Report
Firebird Simulator

1) INTRODUCTION:

Embedded system is becoming one of the most useful area nowadays.
Everything is becoming automated. Robot can also be used for many day-to-day activities. Training a robot or making your own code burn on the robot is a big task and the most time consuming task. So the simulated robot can be used first to check the performance of your code. Hence, simulator is directly or indirectly a performance enhancing step towards embedded system field.

2) Problem statement:

Adding the functionality of gripper arm on the simulated robot using player stage open source simulator.

3) Requirements:

1. Player and Stage open source s/w.
2. The simulated robot should collect the object of specified color and place it on the point specified.
3. The code should be reusable for gripper.

4) IMPLEMENTATION

For implementing gripper arm in the simulated robot, we have used inbuilt gripper module of player and stage.

Some of the inbuilt functions of gripper module are as follows:

1. gripper_return: the value is set if we want the object to get gripped and get placed.
2. PADDLE_OPEN : returns true if gripper arm is in open state.
3. PADDLE_CLOSE : returns true if gripper arm is in close state.
4. PADDLE_OPENING : returns true if gripper arm is opening.
5. PADDLE_CLOSING : returns true if gripper arm is closing.

Using above functions, we have implemented a robot which can grip a grippable object in its way and places it on specified location.

Code:

1.Code for line follower-

```
if(left > THRESHOLD && middle > THRESHOLD && right > THRESHOLD)
        stop();
else if(left > THRESHOLD && right < THRESHOLD)
        myRight();
else if(right > THRESHOLD && left < THRESHOLD)
        myLeft();
else if (left < THRESHOLD && right < THRESHOLD)
        forward();
else
        forward();
```

2.Code for gripper –

```
int GripperUpdate( ModelGripper *mod, robot_t* robot)
{
  gcfg = mod->GetConfig();

  if ( gripperAction == 1 && gripperIsClosed() ){
      mod->CommandOpen();
  }
  if ( gripperAction == 2 && gripperIsOpen() ){
      mod->CommandClose();
  }
  if ( gripperAction == 3 && gripperIsDown() ){
      mod->CommandUp();
  }
  if ( gripperAction == 4 && gripperIsUp() ){
      mod->CommandDown();
  }

  return 0;
}
```

3. Code for Robots Position –

```
int PositionUpdate( Model* mod, robot_t* robot )
{       robot->pos->SetSpeed( forwardSpeed, 0, dtor(turnSpeed) );
        Pose pose = robot->pos->GetPose();
        return 0; // run again
}
```

5) Testing strategy and data:

We have positively tested our simulated robot for the following test cases:

1. Line Follower
2. Pick and place
3. Obstacle sensing

6) Discussion of system

Project proposal:

Basically the main task of our project was to incorporate the gripper arm on the simulator robot using player stage open source simulator for robot. We were supposed to add the obstacle detection and white line follower functionality in our simulator.

Actual Implementation:

We have implemented the above things. We have made the reusable code for gripper which can be used as an API for others in case if someone is using gripper functionality just by calling our functions.

7) Future Work:

Implementation of distributed Robots for industrial automation purpose in which robots will be able to communicate with each other and more than one robot will be used for picking and placing the objects at required place.

e.g simulation of automation robot project (e-yantra site)

8) Conclusion:

We have implemented the simulated robot having gripper arm functionality which can be used by others for pick and place objects.

9) References:

1. www.sourceforge.net
2. http://www-users.cs.york.ac.uk/jowen/player/playerstage-tutorial-manual-2.1.pdf
3. http://www-users.cs.york.ac.uk/jowen/player/playerstage-manual-update.html
4. http://web.eecs.utk.edu/~parker/Courses/CS594fall07/handouts/PlayerStageGettingStarted.html
5. http://www.cse.iitb.ac.in/~anirban/projects/seminarReport.pdf