Tire Pressure Monitoring

4588745, 2384618

20.03.19

Contents

All of the matlab exercises can be run by running script.m.

Authors info

- 4588745 [Email]
- 2384618 [Email]

D1

Three Point estimates in Minutes

Task No.	Best Case	Likely Case	Worst Case
D1	10	25	40
D2	60	80	160
D3	80	150	200
D4	40	80	160
D5	30	70	150
D6	50	100	160
D7	40	70	120
D8	30	50	120
D9	40	60	100
D10	30	60	120
D11	40	60	100
D12*	40	70	100
D13*	20	40	60
D14*	30	50	70
Sum (min)	540	965	1660
Sum (hours)	9	16.08	27.67
Result (min)	1010		
Result (hours)	16.83		

D2

Files: findRadius.m

Given the formulas

$$R_{RR} = W + R_{RL}$$

and

$$\frac{V_x}{V_y} = \frac{R_x}{R_y}$$

and velocities for all the wheels during the entire duration of the test data one can determine the curve radiuses easily as such:

$$R_{RL} = \frac{W}{\frac{V_{RR}}{V_{RL}} - 1}$$

$$R_{RR} = \frac{R_{RL} \cdot V_{RR}}{V_{RL}}$$

$$R_{FR} = \frac{R_{RL} \cdot V_{FR}}{V_{RL}}$$

$$R_{FL} = \frac{R_{RL} \cdot V_{FL}}{V_{RL}}$$

Another approach is the pythagoras formula which we ended up using in our final version.

$$R_{RL} = \frac{W}{\frac{V_{RR}}{V_{RL}} - 1}$$

$$R_{RR} = R_{RL} + W$$

$$R_{FR} = \sqrt{W^2 + R_{RR}^2} \cdot signum(R_{RR})$$

$$R_{FL} = \sqrt{W^2 + R_{RL}^2} \cdot signum(R_{RR})$$

In MatLab:

```
w_m=zeros(size(tv));
w_m(:,1) = w;
R_RL=(w_m./((vrr./vrl)-1));
R_RR=R_RL+w;
% sign function ensures that all signs are the same
R_FR=(sqrt(w^2 + R_RR.^2)) .*sign(R_RR);
R_FL=(sqrt(w^2 + R_RL.^2)) .*sign(R_RR);
```

