Minimal DDF Realizations of DDFs, DDEs and NDSs

Matthew Peet
Arizona State University
Tempe, AZ USA

American Control Conference
New Orleans, LA

May 18, 2021

The DDE Model of Delay (Zero Initial Conditions)

The Class of Delay Differential Equations (DDEs):

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

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

Note: Delays present in signals: $\{x(t-\tau_i), w(t-\tau_i), u(t-\tau_i)\}$

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 Problems are Tractable with the number of infinite-dimensional components is less than 50 (Here: $(n+m+p)K < 50$).

Minimal DDFs

The DDE Model of Delay (Zero Initial Conditions)

The DDE Model of Delay (Zero Initial Conditions)

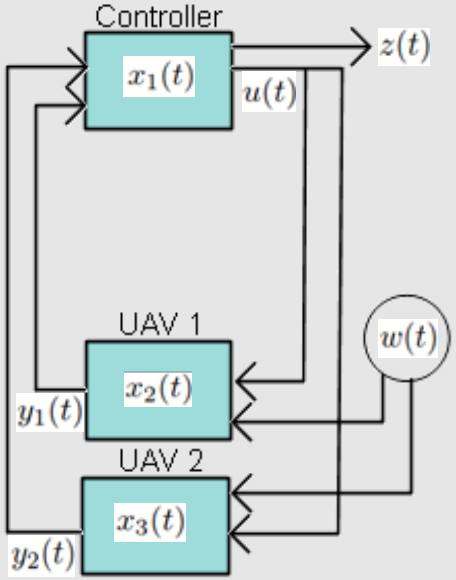
The Class of Delay Differential Equations (DDEs):

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\end{bmatrix} = \begin{bmatrix}A_0 & B_{10} & B_{20}\\ C_{10} & D_{11} & D_{12}\\ C_{20} & D_{21} & D_{22}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix} + \sum_{i=1}^{K}\begin{bmatrix}A_i & B_{1i} & B_{2i}\\ C_{1i} & D_{11i} & D_{12i}\\ C_{2i} & D_{21i} & D_{22i}\end{bmatrix}\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\end{bmatrix}$$
$$+\sum_{i=1}^{K}\int_{-\tau_i}^{0}\begin{bmatrix}A_{di}(s) & B_{1di}(s) & B_{2di}(s)\\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s)\\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s)\end{bmatrix}\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\end{bmatrix} ds$$

Note: Delays present in signals: $\{x(t-\tau_i), w(t-\tau_i), u(t-\tau_i)\}$

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 Problems are Tractable with the number of infinite-dimensional components is less than 50 (Here: $(n+m+p)K < 50$).



Model of a fleet of UAVs:

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

The NDS Model of Delay (Zero Initial Conditions)

The Class of Neutral Delay Systems (NDSs):

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

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

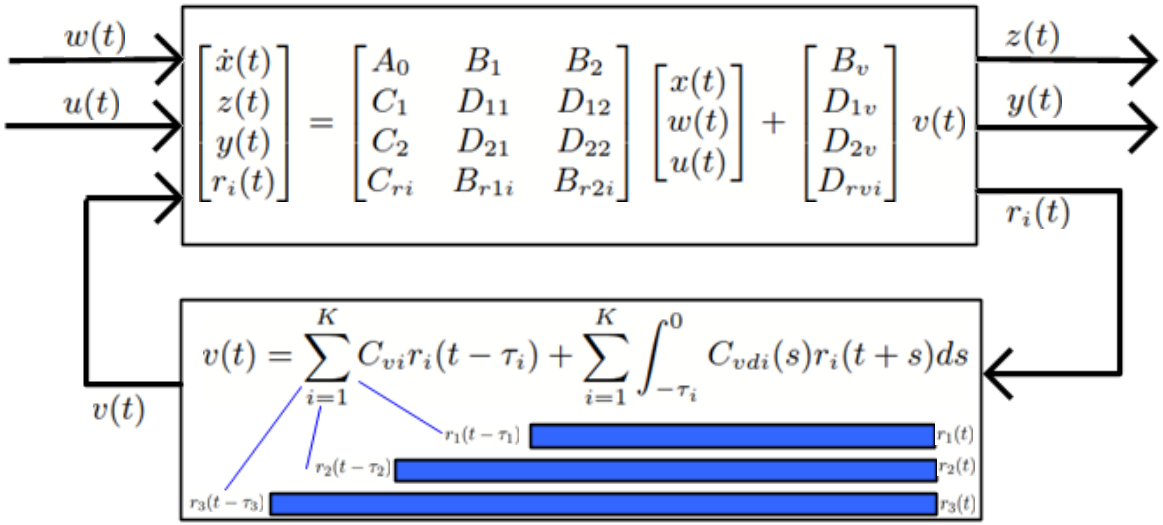
Note: Delays present in signals: $\{x(t-\tau_i), \dot{x}(t-\tau_i), w(t-\tau_i), u(t-\tau_i)\}$.

