Abhay Dalmia

November 27, 2015

AE 2220: Dynamics Project

Daredevil on a Rocket-Propelled Bicycle

The aim of this project is to do a dynamical analysis of an experiment that was conducted in Switzerland on November 7, 2014. In this experiment, the rider Francois Gissy, rode a rocket propelled bike to a top speed of 92.5 m/s.

Most of the information provided from this experiment deals directly with kinematics, as the velocities and accelerations of the bike at several different points is provided. The information provided is summarized below

Summary of Information Provided[1]

Thrust due to Rocket $= 4.2\ kN$

Maximum In-Line Acceleration $= 3.1 \cdot g\ m/s^2$

Peak Deceleration $= 1.5 \cdot g\ m/s^2$

Peak Speed $= 92.5\ m/s$ at $\approx 250\ m$

All Time Data in seconds, Velocities in meters/second and distances in meters.

Time $= 0$	Time $= 1.1$	Time $= 2.4$	Time $= 4.1$	Time $= 4.8$	Time $= 6.2$
$v = 0$	$v = 27.7$	$v = 55.6$	$v = 83.3$	$v = 92.5$	$s = 402.3$
				$s \approx 250$	

This is a summary of all the data provided. No data is provided on the other forces affecting the bicycle (mainly resistive forces). No data is given on the modulation/thrust profile of the particular custom rocket that he used to reach these speeds.

However, using numerical methods on MATLAB and GeoGebra (Graphing Software), an analysis can be conducted by making educated assertions and assumptions to simplify this problem.

[1] "Francois Gissy World Record." Kamikaze III. N.p., n.d. Web. 22 Nov. 2015. <http://www.swissrocketman.fr/kamikaze-iii,fr,8,91.cfm>.

Analysis of Problem

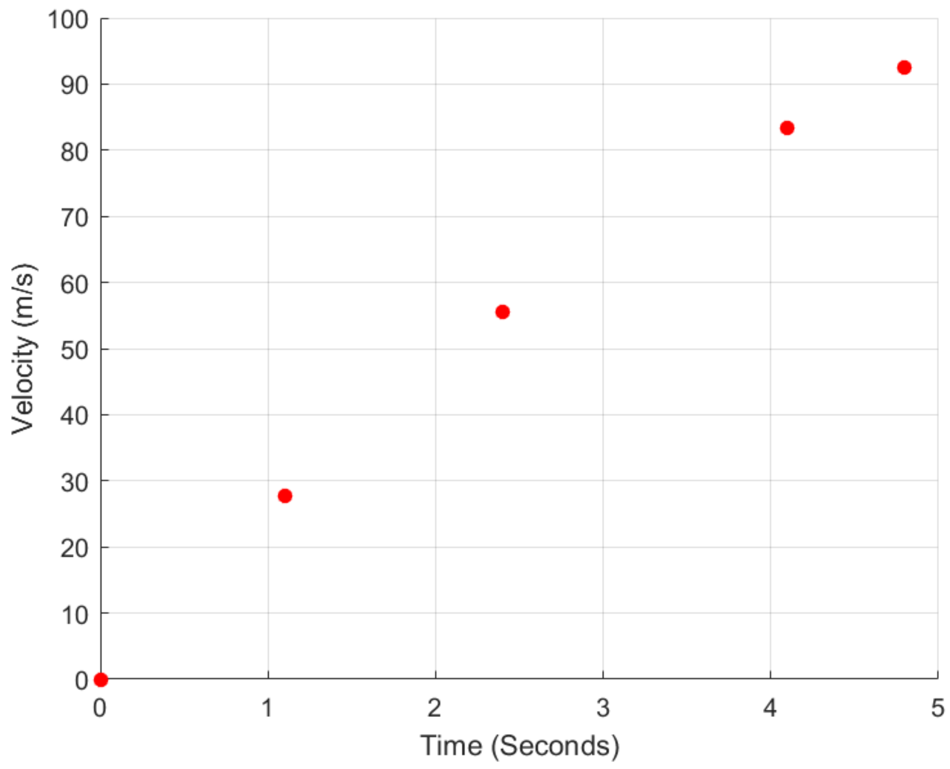
The values provided are basically the thrust/propulsive force on the bicycle and the consequent velocities and accelerations that this thrust creates. So we can conclude that this thrust can definitely cause a top speed of 92.5 m/s but the real question to answer is for what maximum mass the thrust force can create this velocity. A force of any magnitude can cause a kinematic constraint of infinitely different values/varieties as the mass changes. The same force will be able to accelerate a small mass to greater speeds than a large mass. Hence, this thrust and velocity profile is possible, but only for a certain mass. We will prove that this experiment was correct by finding this maximum mass and showing that the bicycle and rider mass is around or below this value.

A crucial assumption that can also be made, is that the **thrust force acts on the bicycle for 4.8 seconds**, until it reaches its maximum speed. Because at this time, the bicycle starts rapidly losing speed and that would suggest that there is a sudden loss of forward/propulsive force, hence the backward forces of drag, resistance etc. take over and create deceleration. **(Assumption 1)**

