Numerical optimal control for delay differential equations: A simultaneous approach based on linearization of the delayed state

Tobias K. S. Ritschel, Søren Stange

Department of Applied Mathematics and Computer Science,
Technical University of Denmark

17.30 – 17.50, WeC1.4, June 25, 2025
European Control Conference, Thessaloniki, Greece

Research question

Delay differential equation with absolute time delay

$$\dot{x}(t) = f(x(t), x(t-\tau))$$

Linearize delayed state

$$x(t-\tau) \approx x(t) - \dot{x}(t)\tau$$

Question: Can we develop numerical optimal control methods based on this approximation?

Optimal control problem

$$\min_{\{u_k\}_{k=0}^{N-1}} \quad \phi = \phi_x + \phi_{\Delta u}$$

$$\phi_x = \int_{t_0}^{t_f} \Phi(x(t), u(t), d(t), p)\, \mathrm{d}t,$$

Tracking/economic objective

$$\phi_{\Delta u} = \frac{1}{2} \sum_{k=0}^{N-1} \frac{\Delta u_k^T W_k \Delta u_k}{\Delta t}$$

Rate-of-movement penalization

subject to

$$x(t) = x_0(t), \quad t \in [t_0 - \tau_{\max}, t_0],$$

Initial state function

$$\dot{x}(t) = f(x(t), z(t), u(t), d(t), p),$$

Delay differential equations

$$z(t) = \begin{bmatrix} r_1(t - \tau_1) \\ \vdots \\ r_m(t - \tau_m) \end{bmatrix},$$