Tractable? Now we need $2n+m+p<50$.

Problem: Neither the DDE or NDS format allows you to specify which information gets delayed.

- General -Purpose methods and software must be designed for the worst case, where everything is delayed.
- Limits most software tools to the study of very small toy problems.
- Also an issue in numerical simulation.

The DDF Model of Delay (0 Initial Conditions)



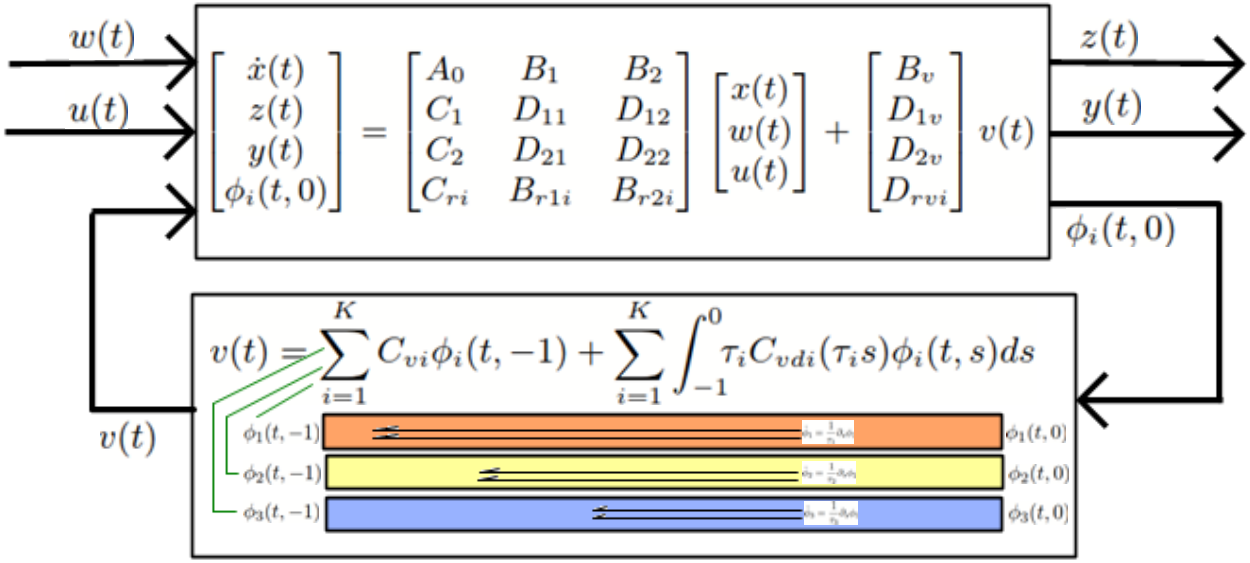
The Class of Differential Difference Equations (DDFs):

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

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

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

Alternative ODE-PDE Representation of the DDF Model



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.

Alternative PIE Representation of the DDF Model

The Class of Partial Integral Equation (PIE) Systems:

$$\mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{B}_{T_1}\dot{w}(t) + \mathcal{B}_{T_2}\dot{u}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)$$
$$z(t) = \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_{11}w(t) + \mathcal{D}_{12}u(t), \quad \mathbf{x}(t) = \begin{bmatrix} x(t) \\ \partial_s\phi_1(t,s) \\ \vdots \\ \partial_s\phi_K(t,s) \end{bmatrix}$$
$$y(t) = \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_{21}w(t) + \mathcal{D}_{22}u(t)$$

