Protocol Validation

X_400117

Homework Assignment:
Movable Patient Support for an MRI Scanner

Team members:
Group 17

Name	Student ID	Email
Chao Zhang	2619800	yellywong7@gmail.com
Siying Zhang	2611811	siying.zhangcn@gmail.com

Resubmitted on
3 January 2020

Content

Introduction

MRI scanners are used to acquire images of patient issue. The scanner has a limited scan range, thus the patient has to be moved into and out of the scanner during a scan procedure. To accomplish a precise and (semi-)automatic positioning, a Patient Support Platform is used. This report aims to design a set of controllers to for a small distributed and/or embedded system of the Moveable Patient Support Platform (MPSP) (see fig.1). The patient system that no harm can ever be done to a patient or to the equipment. In this assignment, it is a strict requirement that at least three separate controllers are used: one for inputs from the console (see fig.2), one for inputs from the sensors, and one for outputs to the motors and the brakes. We assume that three controllers need to communicate with each other and those channels are secure, for instance, no messages are lost.

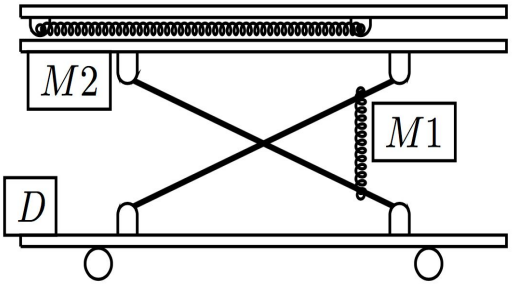

Fig.1: Movable Patient Support Platform (MPSP)

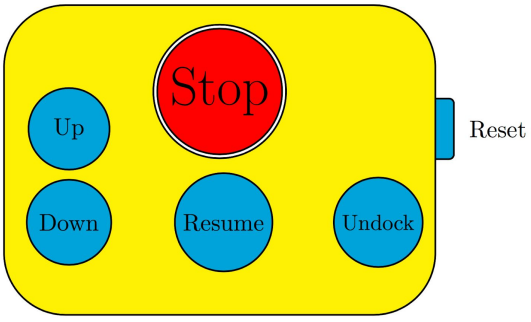

Fig.2: The console of MPSP

In our design, the user operate the console to send inputs to the controllers, the communication between three controllers including Controller, Sensor controller and Motor controller by exchanging data. If there is one variable changes, all of them have to update the new data. (see Section 4.A)

The core idea of control system consists of two primary modes: Emergency mode and Normal mode. The exchange between these two modes by pressing button Stop and Resume. After press Stop button, the system comes into emergency mode, but the bed can be moved manually. Pressing Resume button, the system will change from emergency mode into normal mode which all functions work normally. There are four conditions on normal mode: docked and calibrated, undocked and calibrated, docked and uncalibrated, undocked and uncalibrated. (see Section 4.B)

There are five positions that the bed can be moved in vertical direction under docked and calibrated condition: um(Uppermost), ab(above standard height), sh(standard height), bl(below standard height), dm(lowermost). There are four positions that the bed can be moved in vertical direction Under uncalibrated condition[1]: um(Uppermost), ab(above standard height), bl(below standard height), dm(lowermost). The bed will stop automatically when it reaches to uppermost or lowermost position.

[1] the sh (standard height) will be ignored under uncalibrated condition

Normally, the bed changed vertical positions based on the command from Controller:
Move up: The system will first release vertical brake then bed can be moved up after the Up button pressed.
Stop moving up ：The bed will stop moving up then applied vertical brake after the Up button released.
Move down: The system will first release horizontal brake then the bed can be moved down when the Down button pressed.
Stop moving down: The bed will stop moving down then applied horizontal brake after the Down button released.

In our design, there are four position boundaries: Uppermost, Lowermost, Leftmost and Rightmost.

- When the bed is in the uppermost position, the Up button is pressed, the bed will not do any actions.
- When the bed is in the lowermost position, the Down button is pressed, the bed will not do any actions.
- When the bed is in the leftmost position, the Up button is pressed, the bed will not do any actions.
- When the bed is in the rightmost position, the Down button is pressed, the bed will not move right.

