Modeling Fluid Flow in RDDL for Simulation of Flow Control Systems

Donovan Bisson[1]

School of Computing, Queen's University, Kingston, Canada

Abstract
Coming soon...

1. Introduction

In process control systems, the full state space is typically known but only partially observed. Important values are monitored by sensors in pertinent locations to give a sense of the system at large, but availability of sensor data for a system tends to become more sparse as the system expands. State estimation can fill knowledge gaps by interpolating how the system would function in given conditions. However, such state estimation methods are domain-specific; they need a model that can represent features of the system and simulate the underlying rules of the system.

With respect to control systems, such models are called model-predictive control (MPC) systems, and they are extremely useful for non-linear control problems. One such example would be flow control problems - moving fluid through a system of pipes and reservoirs. These kinds of problems include water treatment and distribution, wastewater treatment, and chemical industry processes. Solving these problems often requires a robust modeling scheme, both for simulating the system to build a policy and for estimating action effects in real time.

The goal of this model is to act as a realistic domain representation of flow control systems, one that can both act as a simulator for training an offline policy and a way to estimate states for an online policy.

2. Background

2.1. Overview - Model Predictive Control

https://link.springer.com/article/10.1007/s00170-021-07682-3

Robust analysis of model-predictive control. Key concepts:

- MPC systems model a state representation of a system, which is used to predict the effects of actions within the system. This turns control problems into constrained temporal optimization problems (great for planning!)
- MPC can control non-linear models that traditional control systems struggle with. Very valuable in process industries.
- Broadly speaking, this article is here to define MPC and to justify use of a planning model for process-based problem spaces such as flow control. Nothing specifically RDDL related, but MPC systems have found use all over the place from my reading - I'll specify some examples in the final report.

2.2. Flow Networks

https://dspace.mit.edu/bitstream/handle/1721.1/49424/networkflows00ahuj.pdf

Outlines how flow networks are defined, as well as common problems they can be used to solve (max flow, min-cost flow, etc.).

(TODO - explain the notation of a flow network (nodes, arcs, source/sink, flow conservation).

2.3. Types of Pumps, Formulas, Pump Curves

- Positive-displacement pumps - move fixed volumes of fluid per cycle, volumetric flow mostly independent of pressure (above threshold, called "Net Positive Suction Head")
- Centrifugal pumps - use impellers to generate flow via kinetic energy, ratio of cycle rate:volumetric flow becomes a function of fluid pressure.
- Pump curves - Represented as a curve, but more of a topology. Pressure on one axis, flow rate on the other, mapping speed of the pump forms a surface in 3D space. This is really important, as systems that use pump speed as the control variable and flow rate/pressure as the process variable need to compensate for changes in one or the other due to environmental factors (turbulence) and upstream/downstream changes.

CISC-813: Automated Planning – Final Project

3. A Planning Model for Partially-observable Fluid Control Systems

Model formulation: RDDL
Controller to build: Deep reactive policy (using JaxDeepReactivePolicy)
Format:
The model will be structured as a flow network, allowing for both closed networks (where resources within the network is conserved) and open networks (where the system can receive from and/or donate resources). Nodes in the network will act as capacitors (specifically fluid tanks), able to store resources (fluid) up to a static capacity. Arcs between nodes will represent paths of controlled flow (pipes with flow generated/maintained by pumps).
Objects and associated values:

- Tanks - CAPACITY, holding, inflow, outflow
- Pumps - TYPE(@pd/@cf), DRAWS(from, to), Hz, pump-flow, (optional) target-flow
- Pressure sensors(pump, @start/@end) - (TODO) Gives a pressure value somewhere between the arc entrance/exit and the pump. These will be most important with centrifugal pumps, where the function of cycle rate to flow is pressure-dependent.

4. Evaluation

Coming soon...

5. Related Work

5.1. IPPC - Reservoir (cont.)

https://github.com/pyrddlgym-project/rddlrepository/tree/main/rddlrepository/archive/competitions/IPPC2023/Reservoir
Strong foundation for RDDL implementation. Key differences:

- System flow is *node-centric* - the release() action cannot control proportion of released water sent to 2+ downstream reservoirs (instead simply proportional). Our model needs to control fluid movement between nodes directly by operating on connections (making it *arc-centric*).
- Stochastic addition/reduction of Reservoir level (by rain and evaporation respectively). Volumetric flow must be conserved throughout our model, so stochasticity will need to be introduced to the simulation elsewhere (specifically through flow resistance due to pressure).
- All levels are known by the agent (fully-observable). We need to modify the model to function off incomplete information provided by sensors, to provide a partial state representation the model can build from.

6. Summary

6.1. Future Work

Coming soon...