---

where $\{\mathcal{T}, \mathcal{A}, \cdots, \mathcal{D}_{22}\}$ are 4-PI Partial Integral Operators:

$$\mathcal{A} = \mathcal{P}\begin{bmatrix} \mathbf{A}_0, & \mathbf{A} \\ 0, & \{I_\tau, 0, 0\} \end{bmatrix}, \quad \mathcal{T} = \mathcal{P}\begin{bmatrix} I, & 0 \\ \mathbf{T}_0, & \{0, \mathbf{T}_a, \mathbf{T}_b\} \end{bmatrix}, \mathcal{B}_{T_1} = \mathcal{P}\begin{bmatrix} 0, & \emptyset \\ \mathbf{T}_1, & \{\emptyset\} \end{bmatrix}, \quad \mathcal{B}_{T_2} = \mathcal{P}\begin{bmatrix} 0, & \emptyset \\ \mathbf{T}_2, & \{\emptyset\} \end{bmatrix}, \mathcal{D}_{ij} = \mathcal{P}\begin{bmatrix} \mathbf{D}_{ij}, & \emptyset \\ \emptyset, & \{\emptyset\} \end{bmatrix}$$

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

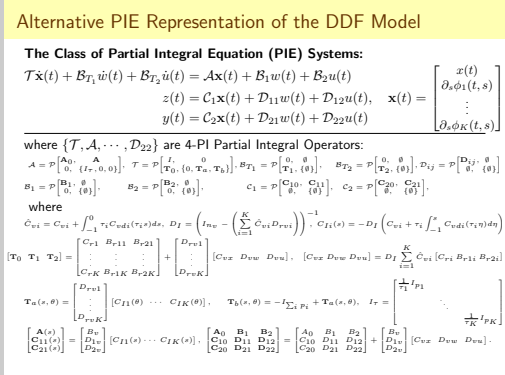
where

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

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

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

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



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

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

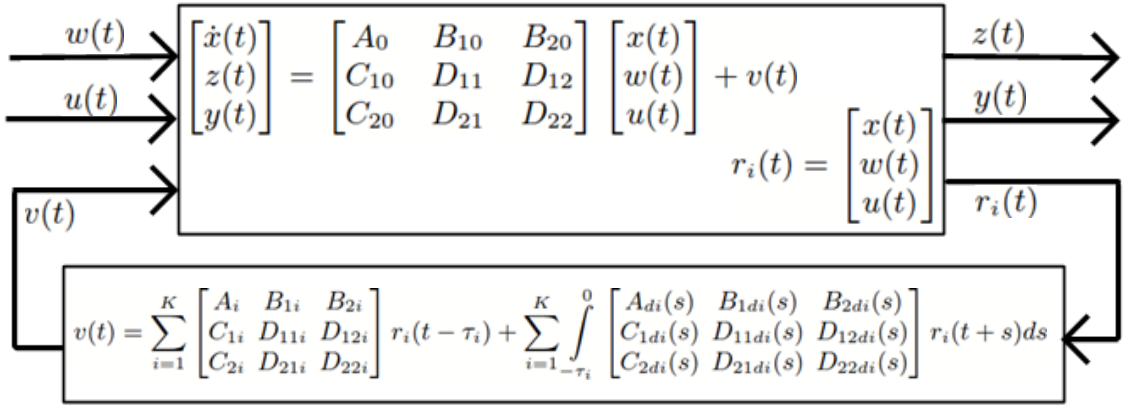
4-PI Operators include a 3-PI Operator, Defined as:

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

PIE Representations are required to utilize PIETOOLS 2021a:

- Construct and Solve Linear PI Inequalities (LPIs)
- Used for
  - Stability Analysis
  - H_∞-gain analysis
  - H_∞-optimal observer synthesis
  - H_∞-optimal controller synthesis.

