Control of Large-Scale Delayed Networks: DDEs, DDFs and PIEs

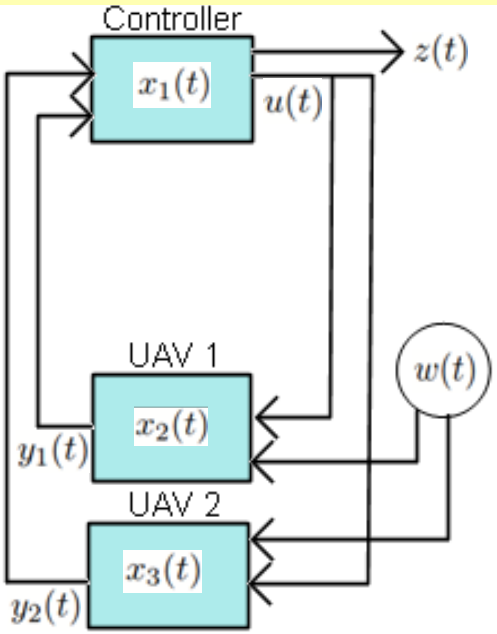
Matthew M. Peet and S. Shivakumar
Arizona State University
Tempe, AZ USA

Mathematical Theory of Networks and Systems
Bayreuth, Germany

Sept. 12, 2022

Networks

Many delays, but small channels



Types of Delay: State delay $(\hat{\tau}, \tilde{\tau})$, input delay (h_i), process delay $(\bar{\tau})$.

$$\dot{x}_i(t) = a_i x_i(t) + \sum_{j=1}^{N} a_{ij} x_j(t - \hat{\tau}_{ij}) + b_{1i} w(t - \bar{\tau}_i) + b_{2i} u(t - h_i)$$
$$z(t) = C_1 x(t) + D_{12} u(t)$$
$$y_i(t) = c_{2i} x_i(t - \tilde{\tau}_i) + d_{21i} w(t - \tilde{\tau}_i).$$

Question: How to leverage network structure to simplify controller design?

DELAY DIFFERENTIAL EQUATIONs (DDEs)

The DDE Model of Delay (Zero Initial Conditions)

The Class of Delay Differential Equations (DDEs):

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_0 & B_{10} & B_{20} \\ C_{10} & D_{11} & D_{12} \\ C_{20} & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^{K} \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix} \begin{bmatrix} x(t-\tau_i) \\ w(t-\tau_i) \\ u(t-\tau_i) \end{bmatrix}$$

$$+ \sum_{i=1}^{K} \int_{-\tau_i}^{0} \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix} \begin{bmatrix} x(t+s) \\ w(t+s) \\ u(t+s) \end{bmatrix} ds$$

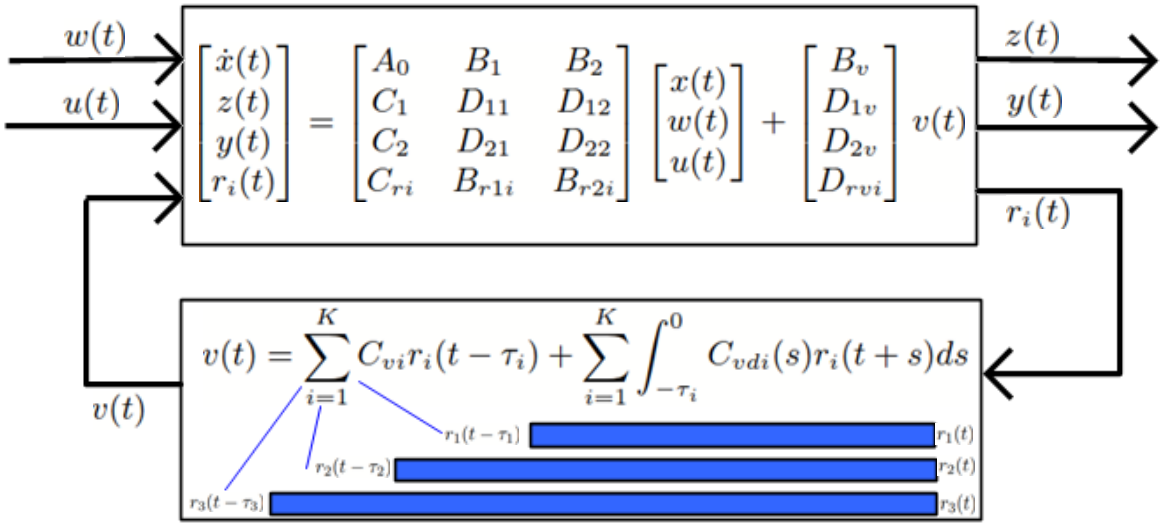
Signals:	Sources of Delay:
• The present state $x(t) \in \mathbb{R}^n$	• State delay: $x(t-\tau)$
• The disturbance or exogenous input, $w(t) \in \mathbb{R}^m$	• Disturbance Delay: $w(t-\tau)$
• The controlled input, $u(t) \in \mathbb{R}^p$	• Input Delay: $u(t-\tau)$
• The regulated output, $z(t) \in \mathbb{R}^q$	• Output Delay: $y(t-\tau)$
• The observed or sensed output, $y(t) \in \mathbb{R}^r$	

Assertion: Analysis and Control is tractable when the number of infinite-dimensional components is less than 50 (Here: $(n+m+p)K < 50$).

