ENEL428 Project 3 – Operating on Cruise Control

Aydin Arik and Sam Leichter

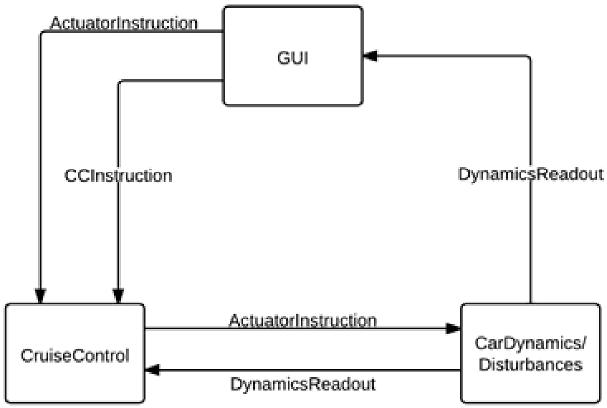

Figure 1. Designed message passing system. Message objects are passed in the directions of the arrows from one thread to another. The main thread is not shown.

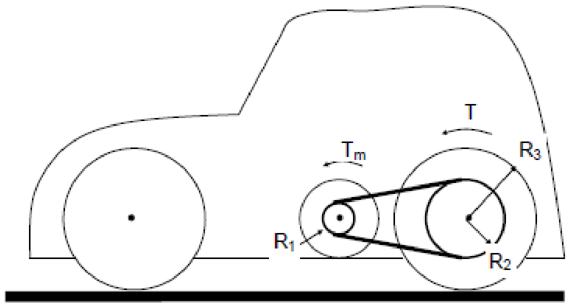

Figure 2. Diagram of car. Note, motor provides propulsion force. Drag and rolling resistance opposes the propulsion force. A braking torque at the wheels will also create an opposing force (as expected).

Message passing was used to communicate between threads. A high level design is shown in Figure 1. There are three threads in total excluding the main thread.

The GUI can be considered as the 'interface' of the car with which the driver interacts. The GUI allows the driver to provide throttle, brake and ignition inputs; each of these *instructions* are passed to CruiseControl as an ActuatorInstruction. Additionally, the driver is able to manipulate the cruise control settings; these *instructions* are passed to CruiseControl as a CCInstruction.

If the cruise control and engine are both on, a new ActuationInstruction is created by the PID controller. However, if at any time there is a driver related ActuationInstruction created by the GUI, this will take precedence over any PID generated ActuationInstruction.

Finally, as shown, discrete-time DynamicReadouts are provided to both the GUI for display and to CruiseControl in case future PID generated ActuationInstructions need to be created. Part of our car model (which is found in Car Dynamics) is shown in Figure 2.

General Notes:
1. CarDynamics 'has a' Disturbance class to generate random hill and wind gust disturbances.
2. PID only adjusts the throttle, as specified. As a result, we have decided only to generate positive hill inclines (uphill). Downhill situations will only show the cruise controller is unable to properly reduce the speed of the car because it doesn't have control over the car's brakes.
3. Hill disturbances use a uniform distribution. Wind disturbances have a Gaussian distribution.
4. The sliders send throttle/ brake information to CruiseControl/ CarDynamics when released.
5. Actual throttle and brake levels (from CarDynamics) are shown below the sliders.
6. We are using the Trapezium Rule to approximate the speed of the car.
7. We are approximating drag using the previous speed of the car.
8. Integral control of the PID controller is zeroed regularly to avoid undesirable effects.