Naïve Conversion From DDE to DDF Model



Define:

$$\begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\end{bmatrix} = I,\ C_{vi} = \begin{bmatrix}A_i & B_{1i} & B_{2i}\\ C_{1i} & D_{11i} & D_{12i}\\ C_{2i} & D_{21i} & D_{22i}\end{bmatrix},\ C_{vdi}(s) = \begin{bmatrix}A_{di}(s) & B_{1di}(s) & B_{2di}(s)\\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s)\\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s)\end{bmatrix},$$

$$D_{rvi} = 0, \qquad \begin{bmatrix}C_{ri} & B_{r1i} & B_{r2i}\end{bmatrix} = I,$$

Then the DDE

$$\{A_0, B_{ij}, C_{ij}, D_{ij}, B_{idj}, C_{idj}, D_{ijdk}\}$$

and the DDF

$$\{A_0, B_{i0}, C_{i0}, D_{ij}, B_v, D_{iv}, B_{r1i}, B_{r2i}, C_{ri}, C_{vi}, C_{vdi}, D_{rvi}\}$$

are equivalent (the size of the infinite-dimensional channel is unchanged).

Naïve Conversion From NDS to DDF Model

Define:

$$C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} & E_i \\ C_{1i} & D_{11i} & D_{12i} & E_{1i} \\ C_{2i} & D_{21i} & D_{22i} & E_{2i} \end{bmatrix}, \; C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) & E_{di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) & E_{1di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) & E_{2di}(s) \end{bmatrix},$$

$$\begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I, \quad \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ A_0 & B_1 & B_2 \end{bmatrix}, \; D_{rvi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I & 0 & 0 \end{bmatrix}$$

Then the NDS

$$\{A_0, B_{ij}, C_{ij}, D_{ij}, E_{ij}, B_{idj}, C_{idj}, D_{ijdk}, E_{ijdk}\}$$

and the DDF

$$\{A_0, B_{i0}, C_{i0}, D_{ij}, B_v, D_{iv}, B_{r1i}, B_{r2i}, C_{ri}, C_{vi}, C_{vdi}, D_{rvi}\}$$

are equivalent using

$$r_i(t) = \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ \dot{x}(t) \end{bmatrix}, \qquad i = 1, \cdots, K.$$

So the size of the infinite-dimensional channel is likewise unchanged.

How to find efficient DDF representations of DDFs?

Theorem 1.

Given a DDF (and choice of $Z(s)$), construct the matrices T_i, T_{di} as

$$T_i := C_{vi}\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix} \qquad Z(s)T_{di} := C_{vdi}(s)\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix}$$

Use the SVD to construct the smallest U_i and V_i such that

$$U_i V_i^T := \begin{bmatrix} T_i \\ T_{di}\end{bmatrix} \qquad i \in [K].$$

Now define:

$$\begin{bmatrix} \tilde{C}_{ri} & \tilde{B}_{r1i} & \tilde{B}_{r2i} & \tilde{D}_{rvi}\end{bmatrix} = V_i^T, \qquad \begin{bmatrix} \tilde{C}_{vi} \\ \tilde{C}_{vdi}(s)\end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & Z(s)\end{bmatrix} U_i.$$

Then the DDF

$$\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, B_{r1i}, B_{r2i}, C_{ri}, C_{vi}, C_{vdi}, D_{rvi}\}$$

and the DDF

$$\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, \tilde{B}_{r1i}, \tilde{B}_{r2i}, \tilde{C}_{ri}, \tilde{C}_{vi}, \tilde{C}_{vdi}, \tilde{D}_{rvi}\}$$

are equivalent DDF representations, where $\tilde{r}_i(t) = V_i^T \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t)\end{bmatrix}$.

How to find efficient DDF representations of DDFs?

Theorem 1.

Given a DDF (and choice of $Z(s)$), construct the matrices T_i, T_{di} as

$$T_i := C_{vi}\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix} \quad Z(s)T_{di} := C_{vdi}(s)\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix}$$

Use the SVD to construct the smallest U_i and V_i such that

$$U_iV_i^T := \begin{bmatrix} T_i \\ T_{di}\end{bmatrix} \qquad i \in [K].$$

Now define:

$$\begin{bmatrix} \tilde{C}_{ri} & \tilde{B}_{r1i} & \tilde{B}_{r2i} & \tilde{D}_{rvi}\end{bmatrix} = V_i^T, \quad \begin{bmatrix} \tilde{C}_{vi} \\ \tilde{C}_{vdi}(s)\end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & Z(s)\end{bmatrix} U_i.$$

Then the DDF

$$\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, B_{r1i}, B_{r2i}, C_{ri}, C_{vi}, C_{vdi}, D_{rvi}\}$$

and the DDF

$$\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, \tilde{B}_{r1i}, \tilde{B}_{r2i}, \tilde{C}_{ri}, \tilde{C}_{vi}, \tilde{C}_{vdi}, \tilde{D}_{rvi}\}$$

are equivalent DDF representations, where $\tilde{r}_i(t) = V_i^T\begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t)\end{bmatrix}$.

