International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

Chief Editor
Dr. J.B. Helonde

Executive Editor
Mr. Somil Mayur Shah

RESEARCHERID
THOMSON REUTERS

[Patel *et al.*, 10(2): February, 2021] ISSN: 2277-9655
IC™ Value: 3.00 Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

MINIMUM LAP TIME CALCULATION USING MATLAB

Dhruv V. Patel*[1] & Sidhesh M. Lokhande[2]
*[1&2]Mechanical Department, C.S.P.I.T- Charusat University, India

DOI: https://doi.org/10.29121/ijesrt.v10.i2.2021.8

ABSTRACT

Formula Student car, to participate in different Formula Student events. Though the car was designed and all designed systems and components were validated using Static Simulation, but it was not known how would car perform on the actual track. By using different dynamic simulation methods and MATLAB programming, team has tried to understand the vehicle behaviour on track. One of the methods used by the team was, to calculate the limiting velocity for each curve on the track and to calculate the time required by the car to complete one Endurance lap of Formula Bharat 2020. This was achieved by making a mathematical model and using concepts of vehicle Dynamics. In the study different cases of the track are also discussed and how does the car behave at different portion of the track is clearly explained in the study. This helps the driver to handle the car properly. As with this information driver would be able to know how his car would behave at different corners of the track.

KEYWORDS: Formula Student Vehicle, Limiting Velocity, Total Lap time, MATLAB, Mathematical Modelling, Vehicle dynamics.

1. INTRODUCTION

Formula Student: The Challenge
Each year the team designs, builds, tests, and eventually races their car against other university teams from all over the world in the Formula Student competition.
The students are to assume that a producing firm has engaged them to supply a prototype car for evaluation. The intended sales market is the nonprofessional weekend auto crosser sprint race and the firm is planning to produce 1,000 cars per year at a cost below 10 lakhs.The car must be low in cost, easy to take care of, and reliable, with high performance in terms of its acceleration, braking, and handling qualities. Watched closely by industry specialists who volunteer their time each team will go through the following rigorous testing process of their car:
Static events: Design, Cost, and Presentation Judging− Technical and Safety Scrutineering − Tilt Test to prevent cars from rolling over − Brake and Noise Test.
Dynamic Events: Skid Pad − Acceleration − Sprint/qualification − Endurance and Fuel Economy – Autocross.

Problem Definition
A typical formula student vehicle can be designed, but to learn its behaviour in actual conditions becomes a difficult task. The
problem was, how the Limiting Velocity for the car for any specific curve could be calculated and what would be the theoretical time taken by car to complete one lap of the track.
All such problems were easily solved by constructing a mathematical model in MATLAB.

2. LIMITING VELOCITY AND MINIMUM LAP TIME

Limiting Velocity.
Limiting velocity is the maximum velocity at which car can drive through the corner.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Assume that car is travelling through the corner and on that car, forces acting are centrifugal force in outward direction, frictional force in inward direction, aerodynamic drag force in downward direction, self-weight of car in downward direction, and normal force in upward direction.

figure

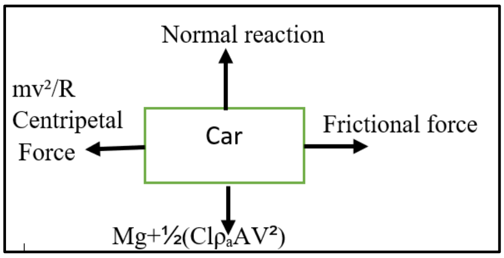

Fig. - 1. (Forces on car)

limiting velocity is the maximum velocity at which car can drive through the corner.
Using equation for limiting velocity,
v=sqrt((μ*m*g)/(m/r-μ*Cl*ρ_a*A/2))
v=limiting velocity
r=Radius of track
μ=coefficient of friction
m=mass of car=320kg
Cl=downforce coefficient
g=gravitational acceleration=9.81m/s^2
A=Aerodynamic drag force area
ρ_a=density of air=1.2

