Automotive Control Systems (EE 5812/ MEEM 5812)
Project #3
Optimal Idle Speed Control of a Spark Ignition Engine
Submit: Simulink Controller as slx-files
Program for generating control gains as an m-file
The one-page, short answers to my questions as a pdf-file

The idle speed controller for a spark ignition engine generates control signals by using the measured idle speed (in RPM) and the measured (or computed) intake manifold pressure. The controller generates two control inputs: the spark timing (changes torque on the next combustion event) and the throttle bypass valve (changes air flow into the intake manifold). The goal is to design an optimal integral LQR controller that will drive the engine speed to 600 RPM. To design this system, you will first have to evaluate the equilibrium with a nominal external load of 340 W. In the controller block, the throttle position that yields the desired speed is given with the spark position set at 20 degrees after MBT spark timing (this is also done in the controller block). You should then get the equilibrium states from the simulation. Next, linearize the plant model around this equilibrium state. Note that the equations of the nonlinear model are given in the notes presented in class. If there is a discrepancy between the Simulink model and the equations in the notes, use the equations in the Simulink model. There is very little change in the volumetric efficiency when the idle speed is close to 600 RPM. Therefore, to make the linearization simpler, the volumetric efficiency is set to a constant 0.7 in the model. This saves a lot of computation and does not significantly change the results.

I have supplied an engine simulation as a Simulink diagram. This will appear in the assignment as a file "IdleSpeedSim." This file has a controller subsystem that has two inputs: Engine Speed (RPM) and Intake Manifold Pressure (Pa). The output of this controller subsystem is the change in spark timing (from MBT spark timing) and the throttle setting. These outputs are currently connected to constant blocks so that you can evaluate the equilibrium states, but you should change this and install your controller within this block. Note that you cannot add any additional information into the control system. Also, there are two Matlab scripts, called "IdleSpeedSymParameters" and "IdleSpeedAutoGrading" which contains the parameters that are used in this simulation. You should download these scripts and locate them in the folder with the downloaded IdleSpeedSim and IdleSpeedController.

1. To design this system, you will first have to evaluate the equilibrium with an external load of 340 W. In the controller block, the throttle position that yields the desired speed is given with the spark position set at 20 degrees after MBT spark timing (this is also done in the controller block). You should then get the equilibrium states from the simulation. Include these equilibrium states in your one-page question answer sheet.

2. Next, linearize the plant model around this equilibrium state. Note that you can ignore the time delay when performing the linearization. Note also that the equations of the nonlinear model are given in the notes presented in class. If there is a discrepancy between the Simulink model and the equations in the notes, use the equations in the Simulink model. Include your linearized model in your one-page question answer sheet.

3. Design an integral LQR controller that includes the integral of the change in spark advance from the set point of 20 degrees after MBT. Note that the integral LQR will operate on the variations of the engine speed and intake manifold pressure from the nominal values, and then generate the variations of the spark timing and the throttle input from the nominal values. But, to implement this controller, you then have to compute the actual throttle value and the actual spark timing. When designing your controller, set the weight the square of the airflow variation to 1 (you can always set one value in a cost function to 1). Use a diagonal control weighting matrix and set the weight on the spark advance variation by noting the difference in the specifications for the two control inputs. The contribution from both inputs should be equal when the inputs are at their maximum values. Vary the weights on the engine speed, the integral of the engine speed, and the integral of spark timing error until you achieve good performance. Good performance is defined as meeting the specifications:

- Maximum RPM error when reducing the load should be less than 20 RPM, i. e., the RPM cannot be greater than 620
- Minimum RPM error when increasing the load should be greater than -30 RPM, i. e., the RPM cannot be less than 570.
- Settling time for the RPM is less than 1 second.
- Maximum Overshoot when reducing the load should be less than 1. Note that Overshoot in this case is defined to be the error below 600 when the load is decreased.
- Maximum Overshoot when increasing the load should be less than 6. Note that Overshoot in this case is defined to be the error above 600 when the load is increased.
- Maximum Steady State RPM error should be less than 0.01.
- The Throttle value should remain in the range from 0 to 0.12.
- The Spark timing should remain in the range from less 0° (with respect to MBT) to -40° (with respect to MBT)
- Maximum absolute value of the Steady State spark timing error should be less than 0.0001°, i. e., the steady state spark timing should be between -20.01° and -19.99°

Maximum spark timing should be greater than -10°

4. Increase the state weighing matrix by a factor of 2 (keeping everything else the same) and see what happens.
5. Increase the control weighting matrix by a factor of 10 (keeping everything else the same) and see what happens.

Upload your controller; Upload your program for computing the controller gains; and Upload your one page of answers to the questions:

1. What are the equilibrium states for the idle speed control system?

2. What is the final linearized plant model?

3. What happens when the state weighting matrix is increased by a factor of 2. What happens to the state trajectories? What happens to the control trajectories?

4. What happens when the control weighting matrix is increased by a factor of 10. What happens to the state trajectories? What happens to the control trajectories?

Comments:

Each student should do this project individually and no student should share Simulink diagrams or code or results with other students. Note that I will be running your results through a similarity checker. If you have shared results and it is detected, you will fail this course! So, please don't share code. But, feel free to ask your instructor or the TAs questions.