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

Matthew M. Peet *, **Sachin Shivakumar** *,

* School for the Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85298 USA. (e-mail: mpeet@asu.edu, sshivak8@asu.edu)

Abstract: Delay-Differential Equations (DDEs) are the most common representation for systems with delay. However, the DDE representation has limitations. In network models with delay, the delayed channels are typically low-dimensional and accounting for this heterogeneity is challenging in the DDE framework. In addition, DDEs cannot be used to model difference equations. In this paper, we examine alternative representations for networked systems with delay and provide formulae for conversion between representations. First, we examine the Differential-Difference (DDF) formulation which allows us to represent the low-dimensional nature of delayed information. Next, we consider the coupled ODE-PDE framework and extend this to the recently developed Partial-Integral Equation (PIE) representation. The PIE framework has the advantage that it allows the H_∞-optimal estimation and control problems to be solved efficiently using the recently developed software package PIETOOLS. In each case, we consider a very general class of networks, specifically accounting for four sources of delay - state delay, input delay, output delay, and process delay. Finally, we use a scalable network model of temperature control to show that the use of the DDF/PIE formulation allows for optimal control of a network with 40 users, 80 states, 40 delays, 40 inputs, and 40 disturbances.

Keywords: Delay, PDEs, Networked Control

1. INTRODUCTION

Delay-Differential Equations (DDEs) are a convenient shorthand notation used to represent what is perhaps the simplest form of spatially-distributed phenomenon - transport. Because of their notational simplicity, it is common to use DDEs to model very complex systems with multiple sources of delay - including almost all models of control over and of "networks".

To illustrate the DDE framework, consider a swarm of N Uncrewed Aerial Vehicles (UAVs) over a wireless network. Each UAV, i, has a state, $x_i(t) \in \mathbb{R}^{n_i}$ which represents displacement from a desired state (the concatenation of all such states is denoted x). Each UAV has local sensors which measure y_i and this information is transmitted to a centralized control authority. There is also an input, u, a regulated output, z, and a vector of disturbances, w - including both process and sensor noise. We model this system as follows.

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

This relatively simple model shows that delayed channels are often low dimensional ($\mathbb{R}^{n_i}$ vs. $\mathbb{R}^{\sum n_i}$) and specifies four separate yet individually significant sources of delay. Specifically, we have: state delay ($\hat{\tau}_{ij}$); input delay (h_i); process delay ($\bar{\tau}_i$); and output delay ($\tilde{\tau}_i$).

This network is modeled as a DDE - a structure formulated in Eqns. (2) in Section 2 and applied to this example problem in Subsection 6.1. If we were to consider control of such a network, however, we find that while there are algorithms for control of DDEs, these algorithms are complex and are memory-limited to a relatively small number of UAVs (perhaps 4-5). The premise of this paper, however, is that the limitations of these algorithms are not caused by inefficient algorithms, but rather the cause is the failure to account for the low dimensional nature of the delayed channels. Specifically, we note that in our UAV model, that while the concatenated state, $x(t)$, is high-dimensional, the individual delayed channels, $x_i(t)$, are of much lower dimension. If we represent the network as a DDE using the formulation in Subsection 6.1, then this low-dimensional nature of the delayed channels is lost. Furthermore, we note that DDEs cannot represent some important system designs - See Subsection 3.1.

For these reasons, we consider the use of Differential Difference Equations (DDFs) in Section 3. The DDF formulation allows for the representation of delayed information in heterogeneous low-dimensional channels. Specifically, the infinite-dimensional component of state-space (as defined in Gu (2010); Pepe et al. (2008)) in this DDF framework is then $\prod_i L_2[-\tau_i, 0]^{n_i}$ as opposed to $\prod_i L_2[-\tau_i, 0]^{\sum n_i}$, which would be the traditional (e.g. Bensoussan et al. (1993)) infinite-dimensional component of the state-space in the DDE model of this network. In addition to providing a more compact notion of state, in Subsection 3.1 DDFs

[1] This work was supported by the NSF and NIH under grants No. 1935453 and 2054354.

will allow us to represent the difference equations which arise in some network models.

From the DDF model, in Section 4 we briefly turn to the class of coupled ODE-PDE models. Backstepping methods have been developed for ODE-PDE models of delay in Krstic and Smyshlyaev (2008); Zhu et al. (2015) and the formulae we present in this section for conversion of DDFs to the ODE-PDE framework may prove useful if the reader is interested in application or further development of these backstepping methods. However, the primary reason for including the ODE-PDE formulation in this manuscript is that it is relatively easy to convert a coupled ODE-PDE model to a PIE.