For the simulink simulation and to prevent errors, bad values like inf an NaN are replaced with 0.

```
% Filter bad values
R_RR(isnan(R_RR)|isinf(R_RR)) = 0.0;
R_RL(isnan(R_RL)|isinf(R_RL)) = 0.0;
R_FR(isnan(R_FR)|isinf(R_FR)) = 0.0;
R_FL(isnan(R_FL)|isinf(R_FL)) = 0.0;
```
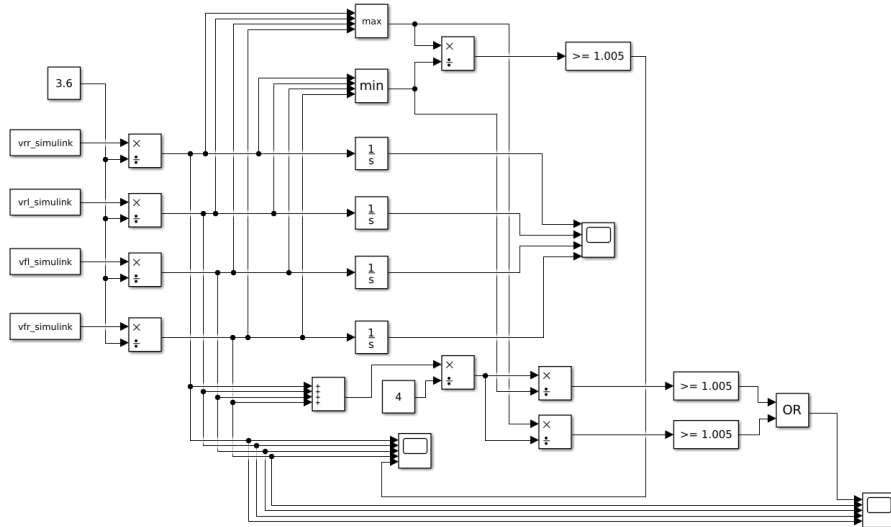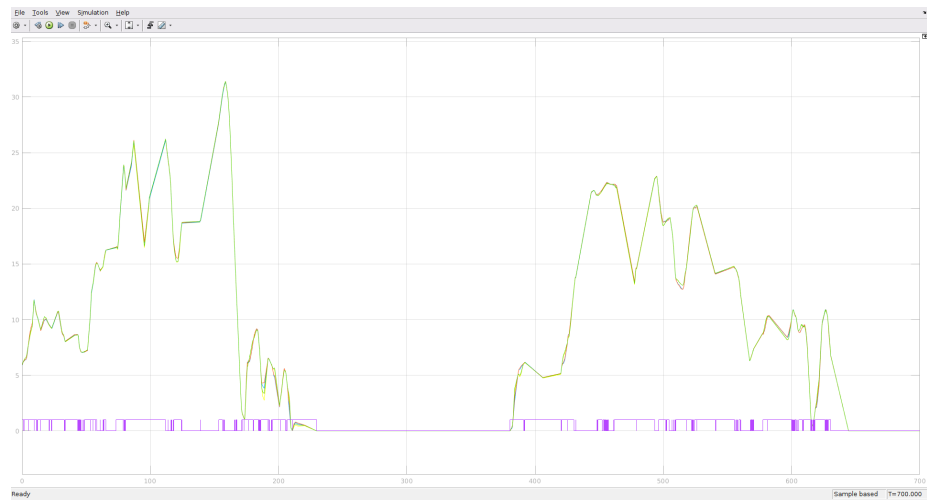
Result:
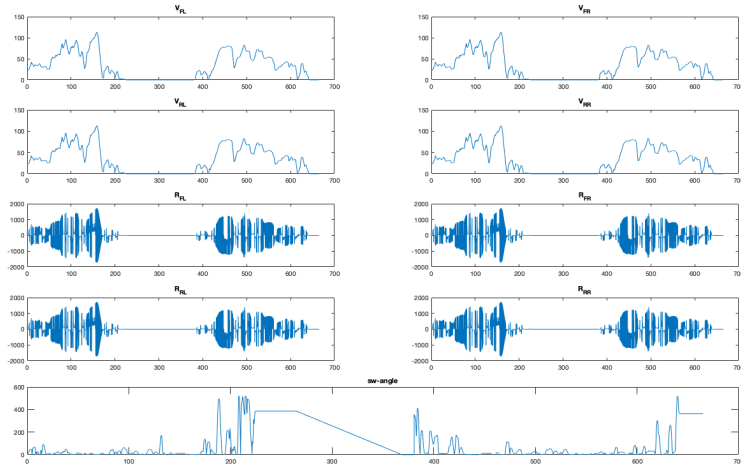
D3

Files: D3.slx

The simulink model for D3 is provided with the velocities of the respective wheels of the vehicle. It converts the values that are given in $\frac{km}{h}$ into $\frac{m}{s}$ by dividing by 3.6. As to the Requirement R2, we think there are at least two interpretations of the word "imbalance". Both are present in the simulink model.

First Interpretation

If any two tires differ by 0.5% in their velocities, a drop is detected. This method is implemented by taking minimum and maximum values of the four velocities at any time and dividing them to see if the two most exteme wheels differ by 0.5%. This produces the following result:


Second Interpretation

The maximum and minimum velocities are compared to the average of all velocities. If either the minimum or maximum velocity differs by 0.5% an imbalance is detected. This produces the following result:
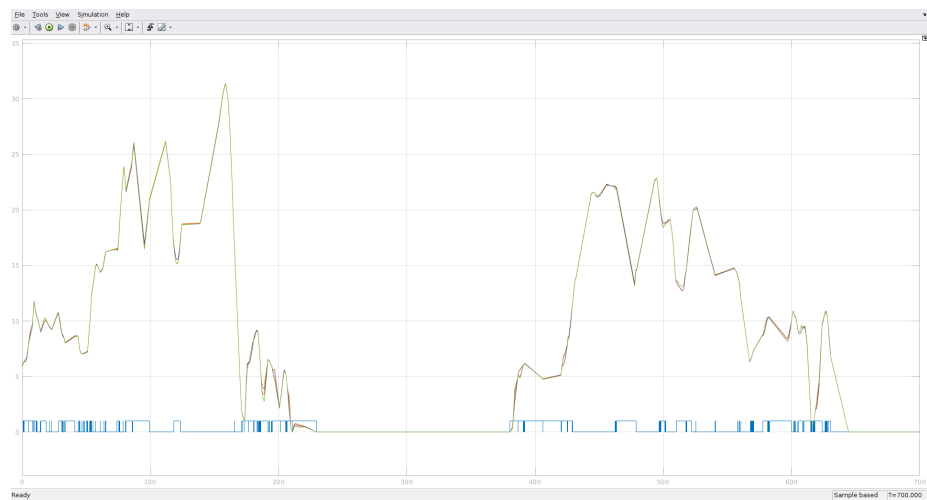
D4