Memory states

Time delays are input-dependent

$$\tau_i = \tau_i(u(t))$$

$$r_i(t) = h_i(x(t), p),$$

Delayed variables

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[,$$

$$d(t) = d_k, \quad t \in [t_k, t_{k+1}[,$$

Zero-order-hold (ZOH) parametrization

$$x_{\min} \le x(t) \le x_{\max},$$

State bounds

$$u_{\min} \le u_k \le u_{\max},$$

Input bounds

Linearize delayed variables

Linearize delayed state

$$x(t-\tau_i) \approx x(t) - \dot{x}(t)\tau_i$$

Approximate system

$$\dot{x}(t) = f(x(t), z(t), u(t), d(t), p),$$

$$z(t) = \begin{bmatrix} r_1(t) \\ \vdots \\ r_m(t) \end{bmatrix},$$

$$r_i(t) = h(x(t) - \dot{x}(t)\tau_i, p), \quad i = 1, \ldots, m$$

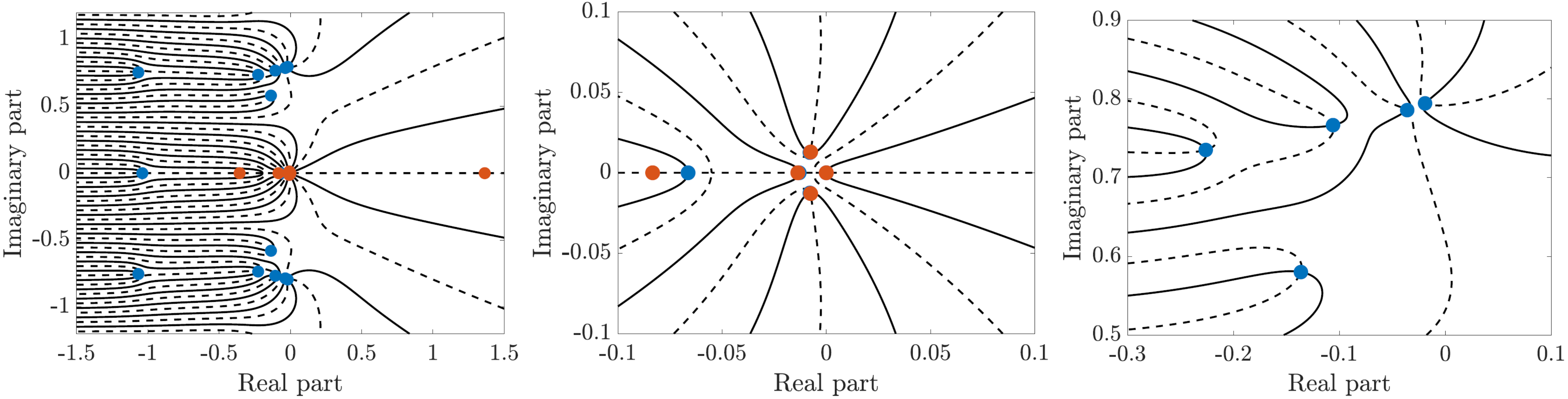
Original characteristic equation

$$\det\left(\lambda I - \frac{\partial f}{\partial x} - \sum_{i=1}^{m} \frac{\partial f}{\partial z}\frac{\partial z}{\partial r_i}\frac{\partial r_i}{\partial x} e^{-\lambda \tau_i(u_s)}\right) = 0$$

Approximate characteristic equation ($e^{-x} \approx 1 - x$)

$$\det\left(\lambda I - \frac{\partial f}{\partial x} - \sum_{i=1}^{m} \frac{\partial f}{\partial z}\frac{\partial z}{\partial r_i}\frac{\partial r_i}{\partial x} (1 - \lambda \tau_i(u_s))\right) = 0$$

The approximate system may be unstable



Main conclusion: The numerical approach must be able to handle unstable systems

- Multiple-shooting methods
- Simultaneous methods

Discretization w. Euler's implicit method

Euler's implicit method

$$x_{k,n+1} - x_{k,n} - f(x_{k,n+1}, z_{k,n+1}, u_k, d_k, p)\Delta t_{k,n} = 0$$

Approximate linearized delayed state

$$v_{i,k,n+1} = x_{k,n+1} - \frac{x_{k,n+1} - x_{k,n}}{\Delta t_{k,n}}\tau_i(u_k)$$

Main conclusion: The <u>dependencies</u> in the discretized equations are fixed (independent of u_k)

Transcribed optimal control problem (nonlinear program)

Nonlinear program

$$\min_{\{\{x_{k,n+1}\}_{n=0}^{M-1}, u_k\}_{k=0}^{N-1}} \quad \psi = \psi_x + \phi_{\Delta u}$$

subject to

$$R_{k,n}(x_{k,n+1}, x_{k,n}, u_k, d_k, p) = 0,$$
$$x_{\min} \leq x_{k,n+1} \leq x_{\max},$$
$$u_{\min} \leq u_k \leq u_{\max},$$

Discretized objective

$$\psi_x = \sum_{k=0}^{N-1} \sum_{n=0}^{M-1} \Phi(x_{k,n+1}, u_k, d_k, p) \Delta t_{k,n}$$

