SimView: Cruise Control

Finn Aakre Haugen, TechTeach. (finn@techteach.no)

04 January 2025

1 SimView simulator

The SimView simulator is available at

https://techteach.no/simview/cruise_control

2 The cruise control system

Figure 1 shows a block diagram of the car cruise control system.

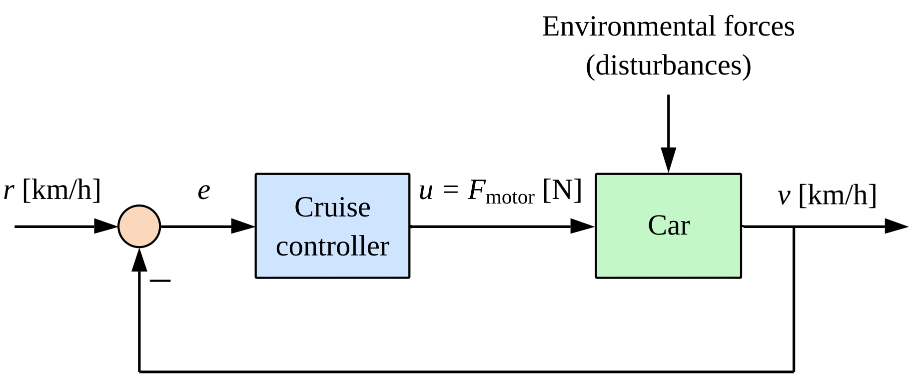

Figure 1: Cruise control system.

3 The car

3.1 Overall description of the car

The car model parameter values used in the simulator are for an electric car, namely a Tesla Y, Long Range All-Wheel Drive, see Figure 2. The mathematical model represents the "surge" motion, i.e. motion along the forward direction of the car.

Figure 2: Electric car being simulated: Tesla Y, Long Range All-Wheel Drive (my own car).

3.2 Variables and parameters

Figure 3 shows a schematic drawing of the car with variables and parameters, which are defined in Table 1.

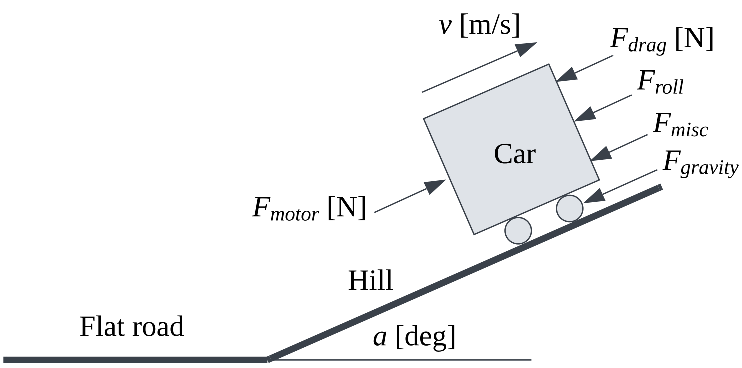

Figure 3: Schematic drawing of electric car with variables and parameters.

3.3 Mathematical model

A mathematical model of the car is based on Newton's 2nd Law, giving

$$v' = (F_{\text{motor}} - F_{\text{drag}} - F_{\text{roll}} - F_{\text{gravity}} - F_{\text{misc}})\,/m \tag{1}$$

Table 1: Electric car (Tesla Y): Variables and parameters.

Symbol	**Value (default)**	**Unit**	**Description**
$v = v_{\text{kmh}}/3.6$	-	m/s	Car speed
$v_{\text{kmph}} = v \cdot 3.6$	-	km/h	Car speed
F_{motor}	-	N	Force generated by motor
$F_{\text{motor, max}}$	10779	N	Max. motor force
F_{drag}	(2)	N	Drag force by air resistance
F_{roll}	(3)	N	Force due to roll friction
F_{misc}	80	N	Resistance force from miscellaneous sources
F_{gravity}	0	N	Force on car in speed direction due to gravity
m_{car}	1979	kg	Weight of car
m_{load}	150	kg	Weight of load, incl. one passenger and various things
$m = m_{\text{car}} + m_{\text{load}}$	2129	kg	Total weight of car + load
ρ	1.29	kg/m^3	Air density
A	2.5	m^2	Front area
C_d	0.24	1	Drag coefficient of air resistance force.
C_r	0.01	1	Roll force resistance coefficient. (Typical value for cars.)
a	0	degrees	Slope angle of hill
g	9.81	m/s^2	Gravity acceleration

where:

$$F_{\text{drag}} = \frac{1}{2}\rho C_d A v^2 \tag{2}$$

$$F_{\text{roll}} = C_r \cdot mg \cdot \cos(\text{a}) \tag{3}$$

$$F_{\text{gravity}} = mg \cdot \sin(\text{a}) \tag{4}$$

$$F_{\text{misc}} = 80\text{N} \tag{5}$$

Miscellaneous force, F_{misc}

F_{misc} represents miscellaneous resistance forces, e.g. friction in the drive system of the motor, and inaccuracies in F_{drag}, F_{roll}, and F_{gravity} in (1). An assumed constant F_{misc}can be calculated from the steady state version of (1):

$$F_{\text{misc}} = F_{\text{motor}} - F_{\text{drag}} - F_{\text{roll}} - F_{\text{gravity}} \tag{6}$$

where F_{motor} can be calculated from the "Highway - Mild Weather" test data on

https://ev-database.org/car/1619/Tesla-Model-Y-Long-Range-Dual-Motor where speed is 110 km/h, and distance driven using the battery capacity of 75 kW (full battery) is 415 km. The forces terms F_{drag}, F_{roll}, and F_{gravity} in (6) can be calculated from information given in the present section. Further details of the calculations are not shown here.

