UPPSALA
UNIVERSITET

Programming Embedded Systems Hand-in 2

Group 10

Nusrat Hossain Fredrik Tåquist

December 27, 2019

1 Lab Question 3 - The Safety Module

Here follows a description of our solution to the environment assumptions.

2. The elevator moves at a maximum speed of 50 cm/s. At that speed it takes 20 ms to go 1 cm. Since the safety module updates every 10 ms, this will not provide enough resolution for accurate measurements. Instead, we measure how long it takes to go 10 cm, which would be 200 ms $\pm$ 20 ms. So to check this, whenever a distance of 10 cm have been covered, check that the elapsed time isn't greater than 180 ms.

```
check(timeSinceLastSpeedUpdate >= 180, "env2");
```

In addition, we must also make sure that each time a position change occurs the change isn't more that 1 cm. Since it should take at most 20 ms to go 50 cm, if it takes less than 10 ms the elevator is definitely going much too fast.

```
check(ABS(pos - prev_pos) == 1, "env2");
```

This is done in a separate step, while calculating the time between each centimeter (used in safety requirement 4 and 6).

3. If the ground floor is put at 0cm in an absolute coordinate system, the second floor is at 400cm and the third floor at 800cm (the at-floor sensor reports a floor with a threshold of +-0.5cm) Solution:

$$AT\_FLOOR \Rightarrow pos \in FLOOR\_POSITION$$

where

$$FLOOR\_POSITION = \{0, 1, 399, 400, 401, 799, 800\}$$

The position tracker is accurate to within ± 1 cm. Translated into C-code using de Morgan's rule:

```
check(!AT_FLOOR || ((pos <= FLOOR_1_UB) ||
                    (pos >= FLOOR_2_LB && pos <= FLOOR_2_UB) ||
                    (pos >= FLOOR_3_LB)), "env3");
```

where FLOOR_x_UB is the upper bound of floor x and FLOOR_x_LB is the lower bound.

4. If the elevator has arrived at a floor, the doors must open at some point before it leaves again (we don't want any passenger to get stuck).

Solution: If the elevator is at a floor and the motor is stopped, record that this has happened. If the doors open at any point while the elevator is stationary at a floor, record that event as well. Once the floor starts moving again, check that the doors have opened, and then reset the variables keeping track of the recorded events.

```
check(doors_have_opened, "env4");
```

Here are our solutions for the safety requirements.

3. The elevator may not pass the end positions, that is, go through the roof or the floor
   Solution: This solution imposes another precondition - the position markers must continue beyond the top and ground floors. If that condition holds then the following is the requirement

$$pos \leq 800 \wedge pos \geq 0$$

```
check((pos >= FLOOR_1 && pos <= FLOOR_3), "req3");
```

4. A moving elevator halts only if the stop button is pressed or the elevator has arrived at a floor
   Solution:

$$\begin{aligned}\neg ELEVATOR\_MOVING \Rightarrow\ & STOP\_PRESSED \vee \\ & \vee STOP\_RECENTLY\_RELEASED \vee \\ & \vee AT\_FLOOR\end{aligned}$$

   In the implementation we assume that the elevator is moving if its position has changed within 1 second (as the minimum speed is 3 cm/s). We also keep track of how long it has been since the stop button was released, since it can take a while for the motor to start running again. 1 second should be ample time. Translated using de Morgan's rule

```
check(stop_recently_pressed || AT_FLOOR ||
      timeBetweenCentimeters * portTICK_RATE_MS < 1000, "req4");
```

5. Once the elevator has stopped at a floor, it will wait for at least 1s before it continues to another floor
   Solution:

$$\neg MOTOR\_STOPPED \Rightarrow \neg idle \vee idle\_time \geq 1 \text{ s}$$

   where

$$idle = \begin{cases} 1 & \text{if elevator has stopped at a floor} \\ 0 & \text{otherwise} \end{cases}$$

   and *idle_time* is the time that the elevator has been at the floor.

```
check(MOTOR_STOPPED || !idle || idle_time >= MAX_IDLE, "req5");
```

6. If the elevator approaches the ground or top floors it must move slowly.
   Solution:

$$\begin{aligned}&(MOTOR\_UPWARD \wedge timeBetweenCentimeters < 300 \Rightarrow pos > 2) \wedge \\ &\wedge (MOTOR\_DOWNWARD \wedge timeBetweenCentimeters < 300 \Rightarrow pos < 798)\end{aligned}$$

Here we measure the time it takes to move 1 cm. As the elevator should be moving slowly, measuring the time it takes to move 10 cm takes far too long. Also, the elevator will be slowing down at this point. Since the minimum speed (which the elevator should be close to at the end points) is 3 cm/s, it takes 333 ms to move 1 cm. When translated using de Morgan's rules this becomes

```
check(timeBetweenCentimeters >= 300 ||
      ((!MOTOR_DOWNWARD || pos > FLOOR_1_UB + 1) &&
       (!MOTOR_UPWARD   || pos < FLOOR_3_LB - 1)),
      "req6");
```