Robot Description

The Robot we need in this simulator is a simple wheeled robot with an IR(infrared) sensor in its front. Basically what it does is the IR sensor sends the signal corresponding to what mark it is in and we can use that signal to develop an algorithm to move the robot in the fixed-line calling it the line follow the robot.

Working of an IR Sensor

Before learning about line follower robots you have to understand what an Infrared sensor is. You can learn in detail about the working of an IR sensor and its implementation in line follower robot here.

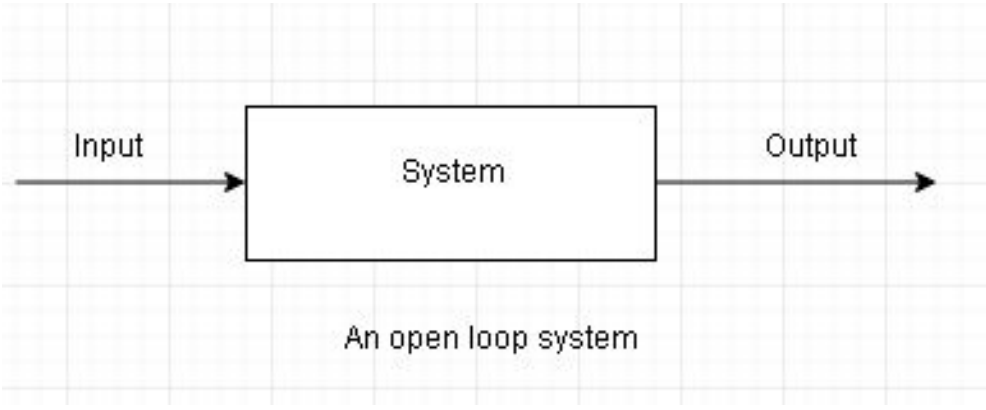
Control System of the Robot

In order to accomplish the task of moving your robot in a black line you must first understand the control system of the robot.

A control system is a mechanism that changes the future state of a system towards the desired state. A control system is concerned with giving appropriate input to a system so that the system might behave in the desired way. A control system can be open or closed.

If we want to move a robot in a specific black line, the robot is the system, the position of the robot with respect to the black line is the state/output of the system and the voltage given to the motor is the input to the system. This process can be achieved using both open and closed-loop systems.

In an open-loop system, the input given to the system does not depend upon the state of the system. In other words, there is no feedback to the system.

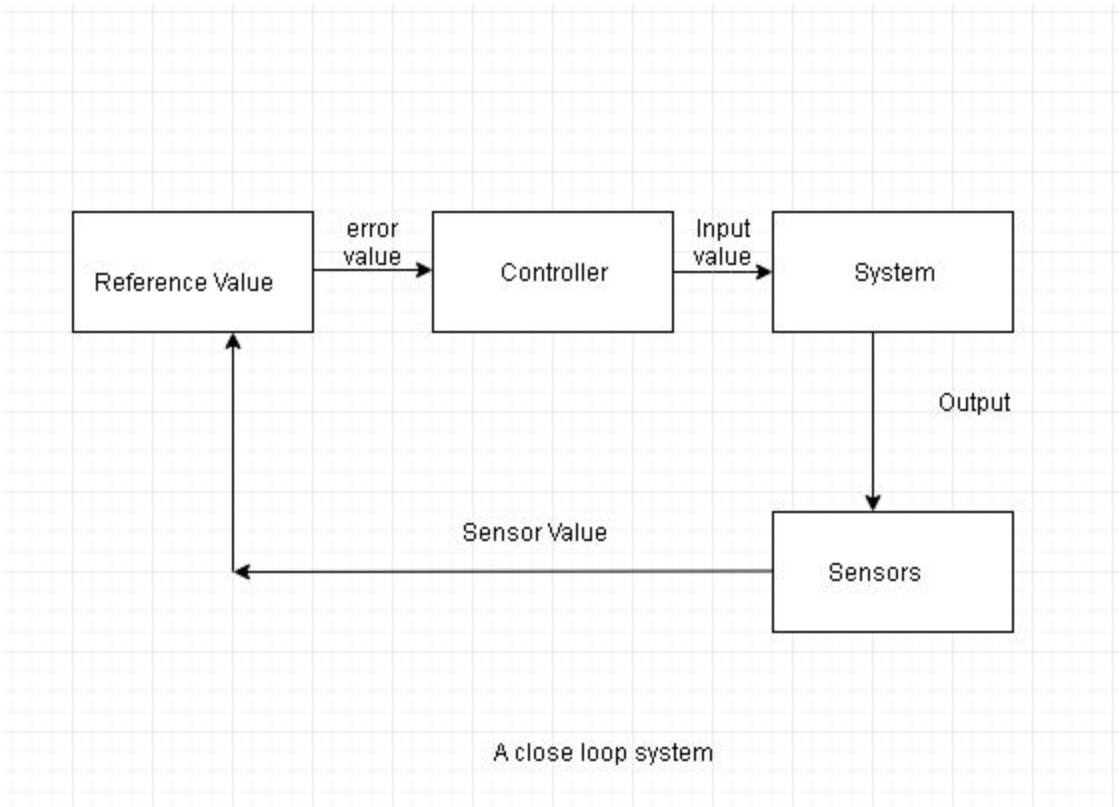


An open loop system

For example, we can make a robot that appears to follow a black line without using any sensors as well. But doing so requires that we measure the length, curvature, etc of the path beforehand. We can then use precise delays and make a robot that fulfills the task. But throughout the path, the voltage given to the motors is never dependent upon where the robot is in reference to the black line. If we use this type of mechanism in our robot then our robot is said to operating under an open-loop system.