Gear ratio: The **ratio** of the angular speed of the initial or driving member of a **gear** train or equivalent mechanism to that of the final or driven member specifically : the number of engine revolutions per revolution of the rear wheels of an automobile.
over gear ratio (Gn)=Ne/Nw
Ne=Engine speed in rpm
Nw =wheel speed in rpm
By this different gear ratios are obtained
Gear ratios of 1st gear=2.78 for velocity ranges from 0 to 23.88 m/s
2nd=2.05 for range of 23.88 to 32.5 m/s
3rd=1.68 for 32.5 to 39.72 m/s
4th=1.45 for 39.72 to 46.11 m/s
5th=1.3 for 46.11 to 51.39 m/s
6th=1.18 from 51.39 m/s

MATLAB code:
```
function [gr]= gearsel(v)
if v>0 && v<=23.89
gr=2.78;
elseif v>23.89 && v<=32.5
gr=2.05;
elseif v>32.5 && v<=39.722
gr=1.68;
elseif v>39.72 && v<=46.11
gr=1.45;
elseif v>46.11 && v<=51.388
```

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
gr=1.3;
else v>51.388
gr=1.18;
end
```

Calculating rpm of wheel from speed in m/s
u=D/2*w
w=2*π*Nw/60
u=π*D*Nw/60

u=velocity of car in m/s
D=Diameter of wheel in m

Now calculating torque from max power output
P=2*π*N*T/60
P=max power output=69kW
T=Engine torque output
N=Max rpm of engine=12000rpm
Therefore, T=59.4Nm

Using 2 equations of torque at wheel (Tw)
Tw=Gn*T
Tw=Ftrac*D/2
Therefore, Fw=T*Gn*2/D
(Ftrac=Traction force at wheel)

And using other basic equation of kinematics
v=sqrt(u^2+2*a*s)
time=distance/velocity
where u=initial velocity
v=final velocity
a=linear acceleration
s=displacement
Mechanics of Tire rolling and Friction

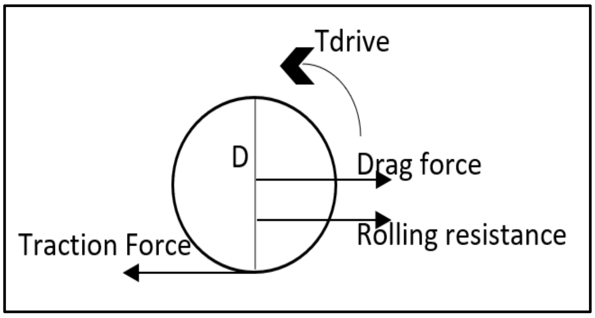

Fig. - 2. Mechanics of Tire rolling and Friction)

Assuming no slip condition,
Forces acting on tire are
Ftrac=T*Gn*2/D
MATLAB code:
function [ftr]=Ftrac(v)

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

ftr=(gearsel(v)*54.9/0.2286); torque=54.9, radius of wheel=0.2286m
end

Fdrag=Cd*ρ*A*u^2/2
MATLAB code:
```
function [fdr]=Fdrag(v)
Da=1.2;
Af=1;
Cd=0.8926;
fdr=(Cd*Da*Af*v^2)/2;
end
```
Froll=urr*((m*g)+(cl*ρ*A*u^2)/2)
MATLAB code:
```
function [froll]=Froll(v)
froll=0.02*((320*9.81)+(1.2*1.2*v^2)/2);%Cl=1.2
end
```

Ftrac=Traction force in forward direction
Fdrag=Drag force in backward direction

Froll=Rolling resistance in backward direction
urr=rolling coefficient of resistance
Cd=drag coefficient

Therefore, Net force=m*a
(Fw-Fdrag-Froll)=m*a
a=(Fw-Fdrag-Froll)/m
MATLAB code:
```
function [an]=anew(v)
acalc=(Ftrac(v)-Fdrag(v)-Froll(v))/320;
if acalc>0
an=acalc;
else
an=0;
end
```
assumed no slip condition so,
Net moment=0
Fw*D/2-T=0

Calculating the minimum lap time
There are 6 conditions in possible in which different can be constructed, all the graphs shown below are Velocity – distance graphs.

Straight-curve
Car entering the straight track with certain velocity then accelerate up to specific length of track and then decelerate to attain limiting velocity at end of track and at the entrance of curve and maintain limiting velocity throughout the curve.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

figure

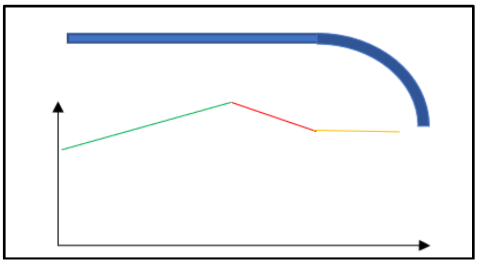

Fig. – 3. (Straight - Curve)

MATLAB code:

```
function [out]=straights(le,r,vi)
v1=vi;
v2=sqrt(3139.2/(302/r-0.72));%m=320kg, µ=1
for x=1:0.25:le
time=0;
v(1)=v1;
for i=1:x
v(i+1)=sqrt(v(i)^2+2*anew(v(i)));
if v(i)~=0
time=time+1/v(i);
end
end
v3=v(end);
v4=sqrt(v2^2+2*24.52*(lex));%deceleration=2.5g
time=time+(v3-v2)/24.52;
if abs((v3-v4)/v4)*100<2
a=[v1,v2,x,v3,v4,time,anew(v(i))];
disp(a)
break;
end
end
end
```

The above code shows that car enters with velocity vi then it accelerated till velocity v3 which is final for acceleration section and calculate time for this section, and then calculated the initial velocity v4 considering v2 as final velocity for deceleration section and then add previous calculated time to this decelerated section time. And at last if v3 and v4 found to satisfy the condition because top velocity at end of acceleration and initial velocity at start of deceleration should match, we get total time.

Curve-straight

Throughout the curve maintain the limiting velocity and at the straight track, accelerate freely.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

figure

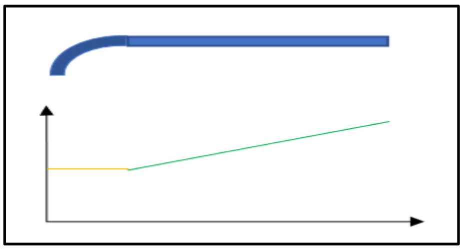

Fig. – 4. (Curve – Straight)

MATLAB code:

```
Function[curstr]=curve_str(le,r)%length of curve
v=sqrt(3139.2/(302/r-0.72));%limiting velocity of curve
time=le/v
end
```

The above code shows time is being simply calculated by dividing velocity to length.

Curve-Curve with increased radius

At the 1st curve maintain the limiting velocity and at 2nd curve having larger limiting velocity compare to 1st because of larger radius than 1st
Curve. So, on 2nd curve having increased radius car can accelerate up to limiting velocity for specific length and maintain the limiting velocity for remaining section of curve.

figure

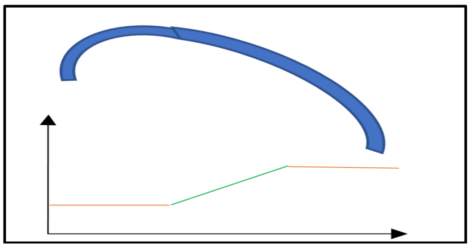

Fig. – 5. (Curve-Curve with increased radius)

MATLAB code:

```
function [curinc]=curvesinc(le,rp,rc)
v2=sqrt(3139.2/(302/rc-0.72));
v1=sqrt(3139.2/(302/rp-0.72));
for x=1:0.25:le
time=0;
v(1)=v1;
for i=1:x
V(i+1)=sqrt(v(i)^2+2*anew(v(i)));
v(i+1)=v(i);
if v(i)~=0
time=time+1/v(i);
end
end
v3=v(end);
if abs((v2-v3)/v2)*100<2
time=time+(le-x)/v3;
```

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
a=[v1,v2,v3,x,time];
disp(a)
else
b=[v1,v2,v3,x,time];
disp(b)
end
end
end
```

The above code shows car enters with limiting velocity of previous curve then accelerate till v3 and time is being calculated, if v3 matches with limiting velocity of current curve i.e. V2 then total time is calculated by adding accelerated previous time.