The special part of our design is that Undock button on console. In our design, we assume that the bed disconnects from the sensor when you press Undock button. When the bed is disconnected from the sensor at first, press Undock button, the bed will connect to sensor again. Under undocked condition and in emergency mode, the bed only can move up or move down, the bed is not allowed to move left or right under these conditions. After pressing Resume button under undocked condition, the system will be recalibrated, the bed could be on above standard height or below stand height position. If emergency mode happens, the user can operate the bed manually, the bed can be moved innermost or outermost position.

Design

1.Global requirements

In our design, the first step is to identify all behavioural requirements or properties on the behaviour . We formulate 11 properties in our design.

RQ1: When the MPSP is disconnected from the scanner, the bed must always be in the rightmost position.

RQ2: When disconnected from the scanner, the horizontal brake must always be applied.

RQ3: Horizontal movement is only allowed when the MPSP is docked.

RQ4: When a motor is on, the corresponding brake must not be used.

RQ5: In order to protect the motors, the motors will not attempt to push the bed beyond its boundaries: Leftmost, Rightmost, Uppermost, Lowermost position.

RQ6: When the MPSM is docked and calibrated, the bed will stop at the standard height.

RQ7: The MPSP must be calibrated by setting the standard height when the system is docked.

RQ8:

a. When the MPSP is docked and calibrated and the bed is at standard height, after the down button pressed, the bed moves rightward, until the horizontal detector indicates that the bed is in the rightmost position (and thus completely outside the scanner).
b. By releasing and pressing the down button again the bed will subsequently move downwards.

RQ9: In emergency mode, the horizontal brake must be released, to allow medical staff to manually drag the patient outside the scanner.

RQ10: When the MPSP is docked and at standard height and the up button is released, the bed moves leftward.(combined with **RQ7**)

RQ11: Deadlock free

2.Interactions

There are three separate controllers are used in our design: console, motor and sensor. Console and sensor control the inputs, motor and brakes control the outputs. And these three controllers can communicate with each other. On Table 1, we identify and list the actions which the controllers communicate with the outside world.

Controllers	**Communications**		**Actions**
1. Console	1a	Reading that the button is pressed	pressUp, pressDown, pressReset, pressResume, pressStop, pressUndock
	1b	Reading that the up button is released	releaseUp
	1c	Reading that the down button is released	releaseDown
2.Motor	2a	Applying a horizontal or vertical brake	aHbrake, aVbrake
	2b	Releasing a horizontal or vertical brake	rHbrake, rVbrake
	2c	Turning a motor1 on and making a movement	mUp, mDown
	2d	Turning a motor1 or motor2 off	m1off, m2off
	2e	Turning a motor2 on and making a leftward movement	mLeft, mRight
3.Sensor	3a	Detecting the bed position	lmost (leftmost), rmost (rightmost), umost (uppermost), dmost (lowermost), shpos (standard height position), abovesh (above standard height), belowsh (below standard height)
	3b	Detecting the platform has been calibrated	DoCali
	3c	undocking system when the system is docked	Undock

	3d	Docking system when the system is undocked	Dock

Table 1: The interaction of the MPSP controllers

3.Translation between global requirements and interactions

The annotation in the bracket is the explanation of the actions on the code.

1. (RQ1): Before action Undock taking place, the bed must be at rmost position(Rightmost).

2. (RQ2): Between an action aHbrake (applyHorizontalBrake) and action unlockDock, no action pressUndock is allowed.

3. (RQ3): Between an action Undock and action mLeft (motorLeft) or mRight (motorRight), the action Dock must be applied.

