Christopher Lum
lum@uw.edu

Lecture 05d
Open Loop Control Via Lookup Tables

Lecture is on YouTube

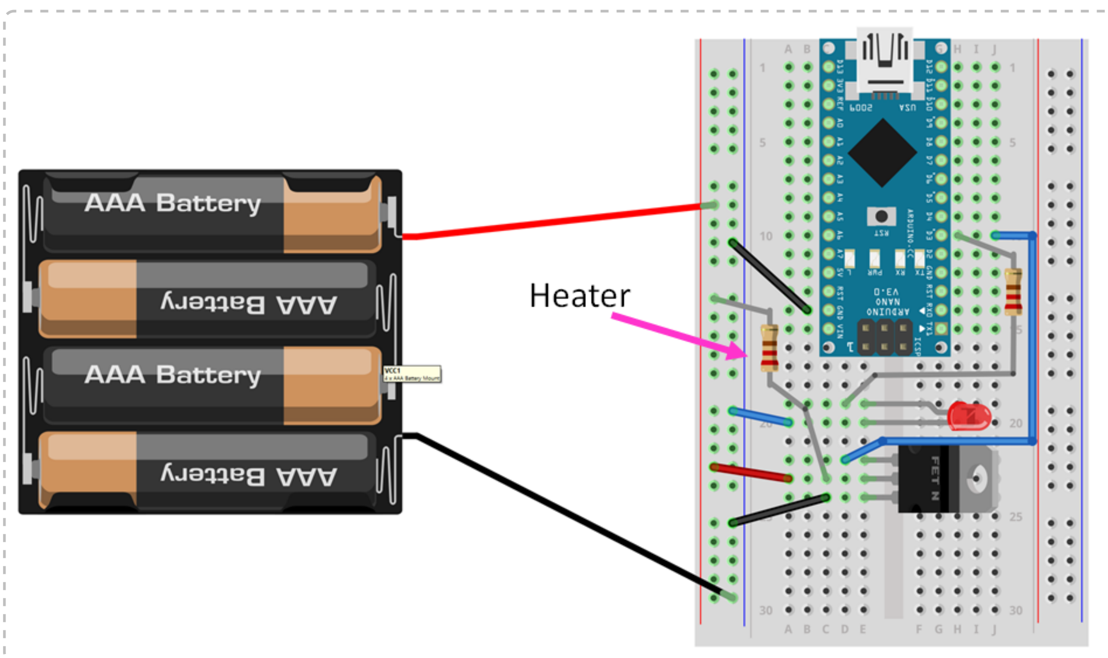
The YouTube video entitled 'Open Loop Control Via Lookup Tables' that covers this lecture is located at TBD.

Outline

-Planar Vehicle Model

Example Thermal System

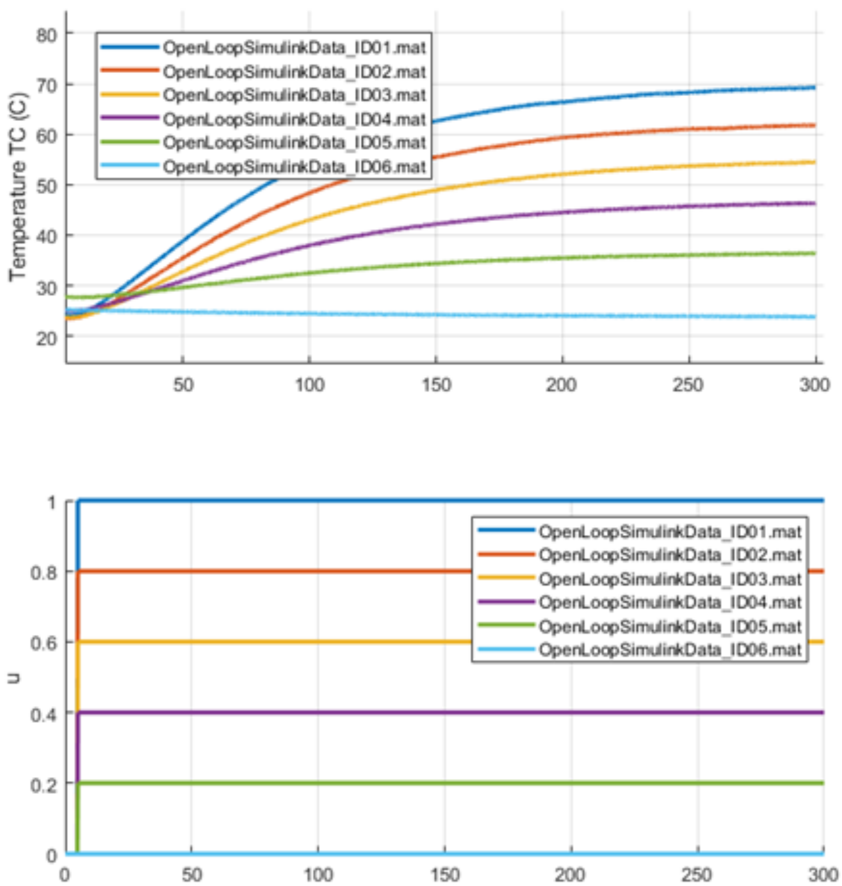
Consider the simple thermal system we discussed previously (INSERT REFERENCE TO THERMAL SYSTEM VIDEO, TBD)