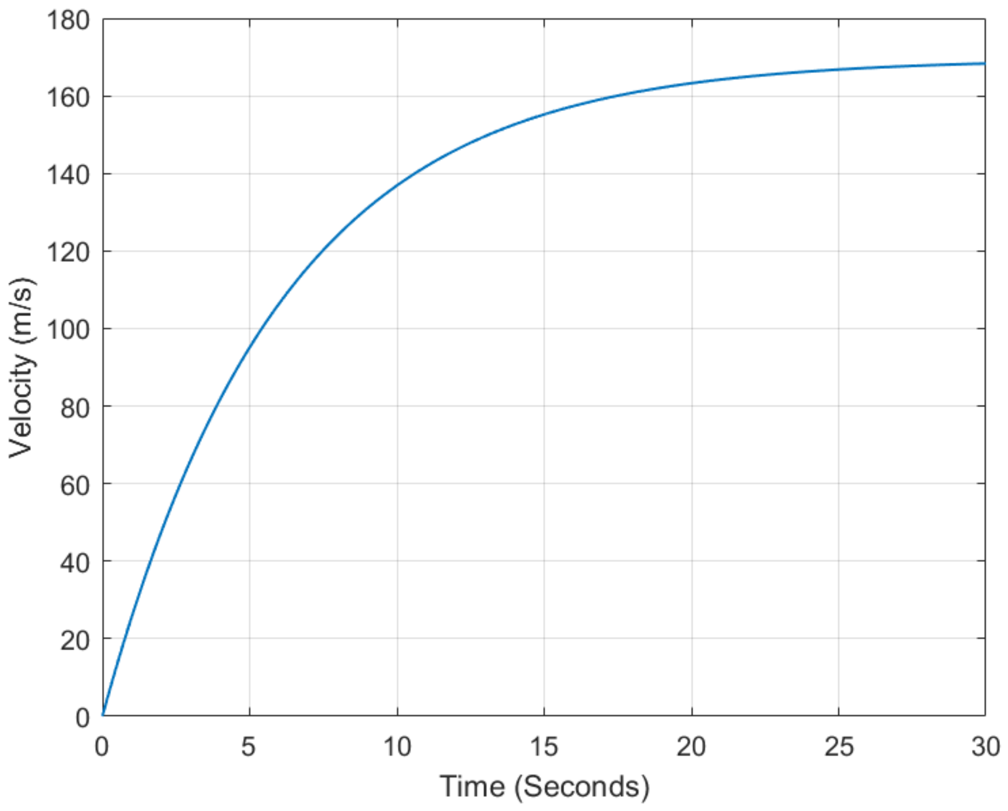
Modelling the Velocity Profile

The velocities given above are for the time interval in which the thrust is acting until time at 4.8 seconds when the bicycle reaches top speed and thrust stops acting on the bicycle.

When plotted the discrete points look like:



The velocity of a rocket propelled body normally takes the shape of an exponential decay (increasing curve)[2] as shown below:



This curve is very similar to a terminal velocity curve. This can be explained because when the thrust just starts, there is great acceleration from the thrust but very little drag (as drag is proportional to Velocity). This leads to great acceleration. But as the speed increases, the drag force and other resistances also increase until at one point, they equal the thrust and this leads to terminal velocity. It is highly unlikely that in our case the cycle reaches terminal velocity before thrust/propulsive force is finished (but we shall soon find out!).

So another assumption we can make is that **the velocity profile while thrust acts on the body can be fitted** with the points **to an exponential decaying curve**. **(Assumption 2)**

Mathematically this curve can be estimated by **(Assumption 3):**

$$v = C \cdot (1 - e^{-kt})$$ [3]

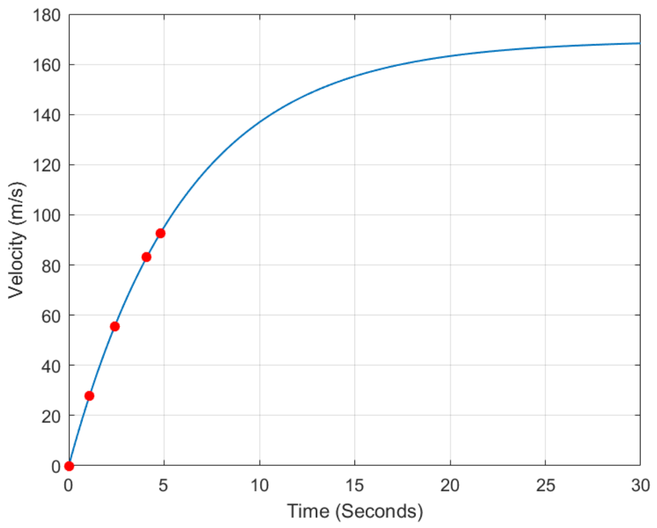
[2] "What Is the Motion of a Flying Rocket?" GCSE PHYSICS. N.p., n.d. Web. 22 Nov. 2015. <http://www.gcsescience.com/pfm21.htm>.
[3] "Exponential and Logarithmic Models." *Mathematical Models*. N.p., n.d. Web. 22 Nov. 2015. <https://people.richland.edu/james/lecture/m116/logs/models.html>.

There are two constants however, in the equation. Next we must find the value of these constants by applying least squares regression analysis of the points and the equation, to calculate the best fit equation through these points. This was done on MATLAB in **VelocityCurveFitting.m**. (**Appendix 1** – at the end of the report).

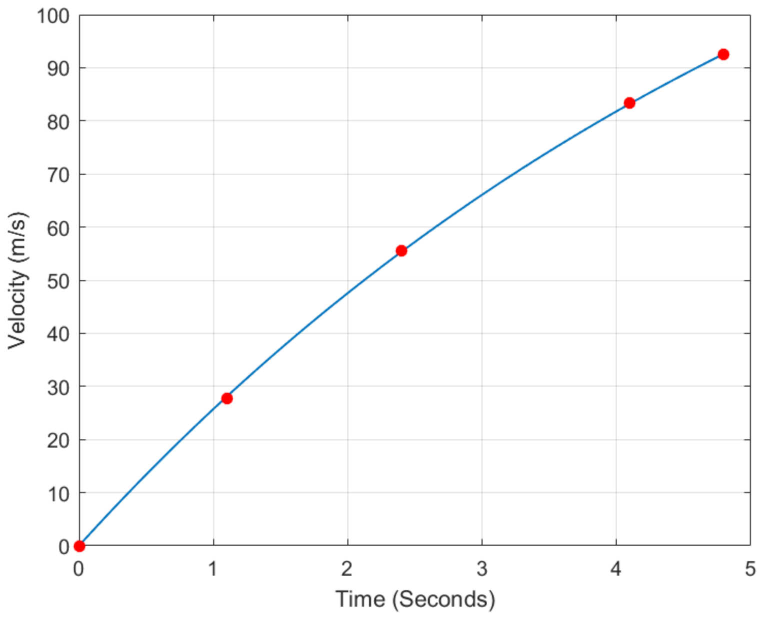
The function outputs the values of the constants so the final velocity curve until time = 4.8s becomes:

$$\boldsymbol{v(t) = 169.5507 \cdot (1 - e^{-0.1646t}) \text{ when } 0 \leq t \leq 4.8}$$