4. (RQ4):
   a. Before an action mUp (motorUp) or mDown (mortorDown), an action rVbrake (releaseVerticalBrake) must be applied.
   b. After an action m1off (motor1off) , an action rVbrake (releaseVerticalBrake) must be applied immediately.
   c. Between an action aHbrake (applyHorizontalBrake) an action mLeft (motorLeft) or mRight (motorRight) taking place, an action rHbrake (releaseHorizontalBrake) must be applied.
   d. Between an action m2off (motor2off) and action aHbrake (applyHorizontalBrake) , an action rHbrake (releaseHorizontalBrake) must be applied immediately.

5. (RQ5):
   a.Between an action rmost (Rightmost) and an action mRight (motorRight), the action pressResume or action mLeft (motorLeft) must be happened.
   b.Between an action lmost (Leftmost) and an action mLeft (motorLeft), the action pressResume or action mRight (motorRight) must be happened.
   c.Between an action umost (Uppermost) and an action mUp (motorUp), the action mDown (motorDown) must be happened.
   d.Between an action dmost (Lowermost) and an action mDown (motorDown), the action mUp (motorUp) must be happened

6. (RQ6): After an action shpos (Standard Height position), no action mDown (motorDown) and action mUp (motorUp) or action mLeft (motorLeft) and action mRight (motorRight) allowed.

7. (RQ7):
   a.Between an action DoCali (calibrated) and shpos (Standard Height position), the UnCali (uncalibrated) should not be applied.

b.When the system is UnCali (uncalibrated) and Undock (undocked), the action shpos (Standard Height position) cannot happen if there is not an Undock (undocked) action.

8. (RQ8):
   a. Before an action mDown (motorDown) and action rmost (Rightmost), an action shpos (StandardHeightposition) must be applied.
   b. Between an action releaseDown and action mDown (motorDown), an action rmost (Rightmost) or an action shpos (StandardHeightposition) must be applied.

9. (RQ9): Between an action pressStop and an action Emer (Emergency), an action rHbrake (releaseHorizontalBrake) must be applied.

10. (RQ10):
    a.Between an action Undock (undocked) and action mLeft (motorLeft), the action Dock (docked) must be happened.
    b. Between a non Dock action and mLeft action, there should be either a pressReset action or a shpos (StandardHeightposition) action applied. (Combined with **RQ6**)

11. (RQ11): No deadlock.

4.Architecture of the control system

In order to preset the control system clearly, we divide the architecture of the control system into : high-level design architecture (see Fig.3) and low-level design architecture (see Fig.4).

A. High level architecture

The high level architecture can give us a basic picture of the control system. First, the Controller receives commands from the user. Then the Controller sends commands to Motor how to change the position of bed. The sensor detects what position of the bed currently. The communication between three controllers including Controller, Sensor controller and Motor controller by sending and receiving data.

- Sensor controller will receive input of sensor about the positions of the bed and then provide information about the status of the bed to Motor controller and Controller.
- The Motor controller will operate the motors and brakes based on the instructions received from the Controller and Sensor controller.
- The Controller will receive the human input from console and the corresponding message to Sensor controller and Motor controller.

The data variables include up, m1, um, left, m2, lm, cali, dock, dm, ab, bl, sh, rm, down, right, emer:

- To be specific, the emer parameter represents the mode including normal mode and emergency mode. When the value of emer is true, the system is in the emergency mode. When the value of emer is false, the system is in the normal mode.
- For m1 and m2 represent vertical and horizontal motor, respectively.
- The dock and cali represent whether the system is docked and calibrated or not.
- For rm, lm, um, dm, sh, bl, ab parameters, they represent the current position of the bed, at rightmost, at leftmost, at uppermost, at lowermost, at standard height, at below standard height, at above standard height position.
- In terms of left, right, down, up, these variable means the direction of movement.

