WATER TANK PLANT MODEL

OBJECTIVE

To model a tank filling process and develop a control system to maintain the level of liquid in the tank.

Dean Lonergan

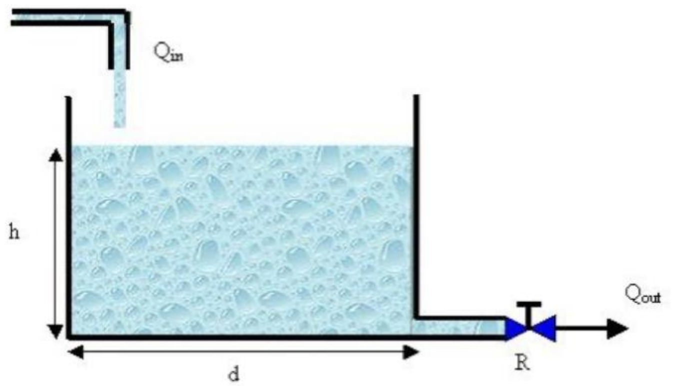
Model Based Development - Applied Computing

TABLE OF CONTENTS

INTRODUCTION

This document will outline the design and testing procedures of a water tank plant model. The model will be built using MATLAB Simulink and will accurately simulate the filling and emptying of a water tank. The complete model will allow the user to control the height of the water in the tank in several ways. Each area of the design process will be documented, the model will be tested at each stage of development to verify it meets the design criteria and is producing valid results. The final model will be presented after a complement of verification and validation tests have been completed.

Design criteria



A liquid flows into a tank at a rate of Q_{in} litres/second. At the same time liquid flows out of the tank at the rate of Q_{out} litres/second through a valve with a flow resistance of R. Closing the valve increases the resistance to flow R, while opening it decreases R. The height of liquid h in the tank at any time is proportional to the difference between the input and output flow rates and the cross-sectional area A of the tank (A = πd metres^2).

The dynamic model of the tank is:

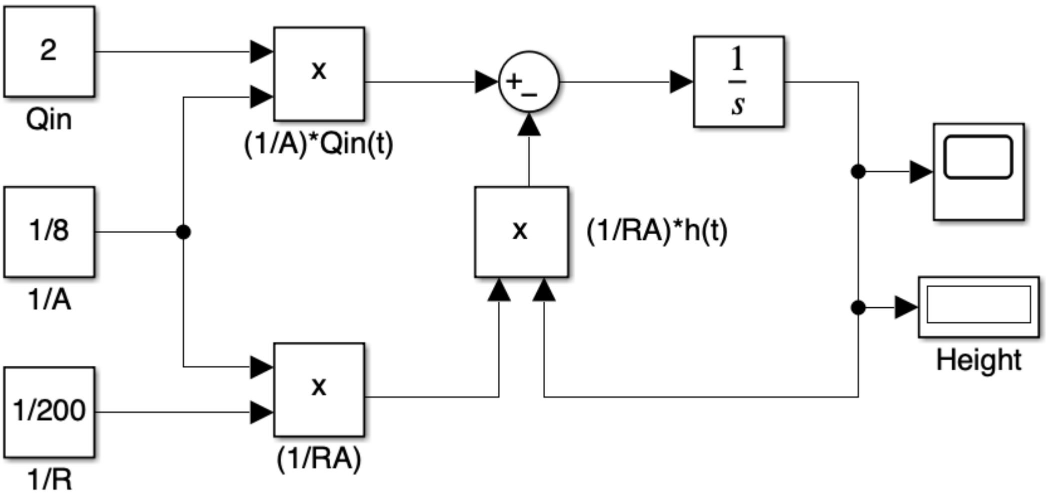
$$\frac{d}{dt}h(t) = \left(\frac{1}{A}\right) Q_{in}(t) - \left(\frac{1}{RA}\right) h(t)$$

Where:
h = height of liquid
R = resistance to flow of output valve
A = cross-sectional area of the tank
Q_{in} = the input flow rate

This model must be controlled by a PID (proportional integral derivative) controller as part of a closed-loop feedback system and must include a transport delay of 0.5 seconds to simulate a reasonable lag in response time. The controller must achieve a response time of approximately 20 seconds and must not demonstrate overshoot characteristics of more than 10%. The user controls must include three buttons to control the water height: Low (height of 2m), Med (height of 8m), and High (height of 16m). There must also be a boost button that increases the current height by 10% for 10 seconds, and an empty button which empties the tank when the button has been pressed for 5 seconds.

