# MATLAB Simulation Codes for [2, Sect. 6]

These MATLAB codes reproduce the simulation results from [2, Sect. 6]. In addition to the script `n2system.m` which runs the simulations and reproduces the results from [2, Sect. 6], the codes contain the script `n2kernelsolver.m` to solve the $n+m$ backstepping kernel equations for the considered parameters when $n = 10$ and $m = 2$.

## Description

The considered class of $n+m$ hyperbolic partial differential equations (PDEs) is of the form

$$\mathbf{u}_t(t,x) + \mathbf{\Lambda}(x)\mathbf{u}_x(t,x) = \frac{1}{n}\mathbf{\Sigma}(x)\mathbf{u}(t,x) + \mathbf{W}(x)\mathbf{v}(t,x), \tag{1a}$$

$$\mathbf{v}_t(t,x) - \mathbf{M}(x)\mathbf{v}_x(t,x) = \frac{1}{n}\mathbf{\Theta}(x)\mathbf{u}(t,x) + \mathbf{\Psi}(x)\mathbf{v}(t,x), \tag{1b}$$

with boundary conditions

$$\mathbf{u}(t,0) = \mathbf{Q}\mathbf{v}(t,0), \qquad \mathbf{v}(t,1) = \mathbf{U}(t), \tag{2}$$

where $\mathbf{u} = (u^i)_{i=1}^n, \mathbf{v} = (v^j)_{j=1}^m$ are the PDE states and $\mathbf{U} = (U^j)_{j=1}^m$ is the control input. In the framework of [2, Sect. 6], the parameters of the system (1), (2) are given by

$$\lambda_i(x) = 1, \qquad \mu_1(x) = 2, \qquad \mu_2(x) = 1, \tag{3a}$$

$$\sigma_{i,l}(x) = x^3(x+1)\left(\frac{i}{n} - \frac{1}{2}\right)\left(\frac{l}{n} - \frac{1}{2}\right), \tag{3b}$$

$$\theta_{1,i}(x) = -3\frac{i}{n}\left(\frac{i}{n} - 1\right), \qquad \theta_{2,i}(x) = -2\frac{i}{n}\left(\frac{i}{n} - 1\right), \tag{3c}$$

$$w_{i,1}(x) = w_{i,2}(x) = x(x+1)e^x\left(\frac{i}{n} - \frac{1}{2}\right), \tag{3d}$$

$$\mathbf{\Psi} = 0, \tag{3e}$$

$$q_{i,1} = 8\left(\frac{i}{n} - \frac{1}{2}\right), \qquad q_{i,2} = -8\left(\frac{i}{n} - 2\right), \tag{3f}$$

for $i, j = 1, \ldots, n$, where $n \in \mathbb{N}$ can be chosen freely and $m = 2$.

A respective $\infty + m$ continuum system of the form

$$u_t(t,x,y) + \lambda(x,y)u_x(t,x,y) = \int_0^1 \sigma(x,y,\eta)u(t,x,\eta)d\eta + \mathbf{W}(x,y)\mathbf{v}(t,x), \tag{4a}$$

$$\mathbf{v}_t(t,x) - \mathbf{M}(x)\mathbf{v}_x(t,x) = \int_0^1 \mathbf{\Theta}(x,y)u(t,x,y)dy + \mathbf{\Psi}(x)\mathbf{v}(t,x), \tag{4b}$$

with boundary conditions

$$u(t,0,y) = \mathbf{Q}(y)\mathbf{v}(t,0), \tag{5a}$$

$$\mathbf{v}(t,1) = \mathbf{U}(t), \tag{5b}$$

is constructed to approximate the original $n+m$ system (1), (2), where the continuum parameters corresponding to (3) can be chosen as

$$\lambda(x,y)=1,\quad \mu_1(x)=2,\quad \mu_2(x)=1, \tag{6a}$$

$$\sigma(x,y,\eta)=x^3(x+1)\left(y-\frac{1}{2}\right)\left(\eta-\frac{1}{2}\right), \tag{6b}$$

$$W_1(x,y)=W_2(x,y)=x(x+1)e^x\left(y-\frac{1}{2}\right), \tag{6c}$$

$$\theta_1(x,y)=-3y(y-1),\quad \theta_2(x,y)=-2y(y-1), \tag{6d}$$

$$\boldsymbol{\Psi}=0, \tag{6e}$$

$$Q_1(y)=8\left(y-\frac{1}{2}\right),\quad Q_2(y)=-8(y-2), \tag{6f}$$

i.e., by replacing $i/n$ and $l/n$ with $y$ and $\eta$, respectively, in (3). The solution to the respective continuum kernel equations [2, (27), (31)–(33)], where we choose $l_{2,1}^{(1)}=\psi_{2,1}=0$, is explicitly given by

$$K_1^1(x,\xi,y)=y(y-1), \tag{7a}$$

$$K_1^2(x,\xi,y)=e^{x-2\xi}y(y-1), \tag{7b}$$

$$K_2^2(x,\xi,y)=e^{2(x-\xi)}y(y-1), \tag{7c}$$

$$L_{1,1}^1(x,\xi)=L_{1,1}^2(x,\xi)=0, \tag{7d}$$

$$L_{1,2}^1(x,\xi)=0,\quad L_{1,2}^2(x,\xi)=-2e^{x-2\xi}, \tag{7e}$$

$$L_{2,1}^2(x,\xi)=0,\quad L_{2,2}^2(x,\xi)=-2e^{2(x-\xi)}, \tag{7f}$$

where $K_1^\star(\cdot,y), L_{1,1}^\star$, and $L_{1,2}^\star$ are defined on $\mathcal{T}_1^1=\{(x,\xi)\in[0,1]^2:\frac{1}{2}x\leq\xi\leq x\}$ and $\mathcal{T}_1^2=\{(x,\xi)\in[0,1]^2:\xi\leq\frac{1}{2}x\}$ for the respective superindex $\star=1,2$, while $K_2^2(\cdot,y), L_{2,1}^2$ and $L_{2,2}^2$ are defined on $\mathcal{T}_2^2=\mathcal{T}=\{(x,\xi)\in[0,1]^2:\xi\leq x\}$, for each $y\in[0,1]$. For solving the $n+m$ kernel equations corresponding to the system (1), (2) with parameters (3), the script `n2kernelsolver.m` is provided.

