Assignment 3: Requirements and Verification

1DT059: Model-based Design of Embedded Software
2020

October 6, 2020

In this assignment you produce Simulink/Stateflow models and text. All models (and code) should be well-structured and readable, and use named constants wherever appropriate. Your solutions to each exercise exercise should be implemented in a .slx-file and/or .pdf-file that is named after the exercise. Include your name at the top of all submitted files.

Assignments are to be solved by students individually. You can (and are encourage to) discuss ideas and concepts with fellow students, but it is absolutely forbidden to share or copy (even parts of) solutions, lines of code, or similar.

Problem 1 Bridge Crossing*:*
(25p) Solve the Bridge crossing problem using Simulink Design Verifier

> A group of four people must cross a bridge. It is dark, and the bridge is fragile. To cross it, one needs to bring a torch, and the bridge can carry at most two persons simultaneously. The group of people have only one torch, meaning that the torch must follow the persons that cross the bridge back and forth across the bridge. The four persons walk at different speeds; it takes them 1, 2, 5, and 10 minutes to cross the bridge, respectively. Since the bridge carries at most two persons, at most two persons can cross in one "move", which takes time equal to that which is needed by the slower person crossing. E.g., if the first move is that the 2-minute person and the 5-minute person cross, then that move takes 5 minutes. Thereafter one of them must go back with the torch before any other person can cross, etc. What is the shortest time they need to cross the bridge?

Make a structured Simulink/Stateflow, using which SLDV can find the solution to this problem. This means that your model should **not** provide any hint about how the group should cross (i.e., in which order the persons should move across). It should only model the above rules of the problem. *Finding* a successful shortest-time plan which gets all persons across the bridge should be done by the Property-proving engine in Simulink Design Verifier. A possible stragety is for the model to contain an integer "parameter" n, and "ask" SLDV whether there is a way to cross the bridge in at most n minutes, and then repeat this "question" for different values of n.

Your solution should include

- A model, which SLDV analyzes to find a shortest time, and a model which allows SLDV to deduce that ther is no shorter-time plan. A description of how the information provided by SLDV is interpreted as a "schedule" for crossing the bridge in a shortest time.
- An explanation of your model and how it captures the rules of the problem.

Problem 2 Requirements for Elevator (50p)*:*
In this problem, the context is the elevator controller of Assignment 2b. For the elevator system (consisting of controller and simple elevator model) that you constructed in Assignment 2b, you should formulate requirements. Each Requirement (numbered R1 – R6 below) should be formulated in a way so that it can be checked using Simulink Design Verifier (SLDV); often this can be done by constructing a Simulink/Stateflow monitor with an output signal that can be sent to a `Proof Obligation` or an `Assertion` block.

a) (30p) The requirements are as follows.

R1: The doors are never open when the elevator is not at some floor. This requirement can be formulated in terms of the signal to the lamp which denotes that doors are open, and the signal *position* from the elevator.

R2: When doors are opened, they remain open for at least 5 seconds.

R3: The elevator stops at floor A only if the button for floor A has been pressed previously.

R4: The elevator stops at floor A only if the button for floor A has been pressed previously and after the last time that the elevator stopped at floor A.

R5: If the button at floor A is pressed before the button at floor B, then the elevator stops at most once at floor B before moving to floor A. For this requirement, you can choose concrete values (floors) for A and B as you like. Your model may or may not satisfy requirement R3 (depending on the algorithm you use for moving between floors).

R6: After a button-push for any floor, the elevator arrives within d time units to that floor.

Here, requirements R3 and R4 are formulated in a slightly ambiguous way. Take care to formulate them in such a way that they are satisfied by your elevator controller, also in corner cases (e.g., that the button is pressed when already at the floor). For requirement **R6**, you should make d as small as possible (how small depends, of course, on your elevator design). Also here, some care is needed for formalizing the requirements: there can be corner cases where its satisfaction is hindered or made complex.

b) (20p) For each of the above requirements, use Simulink Design Verifier (SLDV) to attempt to prove the requirement (e.g. using “Prove Property”). In some cases, SLDV will be able to verify the requirement. In some cases, it can fail. For failing cases, provide a succinct explanation of why SLDV failed. If it failed because your model does not satisfy the requirement then provide a counterexample (i.e., an execution scenario in which it is falsified; this should not happen for R6 since you can choose an appropriate value for d). If it failed because it SLDV was not “smart enough”, suggest how a designer should interpret the failure (regarding whether the requirement is satisfied or not, or needs further inspection).

Submission

Solutions (all files) to this assignment are to be submitted via the Student Portal by **Wednesday, October 14, 2020**