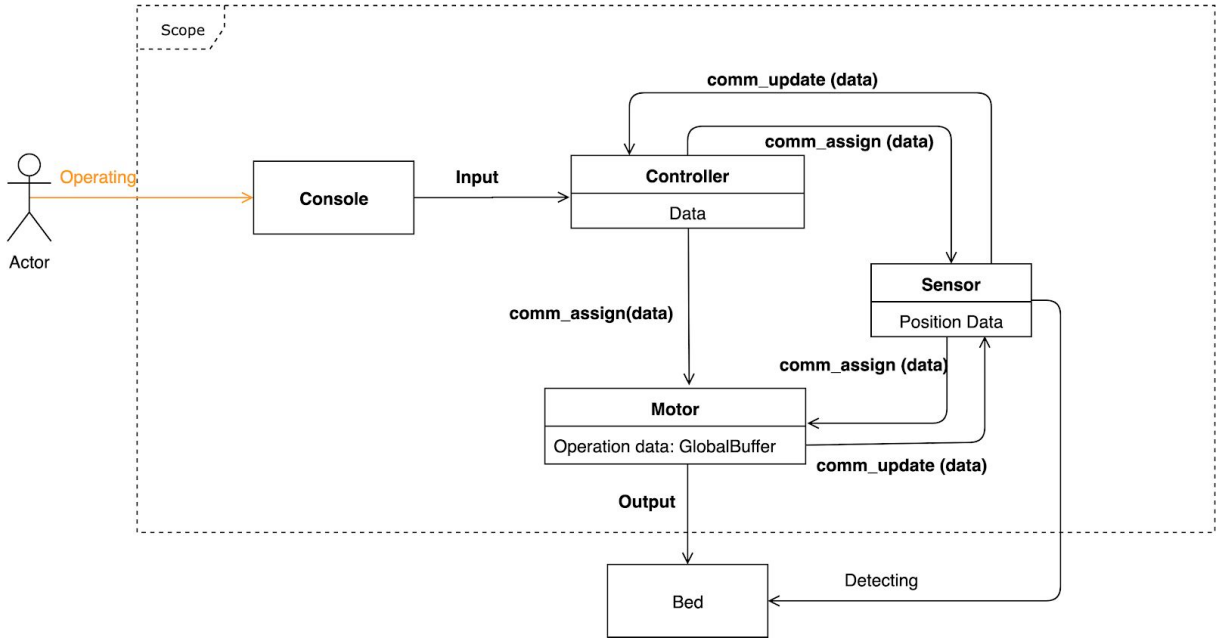

Fig.3 High-level design architecture

B.Low level architecture

On the low level architecture, we will present details of how the control system operates. In emergency mode, the motor2 is off and the horizontal brake is released, and the bed is only responds to pressing the Resume button. If you want to change the bed to normal model, press the Resume button and apply horizontal brake. Then all functions operate normally. Pressing the Stop button is the only way to changing into emergency mode.

There are four conditions on normal mode. Pressing Undock button, the system will from docked condition change to undocked condition. Pressing Reset button, the system will from calibrated condition to uncalibrated condition.

Undocked condition:
The system will be in undocked condition when the bed is in the rightmost position and the Undock button is pressed
Recover docked condition:

After Undock button pressed when the system is undocked, the system will back to docked condition

Press Reset button:

- When the system is undocked, the Reset button is pressed, then the system will be in uncalibrated condition.
- When the system is docked and calibrated and the bed is not in the standard height, the Reset button is pressed, the system will do not have any following reactions.
- When the system is docked and uncalibrated and the bed is in the correct height, which means is not in the um, ab, bl and dm, the Reset button is pressed, the system will be calibrated and the position of the bed is in the standard height position.

Under docked and calibrated condition, the bed could move toward left or right only if it reaches the standard height. After press Undock button, the control system will become undocked calibrated situation, motor1 will be turned on while pressing button Up or Down, the bed will move upward or downward on a steady speed without beyond the uppermost and lowermost position. The bed will stop moving if you release the button or it reaches the uppermost or lowermost position. Under undock situation and uncalibrated, you are allowed to press Undock button only if the bed is at the rightmost position. Then horizontal brake will be applied immediately. Under docked and uncalibrated situation, the bed could move up and down without beyond the uppermost and lowermost position.

