UPPSALA
UNIVERSITET

Model Based Design of Embedded Systems
1DT059
Project

Simulink & Matlab

Vivek Vivian
Jatin Anand
Group - 2

October 28, 2020

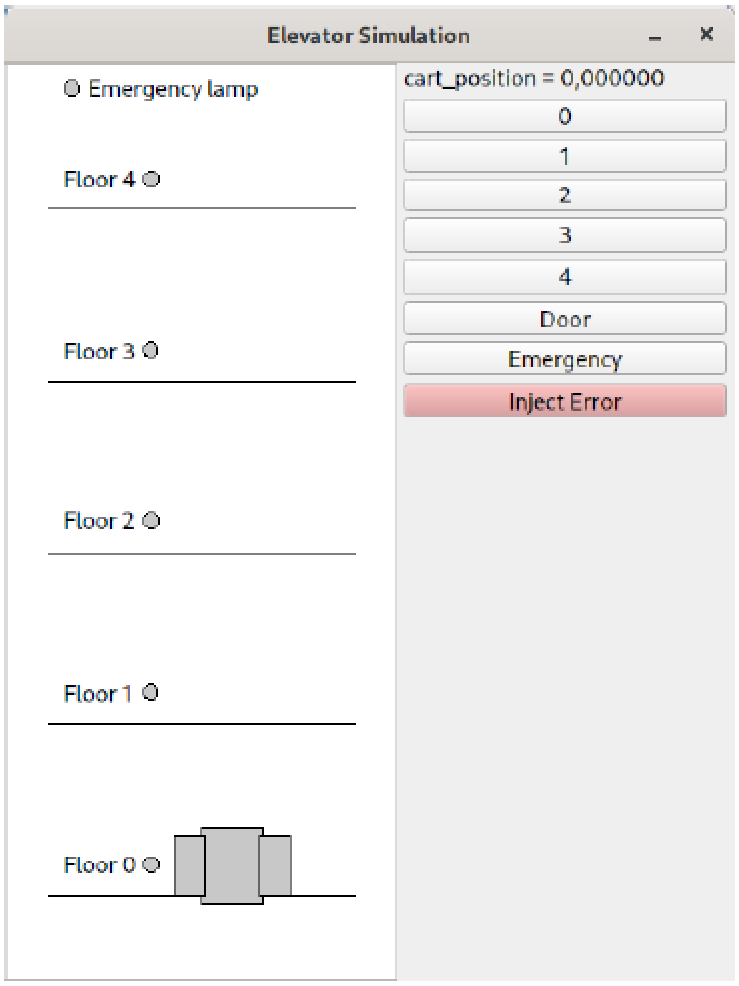
Abstract

The Design and Testing of an Elevator controller has been made using Matlab and Simulink stateflow. The elevator has been designed using a set of user requirements. The stateflow model of the elevator was then created and simulated and tested. Code from this stateflow was generated and it was integrated with the GUI (simulated) Elevator system, programmed in Qt.

Design and Specification

The goal of the design was to design and test an embedded software to control an elevator system using simulation. The elevator can be controlled using the GUI as shown below.
The tasks involved in this project is to design a control algorithm to control the (simulated) elevator, to develop a Simulink model that mimics the (simulated) elevator system, to generate and deploy code from the control algorithm, and testing and tuning of the design. The Design was based on a set of user requirements:

- Once the elevator reaches a floor the doors open, it remains open for at least 5 seconds. The doors close once the elevator start to move
- Whenever a button for floor n has been pressed the elevator eventually stops at that floor and the door opens
- When the emergency button for the elevator has been pressed, the elevator stops and it turns on the emergency lamp. When the emergency button is pressed again the elevator resumes its normal operation from where it stopped before
- A door button is added which when pressed opens the door if the elevator is at a floor
- The elevator stops at all the locations provided in the API which is floor 0, 1, 2, 3, and 4
- The elevator runs smoothly with a slow start and stop at each floor
- When the inject error button is pressed the elevator stops and the door shuts, until the inject error button is pressed again



cart_position = 0,000000

Figure 1: Elevator GUI

Simulink Model

The simulink model used in this project is a stateflow chart. The contents of the state chart are provided below.