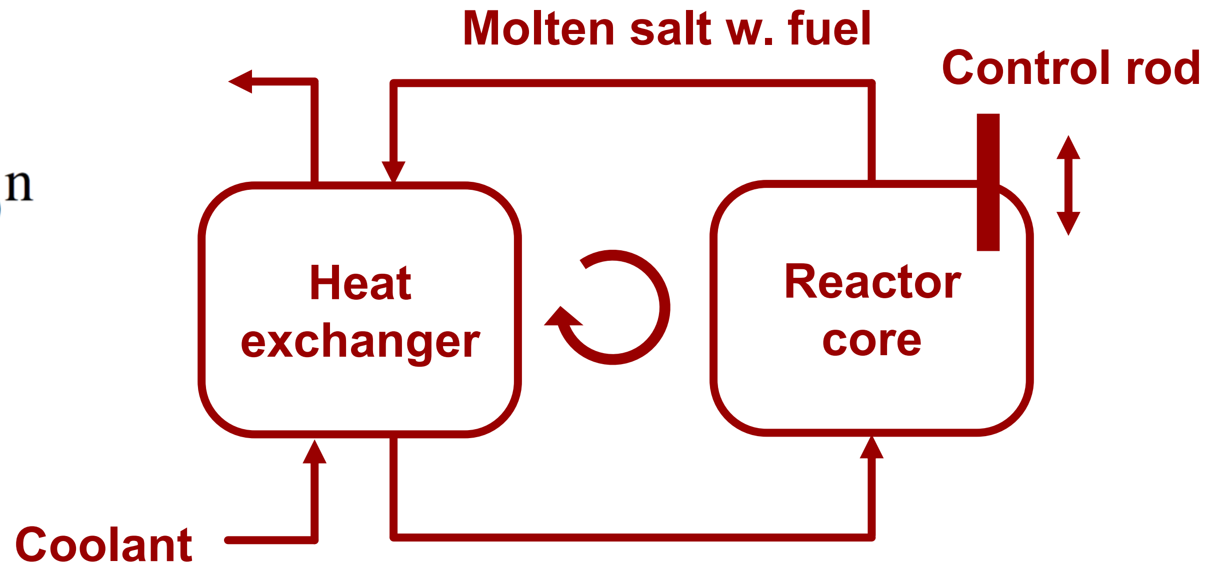
Molten salt nuclear reactor

Fission reaction

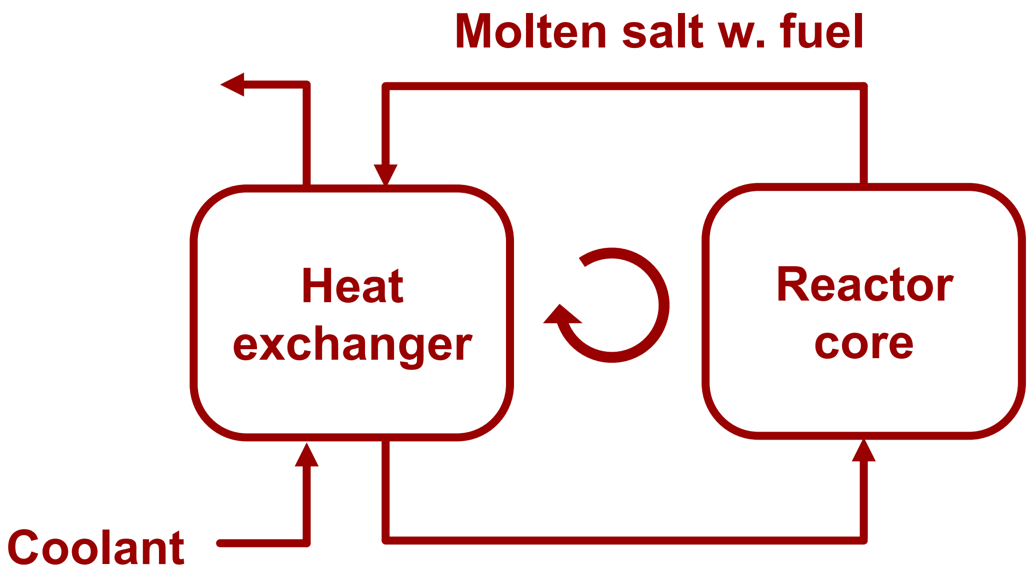
$$ {}^{1}_{0}\mathrm{n} + {}^{235}_{92}\mathrm{U} \xrightarrow{\text{neutron absorp.}} {}^{236}_{92}\mathrm{U} \xrightarrow{\text{fission}} \sum_{i=1}^{N_p} \nu_i \, {}^{a_i}_{z_i}\mathrm{E}_i + \nu_0 \, {}^{1}_{0}\mathrm{n} $$