DESIGNING THE MODEL

Preliminary model



Model verification

figure 3: 2 litres/second, R = 200, t = 60

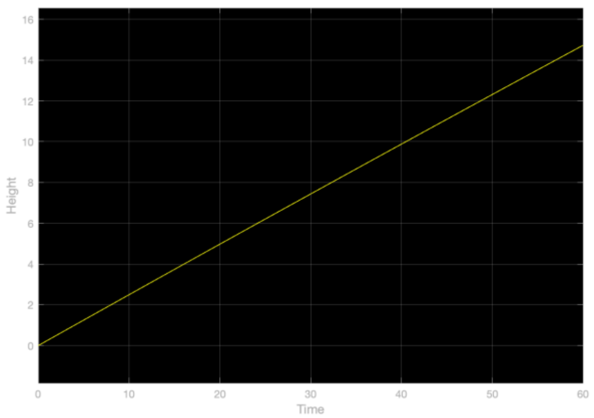


figure 4: 4 litres/second, R = 200, t = 60

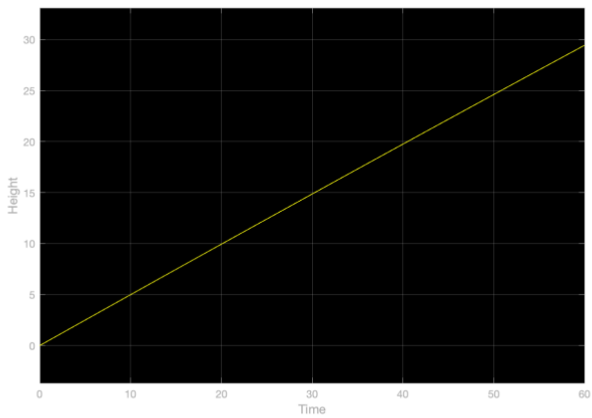


figure 5: 2 litres/second, R = 0.1, t = 15

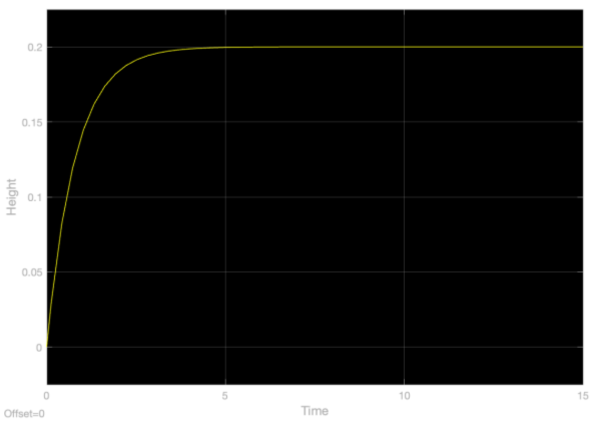
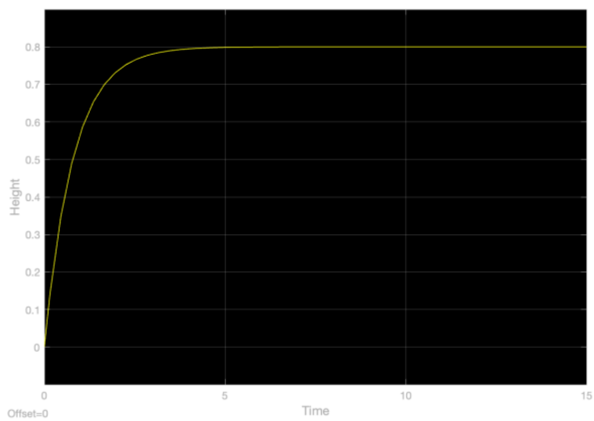


figure 6: 8 litres/second, R = 0.1, t = 15



Verification analysis

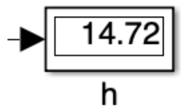
With *resistance* set to 200, the *height* of the water with *Qin* set to 4 is twice that of when *Qin* is set to 2. These results are demonstrated in *figures 3 & 4* and indicate that the model behaves as we would anticipate while the valve is essentially closed. The rate at which the *height* increases is linear and predictable when the flow rate increases. With *Qin* set to 2 and the *resistance* lowered to 0.1, the constant flow of water in causes the *height* to increase non-linearly due to the simultaneous flow of water out.

When the water flowing in reaches an equilibrium with the water flowing out, the *height* levels, as demonstrated in figure 5. The result of increasing *Qin* to 8 while keeping the *resistance* at 0.1 is displayed in figure 6. It behaves, again, as expected, but due to the increased volume of water in, it reaches a greater overall height. This maximum height is four times higher than before, as the amount of water flowing in is four times greater.

These results show that the redesigned model functions correctly and predictably and presents results in alignment with what we would anticipate. While the initial height figures seem reasonable, the model will require further testing and validation to ensure that it is indeed producing accurate results. Testing also shows that the model will not run if provided an *A (cross-sectional area)* or *R (resistance)* value of 0. The final model must have protections to prevent this use case.

