Discussion of the Simulink model `FOQL_PDE_OutputFeedback.slx`

Anton Ponomarev*[1], Julian Hofmann†[2], and Lutz Gröll‡[2]

[1]Saint Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg, 199034 Russia
[2]Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

April 19, 2020

The file `FOQL_PDE_OutputFeedback.slx` available from the repository [1] simulates a *heat exchanger* model under output feedback control. Forced-flow steam-jacketed tubular heat exchanger is a plant where fluid flows through a tube heated from the outside by steam. It can be modeled by the following partial differential equations (PDE) taken from [2]:

$$\frac{\partial T(t,x)}{\partial t} + v\frac{\partial T(t,x)}{\partial x} = -\alpha\big(T(t,x) - T_{\text{jacket}}\big), \tag{1a}$$

$$T(0,x) = 1 - \mathrm{e}^{-5x}, \tag{1b}$$

$$T(t,0) = 0, \tag{1c}$$

$$y(t) = T(t,1). \tag{1d}$$

where T is the temperature within the tube at the coordinate $x \in [0,1]$, T_{jacket} the jacket temperature, v the fluid velocity (manipulated variable), y the boundary output, and α a positive constant representing the product of jacket heat conductivity and jacket area.

The fluid temperature $T(t,1)$ at the tube exit is controlled by the output feedback [2]

$$v(y(t)) = k_1\left(\big(y(t) - y_d\big) + \frac{1}{\tau_I}\left(\int_0^t \big(y(\theta) - y_d\big)\,\mathrm{d}\theta + v_{I,0}\right) - \frac{\alpha}{k_2}\big(y(t) - T_{\text{jacket}}\big)\right) \tag{2}$$

which can be interpreted as PI controller with a feedforward term. Here, y_d denotes constant desired temperature at the tube exit. For simulation purposes, we use the parameters given in [2], namely, $k_1 = 5/3$, $\alpha = 1$, $T_{\text{jacket}} = 10$, $k_2 = 5$, $\tau_I = 2$, and $y_d = 3$. The initial value $v_{I,0}$ of the integrator part is chosen such that $v(0) = 0$, i.e.,

$$v_{I,0} = \tau_I\left(y_d - y(0) + \frac{\alpha}{k_2}\big(y(0) - T_{\text{jacket}}\big)\right) \tag{3}$$

where $y(0) = 1 - \mathrm{e}^{-5}$ which follows from (1b) and (1d).

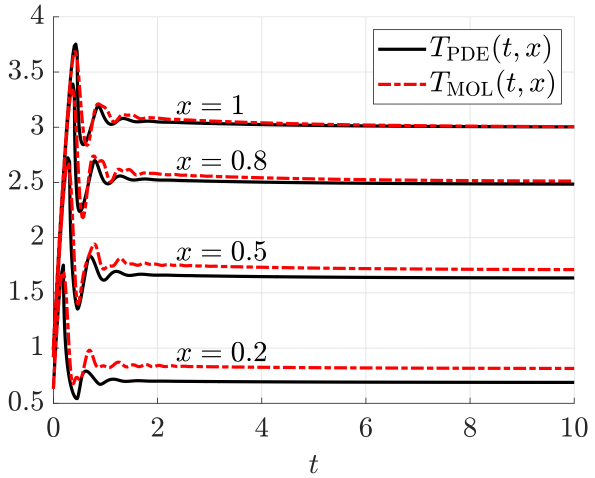
The model `FOQL_PDE_OutputFeedback.slx` employs the `FOQL PDE` block which solves the PDE by the method of characteristics. In order to verify the results of simulating the closed loop (1), (2) using the `FOQL PDE` block, a scheme based on the method of lines (MOL) is implemented. The basic idea of the MOL is spatial discretization of the PDE. By approximating the spatial derivative using finite differences (FDs), an ODE system w.r.t. time t is obtained [3, 4]. Here, we approximate the spatial derivative in (1a) with central FDs (second-order approximation), such that

$$T_{1,x}(t) := \frac{\partial T(t,x)}{\partial x}\bigg|_{x_1} \approx \frac{\frac{1}{2}\left(T_1(t) + T_2(t)\right) - T(t,0)}{\Delta x}, \tag{4a}$$