Delayed release of a neutron

$$ {}^{a_i}_{z_i}\mathrm{E}_i \xrightarrow{\text{neg. beta decay}} {}^{a_i}_{z_i+1}\mathrm{E}_i^* \xrightarrow{\text{neutron emission}} {}^{a_i-1}_{z_i+1}\mathrm{E}_i^* + {}^{1}_{0}\mathrm{n} $$



Molten salt nuclear reactor model



Mass balance equations

$$\dot{C}_i(t) = (C_{i,in}(t) - C_i(t))D(t) + R_i(t), \quad i = 1, \ldots, N_g$$

$$\dot{C}_n(t) = R_n(t)$$

$$\dot{\rho}_{th}(t) = -\kappa \dot{T}_r(t)$$

Delayed variable

$$C_{i,in}(t) = C_i(t - \tau(t))e^{-\lambda_i \tau(t)}$$

Energy balances

$$\dot{T}_r(t) = \frac{f_r(t)}{m_r}(T_{hx}(t - \tau(t)/2) - T_r(t)) + \frac{Q_g(t)}{m_r c_P}$$

$$\dot{T}_{hx}(t) = \frac{f_{hx}(t)}{m_{hx}}(T_r(t - \tau(t)/2) - T_{hx}(t)) - \frac{k_{hx}}{m_{hx} c_P}(T_{hx}(t) - T_c)$$