Model validation

The model functions as expected and presents results in alignment with what is anticipated. To validate that these results are indeed accurate, the model was set up using the following parameters: R = 200, A = 8, Q_{in} = 2, t = 60.
With these parameters the model produced a resulting liquid height of *14.72m.*



These results must now be compared to a calculated result of *height* using the formula:

$$V = Ah$$

Where:
h = height of liquid
A = cross-sectional area of the tank
V = the volume of liquid after 60 seconds at 2litres/second.

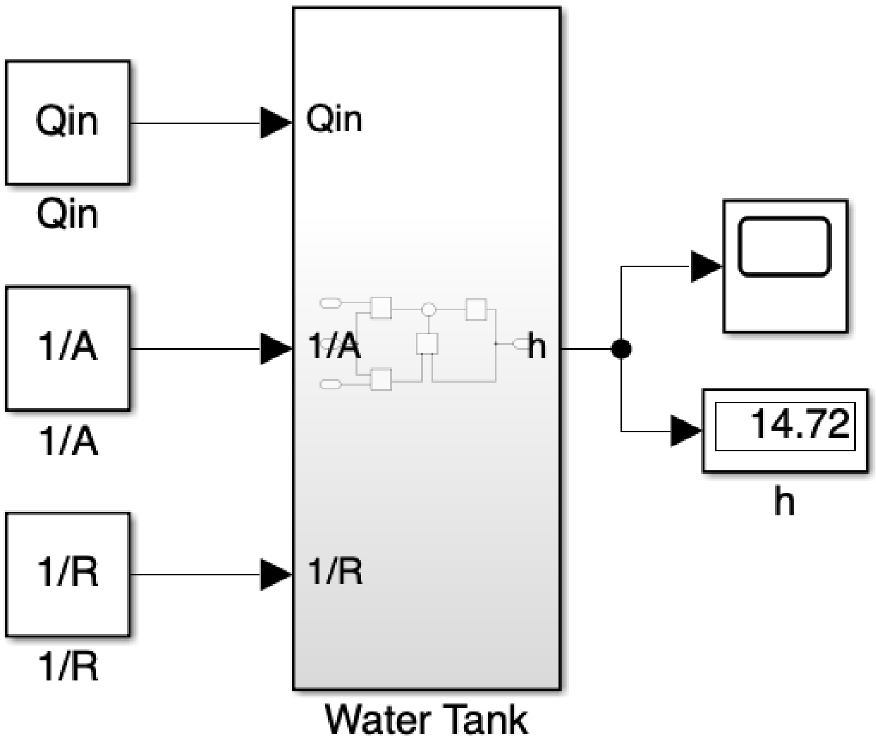
$$120 = 8h \;\therefore h = \frac{120}{8} = 15m$$

This result presents a percentage varience of just *1.8%* between the model and the calculation. While small, this variation can be attributed to the *resistance* setting of 200. Although a *resistance* value this high essentially closes the valve, the model will still allow some water to flow out. Raising the *resistance* to 2000, produces a resulting *height* of *14.97m.*

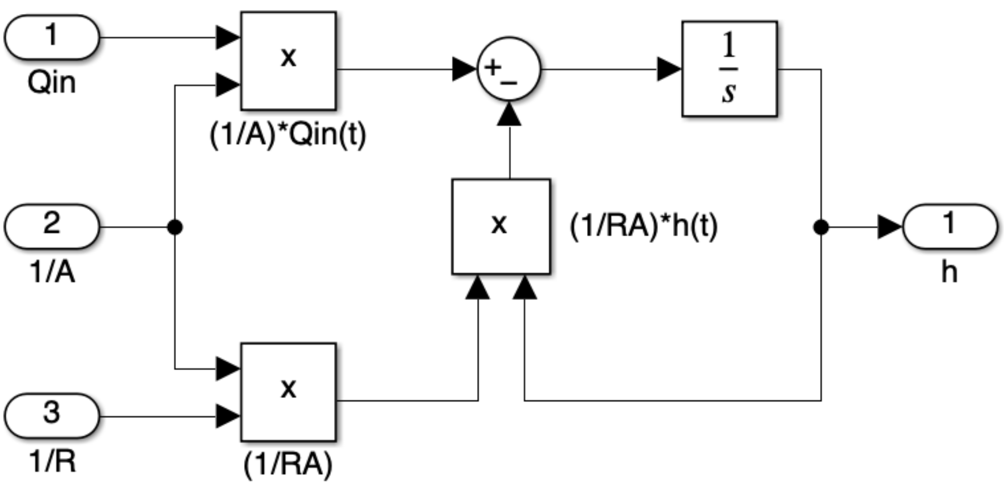
Redesigned model

The redesigned model integrates MATLAB variables and a subsystem to tidy up the model and improve functionality. The subsystem contains the fully verified and validated elements of the redesigned model. The output, *height*, is then presented in the display, *h*, and as a graph via the scope. This version of the model functions as desired and meets the design criteria for the water tank plant model.