Problem: You can't specify which information gets delayed.

DIFFERENTIAL DIFFERENCE EQUATIONs (DDFs)

The DDF Model of Delay (0 Initial Conditions)



The Class of Differential Difference Equations (DDFs):

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \\ r_i(t) \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \\ C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \\ D_{rvi} \end{bmatrix} v(t)$$

$$v(t) = \sum_{i=1}^{K} C_{vi} r_i(t-\tau_i) + \sum_{i=1}^{K} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s) ds.$$

- The delayed channels (infinite-dimensional) isolated in the r_i.
- All other signals are finite-dimensional.

Model Reduction from DDE to DDF

Network structure allows you to reduce size of delay channels

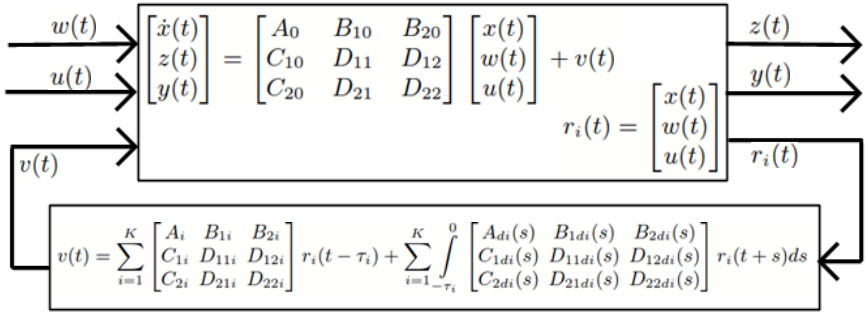


Figure: Naïve Conversion DDE $\rightarrow$ DDF

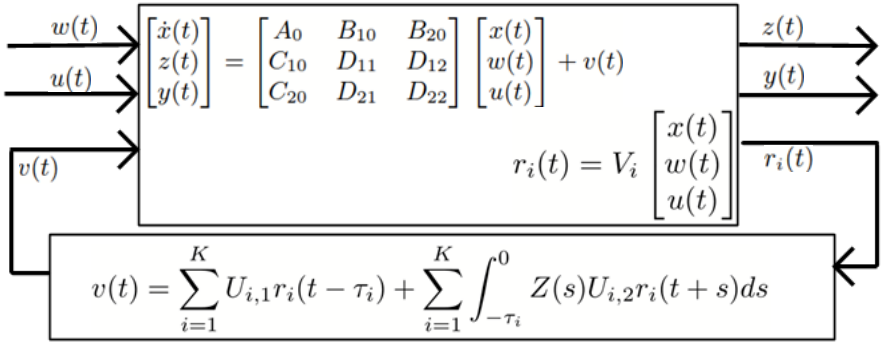


Figure: Minimal Conversion DDE $\rightarrow$ DDF

Define

$$\hat{r}_i(t) = \begin{bmatrix} I & 0 \\ 0 & Z(s) \end{bmatrix} \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \\ \hat{A}_{di} & \hat{B}_{1di} & \hat{B}_{2di} \\ \hat{C}_{1di} & \hat{D}_{11di} & \hat{D}_{12di} \\ \hat{C}_{2di} & \hat{D}_{21di} & \hat{D}_{22di} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & Z(s) \end{bmatrix} \underbrace{U_i}_{\begin{bmatrix} U_{i,1} \\ U_{i,2} \end{bmatrix}} \underbrace{V_i \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}}_{r_i(t)}$$

- Use the SVD to minimize inner dimension of $U_i V_i$.
- For networks, inner dimension will be small (typically 1)

PARTIAL INTEGRAL EQUATIONs (PIEs)

