QR project documentation

Rasyan Ahmed (10784063) & Michelle Appel (10170359)

1 casual model and expected state-graph

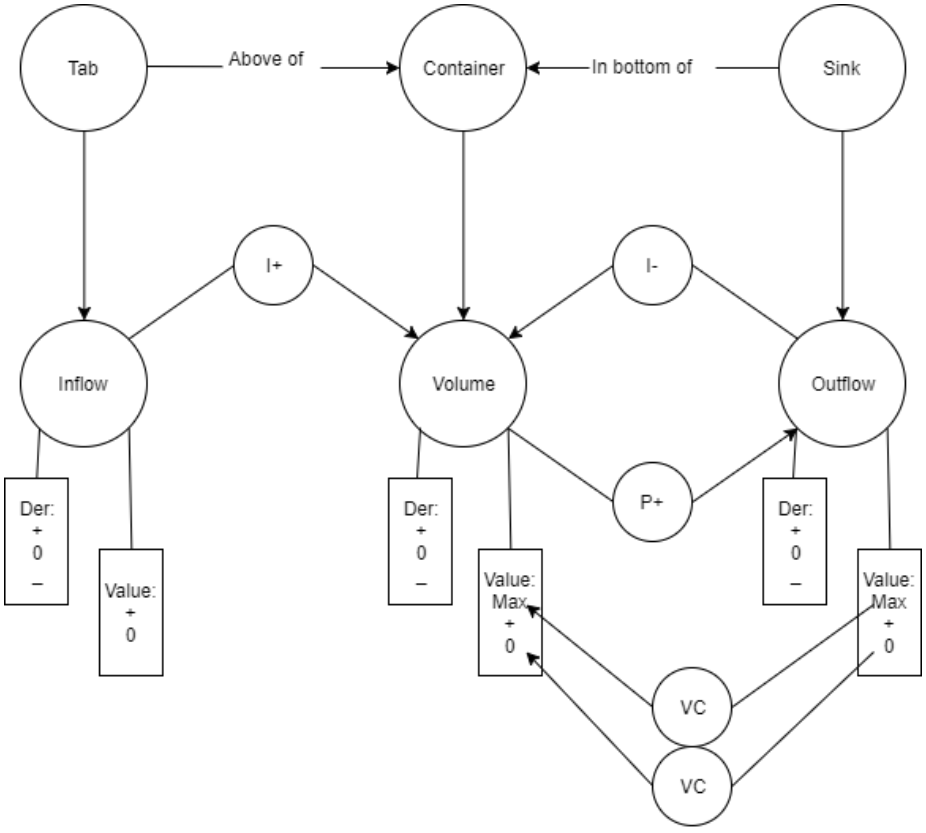

Figure 1: Causal Model showing the relationships between the three variables.

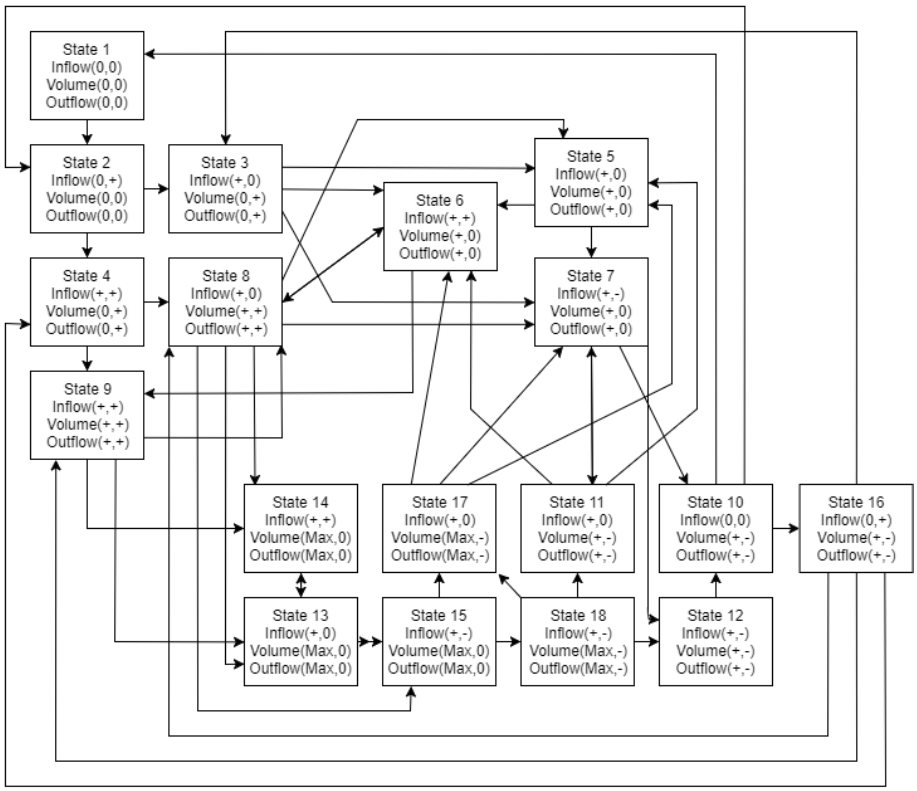

Figure 2: Graph showing all possible states and their transitions.

