Simulate track with simple model

```
clear;
```

Parameters

```
mass = 90; % in kg. Needs to be corrected
air_density = 1.225; % in kg/m�
ftp = 150; % Not exactly sure what it is, but needed to compute wheel force
effective_frontal_area = 0.3; % unit?
rolling_resistance_coefficient = 0.005; % unit?
normal_force = mass * 9.81; % in kg*m/s�
```

Read track information

```
track = importdata('st_augustin_01_2_2km.csv', ';', 0);
travelled_distance = track(:,1);
elevation = track(:,2);
```

Simulate

```
velocity(1) = 1;
command = 0.01;
time(1) = 0;
energy(1) = 0;
overall_energy(1) = 0;
slope(length(track)) = 0;
for index=1:length(track)-1
    slope(index) = track(index+1,2)-track(index,2);
    F = ftp*command/velocity(index) -...
        0.5*1.225*velocity(index)^2 +...
        0.005*9.81*90+...
        9.81*90*slope(index);
    energy(index+1) = energy(index) + F;
    overall_energy(index+1) = overall_energy(index)+energy(index+1);
    time(index+1)=time(index) + 1/velocity(index);
    velocity(index+1) = velocity(index) + F/(90*velocity(index));

end
% energy(length(track)) = ftp*command/velocity(length(track)) -...
%          0.5*1.225*velocity(length(track))^2 +...
%          0.005*9.81*90+...
%          9.81*90*slope(length(track));
```

Result

```
figure(1);
hold on;
subplot(2,2,1);
plot(travelled_distance, energy);
title('energy over time');
```

```matlab
xlabel('travelled distance in m');
ylabel('used energy in ?');
subplot(2,2,2);
plot(travelled_distance, velocity);
title('velocity over time');
xlabel('travelled distance in m');
ylabel('velocity in m/s');
subplot(2,2,3);
plot(travelled_distance, slope);
title('slope over time');
xlabel('travelled distance in m');
ylabel('slope');

subplot(2,2,4);
plot(travelled_distance, overall_energy);
title('overall energy usage over time');
xlabel('travelled distance in m');
ylabel('energy used');
```
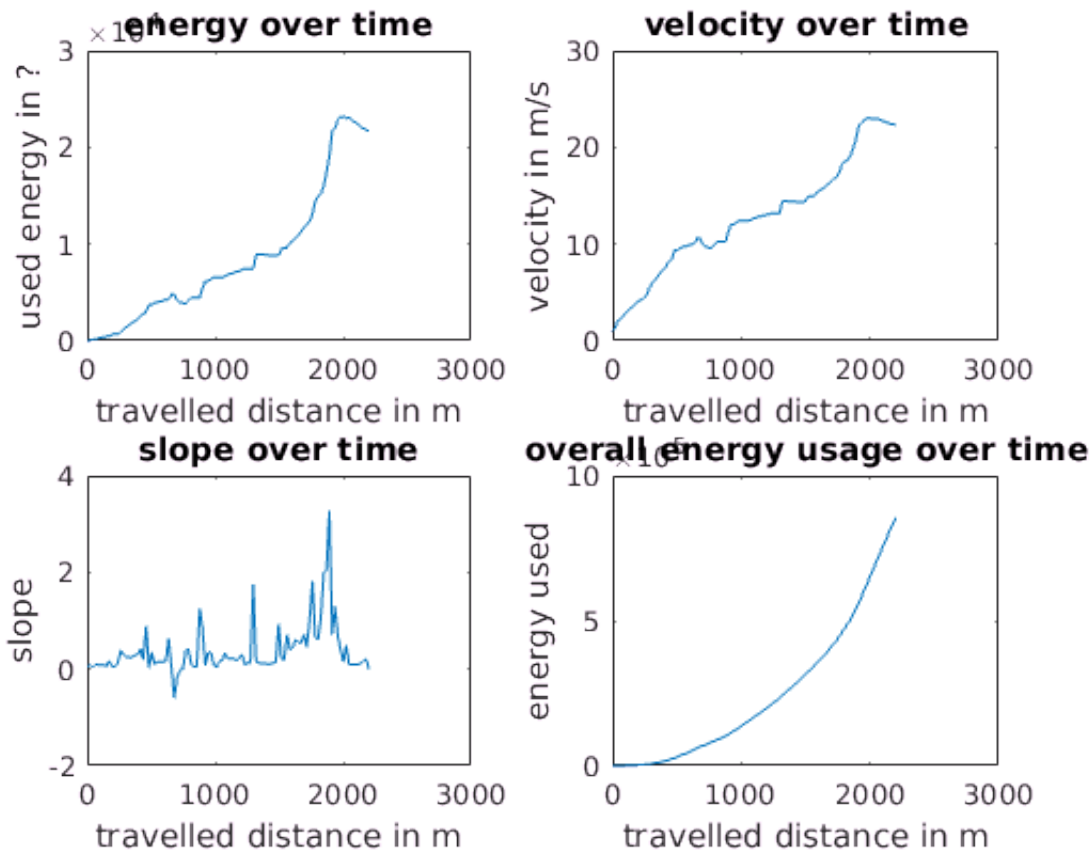


```matlab
n_velocity = velocity/norm(velocity);
n_slope = slope/norm(slope);
n_energy = energy/norm(energy);
n_overall_energy = overall_energy/norm(overall_energy);
figure(2);
hold on;
plot(travelled_distance, n_velocity);
plot(travelled_distance, n_slope);
plot(travelled_distance, n_energy);
plot(travelled_distance, n_overall_energy);
title('normalized comparison of data');
```

```
xlabel('travelled distance');
ylabel('normalized value');
legend('velocity','slope','energy','overall energy used');
```


normalized comparison of data