Curve-curve with decreased radius

For the 1st curve car can maintain limiting velocity up to certain length of curve section then decelerate to attain limiting velocity of curve which is having decreased radius.

figure

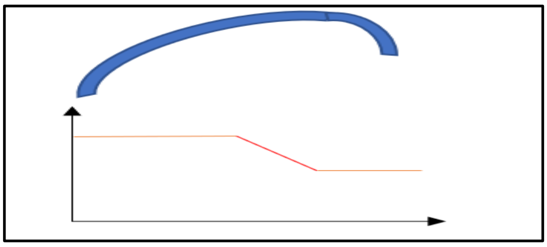

Fig. – 6. (Curve-curve with decreased radius)

MATLAB code:

```
function [para]=curves(le,ri,rf)
v2=sqrt(3139.2/(302/rf-0.72));
v1=sqrt(3139.2/(302/ri-0.72));
for x=1:0.25:le
time=0;
v(1)=v1;
for i=1:x
v(i+1)=v(i);
if v(i)~=0
time=time+1/v(i);
end
end
v3=v(end);
v4=sqrt(v2^2+2*24.52*(le-x));
time=time+(v3-v4)/24.52;
if abs((v3-v4)/v4)*100<2
b=[v1,v2,v3,v4,x,time];
disp(b)
break;
end
end
end
```

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

The above code shows car enters with current curve limiting velocity and continue to travel with same velocity for some part of section, then as limiting velocity of upcoming curve is less so car has to decelerate considering v2 as final velocity and finding v4. And if v3 and v4 matches with in limit we get total time.

Curve-increased radius curve-decreased radius curve

Maintain limiting velocity for 1st curve then as 2nd curve radius is larger its limiting velocity is also larger so on 2nd curve car can accelerate up to its limiting velocity for certain section of curve and at last it should decelerate for remaining section to attain limiting velocity of upcoming curve having decreased radius.

MATLAB code:

```
function [time]=curves(le,rp,rc,rn)
v2=sqrt(3139.2/(302/rc-0.72));%-3.45588
v1=sqrt(3139.2/(302/rp-0.72));%-3.45588
v3=sqrt(3139.2/(302/rn-0.72));
for x=1:0.25:le
time=0;
v(1)=v1;
for i=1:x
V(i+1)=sqrt(v(i)^2+2*anew(v(i)));
if v(i)~=0
time=time+1/v(i);
end
end
v4=v(end);
if abs((v2-v4)/v2)*100<2
for p=1:0.25:(le-x)
v(1)=v4;
for q=1:p
v(q+1)=v(q);
if v(i)~=0
time=time+1/v(q);
end
end
v5=v(end);
v6=sqrt(v2^2+2*24.52*(le-x-p));
time=time+(v5-v6)/24.52;
if abs((v5-v6)/v5)*100<2
w=[v1,v2,v3,x,time];
disp(w)
break;
end
end
end
end
end
```

The above code shows that car enter with previous curve limiting velocity v1 and accelerate till current curve limiting velocity v2 and if it satisfies condition it travel with same velocity till v5 then it starts to decelerate considering v3 velocity as final velocity and finding v6 i.e. starting velocity of deceleration section and if satisfy the condition it shows the total time

Critical radius

Limiting velocity

$v=sqrt((\mu*m*g)/(m/r-\mu*Cl*\rho*A/2))$
here if, $(m/r-\mu*cl*\rho*A/2) =0$
$m/r = \mu*cl*\rho*A/2$
then v becomes infinite that interprets that the car can accelerate freely because it has more than sufficient grip and curve having radius greater than critical radius can be considered as straight track.
$Rcritical=m/(\mu*cl*\rho*A/2)$

Calculating Actual track length
By tracing the track of Formula Bharat Endurance Track in 2D CAD sketch and adding each sketched section length, obtained track length is 2647.7 mm but the actual total length of track is 1000 m or 1km. so there should be multiplication ratio to multiply with sketched section to get actual length.
Multiplication ratio= (total sketch length)/ (actual total length)
= 2647.7/1000
=2.6477
So now multiplying each sketched section with 2.6477, its actual length can be obtained.

