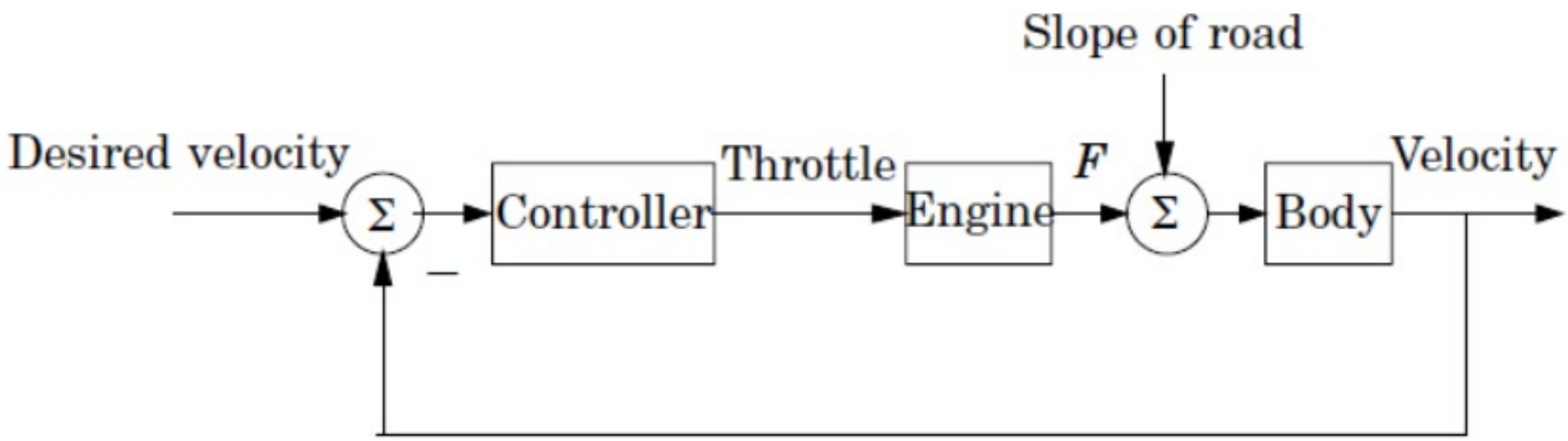
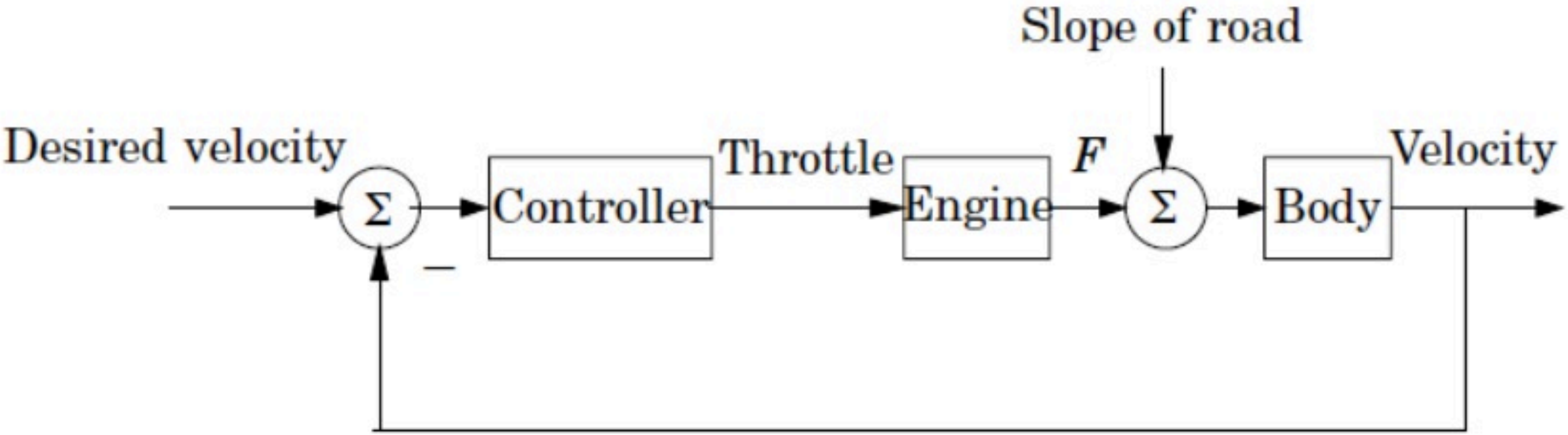
ME528 Control Systems Design

Research Project – Part 1

Automotive Cruise Control Design Analysis





This is the *functional* block diagram of a cruise control scenario for a car. The objective of this project is to turn this diagram into mathematically formal control system block diagrams, incorporating seven different controllers, the car dynamics, any disturbance that is operative, and an appropriate feedback loop. The secondary objective is then to explore the performance of the cruise controllers numerically, then to select the best one and assess it for stability.

We start with the car dynamics

The car's rate of change of momentum is governed by Newton's Second Law and is defined by $m\ddot{x}$. The combined effect of car and rolling resistance constitutes the damping of the vehicle and can be defined by $c\dot{x}$.

The force required to propel the car comes from the engine and transmission and is transmitted to the road by means of the friction vector between the tyres and the surface of the road. We think of this as one collective propulsive force vector pointing forwards, and we can denote it here by F.

The effect of an incline is a retarding force defined as $mgsin\theta$. We can consider the incline as limited to around 20° so that $sin\theta \to \theta$. Physically speaking the engine has to do extra work against gravity in order to lift the car as it travels up the incline, so this force joins the inertia and damping forces as those which have to be overcome by the propulsive force. It is effectively 'just an mg term' but one which varies with inclination, through θ, and as inclination can vary with time then this force can potentially vary with time too.