We are using the SVD to identify the information in $r_i(t) = \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ \dot{x}(t)\end{bmatrix}$ which does not influence

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

To use the SVD to construct the smallest U_i and V_i such that

$$U_iV_i^T := \begin{bmatrix} T_i \\ T_{di}\end{bmatrix},$$

Perform an SVD to obtain unitary U and V and eliminate zero singular values:

$$\begin{bmatrix} T_i \\ T_{di}\end{bmatrix} = U\Sigma V^T = \begin{bmatrix} u_1 & \cdots & u_{2n_v}\end{bmatrix} \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_{p_i} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0\end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_{2n_v}^T\end{bmatrix} = \underbrace{\begin{bmatrix} u_1^T \\ \vdots \\ u_{p_i}^T\end{bmatrix}^T \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{p_i}\end{bmatrix}}_{U_i} \underbrace{\begin{bmatrix} v_1^T \\ \vdots \\ v_{p_i}^T\end{bmatrix}}_{V_i^T}$$

Application to Several DDE Models

Ex. 1: Chain of N Spring-Masses:

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

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

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

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

Ex. 2: Network of N Showers:

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

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

$$z(t) = \begin{bmatrix} \sum_{i=1}^{N} T_{1i}(t) & .1\sum_{i=1}^{N} u_i(t) \end{bmatrix}^T$$

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

Ex. 3: Popular 2-delay system:

$$\dot{x}(t) = \begin{bmatrix} -2 & 0 \\ 0 & -.9 \end{bmatrix} x(t) + \begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} x(t-\tau_2) + \begin{bmatrix} .1 & 0 \\ 0 & .1 \end{bmatrix} \dot{x}(t-\tau_1)$$

Ex. 4: From [Sipahi, et al.]:

$$\dot{x}(t) = \begin{bmatrix} -2 & .2 & -.3 & 0 & -.4 \\ .2 & -3.8 & 0 & .7 & 0 \\ .8 & 0 & -1.6 & 0 & 0 \\ 0 & .8 & -.6 & -2 & .3 \\ -1 & -.1 & -1.5 & 0 & -1.8 \end{bmatrix} x(t)$$

$$+ \begin{bmatrix} -2.2 & 0 & 0 & 1 & 0 \\ 1.6 & -2.2 & 1.6 & 0 & 0 \\ -0.2 & -0.2 & -0.2 & -0.2 & -0.2 \\ 0 & 0.4 & -1.4 & -3.4 & 1 \\ -0.2 & 0.4 & -0.1 & -1.1 & -3.3 \end{bmatrix} x(t-\tau)$$

$$+ \begin{bmatrix} 0.40888 & 0.00888 & 0.20888 & -0.09112 & -0.29112 \\ 0 & 0.2 & 0 & 0 & 0.6 \\ -0.1 & -0.4 & 0 & -0.8 & 0 \\ 0 & 0 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & -0.1 \end{bmatrix} \dot{x}(t-\tau)$$

Numerical Results:

Ex.	Dimension Size nominal	minimal
Ex. 1 (N=5)	60	9
Ex. 1 (N=10)	220	19
Ex. 2 (N=5)	100	5
Ex. 2 (N=10)	400	10
Ex. 3	8	2
Ex. 4	10	5

Table: The total dimension of delayed channels r_i for nominal and minimal DDF realizations.

Recall: We assert problems are tractable when the ∞-dim part has size less than $\sum_i p_i < 50$!