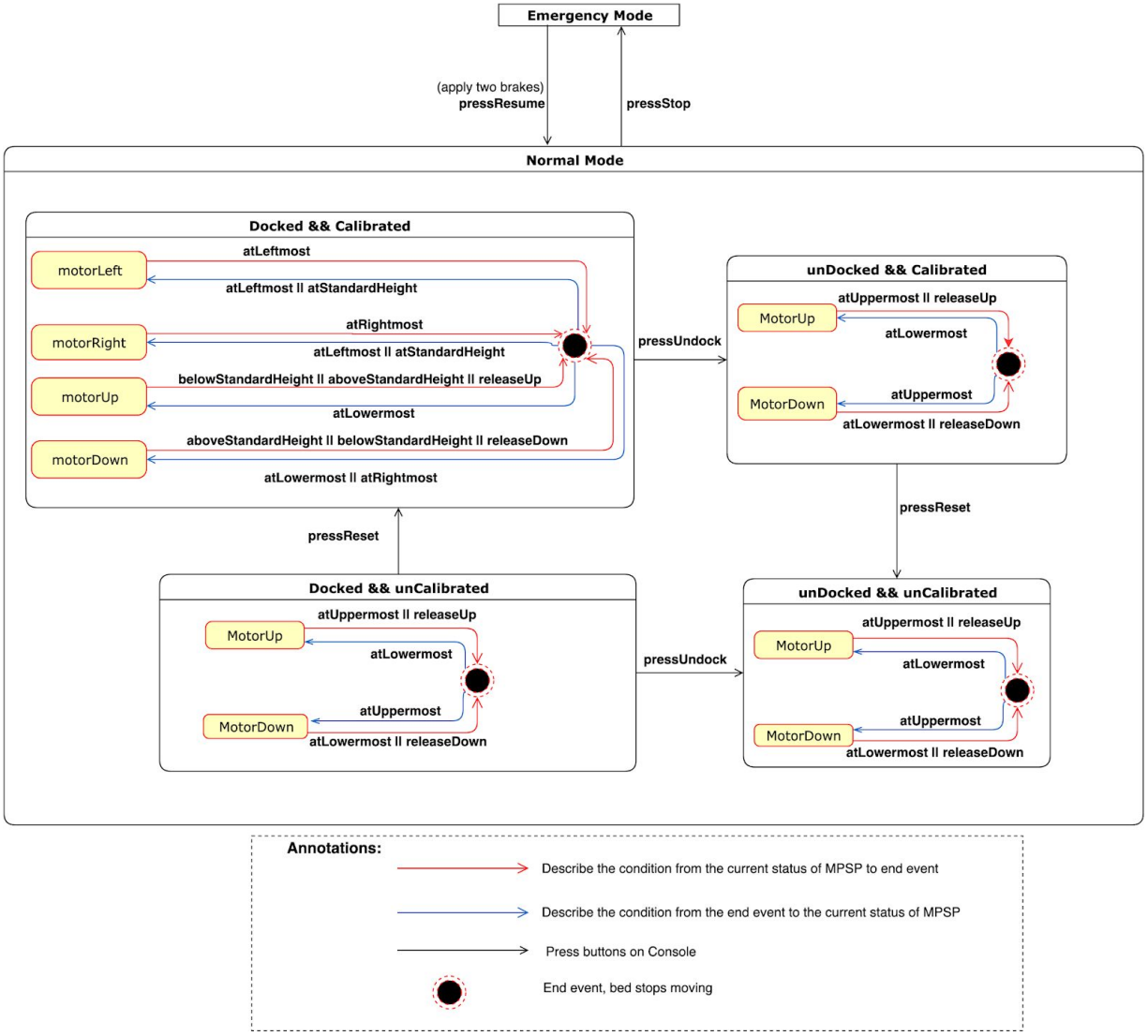

Fig.4 Low-level design architecture

C.Initial stage

In the initial condition, the bed is on the lowest and rightmost position and is connected to sensor while under calibrated, which is shown in Fig 5.

There is no reaction (Noact) if press Reset or Down button. After pressUp button, the vertical brake will be released, the bed moves up to belows (below standard height) or shpos (standard height) position. After the bed reaches the standard height or below standard height, the bed stops moving while releasing Up button. Then motor 1 will be off and Vertical brake will be applied.