ODE-PDE Representation of the DDF Model

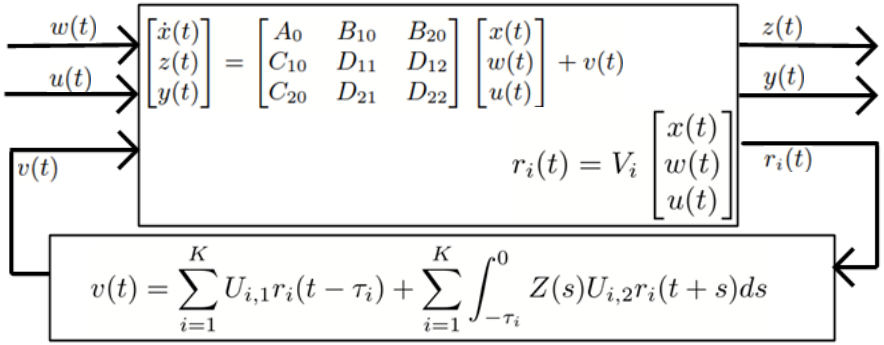


Figure: DDE to be converted

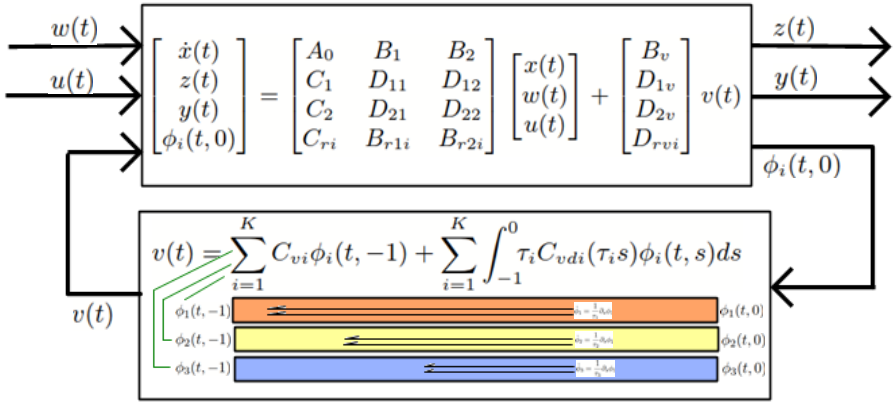


Figure: Equivalent ODE-PDE format

The Class of ODE-PDE Systems (ODE-PDEs):

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\\ \phi_i(t,0)\end{bmatrix} = \begin{bmatrix}A_0 & B_1 & B_2\\ C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\\ C_{ri} & B_{r1i} & B_{r2i}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix} + \begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\\ D_{rvi}\end{bmatrix} v(t) \qquad \phi_i(t,0) = r_i(t)$$

$$\dot{\phi}_i(t,s) = \frac{1}{\tau_i}\phi_{i,s}(t,s), \qquad v(t) = \sum_{i=1}^{K} C_{vi}\phi_i(t,-1) + \sum_{i=1}^{K} \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)\phi_i(t,s) ds$$

- Each ϕ_i represents a pipe of length 1 with flow rate $\frac{1}{\tau_i}$, so $\phi_i(t,-1) = r_i(t-\tau_i)$.
- The conversion from DDF to ODE-PDE is otherwise trivial.

ODE-PDE Representation of the DDF Model

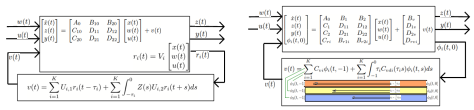


Figure: DDE to be converted

Figure: Equivalent ODE-PDE format

The Class of ODE-PDE Systems (ODE-PDEs):

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \\ \phi_i(t,0) \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \\ C_{ri} & B_{ri1} & B_{ri2} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \\ D_{rvi} \end{bmatrix} v(t) \qquad \phi_i(t,0) = r_i(t)$$

$$\dot{\phi}_i(t,s) = \frac{1}{\tau_i}\phi_{i,s}(t,s), \qquad v(t) = \sum_{i=1}^{K} C_{vi}\phi_i(t,-1) + \sum_{i=1}^{K}\int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)\phi_i(t,s)ds$$

- Each ϕ_i represents a pipe of length 1 with flow rate $\frac{1}{\tau_i}$, so $\phi_i(t,-1) = r_i(t-\tau_i)$.
- The conversion from DDF to ODE-PDE is otherwise trivial.

Compact Version of ODE-PDE:

- Stack the PDE states to remove summations:

Indexed Form:

$$\phi_i(t,0) = r_i(t),\ \dot{\phi}_i(t,s) = \frac{1}{\tau_i}\phi_{i,s}(t,s),\ v(t) = \sum_{i=1}^{K} C_{vi}\phi_i(t,-1) + \sum_{i=1}^{K}\int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)\phi_i($$

Compact Form:

$$\phi(t,0) = r(t) \qquad \dot{\phi}(t,s) = I_\tau \phi_s(t,s) \qquad v(t) = \hat{C}_v\phi(t,-1) + \int_{-1}^{0} \hat{C}_{vd}(s)\phi_i(t,s)ds$$

where

$$\hat{C}_v d(s) = \begin{bmatrix} \tau_1 C_{vd1}(\tau_1 s) & \cdots & \tau_1 C_{vdK}(\tau_K s) \end{bmatrix} \qquad I_\tau = \begin{bmatrix} 1/\tau_1 & & \\ & \ddots & \\ & & 1/\tau_K \end{bmatrix}$$

$$\hat{C}_v = \begin{bmatrix} C_{v1} & \cdots & C_{vK} \end{bmatrix}$$

ODE-PDE System to PIE System

Ignoring Inputs, Outputs and Distributed Delay for now

ODE Subsystem:

$$\dot{x}(t) = A_0 x(t) + v(t)$$

PDE Subsystem:

$$\phi(t,0) = r(t) = Vx(t) \qquad \dot{\phi}(t,s) = I_\tau \phi_s(t,s) \qquad v(t) = U\phi(t,-1)$$

Variable Substitution: $\phi \leftrightarrow \phi_s$

$$\phi(t,s) = \phi(t,0) - \int_s^0 \phi_s(t,\theta)d\theta = Vx(t) - \int_s^0 \phi_s(t,\theta)d\theta$$