2 The Algorithm assumptions

The implementation of this algorithm was done in the python programming language in the Ipython notebook format of Jupyter notebook which is available as part of the anaconda python package. This section discusses how it works and the assumptions it makes.

2.1 Assumptions

There are two assumptions that have been made, Firstly we have assumed that the inflow behaves randomly except that it cant go from + to - and vice versa. This means that almost all nodes have at least two possible child nodes. When the inflow derivative in the previous state is 0, the derivative in the next state could be either -, 0 or +. When the previous inflow derivative was +, you can only get 0 and + and when the previous derivative was - you can only get - and 0.

The second assumption we have made is that the there is no state between the inflow magnitude increasing and the derivative of both the volume and outflow increasing. this means that the inflow derivative not only changes the inflow magnitude in the next state, but often at the same time changes the derivative of the other two variables indirectly. This makes sense from a logical standpoint as the moment there is inflow, you know that there will soon be volume and outflow as a result.

2.2 The Algorithm

The algorithm is at its base a breadth first search algorithm that explores the nodes in its stack and adds any not before seen node it finds to it until the stack is empty. There are two main rules used to generate the children nodes:

1. The first rule simply processes the derivatives into magnitude changes in the next state for all three variables.

2. The second rule handles the derivative for the volume and outflow which due to the proportionality between the two variables are always the same. As both the inflow (positively) and the outflow (negatively) magnitudes of the current state influences the derivative of the volume (and hence that of the outflow), we simply count the inflow as 1 if its not zero and the outflow as -1 if its not zero and add them together, a negative score results in a negative derivation , postive score into a positive derivation and when they cancel each other out then the derivation is set to 0. We also count the inflow derivation of the previous state as either -1,0 and 1 since this could possibly change the inflow amount without changing its symbol.

3. There are another few minor rules that prevent impossible situations from occurring. for example its impossible to have a negative inflow derivative when the magnitude is already 0, or a positive outflow/Volume derivation when they are already at max. (conceptually this can be seen as the water overflowing from the container and hence not increasing the volume nor the outflow)

After applying these rules you are left with an almost finished next state (called a prototype) , the only thing left to do is to copy it with each of the possible inflow derivatives (based on previous derivative). These copy's are the child nodes produced by the previous state.

An important note is that its possible for the algorithm to generate two possible prototypes. This happens when a magnitude can change to 0 or max. As all values between are represented with the symbol +, we have no way of knowing whether the derivative is large enough to change the magnitude to either 0 or max as such both possibility's need to be taken into account. This is why in these kind of situations two prototypes are generated, one in which the magnitude did change, and one for when it did not. Both of these prototypes will

then be expanded into up to 3 actual child nodes which means that a node can have up to 6 possible children.
After the stack has been emptied the algorithm has finished running and found all possible states, it will then proceed to pretty print the trace (which has been submitted as a txt file) and automatically generated network graph below.

3 4b

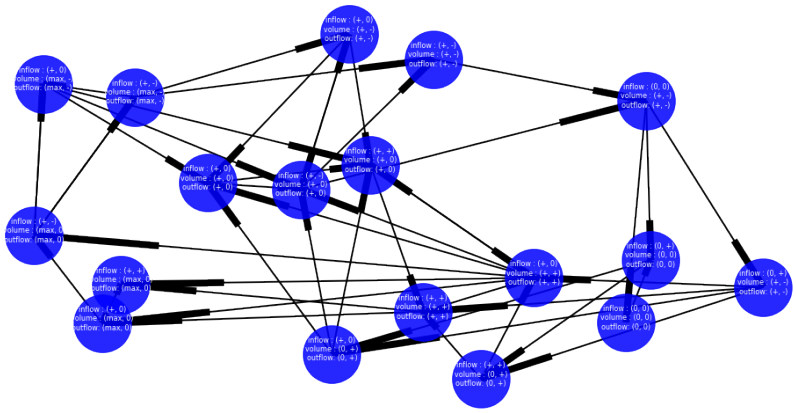

Figure 3: Graph made automatically with the code in Quantitive Reasoning.ipynb file.