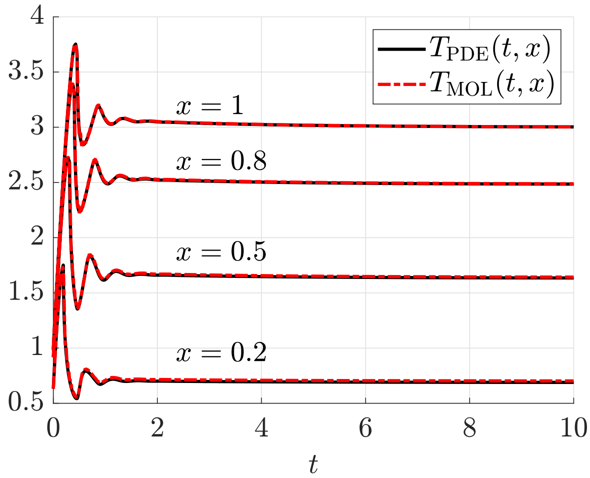
$$T_{i,x}(t) := \frac{\partial T(t,x)}{\partial x}\bigg|_{x_i} \approx \frac{T_{i+1}(t) - T_{i-1}(t)}{2\Delta x}, \quad i = 2,3,\dots,K, \tag{4b}$$

$$T_{K+1,x}(t) := \frac{\partial T(t,x)}{\partial x}\bigg|_{x_{K+1}} \approx \frac{T_{K+1}(t) - T_K(t)}{\Delta x} \tag{4c}$$

*anton.pon.math@gmail.com
†julian.hofmann@kit.edu
‡lutz.groell@kit.edu



(a) $K = 10$



(b) $K = 100$

Figure 1: Solution $T(t,x)$ of (1)-(2) taken at different points in space obtained via the `FOQL PDE` block and the MOL with order K.

where $T_i(t) := T(t, x_i)$, $x_i = (i-1)\Delta x$ and $\Delta x = 1/K$. Thus, we have K segments. Alternatively to (4a), common discretizations of the left boundary are [5]

$$T_{1,x}(t) := \frac{\partial T(t,x)}{\partial x}\Big|_{x_1} \approx \frac{T_2(t) - T(t,0)}{2\Delta x}, \tag{5a}$$

or

$$T_{1,x}(t) := \frac{\partial T(t,x)}{\partial x}\Big|_{x_1} \approx \frac{T_1(t) - T(t,0)}{\Delta x}. \tag{5b}$$

However, as both (5a) and (5b) produce slightly less accurate results than (4a) for the examples of this section, we use (4a) henceforth. With $y(t) = T_{K+1}(t)$ in (2), the PDE model (1) is approximated by the ODE system

$$\frac{\mathrm{d}T_i(t)}{\mathrm{d}t} = -v\big(T_{K+1}(t)\big)T_{i,x}(t) - \alpha\big(T_i(t) - T_{\text{jacket}}\big), \quad i = 1, 2, \ldots, K+1, \tag{6a}$$

$$T_i(0) = 1 - \mathrm{e}^{-5(i-1)\Delta x} \tag{6b}$$

which has straightforward Simulink implementation.

Fig. 1 shows the simulation results of the closed loop (1), (2) using two methods: the `FOQL PDE` block and the MOL. It can be concluded that the MOL results converge to the ones obtained by the `FOQL PDE` block for increasing K. Moreover, note that common numerical problems of MOL solutions using FDs (numerical dispersion and diffusion) do not occur in this example because the initial function (1b) is monotonically increasing. This condition guarantees that the MOL solution does not exhibit wiggles (artifical oscillations) [6].

References

[1] A. Ponomarev, J. Hofmann, and L. Gröll, "PDECharactSimulink." `https://github.com/antonponmath/PDECharactSimulink`. [Online].

[2] H. Shang, J. F. Forbes, and M. Guay, "Feedback control of hyperbolic pde systems," *IFAC Proceedings Volumes*, vol. 33, no. 10, pp. 533–538, 2000.

[3] W. E. Schiesser, *The numerical method of lines: Integration of partial differential equations*. Elsevier, 2012.

[4] A. V. Wouwer, P. Saucez, W. Schiesser, and S. Thompson, "A matlab implementation of upwind finite differences and adaptive grids in the method of lines," *Journal of computational and applied mathematics*, vol. 183, no. 2, pp. 245–258, 2005.

[5] M. B. Carver and H. Hinds, "The method of lines and the advective equation," *Simulation*, vol. 31, no. 2, pp. 59–69, 1978.

[6] M. Zijlema, *Computational modelling of flow and transport*. Faculty of Civil Engineering and Geosciences, Delft University of Technology, January 2015.