However, this type of system is very simplistic and unreliable. For instance, if someone were to suddenly change the pattern of a black line, our robot would still follow the previous pattern. In a way our robot is blind. This defect can be easily overcome using a closed-loop system.

In a closed-loop system, the system continuously monitors its state and keeps correcting itself with reference to a standard value given by the user by continuously varying its input based on the difference between the current state and the reference state.



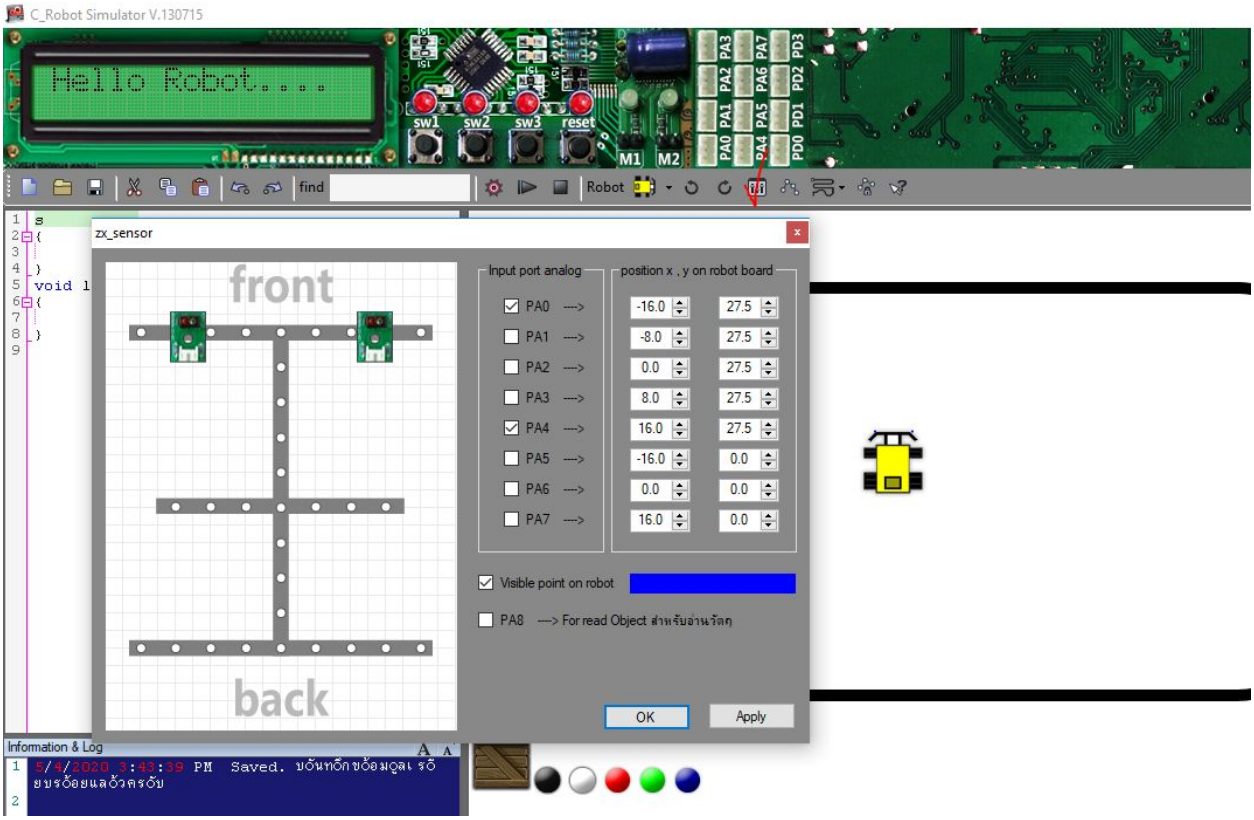
A close loop system

For example, we can make a line following robot using infrared sensors. The infrared sensors can tell us the position of the robot with reference to the black line. Comparing this output with a reference value we can give appropriate input i.e voltage to the motors and thus make the robot continuously follow the black line. Our robot can now be said to be operating under closed-loop control.

Now it will continue to follow the black line even if someone were to change the path. In other words, our robot is now self-operating.

Reading Sensor Value in the Simulator

C simulator allows using IR sensors in the robot. To enable the IR sensors to click on the tab on the menu right next to the clockwise rotation tab. The sensors on the front are named PA0 through PA4 from left to right. Click on the checkboxes to enable them. Now your robot can read sensor values.



Note that the sensors in the simulator are analog in nature meaning that they output value from 0-1024 where 0 is pure black and 1024 is pure white. The return value is of type int and should be stored in an integer variable. For example:

```
int pa0=0;
int pa4=0;//value initialization

pa0=analog(0);//read value from sensor 0 and store it in the variable pa0
pa4=analog(4);//read value from sensor 4 and store it in the variable pa4
```

Now observe the following code to run your robot in a square track in the simulator. Copy the code into the compiler and click the run button on the menu. Remember to drag and place the robot on the track before running the code.

Now replace sl and sr with tl and tr and observe the behavior of the robot.

Bang-Bang Algorithm

Bang-Bang algorithm is the simplest form of line follower algorithm. It is the type of controller which switches between two states. In our case, we are to either turn our robot right or left and the default state of our robot is forward only. As long as the left sensor detects black and the right sensor white our robot has to turn left and vice versa. When both the sensors detect white the robot needs to keep on moving forward. The previous code is the implementation of the bang-bang algorithm.

In the previous example, we only used two sensors. What if we were to add the number of sensors? How would the bang-bang algorithm be implemented then?