Specifically, in Section 5, we consider the Partial Integral Equation (PIE) representation of a delayed network. PIE representations have the advantage that they are defined by Partial Integral (PI) operators. Unlike Dirac and differential operators, PI operators are bounded and form an algebra. Furthermore, as discussed in Peet. (2021), PIE models do not require boundary conditions or continuity constraints - simplifying analysis and optimal control problems. Indeed, it has been recently shown that many problems in analysis, optimal estimation and optimal control of coupled ODE-PDE models can be formulated as optimization over the cone of positive PI operators. Indeed, in Section 7, we will show that use of the PIE formulation allows for optimal control of a 40 user, 80-state, 40-delay, 40-input, 40-disturbance network model of temperature control on a desktop computer with 128GB RAM.

Finally, we note that while subsets of the DDF and ODE-PDE representations of delay systems can be found in, e.g. Bensoussan et al. (1993); Hale (1971); Curtain and Zwart (1995); Gu et al. (2003); Niculescu (2001); Richard (2003), previous models only considered a subset of the possible signals and sources of delay.

2. THE DDE REPRESENTATION

A Delay-Differential Equation (DDE) has the form

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\end{bmatrix}=\begin{bmatrix}A_0 & B_1 & B_2\\ C_{10} & D_{11} & D_{12}\\ C_{20} & D_{21} & D_{22}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix} \tag{2}$$
$$+\sum_{i=1}^{K}\begin{bmatrix}A_i & B_{1i} & B_{2i}\\ C_{1i} & D_{11i} & D_{12i}\\ C_{2i} & D_{21i} & D_{22i}\end{bmatrix}\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\end{bmatrix}$$
$$+\sum_{i=1}^{K}\int_{-\tau_i}^{0}\begin{bmatrix}A_{di}(s) & B_{1di}(s) & B_{2di}(s)\\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s)\\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s)\end{bmatrix}\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\end{bmatrix}ds$$

where the signals are defined as:

- The present state $x(t)\in\mathbb{R}^n$
- The disturbance or exogenous input, $w(t)\in\mathbb{R}^m$
- The controlled input, $u(t)\in\mathbb{R}^p$
- The regulated or external output, $z(t)\in\mathbb{R}^q$
- The observed or sensed output, $y(t)\in\mathbb{R}^r$

For convenience, we combine all sources of delay (state, input, output, process) into a single set of delays $\{\tau_i\}_{i=1}^K$. In Subsection 6.1, the UAV network is formulated in this DDE representation.

2.1 Advantages of the DDE Formulation

The DDE formulation is the prima facie modeling tool for systems with delay and as such is used in almost all network models. The DDE representation has a clear and intuitive meaning. Furthermore, most algorithms and analysis tools are built for this representation. Specifically, Lyapunov-Krasovskii and Lyapunov-Razumikhin stability tests are naturally formulated in this framework.

As mentioned in the introduction, however, the DDE framework does not allow for the representation of difference equations and does not allow us to identify which of the states and inputs are delayed by which amount. For this reason, we consider next the DDF representation.

3. THE DDF REPRESENTATION

A generalization of the DDE formulation is the Differential-Difference (DDF) representation. Simplified versions of this formulation were previously considered in, e.g. Gu (2010); Pepe et al. (2008). In addition to the signals considered in the DDE representation, the DDF representation adds the following.

- The items stored in the signal $r_i(t)\in\mathbb{R}^{p_i}$ are the parts of $x(t)$, $w(t)$, $u(t)$, $v(t)$ which can be delayed by amount τ_i. The signals r_i may be considered the infinite-dimensional parts of the system.
- The "output" signal $v(t)\in\mathbb{R}^{n_v}$ extracts information from the infinite-dimensional signals r_i and distributes this information to the state, sensed output, and regulated output. This information can also be re-delayed by feeding back directly into the r_i.