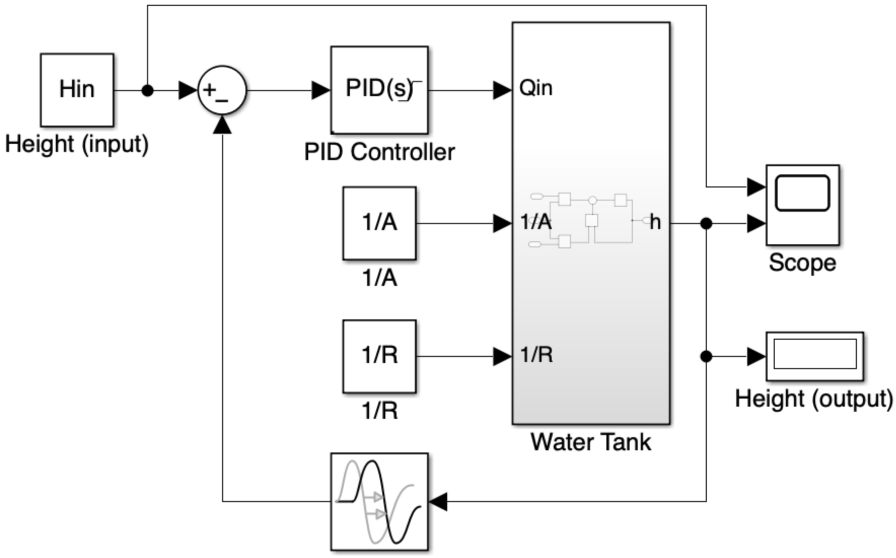
Model and controls



Water tank subsystem



Implementing the PID controller



Preliminary functionality

figure 7: Hin = 8, R = 1, t = 60

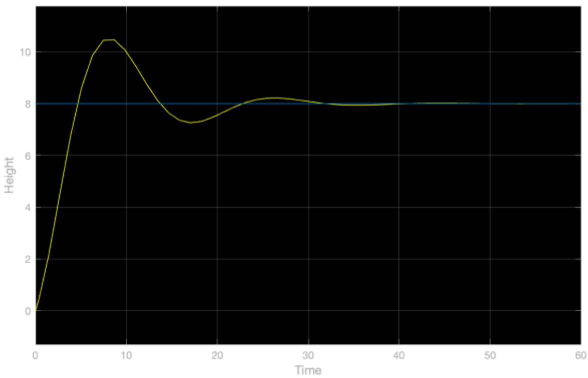
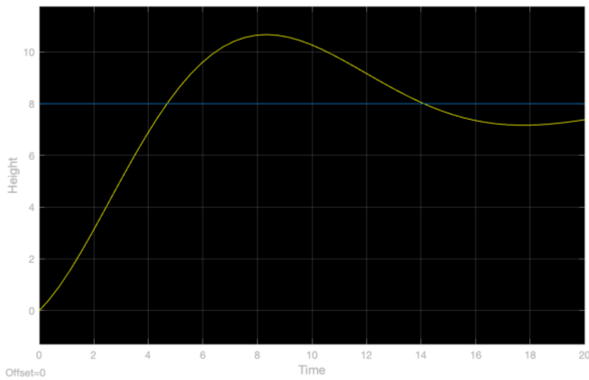


figure 8: Hin = 8, R = 1, t = 20



Preliminary Analysis

It is clear from *figures 7 & 8* above that the controller is underdamped. While the controller does correctly reach the desired height, it takes much longer than the required 20 second period to do so, as demonstrated in *figure 8*. The initial rise is quite fast, but the overshoot far exceeds the required 10% threshold, the height then fluctuates repeatedly before settling after approximately 40 seconds. The controller will need to be tuned using the Ziegler-Nichols method before it functions correctly.

