COMPSYS723 ASSIGNNMENT TWO REPORT GROUP 1

Joshua Gudsell, Sakayan Sitsabesan

Department of Electrical and Computer Engineering
University of Auckland, Auckland, New Zealand

Abstract

This report covers the specifications and design of a cruise control system using Esterel V5 for assignment two of COMPSYS 723 in 2018.

1. Introduction

This report covers group ten's implementation of a cruise control system for assignment two of COMPSYS 723 in 2018. The cruise control system takes inputs from the car and regulates the speed of the car as required. The Cruise controller is responsible for holding a cars speed at a given threshold as long as certain conditions are met which will be discussed in detail later in the report. Following the introduction this report contains information about the specification of the cruise control system (including specification diagrams) and how the specifications were mapped to the Esterel programming language.

2. Specification

The input-output behaviour of the cruise control system is shown in Figure 1. The system reads user inputs from the driver through buttons and the acceleration/braking pedals of the car. The system provides outputs to the car in the form of three signals which will be read by different parts of the car.

The inputs to the cruise controller are six buttons, the acceleration/brake pedals and the current speed of the car. Four of the input buttons which are used to change the state of the cruise control systems consist of: On, Off, Resume and Set. The On and Off buttons are self-explanatory in the fact that they allow the user to turn the cruise control system on and off. The Resume button allows the cruise controller to be re-enabled to an on state if previously the brake was pressed while the cruise controller was on. The set button allows the user to set the current speed of the car to the cruise controllers target speed. There are two other buttons used for input which are the QuickAccel and QuickDecel buttons. The aforementioned buttons allow the user to increase/decrease the cruise controller target speed by a fixed amount. The input from the cars acceleration/brake pedals are used to change the cruise controllers state if they are pressed while the cruise controller is on and is regulating the speed. The final input to the system is the current car speed which is used to set the cruise controllers target speed when the cruise controller is turned on along with ensuring the system keeps the car within reasonable thresholds of minimum and maximum speed.

The outputs from the cruise controller are the CruiseSpeed, ThrottleCmd and CruiseState signals. The CruiseSpeed output is used to give the driver feedback on the currently set target speed for the cruise controller. The ThrottleCmd output is passed to the engine and is either set by the cruise controller or the acceleration pedal of the car. The CruiseState output would be used to inform the driver of the current state of the cruise control system. CruiseState has four possible outputs as follows: ON, OFF, STDBY and DISABLE.

The top-level context diagram shown in Figure 1 is further refined into a lower level context diagram shown in Figure 2. The lower level context diagram shows the split of the cruise controller into a control unit and a cruise speed management unit. The control unit is responsible for managing the CruiseState of the cruise controller between the four possible states. The cruise speed management unit is responsible for correctly setting the ThrottleCmd output while the cruise controller is in the On state. The cruise speed management unit is also responsible for the setting and changing of the target speed of the cruise controller.

The Control unit and Cruise Speed Management Unit each have a finite state machine (FSM) governing their actions shown in Figure 3. The Control Unit FSM shows the logic involved with changing between the different possible CruiseStates. The Cruise Speed Management Unit FSM shows the logic relating to the setting of the CruiseSpeed of the cruise control system.

