inf300: Hybride Systeme

Water Supply Network

Semester Project Summer 2024

Martin Fränzle Paul Kröger

2024-08-09

1 Introduction

The overall idea of this assignment is to model a water supply network and to develop, model, and evaluate controllers for the pumps of that network. The network consists of wells, pumps, tanks, and consumers which are connected via pipes.

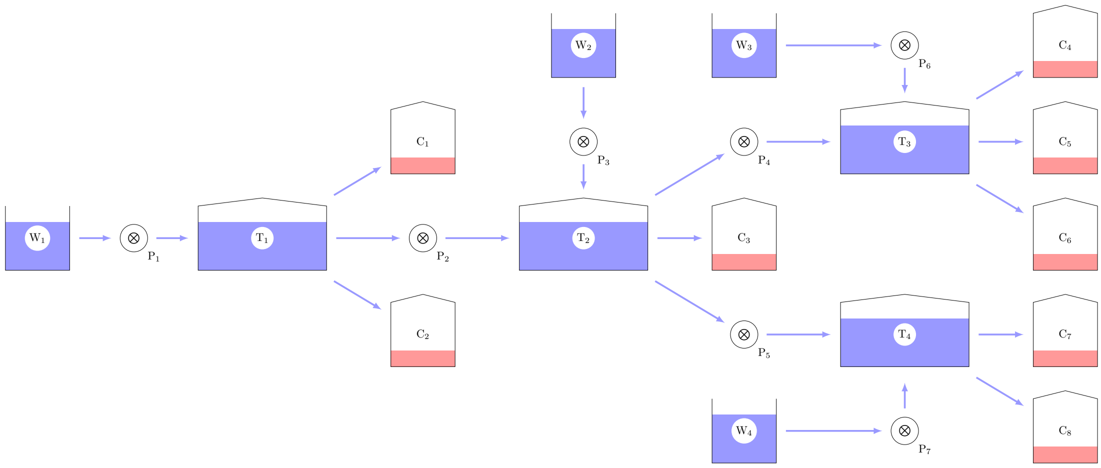

Figure 1: Simple water supply network

Well. A well W_i provides an unlimited amount of water which can be pumped at an arbitrary speed.

Pump. A pump P_i moves some amount of water over time from a well or tank into a tank. Every pump has a certain power p_i defining the maximum number of volume units (VU) per time unit (TU) that the pump can deliver.

Every pump can be switched on or off. When the pump is switched on, it changes its maximum flow rate $r_{i,\max}$ according to the following differential equation:

$$\dot{r}_{i,\max}(t) = \begin{cases} 1 - \frac{r_{i,\max}(t)}{p_i} & \text{iff the pump is on} \\ 0 - \frac{r_{i,\max}(t)}{p_i} & \text{else} \end{cases} \tag{1}$$

Please note that the pump never can pump more water than available. Hence, the actual flow rate r_i depends on $r_{i,\max}$ and the available amount of water. It is permissible for a pump to run dry.

Tank. A tank T_i is a very simple construct: it just has a certain capacity c_i and an initial amount of water $c_{i,\text{init}}$. A tank can (obviously) not have a negative amount of water. If its capacity is exceeded, the additional amount of water filled in will overflow. The filling level of the tank depends on all quantities of water filled in and drawn off.

Consumer. A consumer C_i consumes a certain amount of water. The demand of water is a sine wave specified by three parameters, i.e. the amplitude a_i of the wave, its frequency f_i, and a delay d_i. While amplitude and frequency have their obvious semantics, the delay is just a time delay of the sine wave. This means that the demand of water is constantly a_i for the first d_i time units and then starts to have the sinoid behaviour.

Please note that a sine wave typically ranges from $+a_i$ to $-a_i$. This would cause negative water consumption which is not intended behaviour. The lowest consumption should always be 0. Thus, the sine wave has to be adjusted to range from $2a_i$ to 0.

The actual consumption depends on the demand and the available amount of water: The consumer can consume only water that is available.

Pipe. Pipes do not pose any restrictions on the flow rate.

2 Assignment

Use Matlab Simulink/Stateflow to work on the following tasks:

1. First, model a pump, a tank, and a consumer as "subsystems". You can find an empty "Subsystem" block in the "Library Browser". Make sure that the blocks are configurable in the sense that you can specify the parameters p_i, $c_i, c_{i,\text{init}}$, a_i, f_i, and d_i for every pump, tank, and consumer individually.