When this curve and the points are plotted, the following curve is produced:



As we can see, the fit to the points is very accurate. But we only require the curve until 4.8 seconds, because after that it does not correctly represent the velocity profile (as there is no more propulsive force.) So the above graph becomes:



We can check the accuracy of this model. We know that the time travelled in the first 4.8 seconds is around 250 meters. So this equation/model can be integrated from start to time = 4.8 seconds to find the distance travelled. This is done on MATLAB towards the end of **VelocityCurveFitting.m.** The distance travelled found by the function is 251.33 meters which is very close to the estimated value of 250 meters which was provided.

With this we can conclude that **this model is both precise and very accurate**.

This equation for velocity can be differentiated and integrated for functions of acceleration and distance in time.

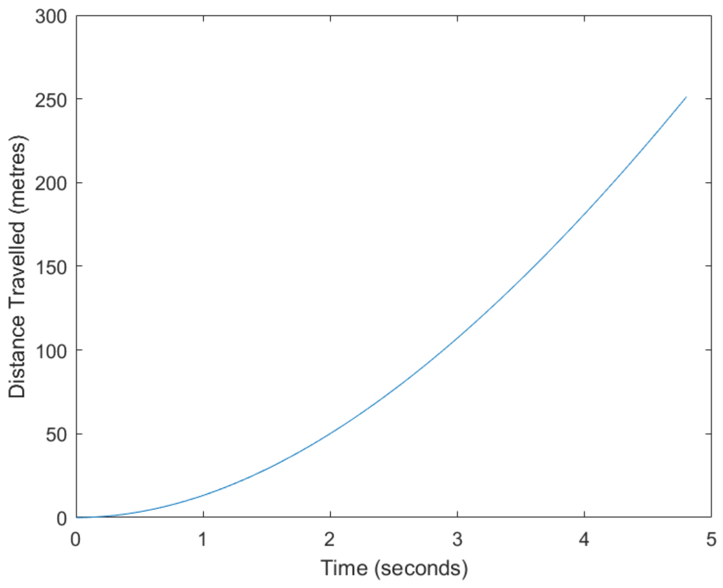
Hence, we conclude with the results:

$$s(t) = 169.5507 \cdot t + 1030.08 \cdot e^{-0.1646t} - 1030.08$$

$$v(t) = 169.5507 \cdot (1 - e^{-0.1646t})$$

$$a(t) = 27.908 \cdot (e^{-0.1646t})$$

$$\mathbf{0 \leq \boldsymbol{t} \leq 4.8}$$



Note: Another proof that this model for distance, velocity and acceleration until time is equal to 4.8 seconds is accurate is the acceleration graph. According to the data, the maximum inline acceleration felt by the rider is 3.1g's. According to our model, the rider experiences 3.0g's at maximum (at $t = 0s$). So this is a very accurate and precise model and hence, can be used further on.

Resistive Forces

There are many resistive forces on a rider, but most of them are negligible. For the purposes of this project, we will assume that only two resistive forces are significant and thus have to be considered: Drag/Wind Resistance and Rolling Resistance. So we are assuming that **only Wind Resistance and Rolling Resistance are significant.**[4] **(Assumption 4)**

Modelling Rolling Resistance

Rolling Resistance is analogous to friction on surfaces, but it applies to rolling bodies. The concept of linear friction can no longer be used, as the contact point of a circular wheel changes instantaneously. Hence, instead the concept of Rolling Resistance must be used. Rolling Resistance itself is complex but it can be estimated to **(Assumption 5)**:

$$F_{Roll.Rest.} = C_R \cdot mg$$ [5]

Where C_R is the coefficient of Rolling Resistance.

Further we can estimate the value of the coefficient of Rolling Resistance for a bicycle tire on an asphalt road (as can be seen in the video) **(Assumption 6)**. For this assumption/estimation, $C_R = 0.004$.

Hence,

$$\boldsymbol{F_{Roll.Rest.} = 0.03924 \cdot m_{total}\ N}$$

Modelling Drag/ Wind Resistance

Wind Resistance/Drag is significant especially at high velocities. Hence, it is a very important factor in this project/investigation.

The Wind Resistance can be estimated by:

$$F_{WR} = \frac{1}{2} \cdot \rho_{air} \cdot C_D \cdot A_{section} \cdot v^2$$

[4] "Forces on Rider." *Analytic Cycling*. Tom Compton, n.d. Web. 23 Nov. 2015. <http://www.analyticcycling.com/ForcesPower_Page.html>.
[5] "Rolling Resistance." *Rolling Resistance*. Engineering Toolbox, n.d. Web. 23 Nov. 2015. <http://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html>.

(Assumptions 7) To simplify this equation, there are many assumptions we can make for the values. Typically$A_{section} = 0.5\ m^2$. Since the rockets added to the bicycle were attached horizontally to the back, it would not increase the Area by much.[6]

So assuming $A_{section} = 0.5\ m^2$.

Further since this experiment took place at a high altitude (the circuit was about 500m above sea level) and at normal levels of wind speed, we can conclude that:

$$\rho_{air} = 1.125\ kg \cdot m^{-3} \text{ and } C_D = 0.5$$

Hence, Drag Force is:

$$\boldsymbol{F_{WR} = 0.140625 \cdot v^2\ N}$$

Force Balance until $t = 4.8s$

For now, we will assume that the mass does not change throughout the process (Assumption 8). Note: Later the change in mass due to the fuel will be discussed and investigated as well but not at this stage.

