Contrôleur de robot ménager

Finite State Machine

Samedi 1 Jan 2022

Keyan FANG

Xin TANG

Chargé de TD
Julien Deantoni

POLYTECH
NICE-SOPHIA

Introduction

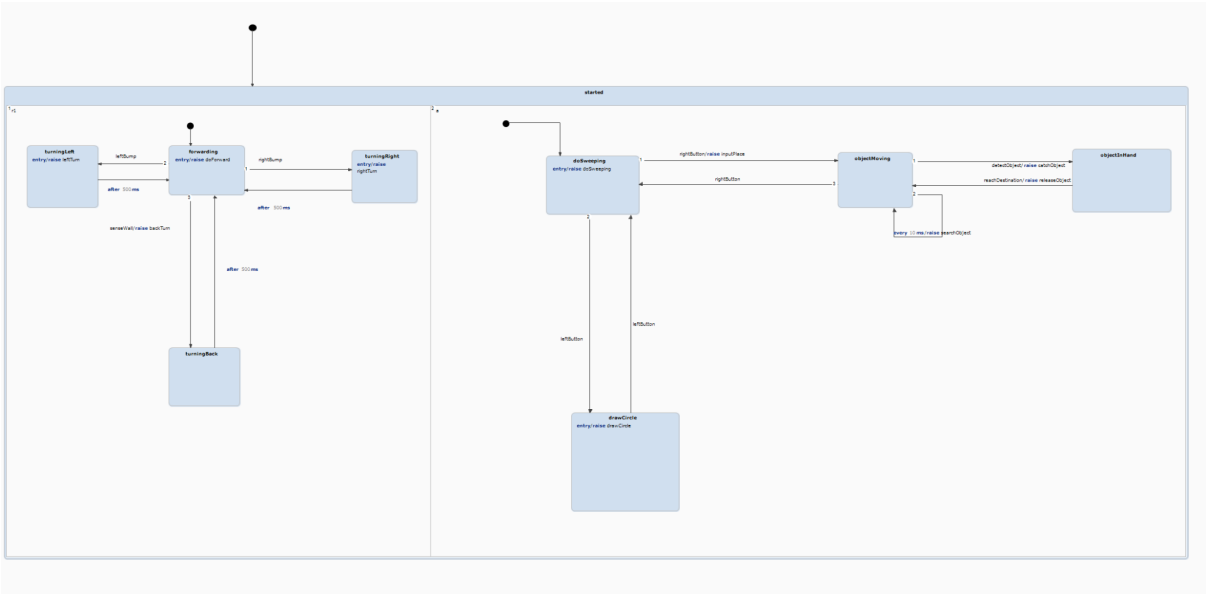
Presentation of subject

The goal of this subject is to simulate a sweeping robot that can vacuum the floor as well as other things like grabbing objects or drawing on the floor. More precisely, the aim is to control the various actuators of the simulated robot in a virtual environment according to the information given by its sensors.

For our project, we have implemented all tasks needed for the MVP and we have chosen to realize the following extension tasks:
1.Contournement propre des objets
2.Gestion du crayon

Presentations of solutions

State chart demonstration



The statechart of our sweeping robot is composed of two main parts: the moving part where the sweeping robot avoids obstacles during its work; and the working part where the sweeping robot performs ground cleaning work and switches the corresponding mode according to customer instructions (buttons) to complete writing, carrying objects, etc.

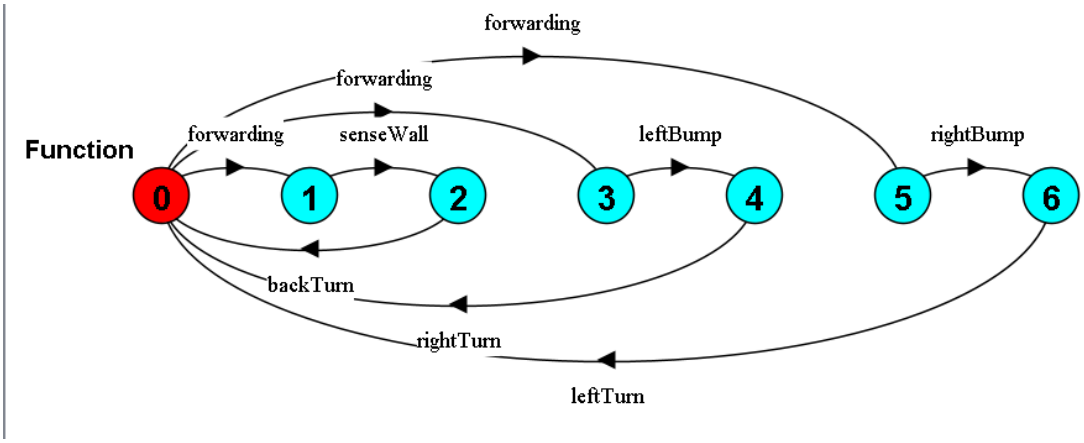
Avoid Obstacle & Clean bypass of objects