PIETOOLS 2020a/2021a Implementation

PIETOOLS User Interfaces:

- `PIETOOLS_DDE`: User Interface for PIETOOLS. Used for declaration of DDEs, conversion to PIEs. LPIs for: Stability Analysis, H_∞-gain analysis, H_∞-optimal observer synthesis, H_∞-optimal controller synthesis.
- `PIETOOLS_DDF`: Alternative Interface used for declaration of DDFs

Executives which can be called from the `PIETOOLS` interfaces:

1. `examples_DDE_library_PIETOOLS`: 25 DDE examples with citations.
2. `examples_DDF_library_PIETOOLS`: 17 DDF and NDS examples w/ citations.
3. `initialize_PIETOOLS_DDE`: Parse user input for DDE elements.
4. `initialize_PIETOOLS_DDF`: Parse user input for DDF elements.
5. `initialize_PIETOOLS_NDS`: Parse user input for NDS elements.
6. `convert_PIETOOLS_NDS2DDF`: Convert NDS input format to DDF.
7. `convert_PIETOOLS_DDE`: Convert DDE input format to PIE.
8. `convert_PIETOOLS_DDF`: Convert DDE input format to PIE.
9. `minimize_PIETOOLS_DDE`: Construct minimal DDF representation of DDE.
10. `minimize_PIETOOLS_DDF`: Construct minimal DDF representation of DDF.

LMIs for ODEs can often be recast as LPIs for PIEs

4-PI Operators have the algebra properties of matrices

ODE:
$$\dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t)$$
$$z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t)$$
$$y(t) = C_2 x(t) + D_{21} w(t) + D_{22} u(t)$$
A, B_i, C_i, D_{ij} : matrices

Stability:
$$P \succ 0$$
$$A^T P + PA \preccurlyeq 0$$

Dual Stability:
$$P \succ 0$$
$$AP + PA^T \preccurlyeq 0$$

H_∞-norm: $\min_{\gamma, P} \gamma, \ P \succ 0$
$$\begin{bmatrix} -\gamma I & D_{11}^T & B_1^T P \\ D_{11} & -\gamma I & C_1 \\ PB_1 & C_1^T & A^T P + PA \end{bmatrix} \preccurlyeq 0$$

H_∞-optimal Estimator: $\min_{\gamma, P} \gamma$
$$P \succ 0$$
$$\begin{bmatrix} -\gamma I & -D_{11}^T & -(PB_1 + D_{21} Z)^T \mathcal{T} \\ -D_{11} & -\gamma I & C_1 \\ ()^T & C_1^T & ()^T + (PA + ZC_2) \end{bmatrix} \preccurlyeq 0$$

H_∞-optimal controller: $\min_{\gamma, P} \gamma$
$$P \succ 0$$
$$\begin{bmatrix} -\gamma I & D_{11} & (C_1 P + D_{12} Z) \\ D_{11}^T & -\gamma I & B_1^T \\ ()^T & B_1 & ()^T + (AP + B_2 Z) \end{bmatrix} \preccurlyeq 0$$

PIE:
$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)$$
$$z(t) = \mathcal{C}_1 \mathbf{x}(t) + \mathcal{D}_{11} w(t) + \mathcal{D}_{12} u(t)$$
$$y(t) = \mathcal{C}_2 \mathbf{x}(t) + \mathcal{D}_{21} w(t) + \mathcal{D}_{22} u(t)$$
$\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$: 3/4PI operators

Stability:
$$\mathcal{P} \succ 0$$
$$\mathcal{A}^* \mathcal{P} \mathcal{T} + \mathcal{T}^* \mathcal{P} \mathcal{A} \preccurlyeq 0$$

Dual Stability:
$$\mathcal{P} \succ 0$$
$$\mathcal{A} \mathcal{P} \mathcal{T}^* + \mathcal{T} \mathcal{P} \mathcal{A}^* \preccurlyeq 0$$

H_∞-norm: $\min_{\gamma, \mathcal{P}} \gamma, \ \mathcal{P} \succ 0$
$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11}^* & \mathcal{B}_1^* \mathcal{P} \mathcal{T} \\ \mathcal{D}_{11} & -\gamma I & \mathcal{C}_1 \\ \mathcal{T}^* \mathcal{P} \mathcal{B}_1 & \mathcal{C}_1^* & \mathcal{A}^* \mathcal{P} \mathcal{T} + \mathcal{T}^* \mathcal{P} \mathcal{A} \end{bmatrix} \preccurlyeq 0$$

