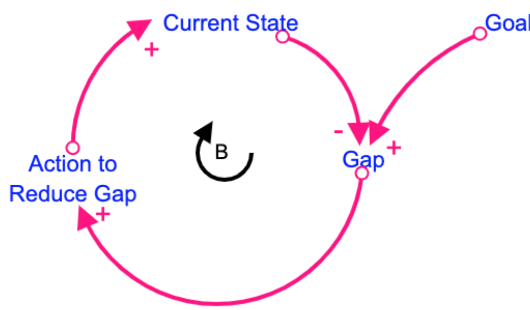
Teaching Robotics with Goal-Seeking Loops

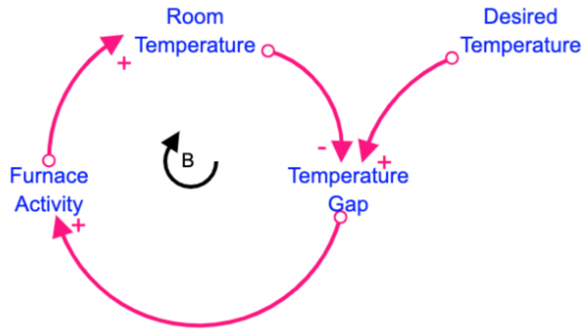
by Chris DiCarlo
Technology Teacher at Innovation Academy Charter School

At work in our thermostats, circadian cycles and even hunger reflexes, the goal-seeking loop is a simple archetype found in systems all around us. We interact with these loops on such a regular basis that teachers can easily activate students' conceptual understanding when introducing a new skill such as maneuvering a robot.

Goal-seeking loops are balancing loops. They seek to bring a quantity to a desired level and keep it there. By evaluating the gap between the actual level and the goal, the system takes corrective action to reduce the gap, and the process repeats. This repetitive nature keeps the current state close to the goal over time.



In our homes, thermostats operate on this concept, measuring the actual temperature in a room and comparing it to a goal temperature. If the actual temperature is too low, the heat turns on in an effort to reduce the gap; if it is too high, the heat turns off, allowing the room to cool. This process repeats indefinitely, causing the heater to turn on and off at just the right times to maintain the desired temperature.



Goal-seeking loops are known as control loops in robotics. They help robots complete complex tasks using relatively simple instructions. A robot takes action to get itself closer to a goal, then evaluates its status and repeats the process. Recently, I challenged my students to program their robots to autonomously maintain a given distance from an object. Their robots have infrared sensors that can determine the distance to something in front of them. Students program a small computer that controls the motors, enabling the robot to move. With no remote controls, students must anticipate which instructions the robot will require in order to take the proper action. Goal-seeking causal loop diagrams help students structure their code and serve as a conceptual graphic organizer for the program. Each element in the loop becomes a line or section of the code. Below is one example of a student's loop and a simplified version of the resulting code.

Causal Loop Diagram	**Robot Program**

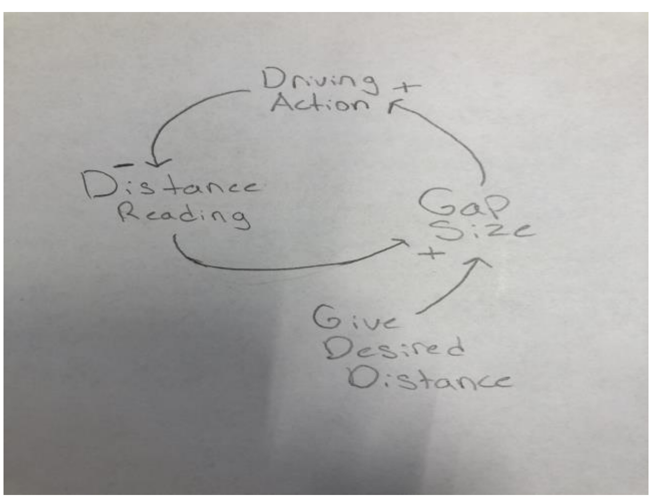
	`loop{`
	`  measure distance`
	`  gap = goal - distance`
	`  if (gap < 0) drive forward`
	`  if (gap > 0) drive backward`
	`} //repeat loop`

Photo Credit: Colin O'Neill	

While the causal loop diagram helps with the structure of the code, the students still need to determine the "driving action" shown in the loop above in response to different gap sizes. Some of my students had difficulty determining if the robot should move forward or backward if the gap was positive. If they had the wrong direction, the robot would drive in a way that would increase the gap each time through the loop creating a *reinforcing* loop instead of a *balancing* one. Students can fix this by swapping the direction that the robot drives. In this case, the code directs the robot to measure the distance to the object, evaluate the gap between actual and desired distance, then take action by driving forward or backward to close the gap. This loop is stable even with a moving target. The robot still takes the proper action to maintain the goal distance. Several students put their robots together in a "conga line" to demonstrate this process in action.

Video of Robot Conga Line

Causal loop diagrams can be powerful for pre-writing code, helping students visualize and understand robotic concepts. The specificity and detail of this type of model complements the high level of precision they need to write functional code.