that data. The prediction of a single decision tree is easier to interpret when compared to a forest of them.

We used the Sci-Kit Learn Library to create our regressor.

Sci-Kit Learn has the following benefits:

Scikit Learn is an open-source library that is available for public use. This is openly available for use or even in this the user can modify or redirect the whole code. Scikit Learn library is free to use. People don't take any license to run this library. The user would not have to worry when they work on the application. In Sci-Kit, the documentation of the task is done in a proper way.

The Sci-Kit learn library gives us multiple parameters to customize our random forest regression model. Some of these are

- `n_estimators` — the number of decision trees you will be running in the model
- `criterion` — this variable allows you to select the criterion (loss function) used to determine model outcomes. We can select from loss functions such as mean squared error (MSE) and mean absolute error (MAE). The default value is MSE.
- `max_depth` — this sets the maximum possible depth of each tree
- `max_features` — the maximum number of features the model will consider when determining a split
- `bootstrap` — the default value for this is True, meaning the model follows bootstrapping principles (defined earlier)
- `max_samples` — This parameter assumes bootstrapping is set to True, if not, this parameter doesn't apply. In the case of True, this value sets the largest size of each sample for each tree.

Our Random Forest Regressor:

For our regression model, the dataset and data transformation techniques are the same as those used for the Neural Network. To get the best values for all the parameters mentioned above, we have done a manual grid search by running a for-loop inside Python and training the regression forest on multiple combinations of the input parameters.

The parameters that we changed for each combination are:

- `n_estimators`
- `max_depth`
- `max_features`

```
# Fit the model for all possible combinations of param_list
n_estimators = [10, 20, 30, 40, 50, 100, 150, 200, 250]
max_features = (range(2, 10))
max_depth=(range(2,10))
param_list = []
for i in n_estimators: for
    j in max_features:
        for k in max_depth: params
            = [i, j, k]
            param_list.append(params)
param_array = np.array(param_list)
oob_dict = {}
for i in param_array:
    forrest = RandomForestRegressor(i[0], max_features = i[1], max_depth
    = i[2], random_state = 1, oob_score = True)
    forrest.fit(X_train, y_train)
oob_dict[str(i)] = forrest.oob_score_
```

In the code snippet above, we have trained a random forest

model for each combination of the parameters mentioned above and stored the combination and its out-of-bag score (oob score) in a Python dictionary.

To obtain the model with the highest oob score, we ran the following code:

```
best_params = {key for key in oob_dict.keys() if oob_dict[key] == max(oob_dict.values())}[0]
print('The best parameters for the random forest regressor are {}'.format(best_params))
>>> The best parameters for the random forest regressor are [250 9]

max(oob_dict.values())
>>> 0.9990323935467239
```

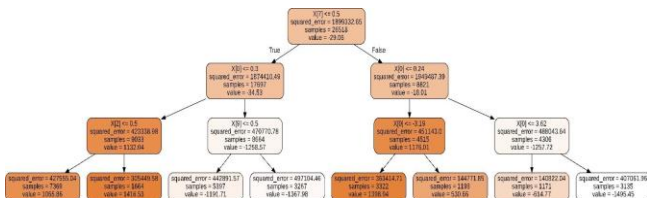
As can be seen by the code snippet, the best combination is at the higher end of the range for each parameter. Hence, we trained another forest regression model for parameters outside the range we had searched in. This is done to check if the OOB score increases further. We created a new Random Forest model with the following parameters:

```
#Create a Random Forest model instance
forest2 = RandomForestRegressor(500, max_features = 10, max_depth = 10
, random_state = 1, oob_score = True) #Fit
the model to the training data
forest2.fit(X_train, y_train)
#Get the oob score of the new model
oob_new = forest2.oob_score_ #Print
the score
oob_new
>>> 0.9992432979980064
```

This shows a marginal increase in the OOB score from the previous value. (0.99924 > 0.99903)

Since there is only a marginal increase, we did not check for higher values due to the time consumed and resources required.

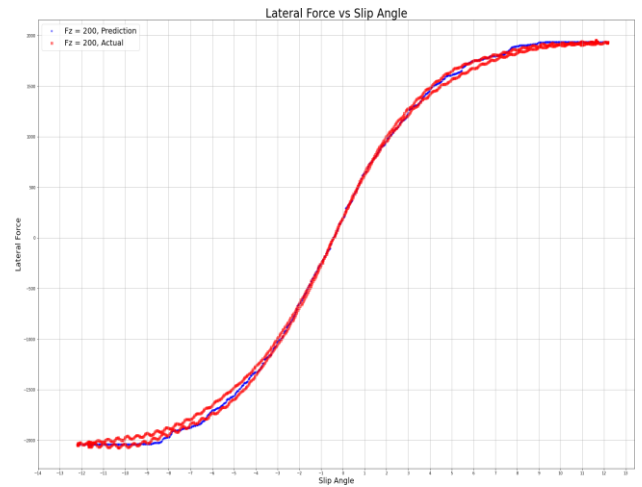
Part of a tree from our Random Forest Model:



Our Random Forest Regressor (One Tree)

Regression Model Predictions:

Below are the predictions of the Random Forest Model for a subset of the data.