TUNING THE CONTROLLER

Ziegler-Nichols tuning method

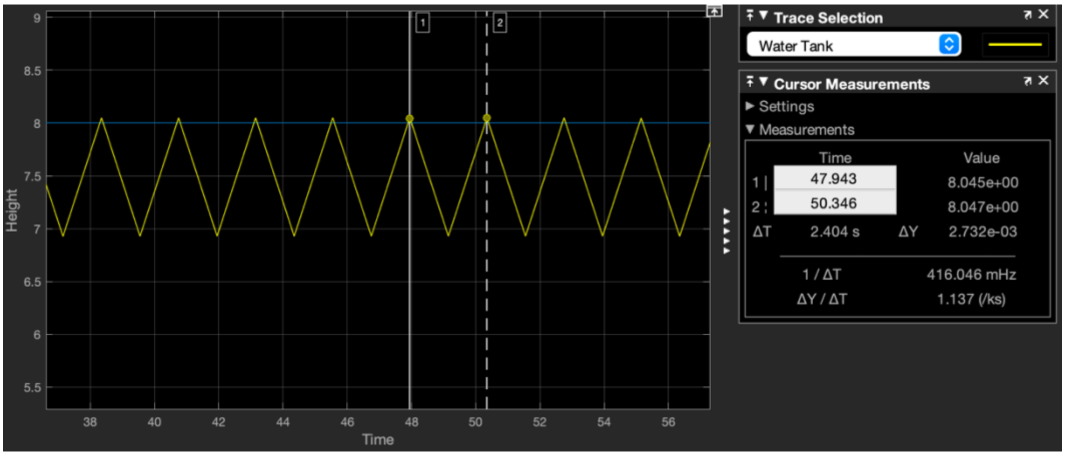
The Ziegler–Nichols tuning method is a method of tuning a PID controller developed by John G. Ziegler and Nathaniel B. Nichols. The method works by setting the integral and differential gains to zero, and then raising the proportional gain until the system is unstable. The value of K_P at the point of instability is called K_{Max}; the frequency of oscillation is f_0. The method then backs off the proportional gain a predetermined amount and sets the integral and differential gains as a function of f_0. The P, I, and D gains are set according to table below.

	K_P	K_I	K_D
P Controller	$0.5\ K_{Max}$	0	0
PI Controller	$0.45\ K_{Max}$	$1.2\ f_0$	0
PID Controller	$0.6\ K_{Max}$	$2.0\ f_0$	$0.125/f_0$

Using this method to tune the controller

With the gains of I and D set to zero, I raised P to 10. This caused somewhat inconsistent oscillations. Reducing this figure to 9 helped but was still not entirely consistent. I eventually reached a P value of 9.6 which provided very consistent oscillations.

figure 9: Hin = 8, R = 1, t = 60



$$K_{Max} = 9.6$$

$$f_0 = \frac{1}{2.404} = 0.416 \therefore 416mHz$$

With these figures, the Ziegler-Nichols method was used to calculate appropriate baseline values for P, I and D.

$$K_P = 0.6(9.6) = 5.76$$

$$K_I = 2.0(0.416) = 0.83$$

$$K_D = \frac{0.125}{0.416} = 0.3$$

Preliminary tuning

figure 10: Hin = 8, R = 1, t = 60

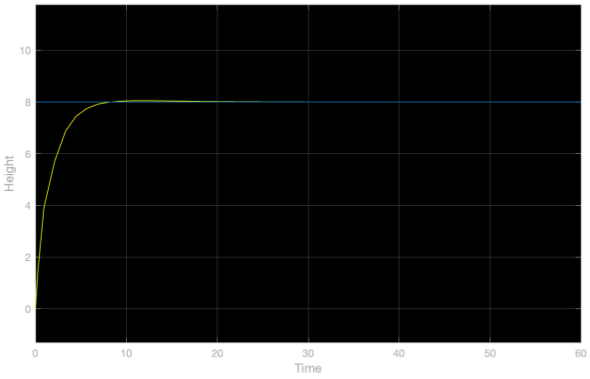
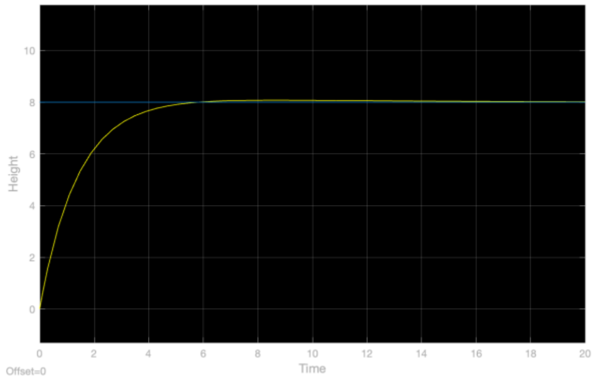
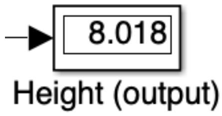


figure 11: Hin = 8, R = 1, t = 20





The values calculated using the Ziegler-Nichols method provided excellent results. The model operates within spec in all areas. As demonstrated in *figures 10 & 11* above, there is less than 1% of overshoot, with an acceptably quick initial rise time. The *height* of the water within the tank reaches the desired *Hin* value of 8 within 20 seconds, being only 0.2% above the desired *height* after this period. The Ziegler-Nichols tuning method has gotten us very much within the desired ballpark, but we can still do some manual tuning from this point to try and improve on these already impressive responses.