Let us consider a few simple control systems to attempt to hold the temperature of the block at a desired value.

Open Loop Controller (Lookup Table)

We can investigate a very simple open loop control scheme. Consider subjecting the system to a series of constant inputs. The resulting temperature traces for various duty cycles are shown below.

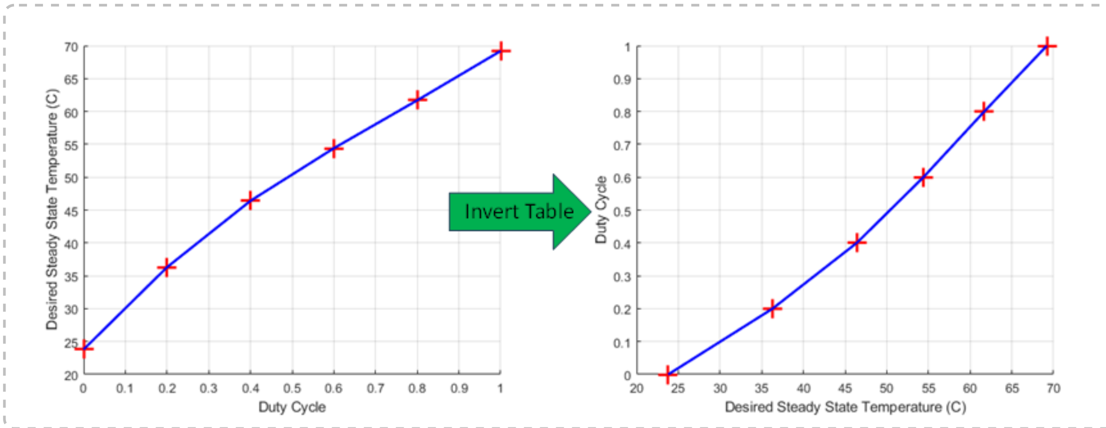


We can tabulate the duty cycle and the resulting output steady state temperature as shown below.