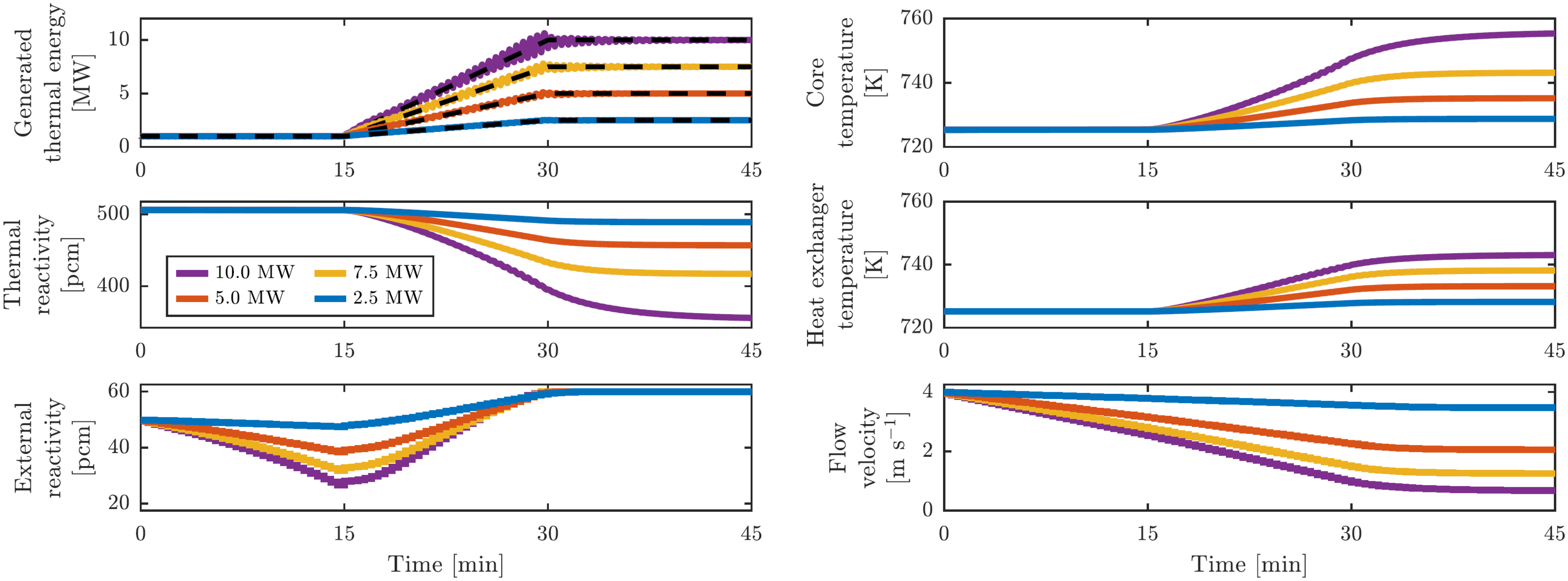
Other variables

$$\tau(t) = \frac{L}{\boxed{v(t)}}$$

$$\rho(t) = \rho_{th}(t) + \boxed{\rho_{ext}(t)}$$

$$Q_g(t) = Q_{g,0}\frac{C_n(t)}{C_{n,0}}$$

Numerical example: Optimal ramping



Summary + future work

Numerical optimal control w.
<u>input-dependent delays</u>

Linearize delayed state

Simultaneous approach

Approximate the state trajectory
w. polynomials

Thank you

Appendix: Molten salt nuclear reactor model