Equivalent Partial Integral Subsystem: No boundary condition needed

$$V\dot{x}(t) - \int_s^0 \dot{\phi}_s(t,\theta)d\theta = I_\tau \phi_s(t,s) \qquad v(t) = UVx(t) - \int_{-1}^0 U\phi_s(t,s)ds$$

Equivalent Partial Integral System:

$$\underbrace{\begin{bmatrix} I & 0 \\ V & -\int_s^0 \end{bmatrix}}_{\mathcal{T}} \underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}_s(t,\cdot) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} A_0 + UV & -U\int_{-1}^0 \\ 0 & I_\tau \end{bmatrix}}_{\mathcal{A}} \underbrace{\begin{bmatrix} x(t) \\ \phi_s(t,\cdot) \end{bmatrix}}_{\mathbf{x}(t)}$$

The *-Algebra of Partial Integral (PI) Operators

$$\underbrace{\begin{bmatrix} I & 0 \\ V & -\int_s^0 \end{bmatrix}}_{\mathcal{T}\in\Pi_4} \underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}_s(t,\cdot) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} A_0 + UV & -U\int_{-1}^0 \\ 0 & I_\tau \end{bmatrix}}_{\mathcal{A}\in\Pi_4} \underbrace{\begin{bmatrix} x(t) \\ \phi_s(t,\cdot) \end{bmatrix}}_{\mathbf{x}(t)\in\mathbb{R}\times L_2}$$

Definition of a 4-PI Operator (Π_4) $\left(\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{smallmatrix}\right]\right)$**:** $\mathbb{R}\times L_2 \to \mathbb{R}\times L_2$

$$\left(\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{smallmatrix}\right]\begin{bmatrix} x \\ \mathbf{\Phi} \end{bmatrix}\right)(s) := \begin{bmatrix} Px + \int_{-1}^0 Q_1(\theta)\mathbf{\Phi}(\theta)ds \\ Q_2(s)x + \left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s) \end{bmatrix}.$$

4-PI Operators include a 3-PI Operator (Π_3), Defined as:

$$\left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s) := R_0(s)\mathbf{\Phi}(s) + \int_{-1}^s R_1(s,\theta)\mathbf{\Phi}(\theta)d\theta + \int_s^0 R_2(s,\theta)\mathbf{\Phi}(\theta)d\theta$$

Seems Unfamiliar? Recall the complete-Quadratic Lyapunov Functional:

$$V(\mathbf{x}_t) = \left\langle \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix}, \underbrace{\begin{bmatrix} U & \int_{-1}^0 U(-\theta-1)A \\ A^TU(-s-1)^T & \int_{-1}^0 A^TU(s-\cdot)A \end{bmatrix}}_{\mathcal{P}\in\Pi_4} \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}$$

The *-Algebra of Partial Integral (PI) Operators

$$\underbrace{\begin{bmatrix} I & 0 \\ V & -\int_s^0 \end{bmatrix}}_{\mathcal{T}\in\Pi_4} \underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}_s(t,\cdot) \end{bmatrix}}_{\mathbf{x}(t)} = \underbrace{\begin{bmatrix} A_0 + UV & -U\int_{-1}^0 \\ 0 & I_\tau \end{bmatrix}}_{\mathcal{A}\in\Pi_4} \underbrace{\begin{bmatrix} x(t) \\ \phi_s(t,\cdot) \end{bmatrix}}_{\mathbf{x}(t)\in\mathbb{R}\times L_2}$$

Definition of a 4-PI Operator (Π_4) $\left(\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_1\} \end{smallmatrix}\right]\right)$: $\mathbb{R}\times L_2 \to \mathbb{R}\times L_2$

$$\left(\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_1\} \end{smallmatrix}\right]\begin{bmatrix} x \\ \mathbf{\Phi} \end{bmatrix}\right)(s) := \begin{bmatrix} Px + \int_{-1}^0 Q_1(\theta)\mathbf{\Phi}(\theta)d\theta \\ Q_2(s)x + \left(\mathcal{P}_{\{R_1\}}\mathbf{\Phi}\right)(s) \end{bmatrix}.$$

4-PI Operators include a 3-PI Operator (Π_3), Defined as:

$$\left(\mathcal{P}_{\{R_1\}}\mathbf{\Phi}\right)(s) := R_0(s)\mathbf{\Phi}(s) + \int_{-1}^s R_1(s,\theta)\mathbf{\Phi}(\theta)d\theta + \int_s^0 R_2(s,\theta)\mathbf{\Phi}(\theta)d\theta$$

Seems Unfamiliar? Recall the complete-Quadratic Lyapunov Functional:

$$V(\mathbf{x}_t) = \left\langle \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix}, \underbrace{\begin{bmatrix} U & \int_{-1}^0 U(-\theta-1)A \\ A^T U(-s-1)^T & \int_{-1}^0 A^T U(s-\cdot)A \end{bmatrix}}_{\mathcal{P}\in\Pi_4} \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}$$