The governing equations may now be represented in the more compact form of Eqns. (3).

$$\begin{aligned}\dot{x}(t)&=A_0x(t)+B_1w(t)+B_2u(t)+B_vv(t)\\ z(t)&=C_{10}x(t)+D_{11}w(t)+D_{12}u(t)+D_{1v}v(t)\\ y(t)&=C_{20}x(t)+D_{21}w(t)+D_{22}u(t)+D_{2v}v(t)\\ r_i(t)&=C_{ri}x(t)+B_{r1i}w(t)+B_{r2i}u(t)+D_{rvi}v(t)\\ v(t)&=\sum_{i=1}^{K}C_{vi}r_i(t-\tau_i)+\sum_{i=1}^{K}\int_{-\tau_i}^{0}C_{vdi}(s)r_i(t+s)ds\end{aligned} \tag{3}$$

Although Eqns. (3) are more compact, they are significantly more general than the DDE in (2). Specifically, if we define the conversion formula

$$D_{rvi}=0,\ B_v=[I_n\ 0\ 0],\ D_{1v}=[0\ I_q\ 0],\ D_{2v}=[0\ 0\ I_r]$$
$$C_{ri}=\begin{bmatrix}I_n\\0\\0\end{bmatrix},\ B_{r1i}=\begin{bmatrix}0\\I_m\\0\end{bmatrix},\ B_{r2i}=\begin{bmatrix}0\\0\\I_p\end{bmatrix},$$
$$C_{vi}=\begin{bmatrix}A_i & B_{1i} & B_{2i}\\ C_{1i} & D_{11i} & D_{12i}\\ C_{2i} & D_{21i} & D_{22i}\end{bmatrix},\ C_{vdi}=\begin{bmatrix}A_{di} & B_{1di} & B_{2di}\\ C_{1di} & D_{11di} & D_{12di}\\ C_{2di} & D_{21di} & D_{22di}\end{bmatrix}, \tag{4}$$

then the solution to the DDF is also a solution to the DDE and vice-versa.

Lemma 1. Suppose u, w, x, y, and z satisfy Eqn. (2). If C_{vi}, C_{vdi}, C_{ri}, B_{r1i} , B_{r1i}, D_{rvi}, B_v, D_{1v}, and D_{2v} are as defined in Eqns. (4), then u, w, x, y, and z also satisfy Eqns. (3) with $r_i(t)=\left[x(t)^T\ w(t)^T\ u(t)^T\right]^T$.

Lemma 2. Suppose u, w, x, y, r_i and z satisfy Eqns. (3) where C_{vi}, C_{vdi}, C_{ri}, B_{r1i} , B_{r1i}, D_{rvi}, B_v, D_{1v}, and D_{2v} are as defined in Eqns. (4). Then u, w, x, y, and z also satisfy Eqn. (2).

It is not possible, in general, to convert a DDF to a DDE as the class of DDFs is more general than the DDEs.

3.1 Advantages of the DDF Representation

The first advantage of the DDF representation is that it is more general than the DDE representation in that it may include difference equations (which are incompatible with the DDE framework). To illustrate, suppose we set all matrices to zero except D_{rvi} and C_{vi}. Then we have the following set of difference equations

$$r_i(t) = \sum\nolimits_{j=1}^{K} D_{rvi} C_{vj} r_j(t - \tau_j) \qquad i = 1, \cdots, K.$$

The second advantage of the DDF representation occurs when the delayed channels only include subsets of the state. For example, if the matrices A_i have low rank (ignoring input and disturbance delay), then $A_i = \tilde{A}_i \hat{A}_i$ for some $\hat{A}_i$, $\tilde{A}_i$ where $\hat{A}_i \in \mathbb{R}^{l_i \times n}$ with $l_i < n$ and we may choose $C_{vi} = \tilde{A}_i$ and $C_{ri} = A_i$. The dimension of $r_i(t)$ now becomes $\mathbb{R}^{l_i}$. This decomposition may be used to reduce complexity in the DDF formulation if $l_i < n$. This reduction is illustrated in detail using the UAV network model in Subsection 6.2 and the temperature control network in Section 7.

A disadvantage of the DDF formulation is that fewer tools are available for systems in this representation. This is partially due to the fact that the class of DDFs is larger than the DDEs and thus the tools must be more general. However, we do note that versions of both the Lyapunov-Krasovskii (Gu (2010)) and Lyapunov-Razumikhin (Zhang and Chen (1998)) stability tests have been formulated in the DDF framework.

4. THE COUPLED ODE-PDE REPRESENTATION

Before proceeding to the PIE representation, we briefly consider the coupled ODE-PDE representation. Use of the DDF formulation facilitiates conversion to the ODE-PDE formulation in that the inputs and outputs to the infinite-dimensional channels have already been identified. Conversion of a DDF to an ODE-PDE can be done directly as follows, where the matrices in the ODE-PDE model are the same ones used to define the DDF.