Manual tuning

figure 12: Hin = 8, R = 1, t = 60

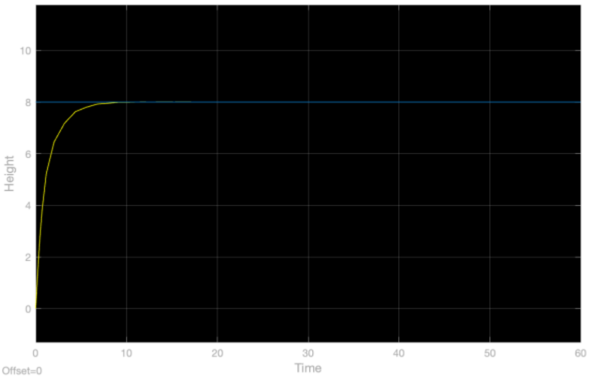
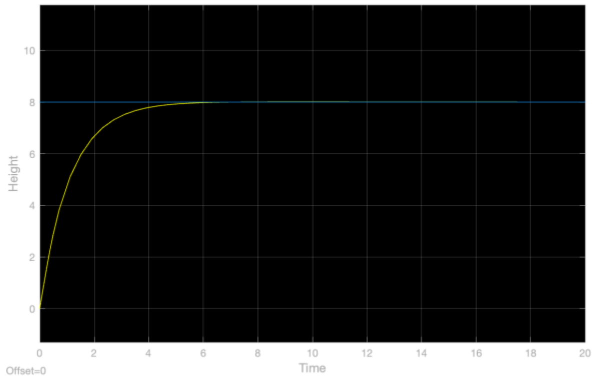
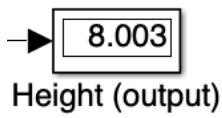