You may also recall the derivative also has this form. If

$$V(\mathbf{x}_t) = \left\langle \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix}, \underbrace{\begin{bmatrix} M_{11} & \int_{-1}^0 M_{12}(\cdot) \\ M_{12}^T(s) & M_{22}(s) + \int_{-1}^0 N(s,\cdot) \end{bmatrix}}_{\mathcal{P}\in\Pi_4} \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t \end{bmatrix} \right\rangle_{\mathbb{R}\times L_2}$$

then

$$\dot{V}(\mathbf{x}_t) = \left\langle \begin{bmatrix} \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t(-1) \end{bmatrix} \\ \mathbf{x}_t \end{bmatrix}, \underbrace{\begin{bmatrix} D_{11} & \int_{-1}^0 D_{12}(s) \\ D_{12}(s)^T & -\dot{M}_{22}(s) - \int_{-1}^0 (\partial_s + \partial_\theta)N(s,\cdot) \end{bmatrix}}_{\mathcal{D}\in\Pi_4} \begin{bmatrix} \begin{bmatrix} \mathbf{x}_t(0) \\ \mathbf{x}_t(-1) \end{bmatrix} \\ \mathbf{x}_t \end{bmatrix} \right\rangle$$

where

$$D_{11} := \begin{bmatrix} G_{11} + G_{11}^T & G_{12} \\ G_{12}^T & -M_{22}(-1) \end{bmatrix}, \quad D_{12}(s) := \begin{bmatrix} A_0^T M_{12}(s) - \dot{M}_{12}(s) + N(0,s) \\ A_{1d}^T M_{12}(s) - N(-1,s) \end{bmatrix},$$

$$G_{11} = M_{11}A_0 + M_{12}(0) + \frac{1}{2}M_{22}(0) \qquad G_{12} = M_{11}A_{d1} - M_{12}(-1)$$

Complete PIE Representation of the *complete* DDF Model

The PIE version of the DDF system model (w/ input delay) is:

$$\mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{T}_w \dot{w}(t) + \mathcal{T}_u \dot{u}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)$$
$$z(t) = \mathcal{C}_1 \mathbf{x}(t) + \mathcal{D}_{11} w(t) + \mathcal{D}_{12} u(t), \quad \mathbf{x}(t) = \begin{bmatrix} x(t) \\ \partial_s \phi_1(t,s) \\ \vdots \\ \partial_s \phi_K(t,s) \end{bmatrix}$$
$$y(t) = \mathcal{C}_2 \mathbf{x}(t) + \mathcal{D}_{21} w(t) + \mathcal{D}_{22} u(t)$$

---

where $\{\mathcal{T}, \mathcal{A}, \cdots, \mathcal{D}_{22}\} \subset \Pi_4$ are given by:

$$\mathcal{A} = \mathcal{P}\begin{bmatrix} \mathbf{A}_0, & \mathbf{A} \\ 0, & \{I_\tau, 0, 0\} \end{bmatrix}, \quad \mathcal{T} = \mathcal{P}\begin{bmatrix} I, & 0 \\ \mathbf{T}_0, & \{0, \mathbf{T}_a, \mathbf{T}_b\} \end{bmatrix}, \quad \mathcal{T}_w = \mathcal{P}\begin{bmatrix} 0, & \emptyset \\ \mathbf{T}_1, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{T}_u = \mathcal{P}\begin{bmatrix} 0, & \emptyset \\ \mathbf{T}_2, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{D}_{ij} = \mathcal{P}\begin{bmatrix} \mathbf{D}_{ij}, & \emptyset \\ \emptyset, & \{\emptyset\} \end{bmatrix}$$

$$\mathcal{B}_1 = \mathcal{P}\begin{bmatrix} \mathbf{B}_1, & \emptyset \\ 0, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{B}_2 = \mathcal{P}\begin{bmatrix} \mathbf{B}_2, & \emptyset \\ 0, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{C}_1 = \mathcal{P}\begin{bmatrix} \mathbf{C}_{10}, & \mathbf{C}_{11} \\ \emptyset, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{C}_2 = \mathcal{P}\begin{bmatrix} \mathbf{C}_{20}, & \mathbf{C}_{21} \\ \emptyset, & \{\emptyset\} \end{bmatrix},$$

where

$$\hat{C}_{vi} = C_{vi} + \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s) ds, \quad D_I = \left(I_{n_v} - \left(\sum_{i=1}^{K} \hat{C}_{vi} D_{rvi} \right) \right)^{-1}, \quad C_{Ii}(s) = -D_I \left(C_{vi} + \tau_i \int_{-1}^{s} C_{vdi}(\tau_i \eta) d\eta \right)$$

$$[\mathbf{T}_0 \;\; \mathbf{T}_1 \;\; \mathbf{T}_2] = \begin{bmatrix} C_{r1} & B_{r11} & B_{r21} \\ \vdots & \vdots & \vdots \\ C_{rK} & B_{r1K} & B_{r2K} \end{bmatrix} + \begin{bmatrix} D_{rv1} \\ \vdots \\ D_{rvK} \end{bmatrix} [C_{vx} \;\; D_{vw} \;\; D_{vu}], \quad [C_{vx} \; D_{vw} \; D_{vu}] = D_I \sum_{i=1}^{K} \hat{C}_{vi} [C_{ri} \; B_{r1i} \; B_{r2i}]$$