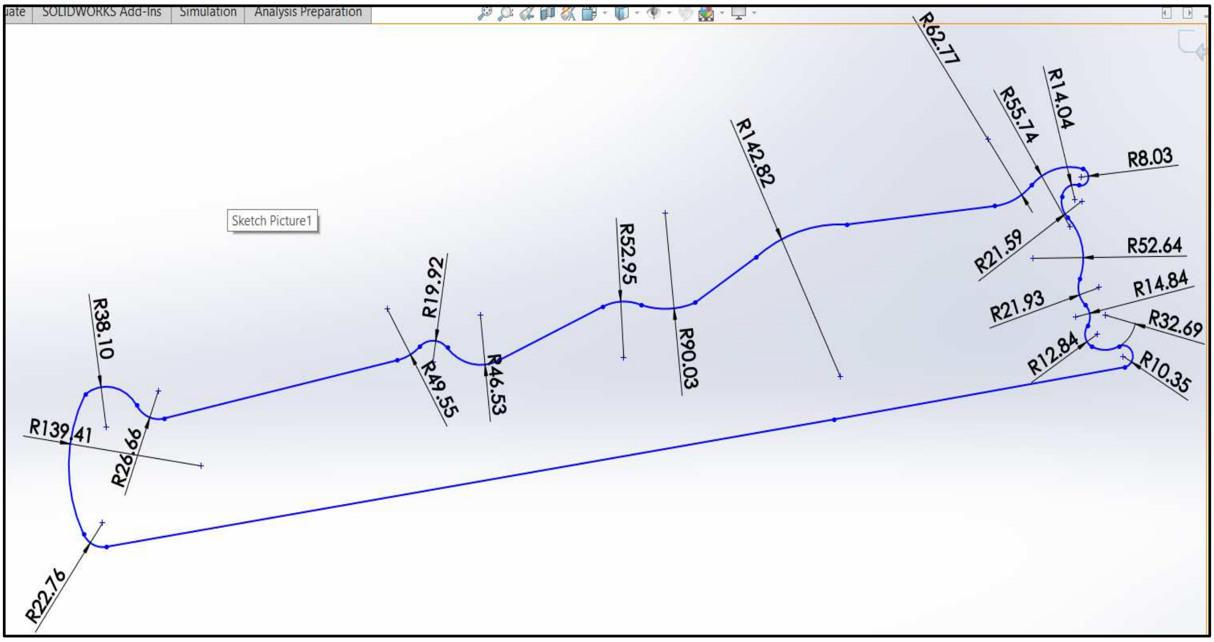

Fig. – 7. (Formula Bharat Endurance track Without slaloms)

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

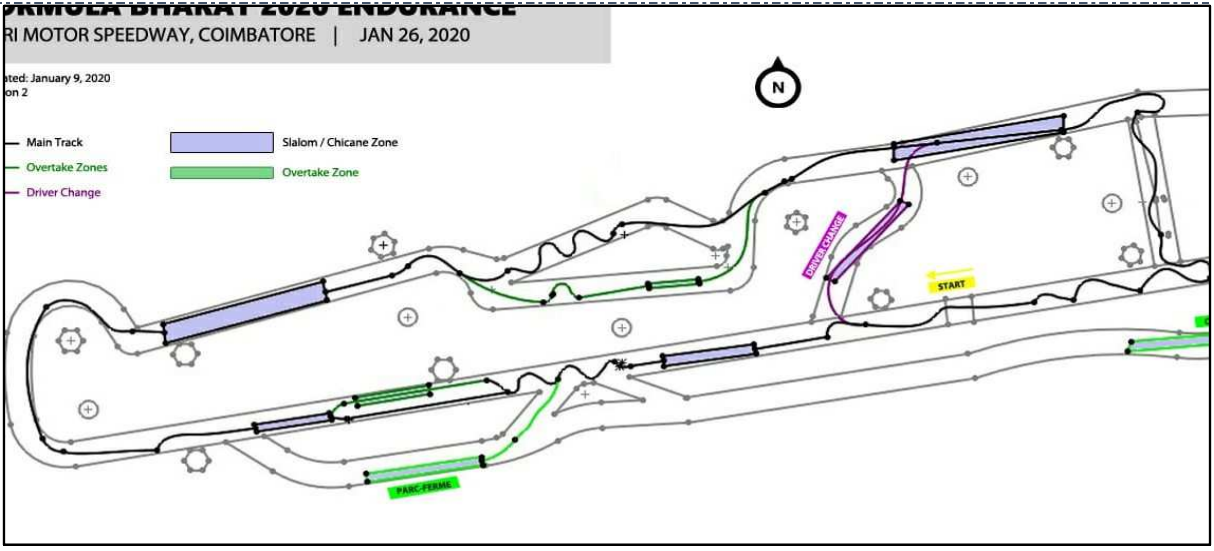

Fig. – 8. (Formula Bharat Endurance track with slaloms)

3. RESULTS AND DISCUSSION

Comparison with actual timing

The total time calculated using MATLAB programing was 65.57 seconds as shown in Fig. – V.B.1.5.
From the original track the part of track which contains Slaloms was removed and it was replaced with straight track, for the ease of calculations.