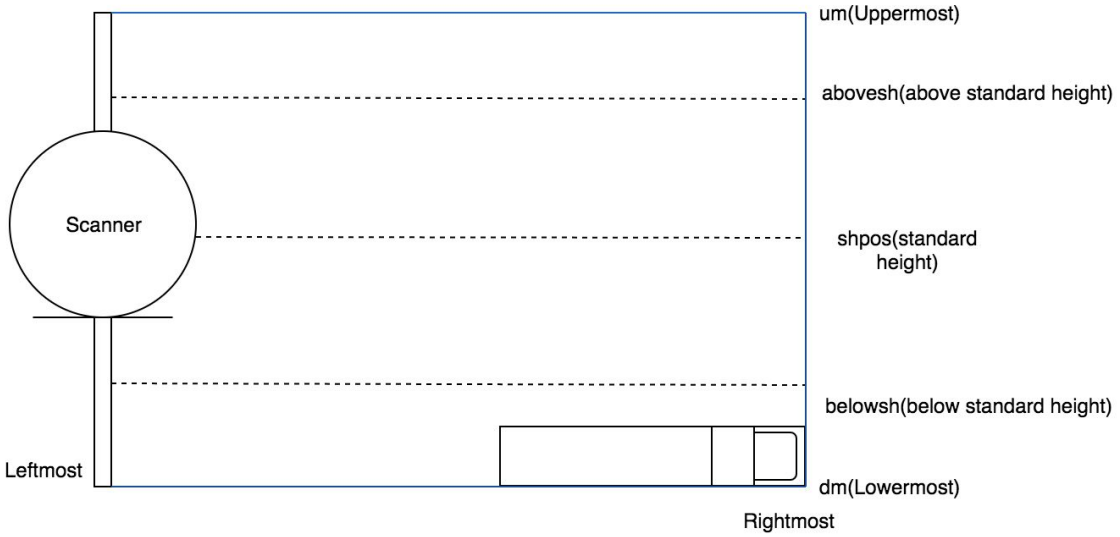

Fig.5 the initial stage of the system

5.mCRL2

See coding file on portfolio.

https://drive.google.com/open?id=1VxIjYZrLgwgCt7DHcyEMISDs64bSwAys

6.Verification

We verify the 11 properties on section 1. All coding files on portfolio.

1.mcf:

[!rmost*.Undock]false

2.mcf:

[true*.pressUndock.!aHbrake*.Undock]false

3.mcf:

[true*.Undock.!Dock*.(mLeft+mRight)]false

4a.mcf:

[!rVbrake*.!rVbrake.(mDown+mUp)]false

4b.mcf:

[true*.m1off.!aVbrake*.rVbrake]false

4c.mcf:

[true*.aHbrake.!rHbrake*.(mLeft+mRight)]false

4d.mcf:

[true*.m2off.!aHbrake*.rHbrake]false

5.mcf:

a.[true*.rmost.!(mLeft||pressResume)*.mRight]false

b.[true*.lmost.!(mRight||pressResume)*.mLeft]false

c.[true*.dmost.!(mUp)*.mDown]false]

d.[true*.umost.!(mDown)*.mUp]false

6.mcf:

[true*.shpos.!(mDown&&mUp)*.(mLeft&&mRight)]false

7a.mcf:

[true*.DoCali.UnCali.shpos]false

7b.mcf:

[true*.UnCali.Undock.!Dock*.shpos]false

8a.mcf:

[true*.shpos.!mDown.mRight*.rmost]false

8b.mcf:

[true*.rmost.!pressDown*.mDown]false

9.mcf:

[true*.pressStop.!rHbrake*.Emer]false

10a.mcf:

[true*.Undock.!Dock*.mLeft]false

10b.mcf:

[true*.!Dock.(pressReset&&shpos).mLeft]false

11.mcf:

[true*]<true>true

7.Modification

We modified all the properties and all of them are verified.