$$\mathbf{T}_a(s,\theta) = \begin{bmatrix} D_{rv1} \\ \vdots \\ D_{rvK} \end{bmatrix} [C_{I1}(\theta) \;\; \cdots \;\; C_{IK}(\theta)], \quad \mathbf{T}_b(s,\theta) = -I_{\sum_i p_i} + \mathbf{T}_a(s,\theta), \quad I_\tau = \begin{bmatrix} \frac{1}{\tau_1} I_{p_1} & & \\ & \ddots & \\ & & \frac{1}{\tau_K} I_{p_K} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{A}(s) \\ \mathbf{C}_{11}(s) \\ \mathbf{C}_{21}(s) \end{bmatrix} = \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} [C_{I1}(s) \cdots C_{IK}(s)], \quad \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_{10} & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_{20} & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_{10} & D_{11} & D_{12} \\ C_{20} & D_{21} & D_{22} \end{bmatrix} + \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} [C_{vx} \;\; D_{vw} \;\; D_{vu}].$$

PIETOOLS: The way to a PIE is through a DDF

Neutral Delay System (NDS) to PIE

```
NDS.A0=[-1]; NDS.Ai{2}=[-2];
NDS.Ei{1}=[.2]; NDS.tau=[.5 1];
NDS=initialize_PIETOOLS_NDS(NDS);
DDF=convert_PIETOOLS_NDS2DDF(NDS);
DDF=minimize_PIETOOLS_DDF(DDF);
PIE=convert_PIETOOLS_DDF2PIE(DDF);
```

Original NDS:

$$\dot{x}(t) = -x(t) + .2\dot{x}(t-.5) - 2x(t-1)$$

Equivalent DDF Data Structure:

```
DDF.A0=-1; DDF.tau=[.5 1];
DDF.Cr{1}=-.2; DDF.Cr{2}=-2;
DDF.Cv{1}=1; DDF.Cv{2}=1;
 DDF.Drv{1}=[.2]; DDF.Bv=1;
```

New DDF Representation:

$$\dot{x}(t) = -x(t) + v(t)$$

$$v(t) = r_1(t-.5) + r_2(t-\tau)$$

$$r_1(t) = -.2x(t) + .2v(t);\ r_2(t) = -2x(t)$$

New PIE Representation: $\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t)$

```
>> PIE.A
ans=

      [-3.7500] | [-1.25,-1.25]
      --------------------------
            [0] | ans.R
            [0] |


ans.R=

    [2,0] | [0,0] | [0,0]
    [0,1] | [0,0] | [0,0]
```

```
>> PIE.T
ans=

             [1] | [0,0]
      ------------------
      [-0.7500] | ans.R
           [-2] |


ans.R=

    [0,0] | [-0.2500,-0.2500] | [-1.2500,-0.2500]
    [0,0] |              [0,0] |             [0,-1]
```

Duality and Optimal Control of PIEs

A Strong Duality Theorem for PIEs

(A) Primal PIE:

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}w(t)$$
$$z(t) = \mathcal{C}\mathbf{x}(t) + \mathcal{D}w(t)$$

(B) Dual PIE/ Adjoint PIE:

$$\mathcal{T}^*\dot{\bar{\mathbf{x}}}(t) = \mathcal{A}^*\bar{\mathbf{x}}(t) + \mathcal{C}^*\bar{w}(t)$$
$$\bar{z}(t) = \mathcal{B}^*\bar{\mathbf{x}}(t) + \mathcal{D}^*\bar{w}(t)$$

For a PIE and its Dual:

1. stability is equivalent; (A) is stable iff (B) is stable
2. L_2-gain is equivalent; $\gamma = \sup_{w \neq 0} \frac{\|z\|}{\|w\|} = \sup_{\bar{w} \neq 0} \frac{\|\bar{z}\|}{\|\bar{w}\|}$

A Dual KYP lemma for PIEs

A **Computational** Test for L_2-gain.

Theorem 1 (Dual KYP lemma and L_2-Gain).

Suppose that $\mathcal{T}, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D} \in \Pi_4$ *and*

$$\begin{aligned} \mathcal{T}\dot{\mathbf{x}}(t) &= \mathcal{A}\mathbf{x} + \mathcal{B}w(t) \qquad \mathbf{x}(0) = 0 \\ y(t) &= \mathcal{C}\mathbf{x}(t) + \mathcal{D}w(t). \end{aligned}$$

If there exists operator $\mathcal{P} \in \Pi_4$*, such that* $\mathcal{P} \succeq 0$

$$\begin{bmatrix} -\gamma I & \mathcal{D} & \mathcal{C}\mathcal{P}\mathcal{T} \\ \mathcal{D}^* & -\gamma I & \mathcal{B}^* \\ \mathcal{T}\mathcal{P}\mathcal{C}^* & \mathcal{B} & \mathcal{A}\mathcal{P}\mathcal{T}^* + \mathcal{T}\mathcal{P}\mathcal{A}^* \end{bmatrix} \preccurlyeq 0.$$

Then $\|y\|_{L_2} \leq \gamma \|\omega\|_{L_2}$.