The time calculated from MATLAB Programming was compared with the time of the other cars. The timing of the other car which had a run on the endurance lap with slaloms was 102 seconds. And the calculated from MATLAB Programming for Ojaswat 2020 car on the same track without slaloms but with straight track was 65.57 seconds and on the other hand for the other cars timing slaloms replaced with straight track was estimated to be 76 seconds.

ISSN: 2277-9655
[Patel *et al.*, 10(2): February, 2021] Impact Factor: 5.164
IC™ Value: 3.00 CODEN: IJESS7

	assumed radius (mm)	actual radius(m)	assumed length (mm)	actual length (m)	limiting velo(m/s)	entering velocity (m/s)	entering velocity (km/hr)	exit velocity (m/s)	max velocity (m/s)	acceleration dist(m)	deceleration dist(starts from end)	time(sec)
S1	0	0	1086.97	410.5337		6.4043	23.05548	9.551	26.15	398.25	398.25	21.0591
C1	22.76	8.59614	28.43	10.73762	9.27315	9.55	34.38	9.55	9.55	0	0	1.1242
C2	139.41	52.65325	136.46	51.53907	24.2063	9.55	34.38	12.4455	15.92	49.5	49.5	3.9956
C3	38.1	14.38985	62.37	23.55629	12.0784	12.4455	44.8038	10.3556	12.4455	0	22.5	1.7609
C4	26.66	10.06912	32.84	12.40322	10.05325	10.3556	37.28016	10.3556	10.3556	0	0	1.1977
S2	0	0	252.03	95.18828		10.3556	37.28016	14.2692	19.45	91.5	91.5	6.2625
C5	49.55	18.71436	27.31	10.31461	13.84406	14.2694	51.36984	8.9237	14.2694	0	7.75	0.488
C6	19.92	7.523511	32.33	12.2106	8.664671	8.664671	31.19282	8.6647	8.66467	0	0	1.3683
C7	46.53	17.57374	59.75	22.56676	13.39761	8.6646	31.19256	12.412	12.412	22.5	0	2.0787
S3	0	0	120.56	45.53386		12.412	44.6832	14.7745	17	45	45	3.0855
C8	52.95	19.99849	41.67	15.73819	14.33279	14.7745	53.1882	14.7745	14.7745	0	0	1.0652
C9	90.03	34.0031	57.45	21.69808	19.0054	14.7745	53.1882	16.7265	16.7265	21.5	0	1.3363
S4	0	0	76.87	29.03275		16.7265	60.2154	18.913	18.913	29	29	1.366
C10	142.82	53.94116	103.43	39.06409	24.54093	18.913	68.0868	21.2357	21.2357	39	0	1.9432
S5	0	0	156.52	59.11546		21.2357	76.44852	16.1615	23.75	52.75	52.75	2.6216
C11	62.77	23.70737	44.14	16.67107	15.67399	16.1615	58.1814	15.1788	15.1788	0	16	0.9873
C12	55.74	21.05223	59.38	22.42701	14.72383	15.1788	54.64368	5.6351	15.25	0	18.25	1.1776
C13	8.03	3.032821	25.58	9.661215	5.473243	5.6351	20.28636	5.6351	5.6351	0	0	1.7145
C14	14.04	5.302716	23.79	8.985157	7.255879	5.6351	20.28636	7.4717	7.4717	7.75	0	1.2155
C15	21.59	8.154247	21.22	8.014503	9.027083	7.4717	26.89812	9.2974	9.2974	8	0	0.9748
C16	52.64	19.88141	62.48	23.59784	14.2888	9.2974	33.47064	9.3718	12.79	22	22	2.0931
C17	21.93	8.28266	26.78	10.11444	9.099224	9.3718	33.73848	7.6844	9.4	0	9.5	0.9574
C18	14.84	5.604865	21.87	8.259999	7.462304	7.6844	27.66384	7.1752	7.6844	0	8.25	1.0288
C19	12.84	5.1868	22.63	8.547041	7.175189	7.1752	25.83072	7.1752	7.1752	0	0	1.2324
C20	32.69	12.34656	29.24	11.04355	11.16157	7.1752	25.83072	6.443	11.5	9.25	9.25	1.305
C21	10.35	3.909053	35.6	13.44563	6.219975	6.4043	23.05548	6.4043		0	0	2.0994
			2647.7	1000						total time=		65.5386

