Modelling a Bathtub system

Akash Raj Komarlu Narendra Gupta

University of Amsterdam

Abstract

In this report, I investigate the behaviour of dynamic container systems such as a bathtub. An algorithm for generating the states from the causal model explaining the system is described.

1. The Causal Model and Setup

The Bathtub modelled here is a container system which can be filled with water from a tap and can be emptied using a drain (which is always open). There is an outflow whenever the volume has a positive value. The following quantities have been modelled in the system - flow, volume, pressure, height, outflow. The causal model is shown in Figure 1.

1.1. Dependencies and Quantity spaces

The quantity space for inflow is $[0, +]$. The quantity space for volume, pressure, height and outflow are $[0, +, max]$. The derivatives for each of these quantities can be $[-, 0, +]$. The following dependencies are considered for this model -

- Inflow has a positive influence over Volume.
- Outflow has a negative influence over Volume.
- Volume is proportional to Height.
- Height is proportional to Pressure.
- Pressure is proportional to Outflow.

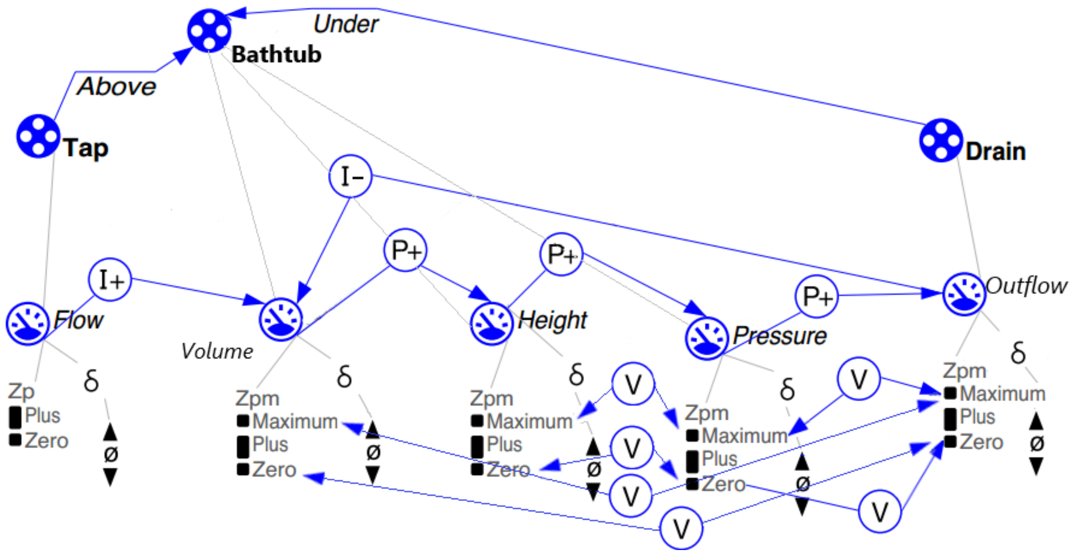

Figure 1: Causal Model showing the various quantities and their relationships

1.2. Value correspondences

I have assumed only bi-directional value correspondences. The following value correspondences are taken into consideration in the model -

- Volume and Outflow - The outflow is maximum when the volume is maximum. If volume is 0, then there is no outflow.
- Height and Pressure - Since pressure is directly proportional to height, when height is maximum, the pressure at the mouth of the drain is maximum. Similarly, when height is 0, there can be no Pressure (negligible)
- Pressure and Outflow - Outflow is maximum when pressure is maximum and outflow is 0 when there is no pressure.

I did not model the value correspondence between Height and Volume. This is because of my assumption that there can a be negligible amount of volume (at the beginning of inflow) and zero height.

1.3. Model Assumptions

The following assumptions have been made - For a moment, there can a non-zero volume with negligible height. Pressure at the mouth of the drain is a function of height only. Hence the logical order for Volume, Height and Pressure.

The start state is fixed with inflow magnitude being 0 and the derivative of inflow, 0. The behaviour of the system is explored when inflow is

1. Increasing
2. Random - The inflow can increase or decrease at anytime (with continuity restrictions enforced on it).

2. The algorithm

For simplicity, I have assumed that the initial state (inflow, volume, pressure, outflow, height, pressure = [0, 0]) of the model is fixed. But the algorithm stated below can be generalized for any initial state. The pseudocode for the algorithm is shown below

```
init step:    available_states = [first_state]
control step: already_visited_states = empty

loop step:
  while available_states is not empty:
    current_state = pop available_states
    if current_state is not in already_visited_states:
        1. states = get all possible states from this state
        2. find the valid state transitions
        3. append valid states to available_states list
        4. mark current_state as visited
```

Listing 1: Pseudocode to generate state graph

2.1. Generate all possible states from current state

For each available state, the algorithm generates all possible state transitions. The point transitions are given priority over interval transitions. If the first derivative is positive (or negative) and the quantity is 0 (or 'max'), then the entity's next state has a positive magnitude. All the combinations over the quantity space is considered here. This is done for each of the quantities for the current state.

For derivative changes, an additional constraint of continuity is applied. If the derivative is positive, then it can only become steady (0) or remain positive. If the current derivative is 0, it can either become positive or negative in the next state.

2.2. Invalid state transitions

Consider the previous state has inflow = [+, +] and volume = [+, +]. Then the next state cannot have volume = [+, 0]. That is, the derivative of volume cannot become 0 because the inflow has a positive magnitude. The generated state is checked for all such invalid state transitions and discarded if any such transition is found.