Solving Linear PI Inequality (LPI) Optimization Problems

How to Enforce positivity of a PI operator?

An LMI for Positivity of PI operators: If Π is a C^* algebra, any positive operator $\mathcal{P} \in \Pi$ can be represented as $\mathcal{P} = \mathcal{A}^*\mathcal{A}$ where $\mathcal{A} \in \Pi$.

Define a vector of bases, $Z(s)$. Then any $\mathcal{A} \in \Pi_4$ (in associated module) may be represented as

$$(\mathcal{A}\mathbf{x})(s) = (Q\mathcal{Z}\mathbf{x})(s) = Q \begin{bmatrix} x(t) \\ Z(s)\mathbf{x}(s) \\ \int_a^s Z(s,\theta)\mathbf{x}(\theta)d\theta \\ \int_s^b Z(s,\theta)\mathbf{x}(\theta)d\theta \end{bmatrix}.$$

for some matrix Q where $Z(s,\theta) = Z(s) \otimes Z(\theta)$. We conclude that if $\mathcal{P} = \mathcal{A}^*\mathcal{A}$, then $\mathcal{P}$ has the form

$$\mathcal{P}\left[\begin{smallmatrix} P, & Q_1 \\ Q_2, & \{R_i\} \end{smallmatrix}\right] = \mathcal{A}^*\mathcal{A} = \mathcal{Z}^*Q^TQ\mathcal{Z} = \mathcal{Z}^*P\mathcal{Z}$$

for some matrix $P \geq 0$.

$$\begin{bmatrix} -\gamma I & \mathcal{D} & \mathcal{C}\mathcal{P}\mathcal{T} \\ \mathcal{D}^* & -\gamma I & \mathcal{B}^* \\ \mathcal{T}\mathcal{P}\mathcal{C}^* & \mathcal{B} & \mathcal{A}\mathcal{P}\mathcal{T}^* + \mathcal{T}\mathcal{P}\mathcal{A}^* \end{bmatrix} \preccurlyeq 0 \quad \text{becomes} \quad \begin{bmatrix} -\gamma I & \mathcal{D} & \mathcal{C}\mathcal{P}\mathcal{T} \\ \mathcal{D}^* & -\gamma I & \mathcal{B}^* \\ \mathcal{T}\mathcal{P}\mathcal{C}^* & \mathcal{B} & \mathcal{A}\mathcal{P}\mathcal{T}^* + \mathcal{T}\mathcal{P}\mathcal{A}^* \end{bmatrix} = -\mathcal{Z}^*P\mathcal{Z},$$

H_∞-optimal static state-feedback control (no input delay*)

$$\begin{aligned}\mathcal{T}\dot{\mathbf{x}}(t)+\mathcal{T}_u\dot{u}(t) &= \mathcal{A}\mathbf{x}(t)+\mathcal{B}_1 w(t)+\mathcal{B}_2 u(t), \quad \mathbf{v}(0)=0\\ z(t) &= \mathcal{C}_1\mathbf{x}(t)+\mathcal{D}_{11}w(t)+\mathcal{D}_{12}u(t),\\ y(t) &= \mathcal{C}_2\mathbf{x}(t)+\mathcal{D}_{21}w(t)+\mathcal{D}_{22}u(t), \qquad u(t)=\mathcal{Z}\mathcal{P}^{-1}\mathbf{x}(t) \qquad (1)\end{aligned}$$

Theorem 2 (H_∞ optimal control, no input delay ($\mathcal{T}_u=0$)).

Decision Variables: $\gamma, \mathcal{P}, \mathcal{Z}$

Optimization problem: $\min_{\gamma,\mathcal{Z},\mathcal{P}\succ 0} \gamma$

$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11} & (\mathcal{C}_1\mathcal{P}+\mathcal{Z}\mathcal{D}_{12}\mathcal{Z})\mathcal{T}^* \\ \mathcal{D}_{11}^* & -\gamma I & \mathcal{B}_1^* \\ \mathcal{T}(\mathcal{C}_1\mathcal{P}+\mathcal{Z}\mathcal{D}_{12}\mathcal{Z})^* & \mathcal{B}_1 & \mathcal{T}\left(\mathcal{A}\mathcal{P}+\mathcal{B}_2\mathcal{Z}\right)^*+\left(\mathcal{A}\mathcal{P}+\mathcal{B}_2\mathcal{Z}\right)\mathcal{T}^* \end{bmatrix} \preccurlyeq 0.$$

Then $\|y\|_{L_2} \le \gamma \|\omega\|_{L_2}$.

- We have cast the optimal control problem as a Linear Operator Inequality.

*For the case with input-delay, see upcoming talk at IFAC TDS, 2022.

Implementing the Controller on a DDF

Then implement the controller

- Method 1: Real-World Implementation on a DDF

$$u(t) = \mathcal{K}\begin{bmatrix} x(t) \\ \partial_s r_i(t+\tau_i s) \end{bmatrix} = K_1 x(t) + \sum_i \int_{-1}^{0} K_{2,i}(s)\partial_s r_i(t+\tau_i s)ds$$

$$r_i(t) = \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t) \end{bmatrix}$$

$$v(t) = \sum_{i=1} C_{vi} r_i(t-\tau_i) + \sum_{i=1} \int_{-\tau_i}^{0} C_{vdi}(s) r_i(t+s)ds.$$

  where the $C_{vi}, C_{vdi}, C_{ri}, B_{r1i}, B_{r2i}, D_{rvi}$ come from the DDF representation.