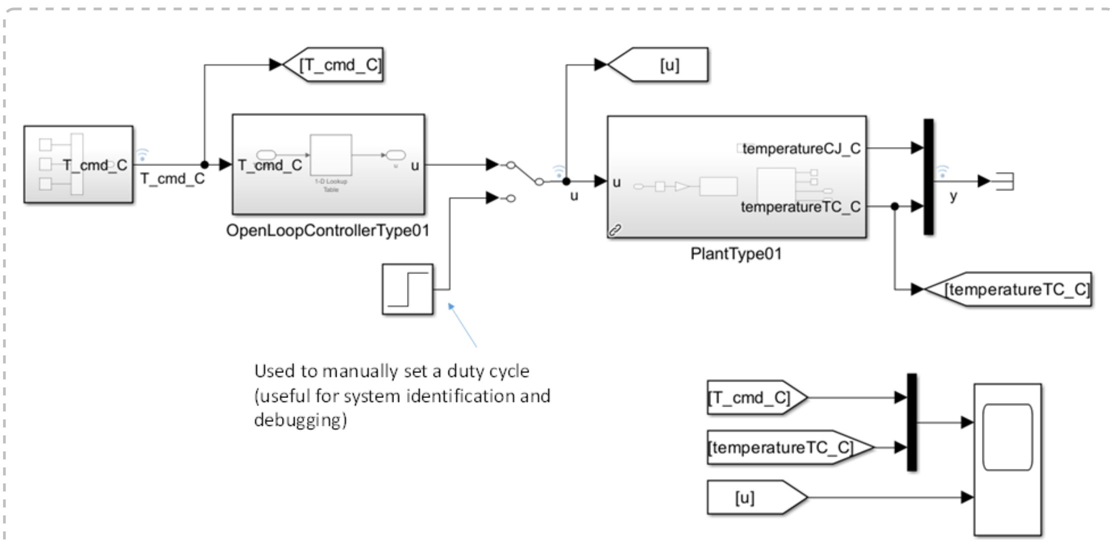
Duty Cycle	Steady State Temperature (C)
0	23.8
0.2	36.3
0.4	46.4
0.6	54.4
0.8	61.7
1.0	69.2

This table embeds knowledge of the system in the sense that we now know what the steady state response of the system is to a steady state input. In this case, we specify the duty cycle and the table returns the steady state temperature.

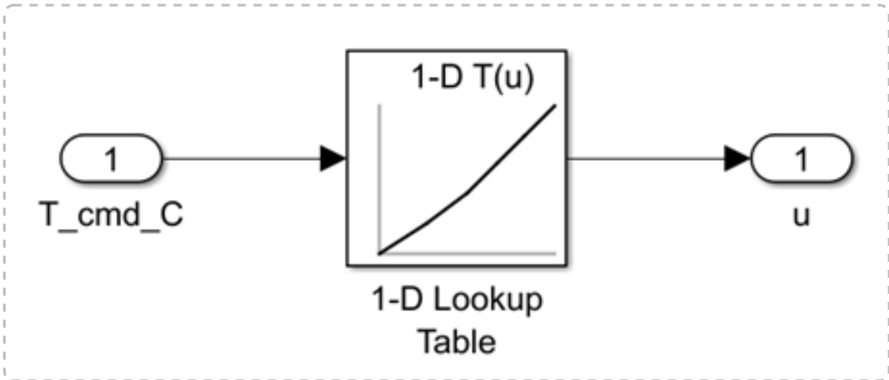
We can generate invert this table to generate a lookup table where we specify desired steady-state temperature and the system will choose the appropriate control input (duty cycle) for us



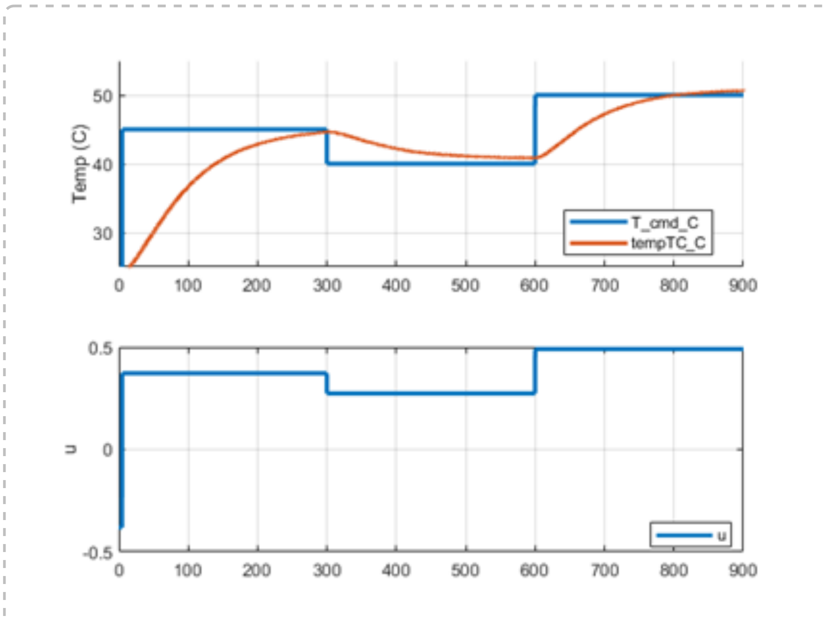
We can build this in Simulink as shown below (see https://github.com/clum/MatlabLum/tree/main/Applications/Arduino/Simulink/TemperatureControl/Step02_OpenLoop)