Forces at negative speed

In (2) it is assumed that the car speed is always positive. If we take into account that the speed may also be negative, the force terms F_{drag}, F_{roll}, and F_{roll} in (1) should be modeled as follows.

$$F_{\text{drag}} = \text{sign}(v) \cdot \frac{1}{2}\rho C_d A v^2 \tag{7}$$

$$F_{\text{roll}} = \text{sign}(v) \cdot C_r \cdot mg \cdot \cos(\text{a}) \tag{8}$$

$$F_{\text{misc}} = \text{sign}(v) \cdot 80\text{N} \tag{9}$$

where $\text{sign}(v)$ is the signum function of v:

$$\text{sign}(v) = 1 \text{ if } v \geqslant 0; \ \text{sign}(v) = -1 \text{ if } v < 0 \tag{10}$$

Neglected time lag

There is certainly some time lag between motor force and its impact on the motor speed. In the simulator this lag is represented with a time delay of 0.5 s, but this lag/delay will have a negligible impact on the responses of the cruise control system if the controller is properly tuned.

4 Controller

4.1 Controller tuning

The controller is assumed a PI controller. It can be tuned using the SIMC method Skogestad (2004) as described below.

You can assume that the car is at zero speed, which is the most critical operating point regarding stability of the control system because there are no damping forces, and should therefore be used as operating point for controller tuning. The forces which do not influence the process transfer function, i.e. transfer function from motor force ($u = F_{\text{motor}}$) to speed (v) can be neglected, i.e. set to zero, in the controller tuning. The process model (1) in the "zero speed" operating point, and neglecting other forces than F_{motor}, is simply

$$v' = F_{\text{motor}}/m = u/m = K_i u \tag{11}$$

where the $u = F_{\text{motor}}$ is the control signal manipulating the car. (11) is the model of an integrator, with integrator gain

$$K_i = \frac{1}{m} \tag{12}$$

In the SIMC method you must specify the closed loop time constant, t_{cc}.

With known K_i and specified t_{cc}, the SIMC PI settings become

$$K_c = \frac{1}{K_i t_{cc}} = \frac{m}{t_{cc}} \tag{13}$$

$$T_i = 2t_{cc} \tag{14}$$

In the simulator, you can adjust t_{cc}.

5 Smooth speed reference tracking

If the speed reference is changed abruptly, for example as a step from 50 to 60 km/h, there will be an abrupt change in the control signal (motor force). This abrupt change of the motor force may be unfortunate. Two methods to avoid the abrupt change are described in the following, namely

- Reference smoothing
- Reducing the weight of the reference in the P term

Both methods can be used in the simulator. I recommend the reference smoothing method as I think it is more flexible, and easier to understand.

Note that the ability of the cruise control system to compensate for disturbances, like driving uphill or downhill, is the same for both methods, i.e. the ability is independent of the methods.

5.1 Reference smoothing

In the this method the reference is smoothed with a smoothing filter, see Figure 4. In the

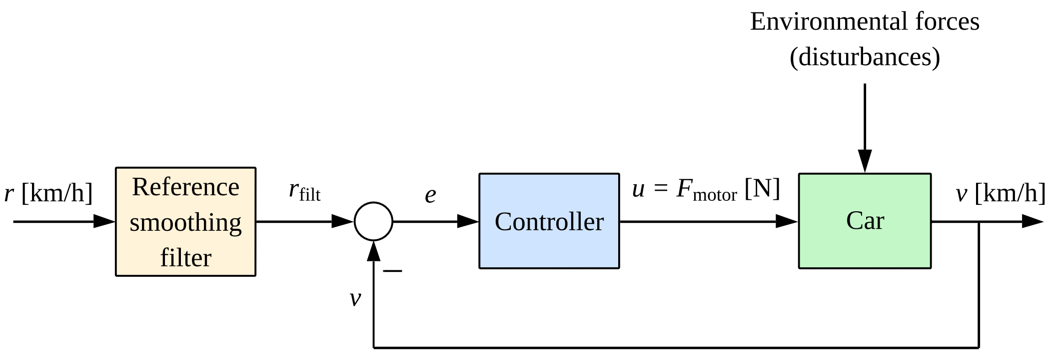

Figure 4: Reference smoothing filter.

simulator, the smoothing filter is in terms of two first order EWMA[1] filters in series, resulting in a second order filter.[2] The (combined) filter is tuned with the filter time constant, which is applied to each of the two filters.

[1] Exponentially Weighed Moving Average

[2] I found from simulations that using one single EWMA filter did not smooth sufficiently, but using two EWMA filters in series works fine.

5.2 Reducing the weight of the reference in the P term

In the simulator, the P term of the PI controller is

$$u_p = K_c\,(w_p r - v) \tag{15}$$

where w_p is the reference weight in the P term. By default,

$$w_p = 1 \tag{16}$$

With (16) a step in r will cause a step in u_p, and therefore a step in the total control signal, u, since u_p is an additive term in u.

To reduce the step in u_p, the weight can be set to a value less than 1. With

$$w_p = 0 \tag{17}$$

there will be no step change in u_p.

Note that with $w_p = 0$, the reference tracking is generally slower than with w_p =1, but ability of the control system to compensate for disturbance is independent of the value of w_p (this is because the speed, v, in (15) is the same whatever value of w_p).

References

Skogestad, S. (2004), 'Simple analytic rules for model reduction and PID controller tuning', *Modeling, Identification and Control* **25**(2), 85–120.