From these propositions we can write down the equation of motion for the car negotiating the incline:

$$m\ddot{x} + c\dot{x} + mgsin\theta = F(t) \tag{1}$$

Therefore, we simply rearrange this to get the propulsive force and the gravitational disturbance force on the right-hand side:

$$m\ddot{x} + c\dot{x} = F(t) - mgsin\theta. \tag{2}$$

In the case of a typical four door saloon in top gear the ratio $\frac{c}{m}$ is commonly around 0.02. Writing $\frac{F(t)}{m} = u(t)$ and dividing through by m leads to:

$$\ddot{x} + 0.02\dot{x} = u(t) - g\theta \tag{3}$$

Cruise control dynamic analyses sometimes involve techniques such as controller saturation and normalisation, for example limiting the control $u(t)$ to $0 \leq u(t) \leq 1$. You may want to display $u(t)$ as you do your calculations and come to your own assumptions about a practical range for it.

The '*control torque*' is u(t) and the '*gradient disturbance*' is $g\theta$.

The next questions are: (i) how do we apply the control effort u(t) and (ii) where does it come from?

A conventional industrial approach to providing robust control of a system such as this is to use Proportional Integral Derivative (PID) control. A PID controller is represented like this mathematically:

$$u(t) = k_p(v_r - v(t)) + k_i \int_0^t \big(v_r - v(t)\big)dt + k_d \frac{d(v_r - v(t))}{dt} \tag{4}$$

where k_p is the gain of the proportional part of the controller, k_i is the gain of the integral part, and k_d is the gain of the derivative part. Furthermore, v_r is the desired velocity of the car, and v(t) is the actual velocity of the car. Obviously $v = \dot{x}$.

The proportional part **P** relates to the current value of the (desired – actual value) quantity (this being the error). The integral part **I** accounts for past values of the error and integrates them over time such that when the error term diminishes to zero the integral term stops growing. The derivative part **D** deals with the future trend of the error based on its current rate of change. So, the more rapid the rate of change the more effective is the control.

So, the PID controller needs to know continuously the speed of the car, and from that, and a knowledge of the required velocity (the set-point velocity) and the three gain constants it can generate the control effort u(t).

From this control effort we can subtract the disturbance $g\theta$, and the difference between them is what drives the car dynamics. In order to find the car dynamics we have to take the equation for the car, setting the quantity $u(t) - g\theta = E(t)$. From this we get the transfer function for the car (i.e. the 'body' in the functional diagram above):

$$\frac{V(s)}{E(s)} = \frac{1}{s+0.02} \qquad (5)$$

The explanation above provides enough information and interpretation of that information for a mathematical block diagram to be drawn. This is effectively where *you* start.

Instructions:

1. Starting with the PID controller in equation (4) and use it to create seven separate block diagrams using standard *Simulink* blocks, each of which should have an input on the left hand side of:

$$e(s) = V_r(s) - V(s)$$

which then creates an output of the following seven forms:

$$u(s) = k_p e(s)$$

$$u(s) = k_i \int e(s)ds$$

$$u(s) = k_d \frac{de(s)}{ds}$$

$$u(s) = k_p e(s) + k_i \int e(s) ds$$

$$u(s) = k_p e(s) + k_d \frac{de(s)}{ds}$$

$$u(s) = k_i \int e(s) ds + k_d \frac{de(s)}{ds}$$

$$u(s) = k_p e(s) + k_i \int e(s) ds + k_d \frac{de(s)}{ds}$$

noting that this is the s-domain equivalent of different combinations of terms from equation (4) giving controllers which may be defined as P, I, D, PI, PD, ID, and PID. You will need to include a feedback line to get the error $e(s)$ in each diagram but as yet you will not have a source for that feedback line, so just leave the other end unconnected at the moment.

2. Your controller output $u(s)$ should then be added/subtracted to/from the gradient disturbance and then the resulting signal, $E(s)$, drives the car 'body', for which the transfer function is given in equation (5). Proceed to extend your block diagrams to include the above mathematics and complete them by connecting the feedback line to the appropriate point to provide the $V(s)$ signal back to the input. Note that the outputs of your block diagrams should be $V(s)$ and these can go straight to a scope, output file, or spreadsheet, whichever you prefer. Then, save your seven block diagrams!

3. Start by assigning unity to each of $V_r(s)$, k_p, k_i, k_d and $g\theta$, and then run the simulation. Explore the effects of all five quantities, for each of the seven controllers, altering each quantity separately, and using orders of magnitude to investigate their individual effects (i.e. 1, 10, 100, also 0.001, 0.01, 0.1). Record what happens in each case, preferably by saving a copy of the output velocity against time in each case.

4. Now put a 'scope or output file onto the diagrams to monitor u(s) and record what this quantity does against time in cases of interest that emerge from task 3 above. You may have to re-run those cases again to get $u(s)$. You should examine and discuss a minimum of three cases for all seven controllers to get down to workable and practical scenarios. Compare them critically and decide which of the seven controllers gives the most robust performance against time.

5. Take your optimal controller and then check it for stability, using Routh-Hurwitz, root-locus, and Bode, to assess its performance for a range of different conditions.

6. At this point return to the lecture and tutorial material supplied on MyPlace for ME528 as far as Bode and write a critique of how your project has applied these topics, stating clearly which have been applied explicitly, which have been useful as a general background to the project, and which have not entered into the project as yet. In the case of the latter provide some examples of what has not been used, and discuss what you would have to do to the project problem statement to be able to introduce it.

7. Write up your work so far so that you have a formal record of all the work done for Part 1 of the project. You should not produce more than 20 A4 pages for Part 1.