Since we now have expressions for all our forces except for thrust, we can do a Force Balance to try to find Force of Thrust and mass of the system.

$$F_{Thrust} - F_{Roll.Rest} - F_{WR} = m \cdot a$$

$$\boldsymbol{F_{Thrust} - 0.03924 \cdot m - 0.140625 \cdot v^2 = m \cdot a}$$

In this equation, the only unknowns are F_{Thrust} and m. Hence, the Thrust Force is a function of the total mass (and time since velocity and acceleration change with time).

We know that the average thrust throughout the region when thrust was produced is 4.2 kN.

Hence,

$$F_{Thrust}\text{Avg.} = \text{mean}(m \cdot a + 0.140625 \cdot v^2 + 0.03924 \cdot m) = 4200N$$

For computing this MATLAB is required again. To compute this, we have to iterate through small values of time between 0 and 4.8 seconds. Then insert a value of mass and calculate the average value of Thrust. We have to keep trying values of mass until the average value of the Force of Thrust comes to $4200N$.

[6] "Forces on Rider." *Analytic Cycling*. Tom Compton, n.d. Web. 23 Nov. 2015. <http://www.analyticcycling.com/ForcesPower_Page.html>.

This is essentially a root finding problem. A root finding problem is when many different values have to be tried systematically to get some value of a function. (Interesting! This is how computers and calculators calculate the square root of 2!).

In this case I used the Newton-Raphson method to conduct the root finding algorithm to solve for the mass in the above equation. **FindMass.m (Appendix 2) and Newton.m (Appendix 3 below).**

When the function is run, the value of mass found is:

$$\text{mass} = 192.1795\ kg$$

Hence, we have essentially solved the problem/investigation.

Final Proof/Analysis

This kinematic/kinetic problem with 4.2kN of average thrust and the velocity and distance values given in the data (first page) is possible for a mass of 192.1795 kg. That means with the thrust calculated and all the assumptions on the forces above, a 192 kg mass can be accelerated to a top speed of 92.5 m/s.

This is also a very logical value because the average weight of a person is around 75kg. That means the bicycle must be a maximum of around 112 kg. Normally bicycles are not that heavy but considering the mass of the rocket thrusters, electrical components and fuel, it is very much possible that the bicycle comes near this value.

However, it should be noted that this is the **maximum value**! One of the assumptions above are that there are no resistive forces other than that due to rolling and drag. If there were other resistive forces considered (because in real life, there are many other small forces and phenomenon that take place mostly because a bicycles tire is NOT rigid), **then this possible mass value will reduce**. So if more resistive forces were taken into account, the mass value would decrease.

Hence, it is proved that this experiment is possible with the laws of dynamics (if the total mass of the system at the start is less than 192 kgs!).

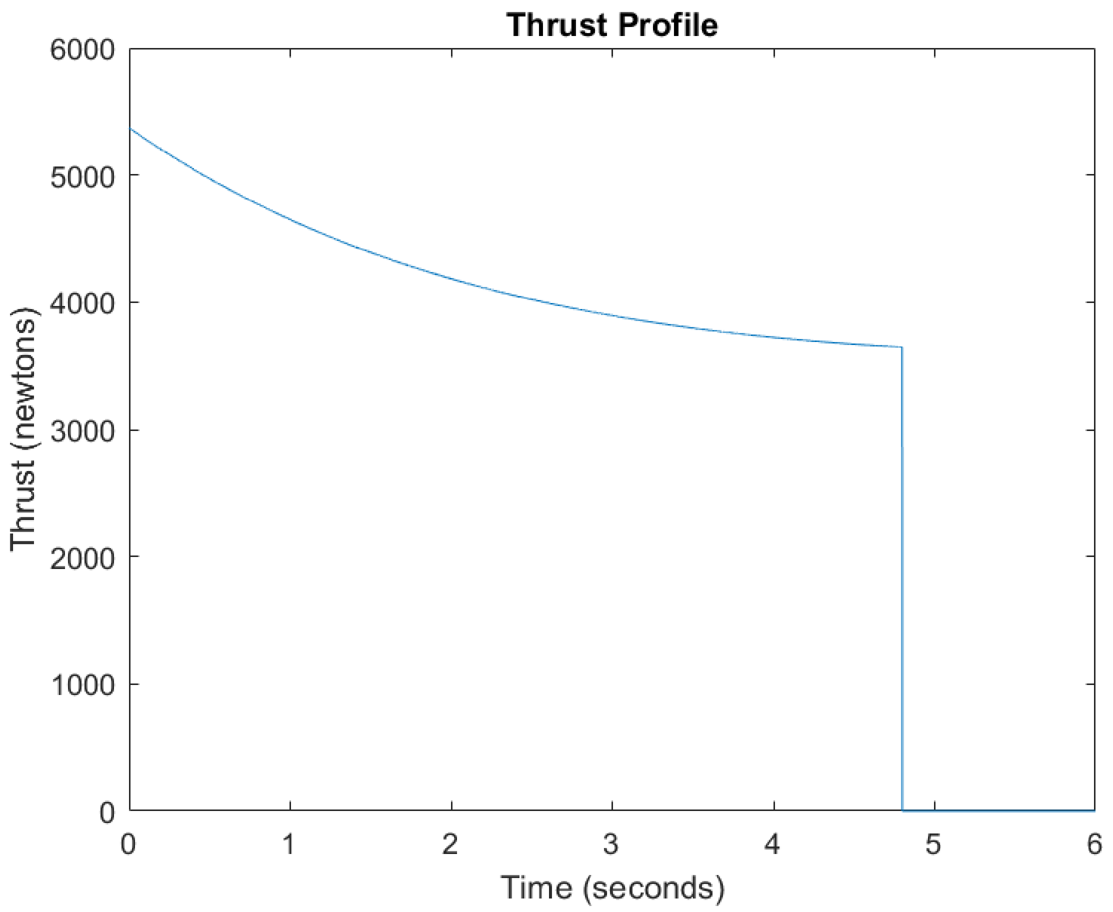
The maximum force his spine has to handle in this is right at the start when the thrust is at a maximum and drag force is virtually 0. **At this point the net Force felt by him is 5300N.**

Thrust Profile

Now that we know the mass, the Thrust Force profile can be created as the Force of Thrust is now dependent on velocity and acceleration, both of which is further related to time:

$$F_{Thrust}(t) = 192.1795 \cdot a(t) + 0.140625 \cdot v(t)^2 + 7.54112)$$

Using MATLAB and the code in **plotThrust.m (Appendix 4 below)**:



So the rocket propulsion only functions for the first 4.8 seconds (until the maximum speed is reached). After that, the fuel burns out, and the lack of forward force, created deceleration and starts slowing the bicycle down.