2.3. Causality, Rule propagation

For each of the above generated states, the dependency rules (proportionality, influence and value correspondences) are applied. For example, if the derivative of volume in the new state changes from 0 to positive, this should be reflected in the derivatives of height, pressure and outflow. The modified new state is considered as a legal state, If this is a unique state, it is added to the list of available states.

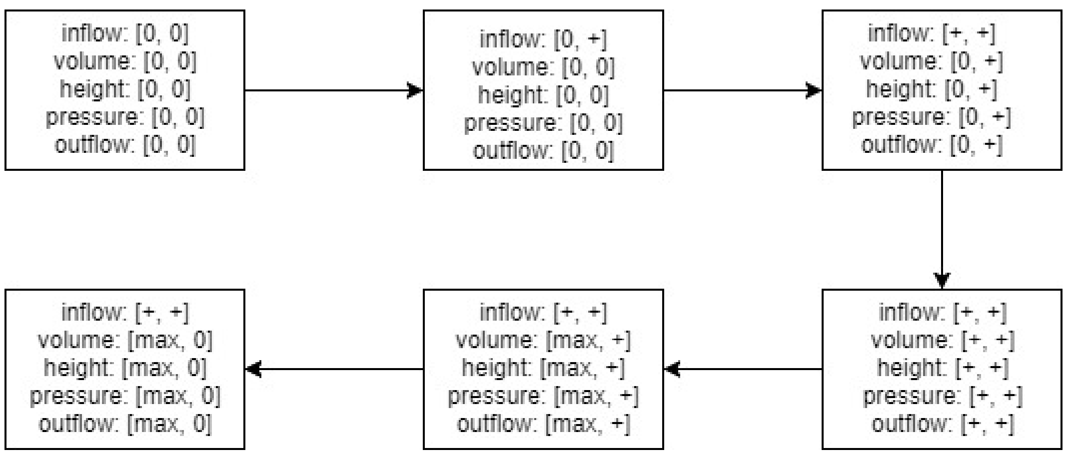

Figure 2: State graph generated when exogenous is an increasing function

3. Results, Conclusion and Future work

Various exogenous functions were used to control the inflow. The results for two of these functions are shown here. Figure 2 shows the state graph generated by the algorithm for an increasing function. The inflow starts from zero magnitude and always increases. There are 6 unique states. In the last state, the overflow of water from the bathtub is not considered and the volume is constant (maximum). This puts a restriction on the amount of pressure generated and hence the outflow.

The state graph generated for a random input is shown in Figure 3. The exogenous is a random function. The initial state is assumed to have 0 volume, and zero inflow. 14 distinct states are generated for this case. The simulation is stopped when the algorithm generates the initial state

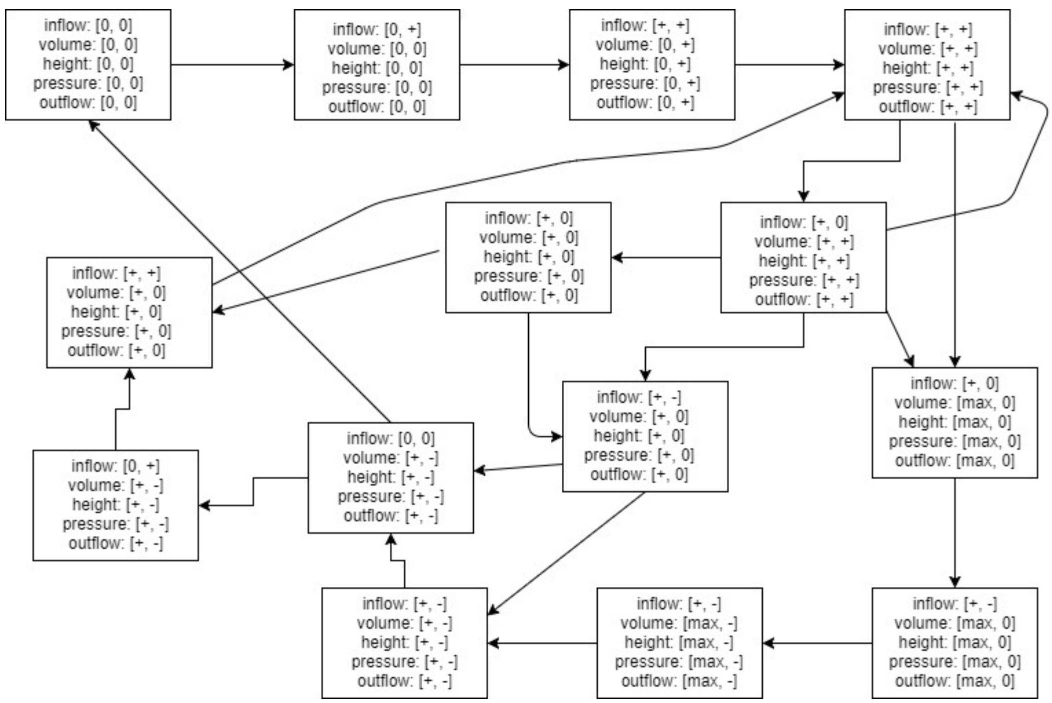

Figure 3: State graph generated when exogenous is a random function

In this report I investigated the behaviour of a container system which has an input and a drain. The resulting state graph contains states which are 'stable' and 'immediate'. Immediate states are those which involve transition from a point to an interval. During these changes, the derivative of the exogenous is assumed to not change.

My github repository contains all the analyses, results and the code (https://github.com/akashrajkn/bathtubs-and-rainbows).