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + B_2 u(t) + B_v v(t) \qquad (5)\\
z(t) &= C_{10} x(t) + D_{11} w(t) + D_{12} u(t) + D_{1v} v(t)\\
y(t) &= C_{20} x(t) + D_{21} w(t) + D_{22} u(t) + D_{2v} v(t)\\
\dot{\phi}_i(t,s) &= \frac{1}{\tau_i} \phi_{i,s}(t,s)\\
\phi_i(t,0) &= C_{ri} x(t) + B_{r1i} w(t) + B_{r2i} u(t) + D_{rvi} v(t)\\
v(t) &= \sum_{i=1}^{K} C_{vi} \phi_i(t,-1) + \sum_{i=1}^{K} \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s) \phi_i(t,s) ds
\end{aligned}$$

In Eqns. (5), the infinite-dimensional part of the state is clearly defined as ϕ_i - which represents a pipe through which information is flowing. This representation presented here is somewhat atypical, however, in that we have scaled all the pipes to have unit length and accelerated or decelerated flow through the pipes according to the desired delay. Clearly, solutions to Eqns. (5) and Eqns. (3) are equivalent, as in the following lemma.

Lemma 3. Suppose u, w, x, r_i, v, y, and z satisfy Eqns. (3). Then u, w, x, v, y, and z also satisfy Eqns. (5) with

$$\phi_i(t,s) = r_i(t + \tau_i s).$$

Similarly, if u, w, x, v, y, ϕ_i and z satisfy Eqns. (5), then u, w, x, v, y, and z satisfy Eqns. (3) with $r_i(t) = \phi_i(t,0)$.

5. THE PIE REPRESENTATION

A Partial Integral Equation (PIE) has the form

$$\begin{aligned}
\mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{B}_{T_1}\dot{w}(t) + \mathcal{B}_{T_2}\dot{u}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t)\\
z(t) &= \mathcal{C}_1 \mathbf{x}(t) + \mathcal{D}_{11} w(t) + \mathcal{D}_{12} u(t),\\
y(t) &= \mathcal{C}_2 \mathbf{x}(t) + \mathcal{D}_{21} w(t) + \mathcal{D}_{22} u(t), \qquad (6)
\end{aligned}$$

where the operators $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ are Partial Integral (PI) operators and have the form

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

where

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

Heretofore, we have shown that the DDE is a special case of the DDF, which is equivalent to a coupled ODE-PDE, where coupling occurs at the boundary. Given a coupled ODE-PDE representation, it is relatively straightforward to convert to a PIE by defining the operators $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ for which solutions to Eqns. (6) also define solutions to Eqns. (3) and Eqns. (5). Specifically, these operators are defined in Eqns. (7)

Lemma 4. Suppose u, w, x, ϕ_i, v, y, and z satisfy Eqns. (5). Then u, w, y, and z also satisfy Eqns. (6) with $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ as defined in (7) and

$$\mathbf{x}(t) := \begin{bmatrix} x(t) \\ \phi_{1,s}(t,\cdot) \\ \vdots \\ \phi_{K,s}(t,\cdot) \end{bmatrix}.$$

Lemma 5. Suppose u, w, y, $\mathbf{x}$ and z satisfy Eqns. (6) with $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ as defined in (7). Then u, w, y, and z satisfy Eqns. (5) with

$$\begin{bmatrix} x(t) \\ \phi_1(t,\cdot) \\ \vdots \\ \phi_K(t,\cdot) \end{bmatrix} = \mathcal{T}\mathbf{x}(t) + \mathcal{B}_{T1} w(t) + \mathcal{B}_{T2} u(t).$$

Note that when $D_{rvi} = 0$, $D_I = I$.

Proof. See Peet (2020b) at Arxiv for the proof.

5.1 Advantages of the PIE Representation

The structure of the PIE representation is inherited from the DDF and ODE-PDE representations and can thus be used to represent low-dimensional delay channels. However, the primary benefit of using the PIE representation is computational. First, PIE representations contain no implicit dynamics. In the DDE formulation, there is an implicit relationship between $x(t)$ and $x(t - \tau_i)$ which is typically leveraged through integration by parts or some other analysis tool. This implicit constraint extends to the DDF representation, although in this case, it is confined to the definition of the vector $v(t)$. In the ODE-PDE representation, the implicit dynamics are defined by the boundary condition and differentiability of the infinite-dimensional state, ϕ. Such implicit constraints are often represented in a compact form as the "domain of the infinitesimal generator". By contrast, in the PIE representation, the state is ϕ_s which is assumed to be in L_2 but is