Distance Travelled, Velocity and Acceleration as t > 4.8s

Previously, the equations that governed the distance travelled, velocity and acceleration of the bicycle before 4.8 seconds was evaluated. Now, with the information on mass and the different resistive forces affecting our bicycle is evaluated, we can evaluate the Velocity, Acceleration and Distance Travelled after 4.8 seconds, until rest.

After 4.8 seconds, when there is no thrust:

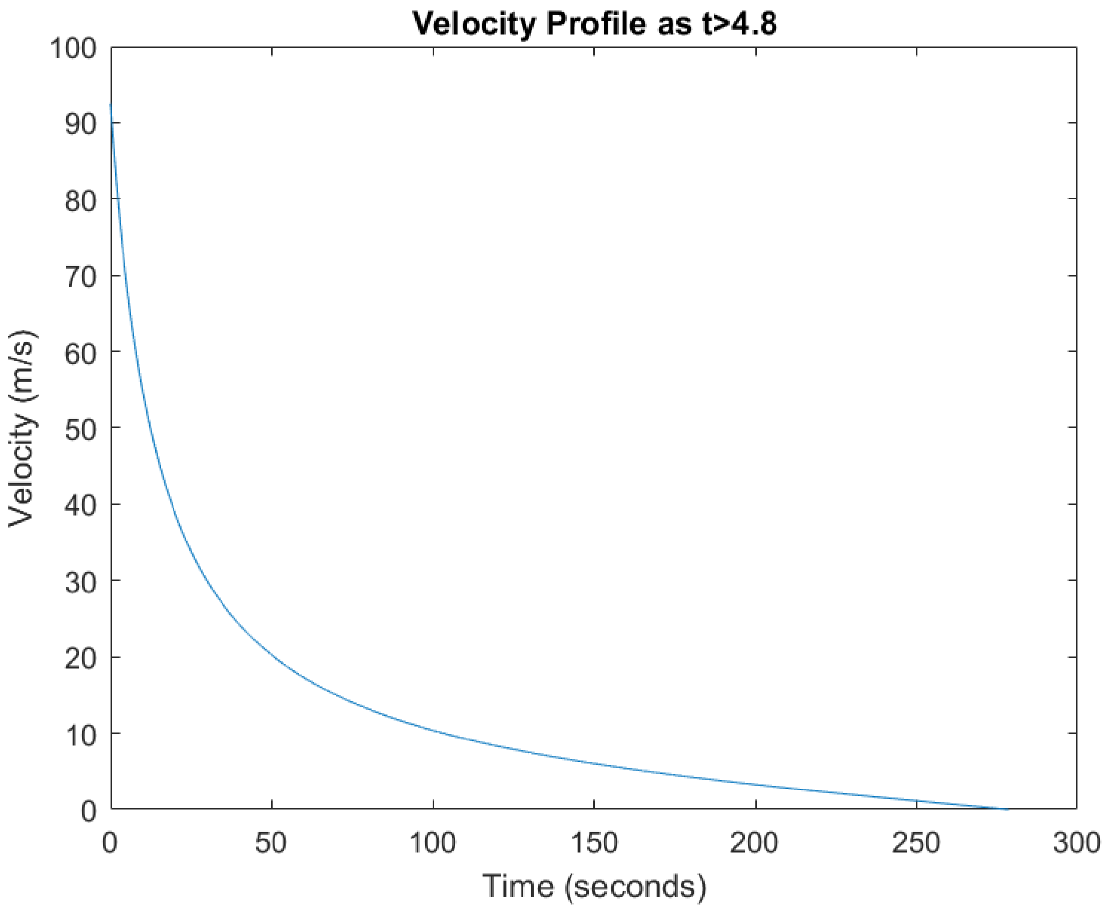
$$\frac{-\mathbf{0.03924} \cdot \boldsymbol{m} - \mathbf{0.140625} \cdot \boldsymbol{v}^2}{\boldsymbol{m}} = \boldsymbol{a}$$

Taking a small time interval of 0.01 seconds (because MATLAB will be used again), then:

$$v_{t=t0+0.01} = v_{t=t0} + a(0.01)$$

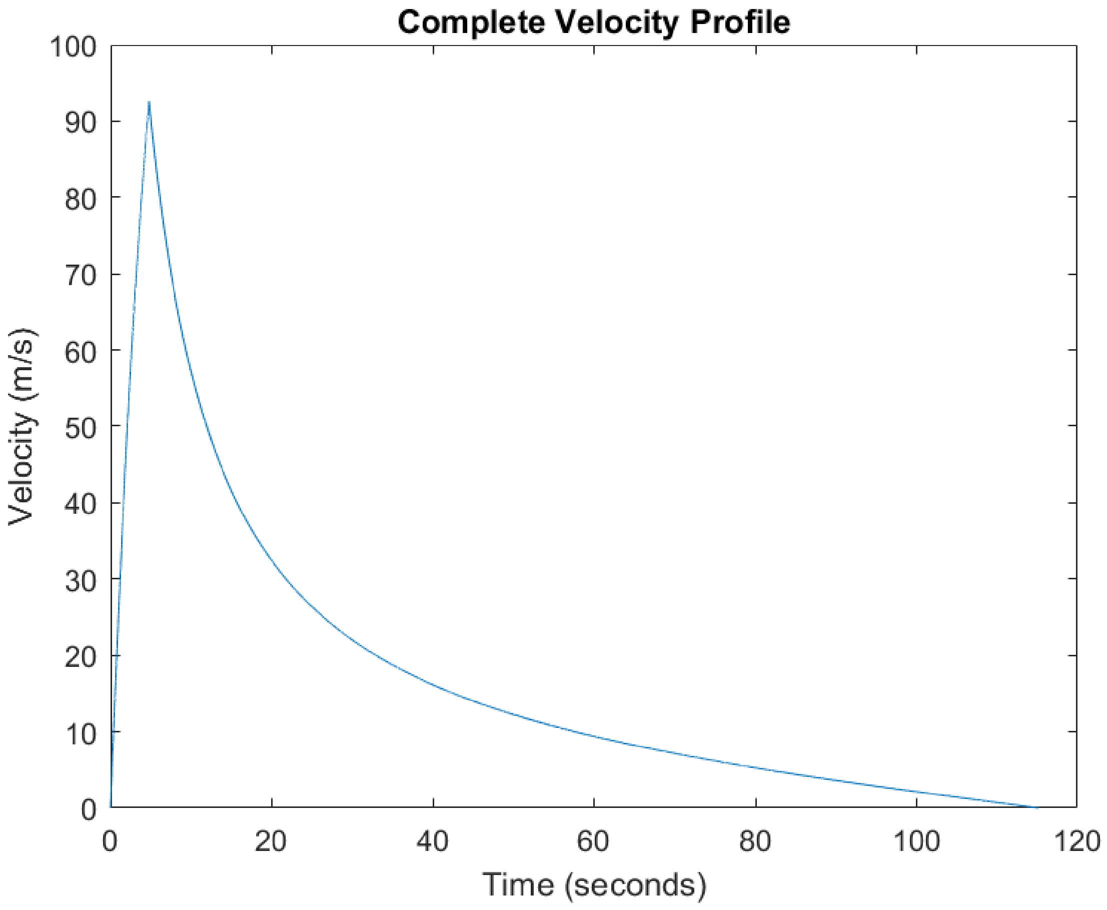
Since, the time interval is tiny, it can be **assumed that within that 0.01 seconds, the acceleration does NOT change. (Assumption 8)**

Using these two equations, the velocity profile for $t > 4.8\ s$ was coded and plotted by the function **VelocityComplete.m (Appendix 5).**



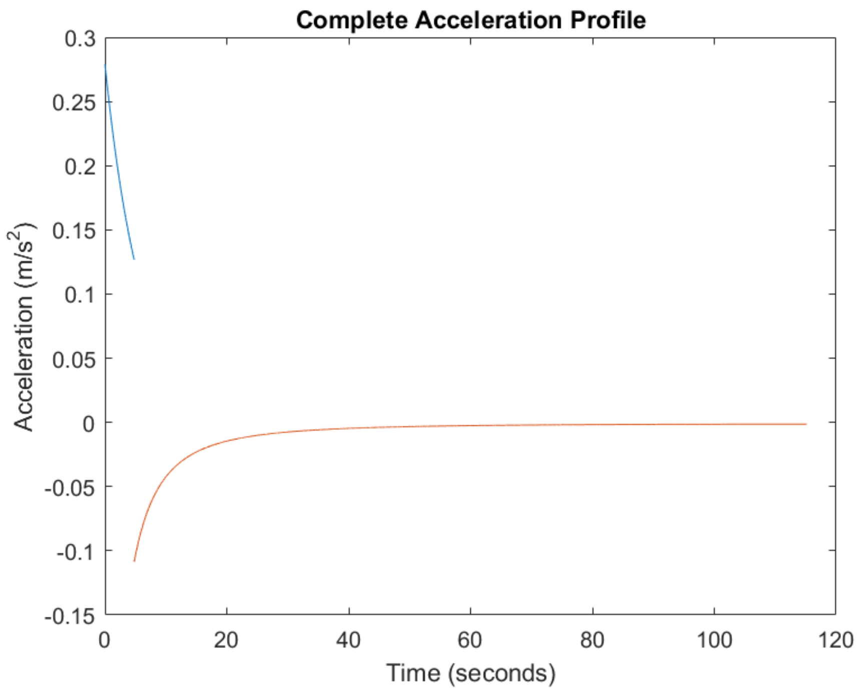
According to this result, the bicycle will only come to a complete stop at around 5 minutes. This does not seem logical and hence, brings attention to the coefficient of Rolling Friction. While drag is correctly modeled for the whole journey, the coefficient will change slightly when the velocities are low. For our model up to 4.8 seconds, it was neglibile because our system only stayed at low speed for a fraction of a second. Hence, for this, the coefficient of rolling friction must be slightly changed as at lower speeds, the effect of static friction of the rubber tires and asphalt road is increased by orders of 10[7].

Hence, once the coefficient of rolling friction is adjusted for low speeds, the following **complete velocity profile is produced by the code in Appendix 5**:



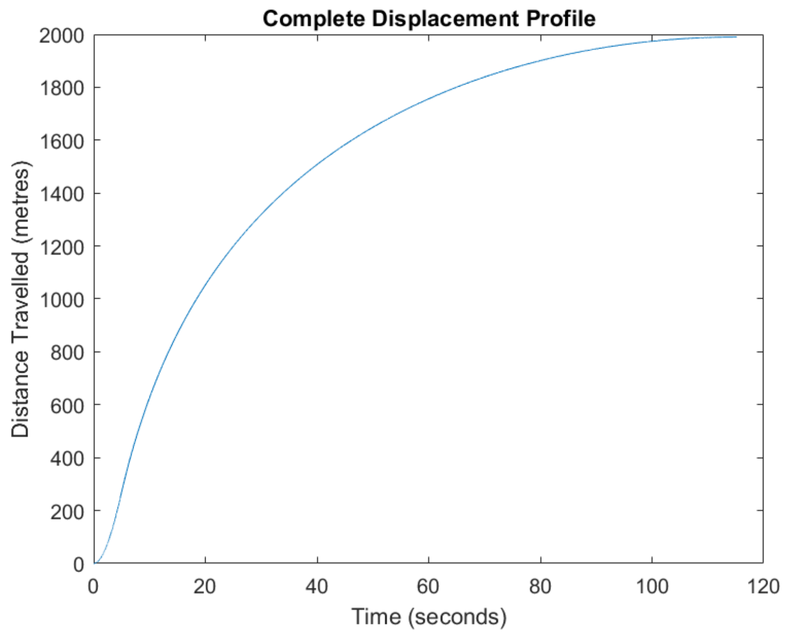
Further, if we numerically integrate and differentiate this complete velocity profile coordinates, then we can similarly get the complete distance travelled and acceleration profiles as well for this entire experiment. The code **AccDisComplete.m (Appendix 6)** does this operation and graphs the other two plots below.

