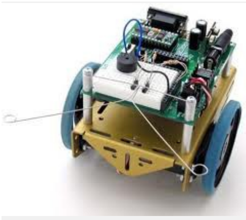
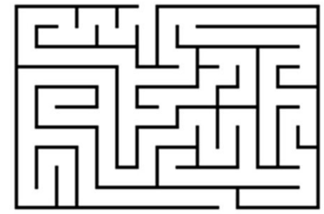


Iona Prep Robotics – Challenge: The Maze



Open Loop and Closed Loop controller:

(Much of what follows is from various articles in Wikipedia)

An open-loop control system, also called a non-feedback control system, is a type of control system that computes its input into a system using only the current state and its model of the system. A characteristic of the open-loop controller is that it does not use feedback to determine if its output has achieved the desired goal of the input. This means that the system does not observe the output of the processes that it is controlling. Consequently, a true open-loop system cannot correct any errors that it could make. It also may not compensate for disturbances in the system.

An open-loop controller is often used in simple processes because of its simplicity and low cost, especially in systems where feedback is not critical. A typical example would be a conventional washing machine, for which the length of machine wash time is entirely dependent on the judgment and estimation of the human operator. Generally, to obtain a more accurate or more adaptive control, it is necessary to feed the output of the system back to the inputs of the controller. This type of system is called a closed-loop system.

For example, an irrigation sprinkler system, programmed to turn on at set times could be an example of an open-loop system if it does not measure soil moisture as a form of feedback. Even if rain is pouring down on the lawn, the sprinkler system would activate on schedule, wasting water.

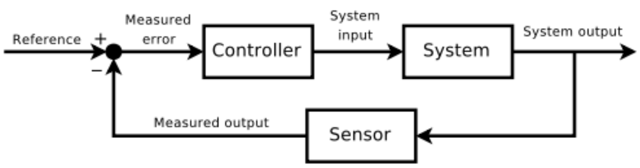
A primitive way to implement cruise control is simply to lock the throttle position when the driver engages cruise control. However, if the cruise control is engaged on a stretch of flat road, then the car will travel slower going uphill and faster when going downhill. This type of controller is called an open-loop controller because no measurement of the system output (the car's speed) is used to alter the control (the throttle position.) As a result, the controller cannot compensate for changes acting on the car, like a change in the slope of the road.

In navigation, dead reckoning (also ded (for deduced) reckoning or DR) is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course.

Dead reckoning is subject to cumulative errors. Advances in navigational aids which give accurate information on position, in particular satellite navigation using the Global Positioning System, have made simple dead reckoning by humans obsolete for most purposes.

ON THE OTHER HAND

In a closed-loop control system, a sensor monitors the system output (the car's speed) and feeds the data to a controller which adjusts the control (the throttle position) as necessary to maintain the desired system output (match the car's speed to the reference speed.) Now, when the car goes uphill, the decrease in speed is measured, and the throttle position changed to increase engine power, speeding up the vehicle. Feedback from measuring the car's speed has allowed the controller to dynamically compensate for changes to the car's speed. It is from this feedback that the paradigm of the control *loop* arises: the control affects the system output, which in turn is measured and looped back to alter the control.



Here is an experiment which will illustrate the point:

Try signing your name on a line while keeping your eyes closed. That could be considered an open loop system. Now sign your name on a line with your eyes open. That would be a closed loop system. Which one worked better for you?

The next assignment: Having your Activity Bot navigate a fairly simple maze.

If you use open loop navigation, errors will accumulate. Since there is no way to correct for this, it will be extremely difficult to arrive at your end point. It is much more efficient to use some form of closed loop navigation.

Your next challenge is to navigate a maze using whiskers to avoid obstacles. The maze will be set up on the lab table in the back of the room.

Construct the whiskers (https://learn.parallax.com/tutorials/robot/activitybot/blocklyprop-robotics-activitybot/navigate-touch) and experiment with your Activity -Bot until you are confident that it can navigate the maze. If you reach the end without causing any fires or injuries to people or property you will get the bulk of the credit. Extra points will be given if you reach the end in a minimum amount of time, or with some original “flair”. Flair is good.