Here, the ‘Controller’ block is implemented using the 'Lookup Table' block



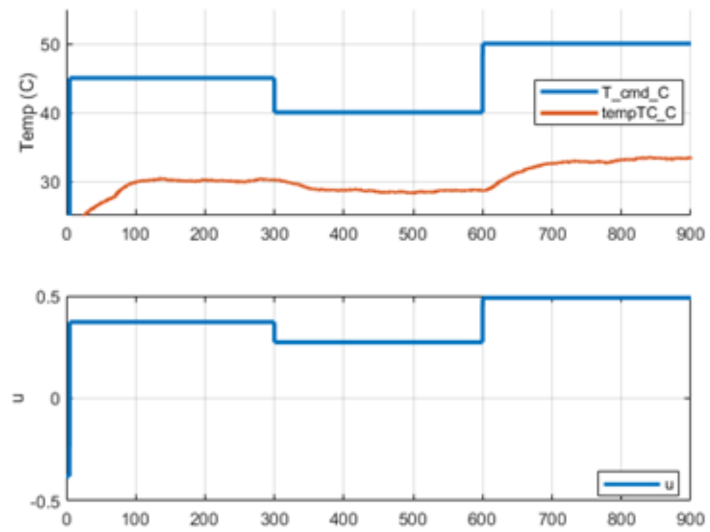
The response of the system is shown below



As can be seen, the performance is surprisingly reasonable. It appears to work in the sense that the actual temperature is close to the desired/specified temperature. That being said, there are some issues including but not limited to:

-Non-trivial steady state error
-Slow response

The performance degrades significantly if the conditions are different. For example, if we add a small breeze across the heater, we obtain the response below.



The performance is now quite poor. Note the following:

- -Steady state error has increased
- -Control signal is unchanged

The fact that the control signal is unchanged means that the system does not respond to change in the plant or environment.

Example: Student Demo

We can investigate an example of this using humans to play the part of the open loop controller and plant.

Student A - controller
Student B - plant

-Draw a point on the white board showing where we would like to place Student B.
-Have Student B stand a certain distance away from the mark.
-Student A looks at student B and the desired location and then blindfolds themselves.
-While Student A is blindfolded, we can switch Student B with another student with longer/shorter legs (to simulate uncertain plant dynamics)
-Student A then issues open loop commands to Student B regarding how many steps to take to get to the mark.

Example: Human Walking (Babies)

Show video of Amara trying to walk up a ramp. Her control system for walking is effectively an open loop controller (sending commands to her legs) and this does not tolerate environmental changes well.

“D:\lum\pictures\lumia_920\WP_20140129_17_57_21_Pro.mp4”

Summary of Open Loop Control

As we have seen, open loop controllers have several advantages and disadvantages

PROS
-Simple.
-No sensor required.
-Economical (since no sensor is required).
-Requires very low computational power.
-Does not require advanced techniques to design controller.

CONS
-Not robust, cannot handle changes in plant, environment, etc.
-Only functions perfectly if conditions are perfect (AKA you require perfect knowledge of plant, environment, etc.)
-Response time may be unsatisfactory.

Relation to Feedforward Control

While this strategy is simple, we see that it has problems, particularly when the plant or environment is not the as the expected. That being said, open loop control can be a powerful technique and is the basis of more sophisticated feedforward control schemes (we will investigate these later).