[7] "Friction." *Friction and Coefficients of Friction*. Engineering Toolbox, n.d. Web. 30 Nov. 2015. <http://www.engineeringtoolbox.com/friction-coefficients-d_778.html>.



The acceleration profile is very interesting and as expected. While the thrust is acting on the bicycle, the acceleration is positive but decreasing because of the effect of drag. Drag increases with time as it is dependent on velocity (which also increases in time for the first 4.8 seconds).

But then, thrust vanishes at 4.8 seconds, and the bicycle starts decelerating instead (red curve). It starts with a steep deceleration, but as velocity is reduced, so is drag and hence, deceleration also approaches zero.



The displacement profile is also as expected. Until 4.8 seconds, the graph is concave up (as there is acceleration and hence, a greater amount of displacement in a unit time. But after 4.8 seconds, it's concave down due to deceleration. **The total displacement at $t = 115$ seconds, when it comes to a stop is about 1990 meters.**

NOTE: This is the maximum amount. Some resistive forces were ignored as negligible, but if they were considered the distance travelled would be slightly less.

Next Version: Spine Crusher

The total amount of force that the spine can withstand is far greater in the horizontal direction rather than the vertical direction (from head to toe). What is more important however, is impulse, or rather how long you are subjected to a particular force or acceleration.

According to NASA[8], in the horizontal axis, the body can take 17g's of force for several minutes before losing consciousness. But this is still not enough to crush his spine. The world record is help by John Stapp at 46.2G's of acceleration but even this was not even close to crushing the spine. At this point, permanent vision damage is caused. This point is 87kN of Force, or about 18 rockets of 4.2kN each.

The amount of force required to snap the spine due to horizontal acceleration is not known. (This is why astronauts during launch sit in the direction of motion and not against it. Otherwise their spines would be crushed due to vertical force.).

So he can add more than 18 rockets in the horizontal direction and his spine wont be crushed (but his vision will be permanently damaged.)

[8] http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19980223621.pdf

References

[1] "Francois Gissy World Record." Kamikaze III. N.p., n.d. Web. 22 Nov. 2015. <http://www.swissrocketman.fr/kamikaze-iii,fr,8,91.cfm>.

[2] "What Is the Motion of a Flying Rocket?" GCSE PHYSICS. N.p., n.d. Web. 22 Nov. 2015. <http://www.gcsescience.com/pfm21.htm>.

[3] "Exponential and Logarithmic Models." *Mathematical Models*. N.p., n.d. Web. 22 Nov. 2015. <https://people.richland.edu/james/lecture/m116/logs/models.html>.

[4] "Forces on Rider." *Analytic Cycling*. Tom Compton, n.d. Web. 23 Nov. 2015. <http://www.analyticcycling.com/ForcesPower_Page.html>.

[5] "Rolling Resistance." *Rolling Resistance*. Engineering Toolbox, n.d. Web. 23 Nov. 2015. <http://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html>.

[6] "Forces on Rider." *Analytic Cycling*. Tom Compton, n.d. Web. 23 Nov. 2015. <http://www.analyticcycling.com/ForcesPower_Page.html>.

[7] "Friction." *Friction and Coefficients of Friction*. Engineering Toolbox, n.d. Web. 30 Nov. 2015. <http://www.engineeringtoolbox.com/friction-coefficients-d_778.html>.

[8] http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19980223621.pdf

Appendix 1: VelocityCurveFitting.m

```
%This function defines the kinematic (Velocity) constraints of the bicycle
%and fits the discrete values of velocity to a particular curve. Then it
%tests the validity of this curve approximation by checking both, the
%accuracy and the preciseness. THIS MODELS THE VELOCITY UNTIL TOP SPEED IS
%REACHED WHERE time<4.8 seconds.

%Author: Abhay Dalmia
%Date: 22 November 2015

%Stating the corresponding times and speeds of the bicycle on its way to a
%top speed of 92.5m/s
time=[0,1.1,2.4,4.1,4.8];
speed=[0,27.7778,55.5556,83.3333,92.5000];

%Initializing the function as an anonymous function with a pointer as
%required by the inbuilt function. The values to be found are in the
%vector B. This function is of the form y=Ce^(-kt). And we are trying to
%find values for C and k that fit the above points by least squares
%regression.
FitFunction=@(B,x) B(1).*(1-exp(-B(2).*x));
%This inbuilt function carres out the curve fitting and outputs the values
%of C and k to a vector "out". The vector [400,2] are simply guess values
%where the curve fitting regression algorithm starts its iteration.
out=nlinfit(time,speed,func,[400,2]);
%Finally we can define the velocity profile from t=0 to t=4.8s
FinalFunction=@(x) out(1).*(1-exp(-out(2).*x));
%Plotting the original points and the new function that we have found.
xFunc=linspace(0,4.8,10000);
yFunc=FinalFunction(xFunc);
plot(xFunc,yFunc,'LineWidth',1)
hold on
plot(time,speed,'r.','MarkerSize',20);
grid on
xlabel('Time (Seconds)');
ylabel('Velocity (m/s)');

%Checking the accuracy of the model. It is given that the bicycle travels
%around 250 meters during the first 4.8 seconds. So our model is
%numerically integrated until 4.8 seconds to find the distance travelled.
distance=integral(FinalFunction,0,4.8);
fprintf('\nThe calculated distance travelled by the modelled velocity')
fprintf('is %0.1f meters in 4.8 seconds.\nThe literature ',distance);
fprintf('value is around 250 metres.\n\n');

%End.
```

