Readme: Implementation progress and what's next

Sept 2, 2018

Margaret Chapman

See also: submitted-Sustech2018.pdf

• For a 1-dimensional LTI system (see Setup_LTI_Dynamics.m),

we have computed $\underline{U_y^r} := \left\{ x \mid \min_{\pi} CVaR_y\left[\sum_{k=0}^{N} e^{m\cdot g(x_k)} \middle| \pi, x_0 = x \right] < e^{mr} \right\}$ (1) via dynamic programming, and

$\underline{S_y^r} := \left\{ x \mid \min_{\pi} CVaR_y\left[\max_{k=0,\dots,N} g(x_k) \middle| \pi, x_0 = x \right] < r \right\}$ via brute force enumeration,

for several $y \subset (0,1)$ and r.

(see Results_LTISystem\compare_DPsoftmax_mis10_vs_BFmax.fig; The figure was generated using the 3rd cell of Script_Compare.m. The previous 2 cells were used to determine grid spacing and the softmax parameter, m.)

• Recall that $U_y^r \subseteq S_y^r$, and our goal is to find a good approximation to S_y^r, and we do this by computing U_y^r via dynamic programming.

• Next, we want to extend the code to a more meaningful system, and compute U_y^r and S_y^r for this system. (branch off Master)

System - pond 1 of our Sustech submission (see figure 2 of Submitted_Sustech2018.pdf) equation 1a, equation 2

$$\dot{x} = \frac{w - q_p(x,u)}{A} \quad (1a)$$

x = water elevation in pond [ft] - state

u = valve setting $\in \{0,1\}$ - control input

w = surface runoff due to rain $\left[\frac{ft^3}{s}\right]$ - disturbance

A = surface area of pond $[ft^2]$

$$q_p(x,u) = \begin{cases} C_d \pi R^2 u \sqrt{2g(x-Z)} & \text{if } x \geq Z \\ 0 & \text{if } x < Z \end{cases} \quad \text{outflow through valve} \quad (2)$$

$\left[\frac{ft^3}{s}\right]$

- The constants are provided in Table I (e.g., $A = 28{,}292\ ft^2$) of the SuStech submission
- The constraint set is $\mathcal{X} = (0\,ft, 5\,ft)$. In other words, if $x > 5\,ft$, then the pond has flooded.
- Water enters the pond due to surface runoff (w), and leaves the pond thru the valve. The valve has 2 settings (open, $u=1$) or (closed, $u=0$).
- The 1st step is to discretize the dynamics to get,

$$x_{k+1} = f(x_k, u_k, w_k), \quad \text{where } \Delta t = 5\text{ min (perhaps)}.$$

$k = 0, 1, \ldots, N-1$

x_k = water level at time k, units of ft

u_k = valve setting at time k (0 or 1, no units)

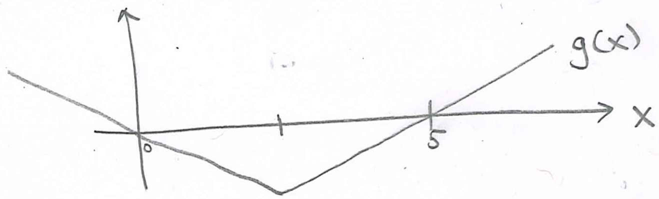
w_k = amount of water entering the pond during $[k, k+1)$, units of ft^3

- We will assume that the probability of w_k taking on a particular value is known at each time point. This will be estimated from a design storm (which is a synthetic storm often based on historical data that is used to design storm water systems).
- While I'm not sure what these probabilities are yet, we should have $\mathbb{E}[w_k] \approx 45000\ ft^3$ if $\Delta t = 5$ min, and the time duration from $k=0$ to $k=N$ is 4 hours.
- For now, let's make up a (time-invariant) distribution $P\{w_k = j\} = p_j$, where the expected value of w_k is about $45000\ ft^3$, and has about 10 samples, so that $\sum_{j=1}^{10} p_j = 1$. We can use this distribution as a placeholder until we get a more accurate one.

- We want the distribution to have a large number of samples, if possible.

- Because the constraint set, $K = (0, 5\,ft)$, the signed distance function will be

$$g(x) = |x - 2.5| - 2.5 .$$



- One needs to define the discretized state space (xs), and the soft-max parameter (m).

- I first started with $m=1$, then played around with grid size (xs, ls). I chose a grid size see top line page 1 so that the computation of U_y^r via dynamic programming and the computation of U_y^r via a brute force enumeration were relatively close.

perhaps use Monte Carlo instead

- After selecting the grid, then I played around with m. Theoretically, bigger m is better, but computationally big m may be problematic.

- Overall, we want to get a figure (like Results_LTI System\compare_DPsoftMax_mis10_vs_BFmax.fig), see top line page 1 showing U_y^r and S_y^r for several y and several r, for the pond system. including $r=0$
We have the results already for the LTI system.

- Because computation of S_y^r via brute force enumeration may not be possible, instead we'll use a Monte Carlo approach (perhaps).

- We'll use our Dynamic programming algorithm to get U_y^r for the pond system.
- In particular, we should show U_y^r, S_y^r for $r = 0$.

- We would like to generate U_y^r, S_y^r for different instances of our pond system to demonstrate how risk-senstive reachability can be used for design of infrastructure in the presense of uncertainty.

Instance 1 (R, passive): $R = 1/3$ ft (pond outlet radius), $u \in \{1\}$ (valve always open)

Instance 2 (R, active): $R = 1/3$ ft, $u \in \{0, 1\}$ "active control" (valve can be either open or closed based on what's optimal) ← Let's start with this instance.

Instance 3 (2R, passive): $R = 2/3$ ft (larger pond outlet radius), $u \in \{1\}$

- The key challenges that I faced while coding up the LTI example:

① choosing min(xs), max(xs) and restrictions on the control input near the boundary of the discretized state space (xs) (to prevent the DP algorithm from interpolating outside the grid)

② making sure that the brute force enumeration code (Main-BruteForce.m) generated the same scenario tree as the DP algorithm (Main-DynProg.m).

- To understand the big picture of the code, please look at

  - Main-DynProgram.m
  - Main_BruteForce.m
  - Script_Compare.m,

and the headers of every .m file in Matlab_Code.

- Please branch off the master node.