- Method 2: Simulation. Simulate as directly a PIE

$$\mathcal{T}\dot{\mathbf{x}}_f(t) = (\mathcal{A} + \mathcal{B}_2\mathcal{K})\mathbf{x}_f(t) + \mathcal{B}_1 w(t)$$

  and reconstruct the solution using $\mathbf{x}(t) = \mathcal{T}\mathbf{x}_f(t)$

Application to Large Networks

Problems with multiple (i.e. K) delay channels

Ex. 1: Spring-Mass Chain: (K delays, 2K states)

$$\dot{x}_1(t) = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix} (x_1(t) + x_1(t-\tau_1)) + \begin{bmatrix} 0 & 0 \\ k & b \end{bmatrix} x_2(t-\tau_2) + u(t)$$

$$\dot{x}_n(t) = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k & b \end{bmatrix} x_{n-1}(t-\tau_n) + w(t)$$

$$\dot{x}_i(t) = \begin{bmatrix} 0 & 1 \\ -2k & -2b \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k & b \end{bmatrix} (x_{i-1}(t-\tau_i) + x_{i+1}(t-\tau_{i+1}))$$

$$y(t) = x_n(t), \qquad z(t) = x_n(t) + .1u(t)$$

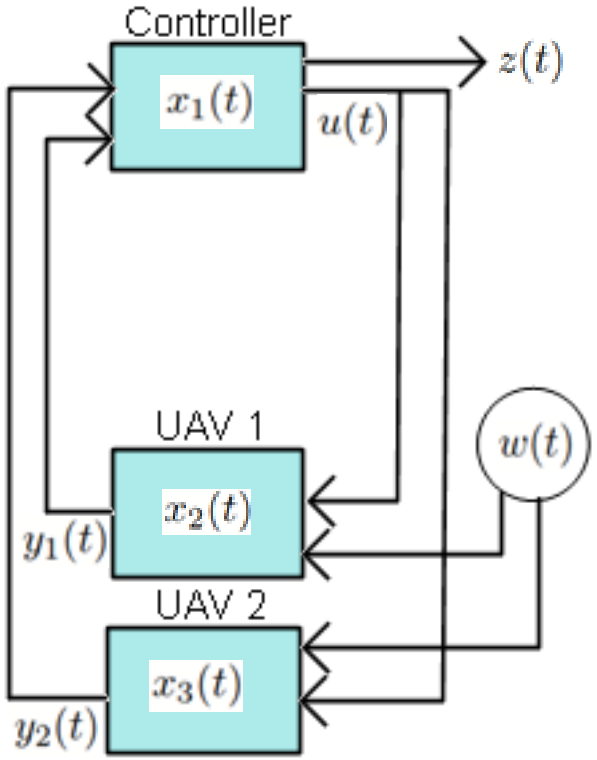
Ex. 2: Showers: (K delays, $2K$ states)

$$\dot{T}_{1i}(t) = T_{2i}(t) - w_i(t)$$

$$\dot{T}_{2i}(t) = -\alpha_i \left(T_{2i}(t-\tau_i) - w_i(t)\right) + \sum_{j\neq i}^{N} \gamma_{ij}\alpha_j \left(T_{2j}(t-\tau_j) - w_j(t)\right) + u_i(t)$$

$$z(t) = \left[\sum_{i=1}^{N} T_{1i}(t) \quad .1\sum_{i=1}^{N} u_i(t)\right]^T$$

$$\alpha_i = 1, \quad \gamma_{ij} = 1/N, \quad \tau_i = i.$$



Reduced Computation Time:

Ex.	Dimension Size nom	Dimension Size min	CPU seconds nom	CPU seconds min
Ex. 1 (K=5)	60	9	N/A	220.6
Ex. 1 (K=10)	220	19	N/A	9,350
Ex. 2 (K=5)	100	5	N/A	2.42
Ex. 2 (K=10)	400	10	N/A	94.7

Table: Computation times for nominal and minimal realizations. Times are H_∞-control.

DDEs are a poor choice for representing delayed networks

Better options include DDFs, and NDSs

PIETOOLS 2021b Implementation:

- Automates conversion between representations
- See `http://control.asu.edu/pietools`
  - User manual, documentation, etc.
- Conversion process is very fast (10ms)
- Provides a standardized representation of DDEs/DDFs/PIEs
  - Applies to a very large class of TDSs
  - Input delays, state delays, dist. delays, output delays
- A helpful input format
- More than 50 examples in the libraries

Advantages of DDFs:

- DDE representation in most software tools is naïve and hence inefficient
- DDFs provide a universal format for representing TDSs
- DDFs allow you to specify delayed channels
- The proposed method automates the task of specifying delayed channels.

Thank you for your attention

(Sponsored by: NSF CNS-1935453)