$$D(t) = F(t)/V$$

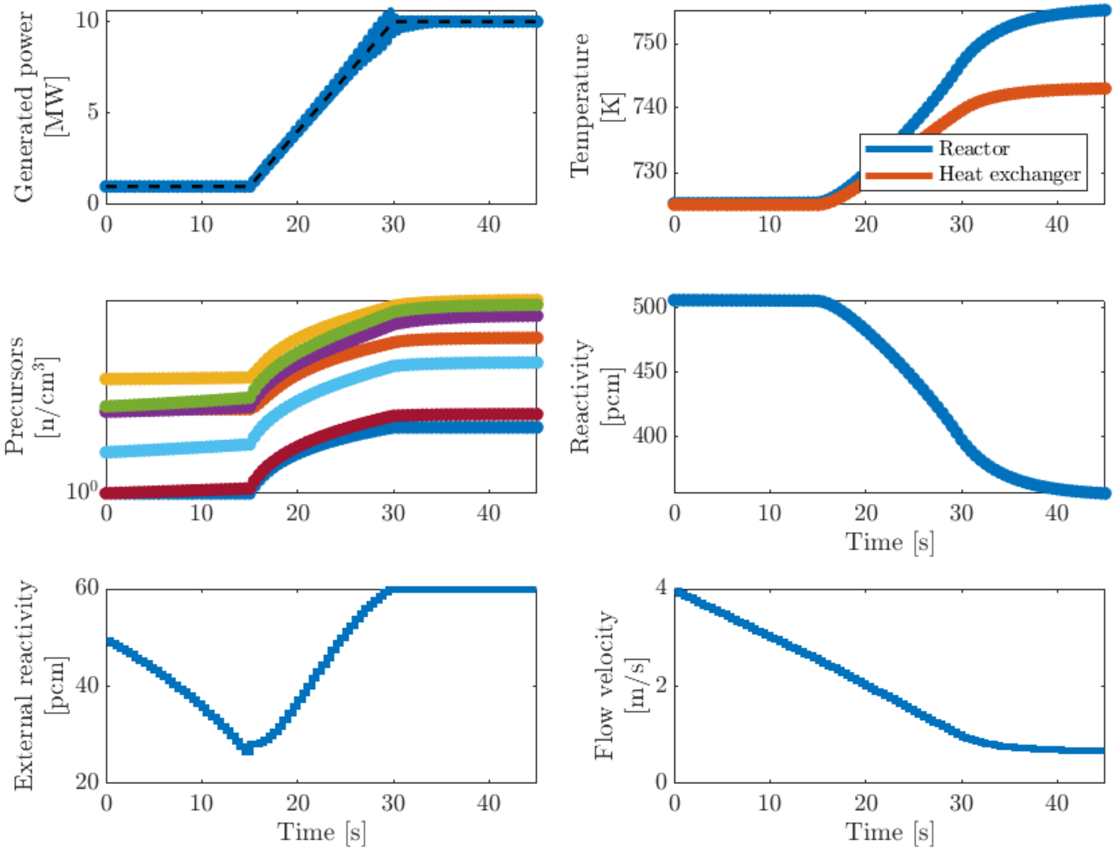
$$F(t) = Av(t)$$

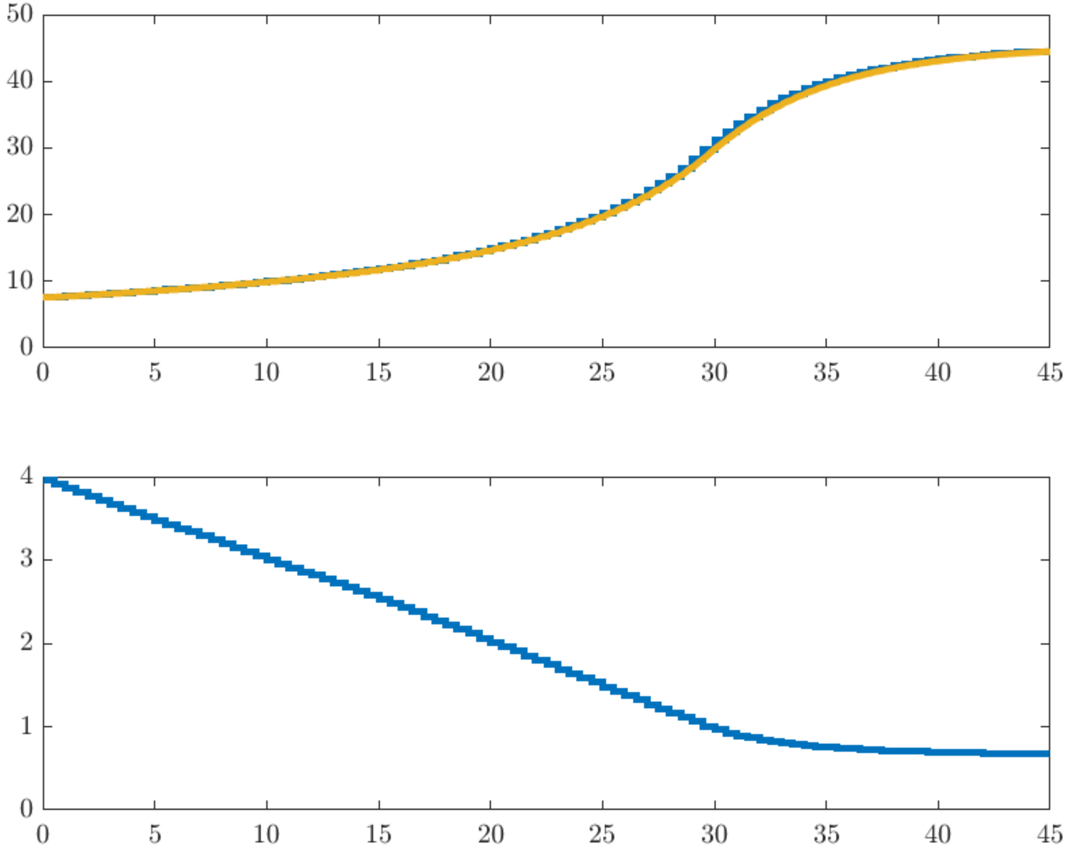
$$f_r(t) = f_{hx}(t) = \rho_s Av(t)$$

$$R(t) = S^T(t)r(t)$$

$$S(t) = \begin{bmatrix} -1 & & & 1 \\ & \ddots & & \vdots \\ & & -1 & 1 \\ \beta_1 & \cdots & \beta_{N_g} & \rho(t) - \beta \end{bmatrix}, \quad r(t) = \begin{bmatrix} \lambda_1 C_1(t) \\ \vdots \\ \lambda_{N_g} C_{N_g}(t) \\ C_n(t)/\Lambda \end{bmatrix}$$