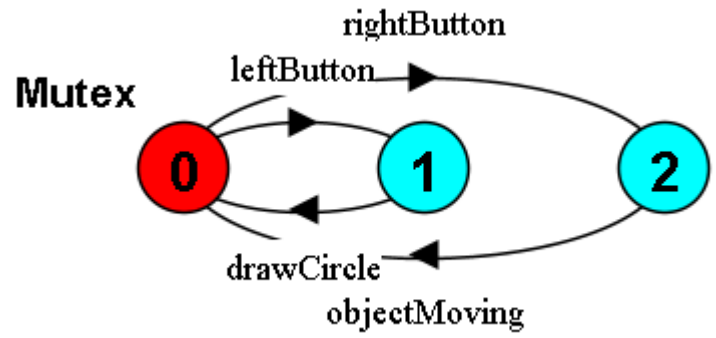
When the sensor of the robot detects an obstacle it turns towards the opposite direction in order to avoid it: turn left when the right sensor detects the obstacle and vice versa. In order to cleanly bypass table legs or other elements, the robot is set to turn at a fixed angle, going on for a small amount of time before returning to the original angle. In this case, once the sweeping robot avoids the obstacle, it will return to the original running track as soon as possible before turning a random angle, so as to ensure that the side of obstacles are cleaned to the greatest extent.

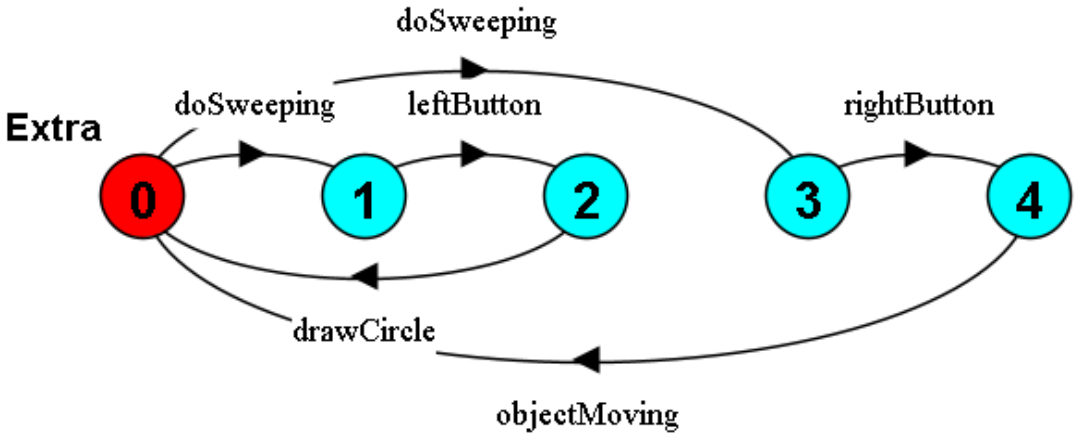
Write on the floor

Using the Pen node of webots to model the pen connected to the mobile robot and display the robot's trajectory. Since it is impossible to simulate the writing with a robot, we choose to use the ‘pen’ to draw circles, squares and other graphics by controlling the speed and direction of the motor, and to achieve human-computer interaction by adding buttons to select the graphics to be drawn through the javaSwing interface. After selecting a certain shape, the robot will begin its drawing and will only end after the user told it to do so.

LTSA and V&V







According to the LTSA code, we can verify that this system works in the correct way and has no error in its programming. And it can change modes between ordinary sweeping and drawing on the ground without extra interference so it's validated to be a state machine which works in good condition.
The result of Check LTL property:

```
Formula !Prop = (true U (initial & (false R !forwarding)))
GBA 2 states 3 transitions
Buchi automata:
Prop = S0,
S1@ =(!forwarding -> S1),
S0 =(true -> S0 |initial&!forwarding -> S1).
Composition:
Prop = initial || forwarding || SYNC || Prop
State Space:
 2 * 2 * 2 * 3 = 2 ** 5
Composing...
-- States: 17 Transitions: 39 Memory used: 14537K
Composed in 11ms
After Tau elimination = 8 state
Prop minimising...
Minimised States: 3 in 1ms
Composition:
DEFAULT = Function || Mutex || Extra || Prop
State Space:
 7 * 3 * 5 * 3 = 2 ** 10
LTL Property Check...
-- States: 105 Transitions: 436 Memory used: 15092K
No LTL Property violations detected.
LTL Property Check in: 1ms
```

As the process goes when the user has chosen mode draw and the machine demanding the type of picture to be drawn will receive a choice so it has liveness property.

What would we have done differently

1. V&V

We regret not being able to use the CADP for the V&V part as the .ysc file didn't work well and there's always a NullPointerException error, given more time we'll be able to fix this problem and use both methods(CADP and LTSA) to check the verification and validation properties.

2. Moving the object

Although we've included the moving object part in the state chart, the code is not developed sufficiently, now the machine can identify the object chosen by the user to be moved and move it when it gets close enough but it cannot search for the object. We can use the front camera to locate the chosen object and turn towards it according to the data returned from the front camera, in that case once the object appears on the camera the robot's priority is to get it rather than to sweep around until the distance in between happens to be small enough. And in order to tell the machine which object to move, the user must put the name of the object in the text panel before clicking the button 'startMoving', next time we can add a checking method so that after the click of the button 'startMoving' if the user didn't put in the name in time the machine will send out error messages.

3. Writing on the floor

We don't know why in the process of interacting with javaSwing, when the method body (circle, square, etc.) called by the monitored button contains the code to change the speed of the motor, the javaSwing interface will freeze. Therefore, after many failed attempts, we only kept a circled method body (drawCircle) and commented out its code.

Auto-évaluation

Name	Xin TANG	Keyan FANG
Finished work	Basic settings of the robot, Write on the floor	Basic settings of the robot, Cleaning bypass obstacle
Points(200/200)	100	100