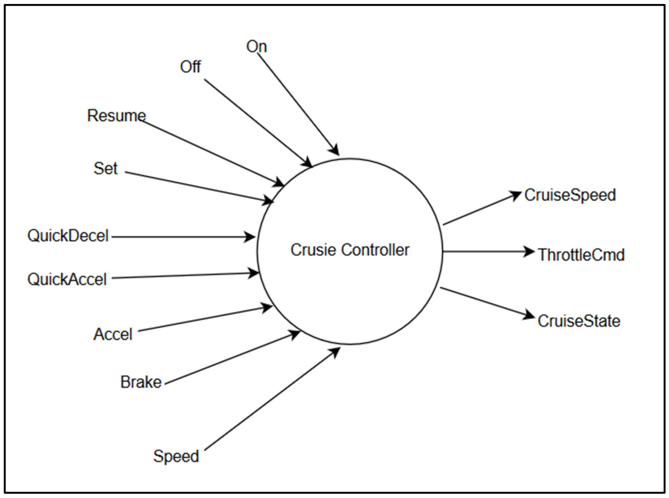

Figure 1:Top-level context diagram

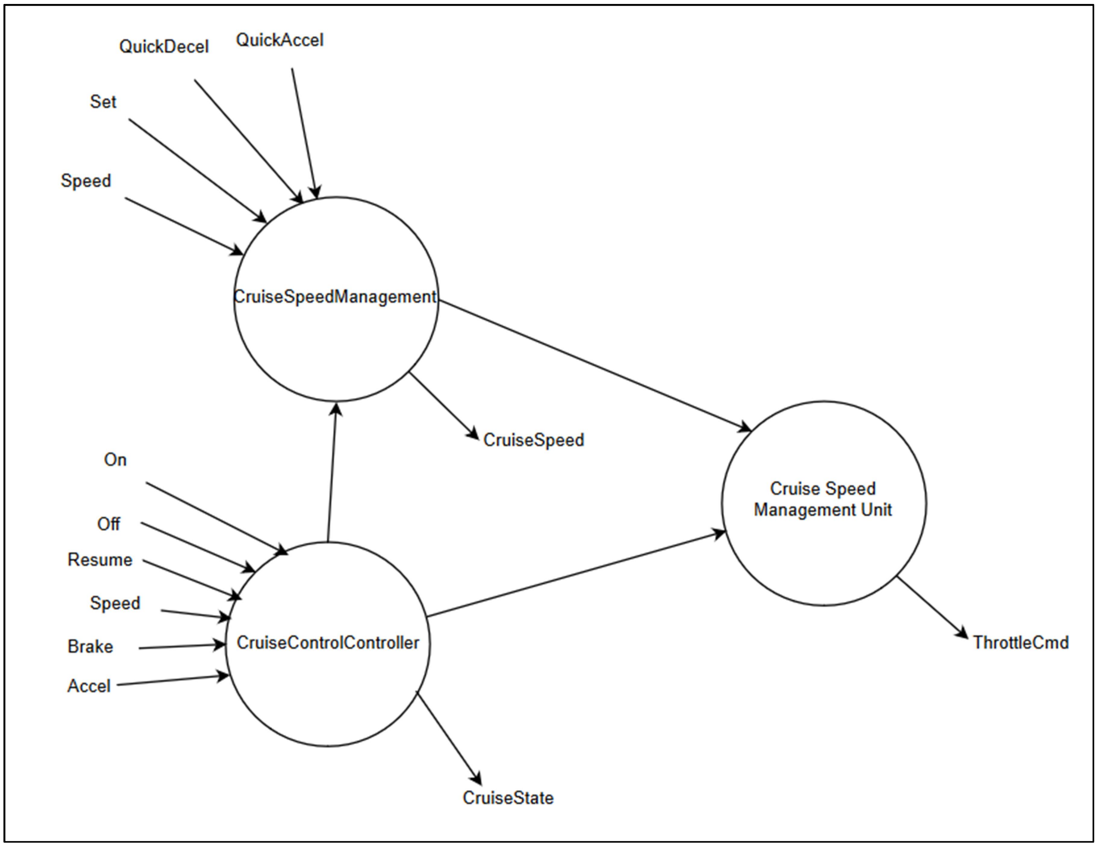

Figure 2: Lower-level context diagram

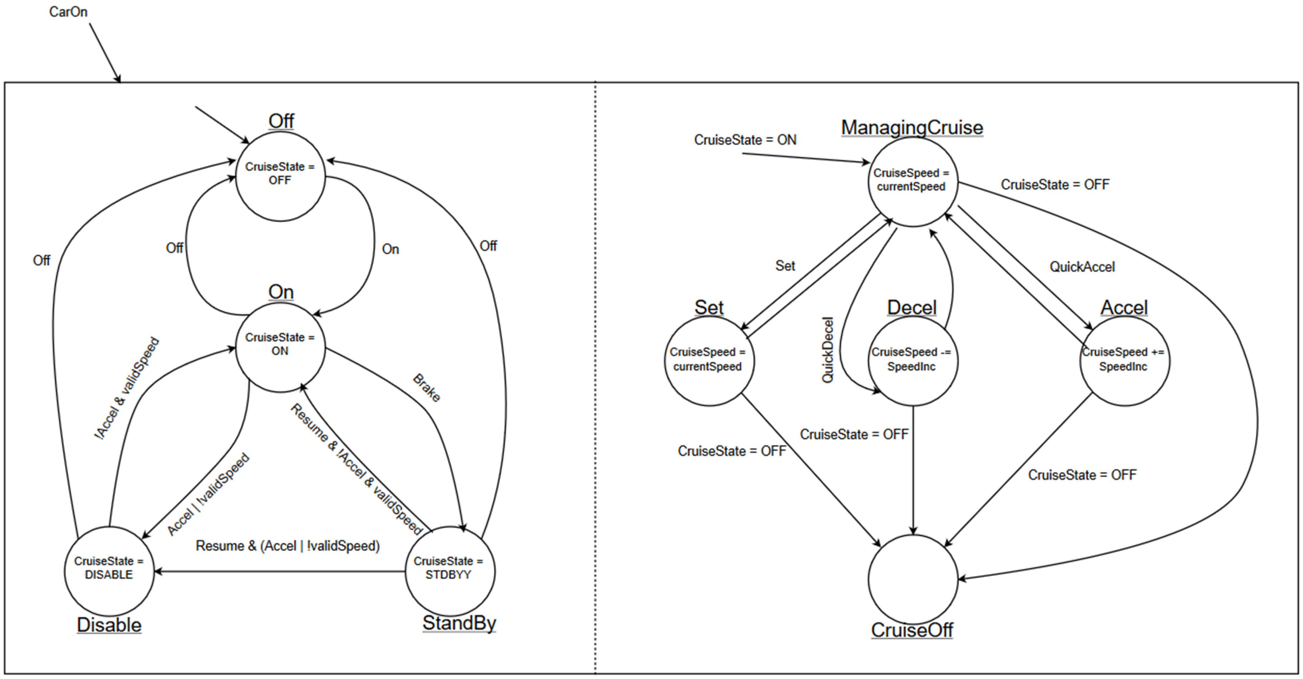

Figure 3: FSM diagrams: CruiseControlController (left) and CruiseSpeedManagement (right)

3. Design in Esterel

The design of the cruise controller is directly based off the specifications in section two. The Esterel program is created with three logic modules along with a top level module. The logic modules consist of: the CruiseControlController module, the CruiseSpeedManagement module and the ThrottleManagement module. The modules are combined to provide overall cruise control system using a top-level module called CruiseControl.

- The CruiseControlController Module: This module executes an FSM that receives inputs relating to the state of the cruise controller (On, Off, Resume, Accel, Brake and Speed) and outputs the current CruiseState of the cruise controller. The logic of the FSM executed is to the left in Figure 3.
- The CruiseSpeedManagement Module: This Module executes an FSM which outputs the CruiseSpeed that the system should be trying to match while the cruise controller is on. This is based upon inputs from the driver such as the Set, QuickDecel and QuickAccel. The CruiseSpeedManagement module is entered when the CruiseState is set to On within the CruiseControlController module.
- The ThrottleManagement module: This is a simple module responsible for selecting whether the ThrottleCmd output is driven by the accelerator pedal or the cruise controller.
- The CruiseControl module: This is the top level Esterel module that runs the CruiseControlController module, CruiseSpeedManagement module and the ThrottleManagement module in parallel. This module is responsible for connecting the systems inputs to the modules where they are required, along with interconnecting the modules as required and collating the outputs from the modules they are returned from to the top level outputs.

3.1. Causality and Data Handling

To ensure causality between modules in Esterel we have accessed the *pre* of signals that are shared between modules. Accessing signals and values using *pre* ensures that causal cycles are removed.

4. Conclusions

This report compliments the design of a cruise control system completed for assignment two of COMPSYS 723. The report details the specifications of the cruise controller along with how it was designed using Esterel.