figure 13: Hin = 8, R = 1, t = 20

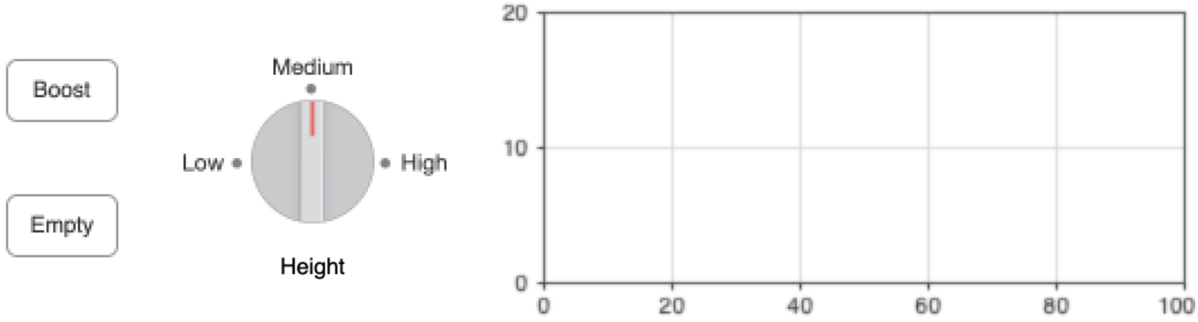


$$P = 7.2, \qquad I = 0.92, \qquad D = 0.29$$



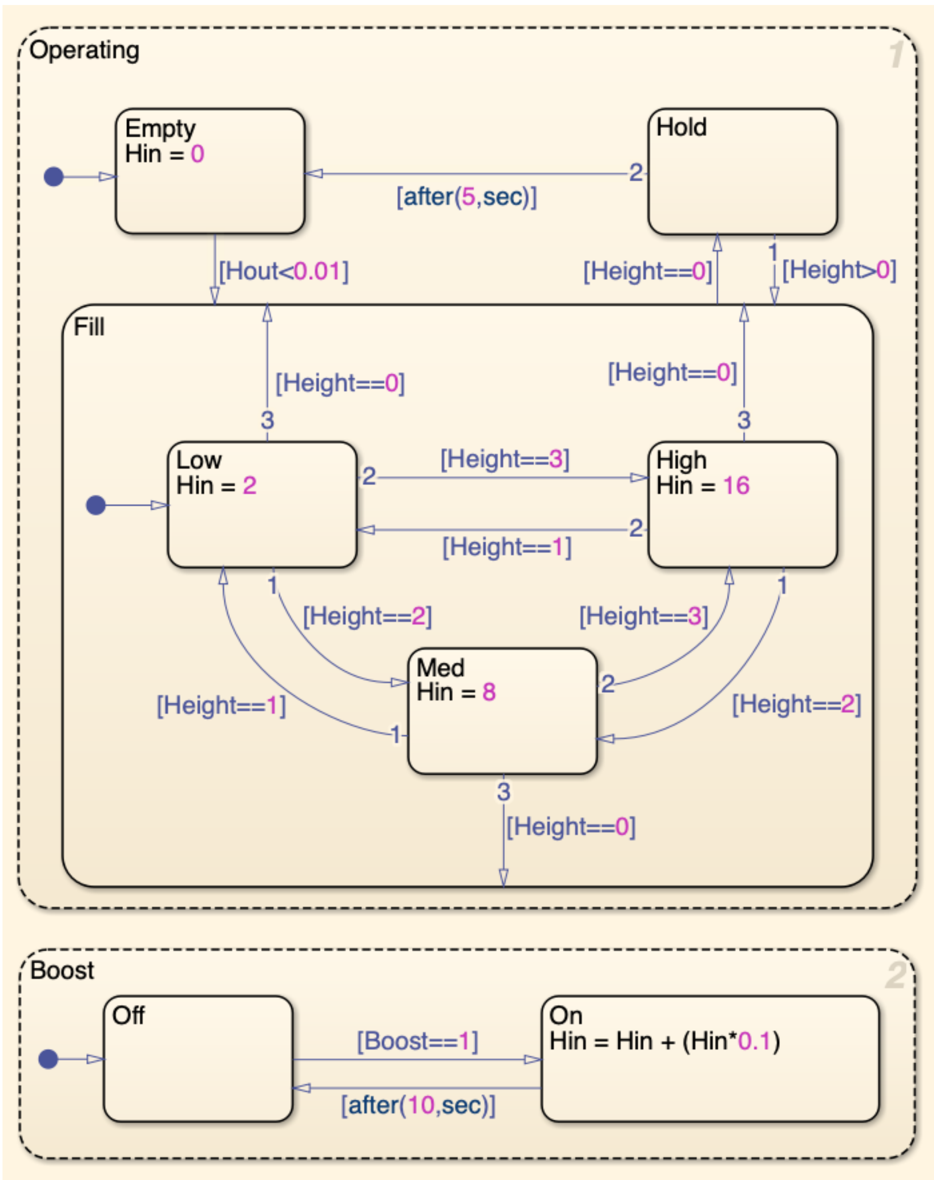
Manually tuning the controller from the baseline lead to further improvements to the already impressive responses the model was demonstrating. The initial rise was made slightly quicker, with any overshoot removed almost entirely. The *height* still reaches the desired *Hin* value of 8 within 20 seconds, this time being less than 0.04% above the desired *height* after this period. These improvements, while small, were far easier to make with the Ziegler-Nichols figures as a baseline. Dialing-in took very little time and has resulted in nearly perfect response characteristics from the model, it is critically damped.

User controls



This control scheme covers each of the requirements set by the design criteria and includes a scope to visualize the filling/emptying of the tank in approximately real time. Implementing the required logic for each control will require designing a Statechart.

Statechart



States

Operating

- Empty – This state empties the tank.
- Hold – While the *Empty* button is held, this state waits for 5 seconds.
- Fill – This is the superstate containing the filling substates.
    - Low – This state sets the desired *height* to 2m.
    - Med – This state sets the desired *height* to 8m.
    - High – This state sets the desired *height* to 16m.

Boost

- Off – This state is the default boost state.
- On – This state increases the desired *height* by 10%.