Appendix 2: FindMass.m

```
function mass = FindMass
%This function initializes the force of thrust function in the report and conducts the
root finding algorithm (Newton-Raphson Method) to calculate mass.

%Author: Abhay Dalmia
%Date: November 23, 2015

%The time, acceleration and velocity vector is initialized
time=linspace(0,4.8,10001);
a=acceleration(time);
v=velocity(time);
%The function that we want to solve for is passed through to the newton
%raphson algorithm. We want the mean of the final force vector to be equal
%to 4200. In other words, the mean subtracted by 4200N to be equal to 0
%which is the root finding problem.
mass=fzero(@(mass) mean(mass.*acceleration(time)+0.140625.*(v.^2)+0.03924*mass)-
4200,190);
end
```

Appendix 3: newton.m

```
function out = newton(fun,x,TolX,itmax)
%This function is a simplified algorithm for the Newton Raphson method, for finding
roots of an equation. In this case it was used for finding the roots of the force
equation where the roots represent the mass of the system.

%Author: Abhay Dalmia
%Date: 24 November 2015

%The variables are initialized to prevent the function from continuing
%forever. Symbolic Derivative of the function is taken as well. NOTE: MUST
%HAVE SYMBOLIC TOOLBOX OF MATLAB FOR THIS!
reached=false;
i=1;
syms y;
deriv=diff(fun,y);

while ~reached && i<=itmax
    f=fun(x);
    d=deriv(x);

    %Calculating the derivate and then the root at y=0.
    xr=x-(f/d);

    %An absolute error to exit at. Since the data type double can handle 64
    %bits, this value normally set to 10^-16 as this is maximum accuracy
    %possible.

    rerr=abs((xr-x)/xr);

    %If new x value within tolerance, the method is complete.
    if rerr<TolX
        out=xr;
        reached=true;
    else
        x=xr;
    end
end
end
```

Appendix 4: plotThrust.m

```
function plotThrust
%This function plots the thrust profile by inserting numerous time values into the
simplified equation for output of different thrust values.

%Author: Abhay Dalmia
%Date: 24 November 2015

%Initializing values and calling functions velocity and acceleration (with
%equations found before in the report). Taking 10001 values of time between
%0 and 4.8.
mass=192.1795;
time=linspace(0,4.8,10001);
a=acceleration(time);
v=velocity(time);

%Calculating the respective Force.
Force=mass.*acceleration(time)+0.140625.*(v.^2)+0.03924*mass;

%After 4.8 seconds, the force of thrust is 0 as all the fuel has been
%burned.
time=[time 4.8005 6];
Force=[Force 0 0];
plot(time,Force);
xlabel('Time (seconds)')
ylabel('Thrust (newtons)')
title('Thrust Profile');
end
```

Appendix 5: VelocityComplete.m

```
function VelocityComplete
%This function find the velocities for time after 4.8 seconds and then plots the
entire velocity profile from time is equal to 0 seconds.

%Author: Abhay Dalmia
%Date: 25 November 2015

%Values are initialized, of the forces and the mass
mass=192.1794954107037;
roll=0.1392.*mass;
FuncWind=@(v) 0.240625.*(v.^2);
%Time increment and intitial values of velocity and time are set.
inc=0.01;
v=[92.5];
t=[0];
%This loop will continue to increment time by 0.01 seconds, until the end
%velocity is equal to 0.
while v(end)>0
    %Time is incremented and for this new value of time, the new values of
    %the resistive forces are calculated.
    t=[t t(end)+inc];
    wind=FuncWind(v(end));
    force=wind+roll;
    %The acceleration between this 0.01 second is computed.
    a=force/mass;
    %The velocity is reduced by the acceleration multiplied by 0.01
    %seconds. Since it is a very small time interval, the assumption that
    %acceleration does not change is viable and accurate.
    vnew=v(end)-a*inc;
    %New velocity at the new incremented time is also added.
    v=[v vnew];
end

%While the time was initialized at 0 before, this is really at 4.8 seconds
%(After the thrust has stopped working). So all-time values increased by
%4.8 seconds, and the time and velocity values before 4.8 seconds is added
%to their respective vectors.
t=t+4.8;
told=linspace(0,4.8,10001);
vold=velocity(told);
t=[told t];
v=[vold v];
%Time and Velocity for the whole experiment is plotted.
plot(t,v);
xlabel('Time (seconds)');
ylabel('Velocity (m/s)');
title('Complete Velocity Profile');
```

Appendix 6: AccDispComplete.m

```
function AccDisComplete
%This function plots the complete acceleration and displacement profile by
%numerically integrating and differentiating the velocity vectors.

%Author: Abhay Dalmia
%Date: 25 November 2015

%The x and y coordinates for velocities is retreievd.
[x,y]=VelocityComplete;

%First figure, the coordinates are numerically integrated with cumtrapz and
%plotted.
figure
distanceTravelled=cumtrapz(x,y);
plot(x(1:end),distanceTravelled);
xlabel('Time (seconds)')
ylabel('Distance Travelled (metres)')
title('Complete Displacement Profile')

%Second figure, the numerical differential is computed. The output
%differential vector is one length shorter (as it is the gradient between
%points. So x vector shortened by one.
figure
acceleration=diff(y);
plot(x(2:481),acceleration(1:480))
hold on
plot(x(482:end),acceleration(481:end))
xlabel('Time (seconds)')
ylabel('Acceleration (m/s^2)')
title('Complete Acceleration Profile')
end
```