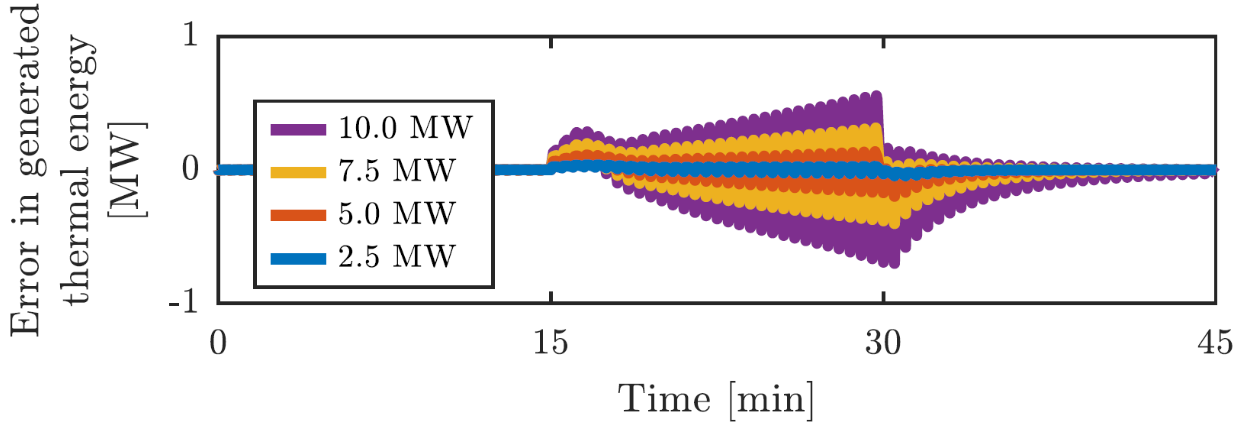
Appendix: Rigorous time-varying time delays

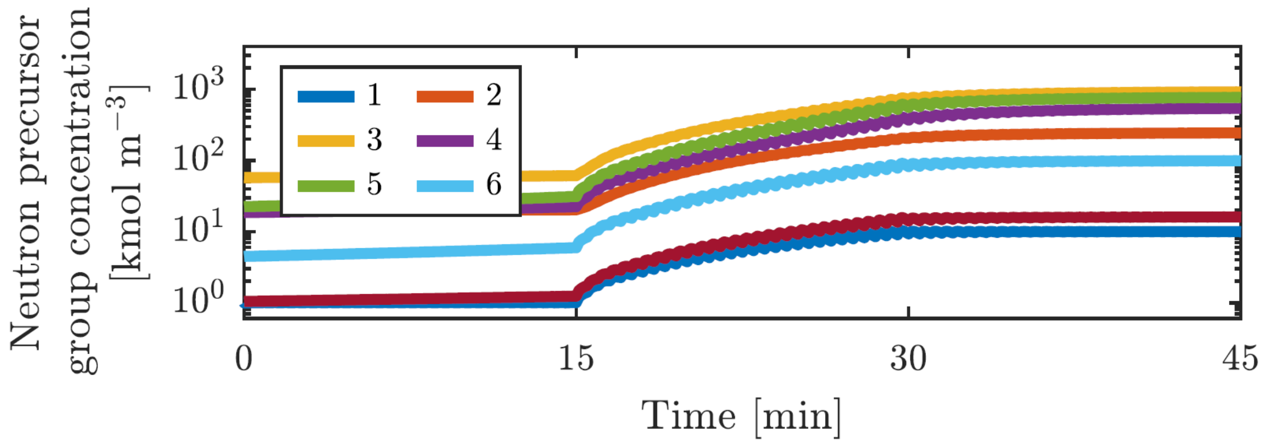




Appendix: Numerical example

Distributed time delays





Absolute time delays