Statechart description

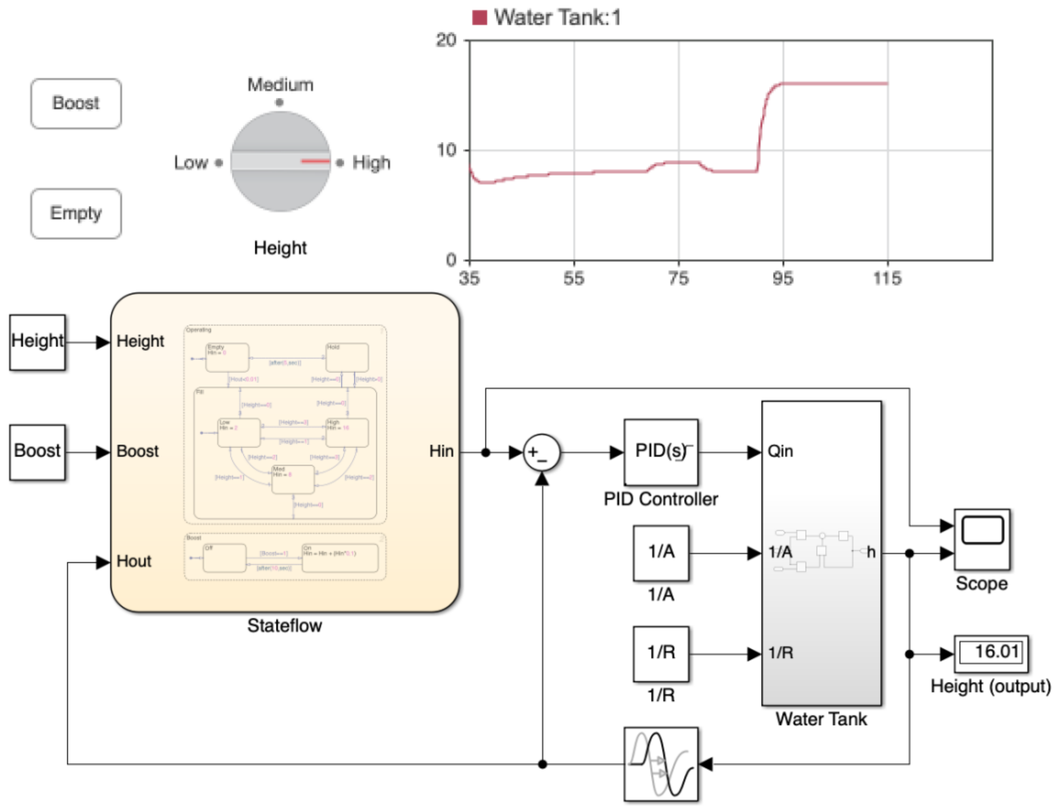
The Statechart requires 3 data points: *Height, Boost,* and *Hout*. *Height* is a variable controlled by the position of the *Height* rotary switch: Low = 1, Med = 2, High = 3. *Height* is also controlled by the *Empty* button, which sets *Height* to 0. *Boost* is a variable controlled by the *Boost* button. When pressed, *Boost* is set to 1, 0 otherwise. *Hout*, or *Height-out,* is the height value calculated by the model. This is used in the Statechart to determine when the tank has been fully emptied.

The Statechart makes use of two parallel sates, *Operating* and *Boost*. This design allows for each filling state to be reached in parallel with the *Boost* state. This means any fill state can be 'boosted' and allows the *Boost* button, which uses an *[after(10,sec)]* logic transition, to achieve the correct boost period. Each fill state can be reached by any other fill state, with simple state logic controlling each transition. The state being tracked is the *Height* variable.

While the *Empty* value of the *Height* variable can be achieved by pressing the *Empty* control button. The *Empty state* is only reached after the *Empty* control button is held down for 5 seconds. This is achieved by the use of the *Hold* state, in conjunction with an *[after(5,sec)]* logic transition. When in the *Empty state*, *Hin* is set to 0. This state is only left after the *height* of the water in the tank is less than 0.01m. Absolute zero is not used as during testing it was found that the *height* value would move towards absolute zero almost asymptotically.

FINAL MODEL

Statechart and controls implemented



Final verification

figure 14: R = 1, t = 120

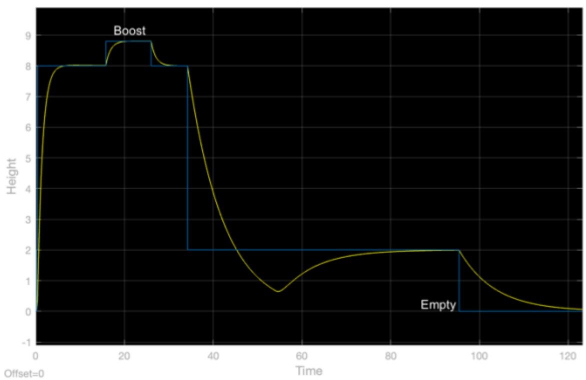
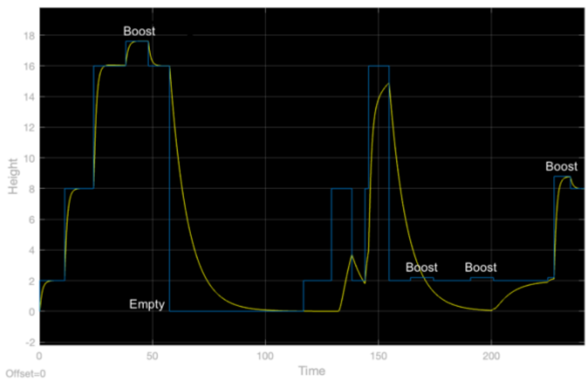


figure 15: R = 1, t = 240



The final model meets all the design criteria and can be controlled as required. The results of "real-world" use are demonstrated in *figures 14 & 15* above; the blue line represents the desired *height,* while the yellow line represents the actual *height* of the water in the tank. The more predictable inputs demonstrated in *figure 14* display the response characteristics of the model during what would be considered "normal" use. *Figure 15* shows somewhat more erratic user inputs with several quick increases in height, followed by emptying the tank. In both scenarios the model behaves as expected, producing an accurate simulation of the filling process, and responding to user inputs quickly and accurately.