   You don't need to but may use variables for these parameters. You can add such variables in the "Model workspace" or "Base workspace" in the "Model explorer" and refer to those variables by simply writing their names in the input fields of basic blocks, e.g. in the field "Constant value" of a "Constant"-block.

   You will need to add input and output ports within your subsystems. These can be found as basic blocks in the "Library browser" named "Input" and "Output".

   Make sure that your tank subsystem provides an information about the amount of overflowing water and that the consumer subsystem provides information about the lack of water, i.e. the amount of water that is demanded by the consumer but not available.

   In this setting, all pumps are always on. But prepare your pumps to be switched on and off.

i	2	2	3	4	5	6	7
p_i	15	20	5	7	3	5	5

Table 1: Pump parameters

2. Use your subsystems (via copy and paste) to model the simple water supply network from Fig. 1. Assume the parameters from Tab. 1 and Tab. 2 plus $c_i = 100$ and $c_{i,\text{init}} = 50$ for $i \in \{1, 2, 3\}$.

3. Measure the overall amount of water that was wasted by overflows.

4. Measure the overall amount of water that was demanded by consumers but not provided by the tanks.

5. Implement a controller for each pump that switches the pump on and of. The goal of the controllers is to minimise the amount of wasted water as well as the lack of water for the consumers. Try to save as much water as you can save and to reduce the lack of water as far as you can.

   The controllers should be identical for all pumps but may have individual parameter values for each pump or ignore certain inputs.

   Demonstrate your achievements by comparing simulation runs with and without control strategy.

6. **(Optional.)** Assume one of the urban settings from Fig. 2 or come up with one of at least similar complexity. Assume at least one tank and at least four consumers for each road segment (the space between intersections plus the space between intersections and end of roads). Wells are located at the end of roads only.

   Find a reasonable variation of parameters such that the water supply problem is not trivial, i.e. controlled pumps are required to save water and reduce the lack of water for consumers, it is not sufficient to always enable / disable certain pumps, and the tanks do initially not provide enough water for more than 5% of the length of the simulation run.

   Configure your pump controllers such that they save as much water as possible and the lack of water is reduced as much as possible. Demonstrate your achievements by comparing simulation runs with and without control strategy.

7. **(Optional.)** Extend your simple water supply network from Task 5 by one additional tank T_A of capacity $c_i = 100$ and initial filling level $c_{i,\text{init}} = 0$. Extend pumps P_2, P_4, and P_5 such that they are additionally connected to T_A. The pumps may

   a) fill water into the additional tank whenever the corresponding original target tank is in danger of overflowing instead of being stopped (every pump is allowed to fill only one tank at a time!), and

   b) take water from T_A whenever the corresponding original source tank is in danger of running out of water instead of being stopped (every pump is allowed to take water from only one tank at a time!).

   Pumps can of course be stopped whenever necessary.

   Try to find a control strategy that reduces the amount of wasted water and the lack of water for the consumers further.

i	2	2	3	4	5	6	7	8
a_i	3.0	8.0	5.0	5.0	2.0	7.0	2.0	5.0
f_i	0.4	0.6	0.2	1.0	0.4	0.1	0.9	0.6
d_i	2.0	0.5	4.0	3.0	4.0	5.0	7.0	4.0

Table 2: Pump parameters

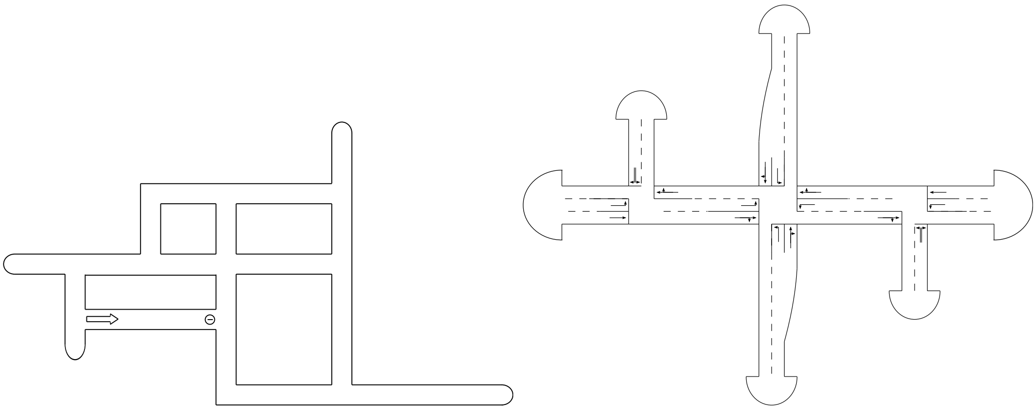

Figure 2: More complex urban scenarios

3 Report requirements

In order to pass the semester project successfully, it is mandatory to satisfy the following requirements:

- Write a well structured and clearly readable report. Document your ideas and design decisions. Don't be afraid of describing where you failed, in particular if you reached the limits of your tools for modelling, simulation, and/or verification.
- Comment on your approach, problems you experienced, and your approaches to solve them. Assess your design decisions and interprete your results.
- All model files have to be packed into a zip- or tar.gz-archive and have to be sent to the e-mail addresses provided at the end of this document. After unpacking, the various files must be clearly identifiable.
- Document all results and problems. For any observations from simulation, document the order of transitions you executed s.t. the simulation can be reproduced without problems.

The report should be written in a style similar to a scientific paper in German or English. It should thus include the following formal parts:

- A title page identifying the course, the assignment, and the participating students. For legal reasons, it is mandatory to include a page declaring the individual contributions of the participating students, naming any means of support used, and being hand-signed by *all* participants.
- A short introduction motivating the problem which you solve with your work.
- An evaluation of related work justifying the need for your own work.
- A detailed description of the developed model(s).

- The results of your Simulink/Stateflow runs, including an interpretation of the output.
- The sufficient relations for correctness of the protocol, together with supporting argumentation.
- Descriptions of any optional work.
- A conclusion in which you should summarise your findings obtained on this project.

This the semester project only simulates writing a scientific paper, you will probably not be able to really prove that no one did before what you are going to do with your project. Please take the following perspective: You were given the specification of components of a fictitious water supply network and you had to develop a controller (or a set of controllers if you use single controllers for each pump). You demonstrate the usefulness of your controller(s) on the toy example (Fig. 1) and possibly some larger use cases.

In this sense, the one option is to try to find related work that provides controllers that in principle could be used for your setting and argue (for example) why your way of implementing the controller is more useful.

A second option is to try to find related work that uses Simulink/Stateflow for similar tasks and a) to reason about the usefulness of Simulink/Stateflow for modelling and evaluating such systems and b) to compare your findings to the findings in the related work.

The goal wrt. related work is to train scientific writing and to come up with a coherent argumentation. Consequently, it will not be evaluated whether the argumentation for the need of your work is completely accurate for your grading, but make sure that you have a reasonable argumentation evaluating factually related work. To this end, you may also derive a leitmotif different to the two options mentioned above as long as the story is consistent.

The report should not exceed 15 pages excluding title page and the declaration about contributions. You may add an appendix with, e.g., additional screenshots or outputs.

You have to submit the model files etc. as well as an electronic version of your report in the PDF file format to *both* of the following two addresses:

`martin.fraenzle@informatik.uni-oldenburg.de`
`paul.kroeger@informatik.uni-oldenburg.de`

The report and the models plus sublementary files have to be submitted until:

September 8, 2024

This is a hard deadline. No reports handed in later will be considered unless consent for late submission has been obtained well before the deadline. If you for some very good reason cannot meet the deadline you must contact us in due time before September 2, 2024.

General information

The semester project supposed to be a group projects of 2 or 3 students. Exceptions are possible but require prior agreement.

Please note:

- The evaluation will take place as announced (based upon the reports AND the colloquium).
- You have to defend your results in a final colloquium.
- The structure of the finial colloquium is as follows:
  - You present your project in a presentation of about 20 minutes. Make that the presentation is equally distributed over all group members. Electronic slides are appreciated.
  - Your presentation might be interrupted by the examiners to clarify certain points or to explore your deeper understanding.
  - After the presentation, there will be a discussion of about 30 minutes. Questions by the examiners may be related to your project but may also cover the contents of the lecture.
- If you exchanged content or knowledge with other groups on a smaller scale, you have to state this (precisely what was exchanged into which direction) on the declaration page as well as all other sources and utilities have to be stated in general. A collaboration on a larger scale cannot be accepted!

If you have any further questions, contact us via e-mail to:

`paul.kroeger@informatik.uni-oldenburg.de`

In particular do not make any assumptions about formalities. Instead, please contact us s.t. we can eliminate any ambiguities.

Also note that clarifications, FAQs, and practical details may be put in the Stud.IP course. You should consult this if you encounter problems.