Running the MATLAB script

```
n2system.m
```

reproduces the simulation results from [2, Sect. 6]. The script simulates a finite-difference approximation (with 256 grid points) of the $n+2$ system (1), (2) with parameters (3) for $n=2,6,10$ in closed-loop with the continuum kernels-based control law

$$U^i(t)=\sum_{l=1}^{n}\sum_{p=i}^{m}\int_{\rho_i^{p+1}(1)}^{\rho_i^p(1)}\frac{1}{n}\tilde{k}_{i,l}^p(1,\xi)u^l(t,\xi)d\xi+\sum_{j=1}^{m}\sum_{p=i}^{m}\int_{\rho_i^{p+1}(1)}^{\rho_i^p(1)}\tilde{\ell}_{i,j}^p(1,\xi)v^j(t,\xi)d\xi \tag{8}$$

where $\rho_i^{m+1}(1)=0$ and $\rho_i^i(1)=1$ for $i\in\{1,2\}$, $\rho_1^2(1)=\frac{1}{2}$, and the kernels are obtained based on the continuum kernels (7) as

$$\tilde{k}_{i,l}^p(1,\xi)=K_i^p(1,\xi,l/n),\qquad \tilde{\ell}_{i,j}^p(1,\xi)=L_{i,j}^p(1,\xi), \tag{9}$$

for $i, j \in \{1, 2\}$ and $l \in \{1, \dots, n\}$. The integrals in (8) are approximated using the trapezoidal rule `trapz`. In addition to using the continuum kernels-based control law (8), the script additionally simulates the system for $n = 10$ in closed-loop with the $n + m$ kernels-based control law, where the solution to the $10 + 2$ kernel equation is computed (approximated) by running the script `n2kernelsolver.m`. The control law is of the same form as (8), but with $\tilde{k}, \tilde{\ell}$ replaced with the computed solution to the $10 + 2$ kernel equations (10), (11) for parameters (3).

Running the MATLAB script

```
n2kernelsolver.m
```

utilizes finite differences and successive approximations (see, e.g., [1, App. F]) in computing the solution to the $n + 2$ kernel equations [1, (19.3)–(19.5)] (for parameters (3) with $n = 10$)

$$\mathbf{M}(x)\mathbf{K}_x(x,\xi) - \mathbf{K}_\xi(x,\xi)\mathbf{\Lambda}(\xi) - \mathbf{K}(x,\xi)\mathbf{\Lambda}'(\xi) = \mathbf{K}(x,\xi)\mathbf{\Sigma}(\xi) + \mathbf{L}(x,\xi)\mathbf{\Theta}(\xi), \tag{10a}$$

$$\mathbf{M}(x)\mathbf{L}_x(x,\xi) + \mathbf{L}_\xi(x,\xi)\mathbf{M}(\xi) + \mathbf{L}(x,\xi)\mathbf{M}'(\xi) = \mathbf{K}(x,\xi)\mathbf{W}(\xi) + \mathbf{L}(x,\xi)\mathbf{\Psi}(\xi), \tag{10b}$$

with boundary conditions

$$0 = \mathbf{K}(x,x)\mathbf{\Lambda}(x) + \mathbf{M}(x)\mathbf{K}(x,x) + \mathbf{\Theta}(x), \tag{11a}$$

$$0 = \mathbf{M}(x)\mathbf{L}(x,x) - \mathbf{L}(x,x)\mathbf{M}(x) + \mathbf{\Psi}(x), \tag{11b}$$

$$L_{i,j}(x,0) = \frac{1}{\mu_j(0)} \sum_{\ell=1}^{n} \lambda_\ell(0) K_{i,\ell}(x,0) Q_{\ell,j}, \quad \forall i \le j, \tag{11c}$$

$$L_{i,j}(1,\xi) = l_{i,j}(\xi), \quad \forall j < i, \tag{11d}$$

where we choose $l_{i,j} = 0$ for $j < i$ in accordance with $\mathbf{\Psi} = 0$. The solution is evaluated at $x = 1$ in order to provide the control gains that are used in the script `n2system.m`. The successive approximation procedure for solving (10), (11) is set to terminate when two successive approximations differ less than $10^{-7}$ in norm, at which point a convergence message is displayed. Moreover, the function is set to terminate after $400$ iterations regardless of convergence, which is to avoid potential infinite loops, albeit in the $10 + 2$ case with parameters (3), the successive approximations procedure reaches the set solution tolerance of $10^{-7}$ in 199 iterations.

## Examples

The simulation results for [2, Sect. 6] can be reproduced by running the script `n2system.m`.

# References


[1] H. Anfinsen and O. M. Aamo. *Adaptive Control of Hyperbolic PDEs*. Springer, 2019.

[2] J.-P. Humaloja and N. Bekiaris-Liberis. Backstepping control of continua of linear hyperbolic PDEs and application to stabilization of large-scale $n + m$ coupled hyperbolic PDE systems. *Automatica*, 2025 (to appear).