Table 1: Input and Output Ports

Input Ports	Output Ports
Floor 0	queue(0)
Floor 1	queue(1)
Floor 2	queue(2)
Floor 3	queue(3)
Floor 4	queue(4)
Emergency	Emergency Lamp
Door	Door Lamp
Position	Speed
Tic	

Each queue() is an output for floor lamps.

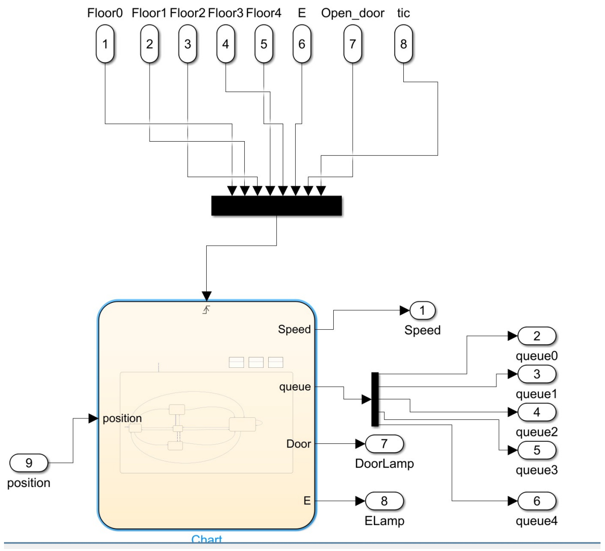


Figure 2: Simulink Model

The model consists of 5 different states as mentioned below:

- Initial State - The state flow starts from this state. It remains in this state when the Speed and Direction is 0
- Up - When the direction changes to 1, the elevator is designed to move

upwards and it enters the UP state. Here since we have to move upwards speed is set to a positive value

- Down - When the direction changes to -1, the elevator is designed to move downwards and it enters the DOWN state. Here the Speed is set to a negative value
- Door - When the elevator reached a particular floor, the doors open and it remains open for at least 5 seconds and then it closes
- Emergency - When the emergency button in the GUI is pressed, the elevator enters this state and it stays here until the emergency button is pressed again

Control Algorithm

The decision algorithm checks the direction of the movement of the elevator and stops at every floor it passes by, if the button for that floor is pressed. Same works for upwards and downwards direction. The Speed is set to 0.2 for upwards (*dir=1*) and -0.2 for downwards (*dir=-1*) direction. When the position of the elevator reaches the desired floor, it stops the elevator and opens the door for at least 5 seconds. The elevator waits at the current floor unless the button for another floor is pressed. If the button for current floor is pressed while the elevator is at a stop, it opens the door for at least 5 seconds again.
The door button opens the door for at least 5 seconds if the elevator is not moving.
Emergency button stops all the functions of the elevator and turns the emergency lamp to 'on' state. It resumes the elevator back to its state from where it had stopped, if the emergency button is pressed again.

There are three MATLAB functions that help the controller work smoothly.
floor_check checks for which direction the elevator should move in according to where its coming from and what floor buttons are pressed.
current_floor checks when the elevator is at halt at a floor and the button for the same floor is pressed, it opens the door for that floor.
Error_check checks if there is an error injected into the controller, i.e. if the controller behaves in a way it is not supposed to, then the Speed is turned to zero.

Additional Changes

Elevator Timing & Simulation

- Upon completion and simulation it was observed that the timing of the elevator was not synchronized with the Qt simulation and had few changes to be made. The error was noticed as the fixed step size in the Simulink settings were set to a sample time of 0.1 seconds while the Qt simulation was set to a sample time of 1 sec. This was causing the simulation to run in an asynchronous manner. The changes have been made and the elevator simulation is smooth and the elevator now satisfies all the requirements.

- The *sleep(1)* which was being used, was causing instability in the simulation. This has been corrected by replacing the *sleep(1)* with *usleep(10000)*, which makes the simulation synchronized.

- The position and speed of the elevator was being read at multiple times in the *main.c* and this has also been corrected.

The elevator now works with better accuracy in terms of calibrating the position and stopping at the precise locations as provided in the API.