H_∞-optimal Estimator: $\min_{\gamma, \mathcal{P}} \gamma$
$$\mathcal{P} \succ 0$$
$$\begin{bmatrix} -\gamma I & -\mathcal{D}_{11}^* & -(\mathcal{P}\mathcal{B}_1 + \mathcal{D}_{21} \mathcal{Z})^* \mathcal{T} \\ -\mathcal{D}_{11} & -\gamma I & \mathcal{C}_1 \\ ()^* & \mathcal{C}_1^* & ()^* + \mathcal{T}^* (\mathcal{P}\mathcal{A} + \mathcal{Z}\mathcal{C}_2) \end{bmatrix} \preccurlyeq 0$$

H_∞-optimal controller: $\min_{\gamma, \mathcal{P}} \gamma$
$$\mathcal{P} \succ 0$$
$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11} & (\mathcal{C}_1 \mathcal{P} + \mathcal{Z} D_{12} \mathcal{Z}) \mathcal{T}^* \\ \mathcal{D}_{11}^* & -\gamma I & \mathcal{B}_1^* \\ ()^* & \mathcal{B}_1 & ()^* + (\mathcal{A}\mathcal{P} + \mathcal{B}_2 \mathcal{Z}) \mathcal{T}^* \end{bmatrix} \preccurlyeq 0$$

Application to Several DDE Models

Ex. 1: Chain of N Spring-Masses:

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

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

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

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

Ex. 2: Network of N Showers:

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

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

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

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

Ex. 3: Popular 2-delay system:

$$\dot{x}(t) = \begin{bmatrix} -2 & 0 \\ 0 & -.9 \end{bmatrix} x(t) + \begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} x(t-\tau_2) + \begin{bmatrix} .1 & 0 \\ 0 & .1 \end{bmatrix} \dot{x}(t-\tau_1)$$

Ex. 4: From [Sipahi, et al.]:

$$\dot{x}(t) = \begin{bmatrix} -2 & .2 & -.3 & 0 & -.4 \\ .2 & -3.8 & 0 & .7 & 0 \\ .8 & 0 & -1.6 & 0 & 0 \\ 0 & .8 & -.6 & -2 & .3 \\ -1 & -.1 & -1.5 & 0 & -1.8 \end{bmatrix} x(t)$$

$$+ \begin{bmatrix} -2.2 & 0 & 0 & 1 & 0 \\ 1.6 & -2.2 & 1.6 & 0 & 0 \\ -0.2 & -0.2 & -0.2 & -0.2 & -0.2 \\ 0 & 0.4 & -1.4 & -3.4 & 1 \\ -0.2 & 0.4 & -0.1 & -1.1 & -3.3 \end{bmatrix} x(t-\tau)$$

$$+ \begin{bmatrix} 0.40888 & 0.00888 & 0.20888 & -0.09112 & -0.29112 \\ 0 & 0.2 & 0 & 0 & 0.6 \\ -0.1 & -0.4 & 0 & -0.8 & 0 \\ 0 & 0 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & -0.1 \end{bmatrix} \dot{x}(t-\tau)$$

Reduced Computation Time:

Ex.	Dimension Size nom	Dimension Size min	CPU seconds nom	CPU seconds min
Ex. 1 (n=5)	60	9	N/A	220.6
Ex. 1 (n=10)	220	19	N/A	9,350
Ex. 2 (n=5)	100	5	N/A	2.42
Ex. 2 (n=10)	400	10	N/A	94.7
Ex. 3	8	2	22.56	.332
Ex. 4	10	5	147.3	4.915

Table: Computation times for nominal and minimal realizations. Times are H_∞-control for Exs. 1 and 2 and stability analysis for Exs. 3 and 4.

Conclusion

An Automated System for Constructing Efficient Representations of DDEs, DDFs, and NDSs

Note: The arxiv paper includes the case of non-zero initial conditions.

PIETOOLS 2021a Implementation:

- Don't need to like PIEs to use PIETOOLS
- See `http://control.asu.edu/pietools`
  - Self-installing
  - User manual, documentation, etc.
- Conversion process is very fast (10ms)
  - Applies to a very large class of TDSs
  - Input delays, state delays, dist. delays, output delays
- A helpful input format
- Lots of examples (add yours too!)

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.

Good Luck

(Sponsored by: NSF CNS-1739990)

With Luck, you won't need luck

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As the
some unprocessed data that should have been added to the final page
page has been added to receive it.
If you rerun the document (without altering it) this surplus page will g
because LaTeX now knows how many pages to expect for this documen