If the difference between the maximum distance and the minimum distance is greater then 0.5% the system will detect tire pressure drop. This is essentially the same as D3 but with the travelled distance of the wheels instead of their velocities. A delay block is used to get a $\delta S = S(t) - S(t-10)$ which is the traveled speed in the last 10 time units. As in D3 there are two approaches to detecting an "imbalance" both of which can be seen below.
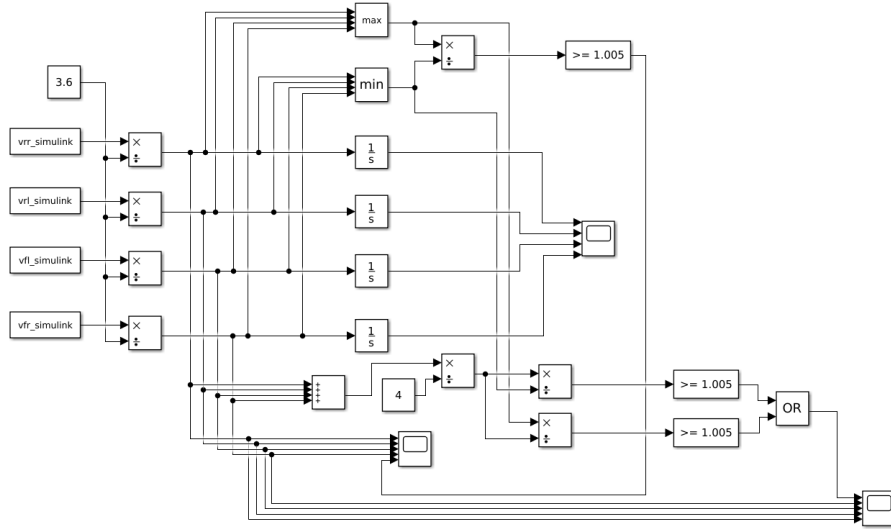
D5

The m and c parameters of the given random number generator are the upper and lower limit for the values generated. An initial value has to be chosen and functions as seed.

```
%randomNumberGenerator.m

function [dataset] = randomNumberGenerator(a, min, max, seed, n)
% Generater random test data for D5
% seed = X(0)
% n = length
% a,c,m according to formula of task D5

% Preallocate return array
dataset = zeros(1, n);
% Set seed
dataset(1) = seed;
```

```matlab
for i = 2:1:n
    dataset(i) = mod(round((a * dataset(i-1) + min)), round(max))
    %fprintf('(%d * %d + %d) mod %d\n',a,dataset(i-1),c,round(max))
end
```

By varying this seed, random data for the wheel velocities can be generated by only varying the seed.

```matlab
% Generate own tire velocity data for testing using randomNumberGenerator()
% Will generate data up to 80 km/h per wheel

vrl_rand = randomNumberGenerator(5, 3, 80, 1, 56093)
vrr_rand = randomNumberGenerator(5, 3, 80, 2, 56093)
vfl_rand = randomNumberGenerator(5, 3, 80, 3, 56093)
vfr_rand = randomNumberGenerator(5, 3, 80, 4, 56093)
```

D12

D13

Using the steering wheel angle and lateral acceleration provided it should be possible to remove bias in the wheels speeds caused by curve driving. However, seeing as there are almost always significant intervals of straight driving, it would be easier to just use these to measure (relatively) unbiased wheel speeds for pressure monitoring.

D14