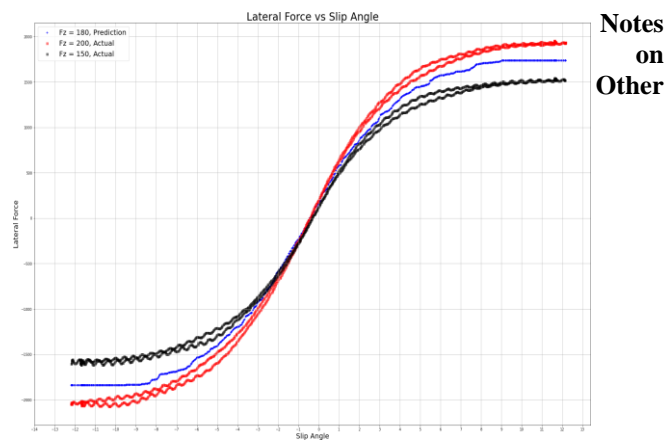
Lateral Force vs. Slip Angle (Random Forest Predictions)

Notes on Regression predictions:

As can be seen above, the model predicts accurately for higher absolute values of slip angles, unlike the neural network. The model has replicated the actual data and hence has an accuracy of 99.9%

Other predictions:

Lateral Force vs. Slip Angle (Random Forest Predictions)



Notes on Other

predictions:

Since predictions were accurate for the previous model, we tried model predictions for unseen data with a normal load '180'. As can be seen above, the predictions for unseen data are satisfactory and lie in the correct range.

VI. RESULTS AND DISCUSSION

In this paper we first studied the effects of tires on the car, from which we inferred what an optimum tire for a given car should look like and how it should behave. Next, we made a sorting and data filtering code and then used that code to infer useful data about the tire performance and its range to select the optimum tire. Lastly, we made two algorithms via two different approaches of neural networks using a tan function

and random forest regressor, and from our project, we inferred the following things.

The usage of optimum tires for a car is very important, improper tire management can lead to the following things, namely balding, graining, uneven tire tread wear, center wear, and flat spots. We also learned to identify an optimum tire where there is no graining and uniform tire wear.

We then moved on to the sorting and comparison of tire data in which we first plotted some graphs. From our filtering and sorting codes, we came to the conclusion that Hoosier R25b 16x6x10 was the appropriate tire for us, the reason being, that it has a high peak lateral force of 2313N, it also has high tire load sensitivity with a coefficient of friction 2.6286, it has biased ply in construction, and it has 10-inch rims which provide enough space for packaging of wheel assembly.

We first made our model for the given lateral force as an output for an input of camber, pressure, and slip angle. We made the model of our tire based on the neural network of Tanh function and we found that the model was accurate 99.6% of the time in the range of slip angles (-4,4) but became linear after that range. This could be attributed to the nature of the tanh function. From our previous model, we concluded that we might have to use a new approach and so we made a new model on the random forest regression function. This function turned out to be 99% accurate.

We then used the random forest regression to predict the lateral force at different load conditions in the range of 50 lbs difference and the results came out to be satisfactory.

VII. FUTURE PROSPECTS

The tire data obtained has many advantages. It helps provide crucial values like slip angle, the moment acting on tires, and the peak lateral force which helps in designing a car suspension and tuning the car. Moreover, the Engine/ motor drivetrain can be set in such a way as to provide peak forces obtained by the tire. Therefore, the codes and algorithms to select tires will consider the maximum force that a tire can exert under various load-bearing conditions and at different pressure conditions for the optimum selection of tires.

Currently, the model only takes into account the tire temperatures ranging from 52 degrees Celsius to 72 degrees Celsius, as TTC only has data available in the said range but in the future, the temperature range can be increased with more testing. The sorting codes can also consider new tires which will be tested and added later by the TTC, thus improving the matrix of comparison.

From the tire model created the values of lateral force and slip angle could be used to make a simulator, As more and more models are made, more data could be used to feed accurate forces into the simulator of the car hence reducing the overall testing time required for a car.

VIII. CONCLUSION

This paper successfully created a method to segregate the tires and select the optimum tire. The sorting and the filtering code was made on MATLAB and the code successfully plotted graphs of FY vs SA, NFY vs SA, NZ vs SA, and lateral load at different pressures and camber. From these graphs, the optimum tire was selected. An algorithm was made to predict the lateral force of the tire with the inputs of slip angle, camber, and pressure. A model was also made to find lateral force at varying loads and the result was satisfactory.

REFERENCES

- [1] Milliken & Milliken, "Race Car vehicle dynamics", SAE International, 1995, ISBN I- 56091-526-9.
- [2] Thomas D. Gillespie, "Fundamentals of Vehicle Dynamics", ASIN: B07Z8H1LT3.
- [3] John Dixon, "Suspension Geometry and Computation", ISBN-13: 978-0470510216.
- [4] Biswal, Sadjyot & Prasanth, Aravind & Udayakumar, R. & Deva, Shobhit & Gupta, Aman. (2017). Design of a suspension system and determining suspension parameters of a medium downforce small Formula-type car.
- [5] Learning Centre - Mould Construction Guide. (n.d.). Fibre Glast.
- [6] Pratik Bhagwat and others, "Design of 10" carbon wheel mold", internal ME 5358-001, the University of Texas, Arlington, 2016.
- [7] Hans Walther, Development of a lightweight laminated composite wheel for FormulaSAE race vehicle, ProQuest LLC, 2016.
- [8] Evan Drew Flickinger, "Design and Analysis of Formula SAE Car SuspensionMembers". Master thesis, Carolina State University, May 2014.
- [9] Hindawi Journal of Engineering Volume 2018, Article ID 4143794
- [10] Tire testing consortium,
- [11] Calspan automotive research center
- [12] Vehicle testing and tuning by Claude Rouelle
- [13] Deep Learning with Python - by François Chollet
- [14] Research Paper on the Basic of Artificial Neural Network
- [15] Practical Deep Learning for Coders
- [16] Random Forests - Mathematics and Statistics
- [17] Are Random Forests Truly the Best Classifiers? -
- [18] Deep learning: adaptive computation and machine learning