Fig-9. (Total time calculations)

4. CONCLUSION

Practically constructing the track and after updating the parameters in the car according to the study and after driving on track, it was concluded that the lap time obtained was 75 seconds which is close to time calculated of other car for which slaloms was replaced by straight track. So it can be concluded now after this many changes in car that lap time can be majorly decreased by driver's skills and practice and minorly dependent on cars efficiency.

- MATLAB- coding using loops, conditional statements, user define functions, and other basic topics. Mathematical modelling for calculating minimum lap time.
- Textbooks-Fundamentals of vehice dynamics by Thomas DGillespie-Tire's physical forces theory.
- William and Douglas-Milliken-forces on cars
- Research papers- concept of programming algorithm and its importance.
- Internet sources-gear ratio for calculating the speed from gearcommander.com
- This study also helped us to learn and enhance our skills in Solid works and Autodesk Fusion 360-inserting image and sketching the map of track.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

These all FSAE competition holds at national as well as international platforms, it serves the students to share and gain knowledge from recognized teams and also it gives exposure to judges like Pat Clarke & Claude Rouelle and famous industrialists.

This fragment should obviously state the foremost conclusions of the exploration and give a coherent explanation of their significance and consequence.

5. ACKNOWLEDGEMENTS

We express our gratitude to our guide Mr. Anand P Patel for his invaluable guidance and overwhelming encouragement throughout this project work. We are impelled as our respected guide has dedicated his valuable time and shared his expert knowledge.

We are gratefulness to the Mechanical engineering Department of C.S.P.I.T -CHARUSAT University –Changa for permitting us to execute the project work. We also pay our tributes to all teammates of Team Ojaswat. who have helped us in the execution of our work.

We prolong our sincere thanks to HOD –Dr. Vijay Chaudhary, Department of Mechanical Engineering, Principal –Dr. A. D. Patel of C.S.P.I.T and Dean, Faculty of Technology & Engineering, CHARUSAT for allowing us to carry our work.

We also like to thank our friends and colleagues for supporting us and constantly encouraging us to successfully complete our work.

REFERENCES

[1] William and Douglas Milliken.
[2] Fundamentals of Vehicle Dynamics by Thomas D Gillespie

Research papers

[3] Gadola, M.; Vetturi, D.; Cambiaghi, D.; Manzo, L. A Tool for Lap Time Simulation (No. 962529); SAE Technical Paper; SAE International: Warrendale, PA, USA, 1996
[4] Malcher, S.; Bargende, M.; Grill, M.; Baretzky, U.; Diel, H.; Wohlgemuth, S. Virtual Optimization of Race Engines Through an Extended Quasi Steady State Lap Time Simulation Approach (No. 2018-01-0587); SAE Technical Paper; SAE International: Warrendale, PA, USA, 2018.

Websites

[5] https://www.solidworks.com/
[6] https://www.youtube.com
[7] https://www.google.co.in
[8] https://in.mathworks.com/
[9] https://www.autodesk.com/
[10] www.gearingcommander.com
[11] www.engineeringtoolbox.com
[12] www.hpwizard.com

Web course

[13] MATLAB onramp
[14] NPTEL vehicle dynamics
[15] NPTEL Fundamentals of Automotive Engineering

Forums

[16] MATLAB forum
[17] Fusion 360 forum
[18] SAE forum
[19] FSAE forum
[20] Solidworks forum

Resources

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[21] Fusion 360
[22] Solidworks
[23] MATLAB
[24] Microsoft office (excel and word)