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

$$
\mathcal{B}_1=\mathcal{P}\begin{bmatrix}\mathbf{B}_1, & \emptyset\\ 0, & \{\emptyset\}\end{bmatrix},\quad \mathcal{B}_2=\mathcal{P}\begin{bmatrix}\mathbf{B}_2, & \emptyset\\ 0, & \{\emptyset\}\end{bmatrix},\quad \mathcal{C}_1=\mathcal{P}\begin{bmatrix}\mathbf{C}_{10}, & \mathbf{C}_{11}\\ \emptyset, & \{\emptyset\}\end{bmatrix},\quad \mathcal{C}_2=\mathcal{P}\begin{bmatrix}\mathbf{C}_{20}, & \mathbf{C}_{21}\\ \emptyset, & \{\emptyset\}\end{bmatrix},\quad \mathcal{D}_{ij}=\mathcal{P}\begin{bmatrix}\mathbf{D}_{ij}, & \emptyset\\ \emptyset, & \{\emptyset\}\end{bmatrix} \tag{7}
$$

where

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

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

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

$$
\begin{bmatrix}\mathbf{A}(s)\\ \mathbf{C}_{11}(s)\\ \mathbf{C}_{21}(s)\end{bmatrix}=\begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\end{bmatrix}[C_{I1}(s)\ \cdots\ C_{IK}(s)],\ \begin{bmatrix}\mathbf{A}_0 & \mathbf{B}_1 & \mathbf{B}_2\\ \mathbf{C}_{10} & \mathbf{D}_{11} & \mathbf{D}_{12}\\ \mathbf{C}_{20} & \mathbf{D}_{21} & \mathbf{D}_{22}\end{bmatrix}=\begin{bmatrix}A_0 & B_1 & B_2\\ C_{10} & D_{11} & D_{12}\\ C_{20} & D_{21} & D_{22}\end{bmatrix}+\begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\end{bmatrix}[C_{vx}\ D_{vw}\ D_{vu}]. \tag{8}
$$

otherwise unconstrained. As a result, the PIE representation is well-suited for computation. Furthermore, the representation is defined using the algebra of Partial Integral (PI) operators. If we define the sub-algebra of PI operators parameterized by polynomials, then the software package PIETOOLS described in Shivakumar et al. (2020a) allows for: manipulation of PI operators as a class object; declaration of PI operator variables; enforcement of PI operator positivity constraints; and solution of convex optimization problems defined by linear operator inequality constraints. For a more extensive discussion of the optimization of PI operators and their use in analysis and optimal estimation and control of infinite dimensional systems, we refer to the PIETOOLS manual in Shivakumar et al. (2021) or any of the recent papers on analysis and control in the PIE framework - e.g. Shivakumar et al. (2020b); Wu et al. (2019). Without embarking on an exhaustive discussion of these results, we note that the consensus seems to be that analysis and control in the PIE framework is possible when the distributed-parameter part of the state is in L_2^N where $N \le 50$.

We also briefly note some disadvantages of the PIE framework. The disadvantage is primarily due to the LHS of Eqn. (6) which is of the form

$$
\mathcal{T}\dot{\mathbf{x}}(t)+\mathcal{B}_{T_1}\dot{w}(t)+\mathcal{B}_{T_2}\dot{u}(t).
$$

The presence of $\dot{u}$ in the LHS can be eliminated if the feedback controller is of the form $u(t)=\mathcal{K}\mathbf{x}(t)$. However, if we have process delay ($\bar{\tau}_i$), then $\mathcal{B}_{T_1}\neq 0$ and hence $\dot{w}$ appears in the equation. Accounting for the relationship between w and $\dot{w}$ is an unsolved problem in the analysis and control of systems in the PIE representation.

6. MODELING OF A NETWORK OF UAVS

To illustrate some of the differences between the DDE, DDF and PIE representations, we again consider control of a network of UAVs. In this section, we focus on the DDE and DDF representations, as the state dimension in the PIE formulation is inherited from the DDF and conversion is straightforward using the formulae provided. For simplicity, we initially ignore the state delays governing interactions between UAVs. Furthermore, we map the process, input, and output delays to a common set of delays, $\{\tau_j\}_{j=1}^{3N}$ where we identify the index for the process delay for state x_i as τ_i, the index for input delay in state x_i as τ_{N+i}, and the index of the output delay from state x_i as τ_{2N+i}. The process noise is dimension $w(t)\in\mathbb{R}^m$, the input is dimension $u(t)\in\mathbb{R}^p$, all states are dimension $x_i(t)\in\mathbb{R}^n$ and the outputs are all dimension $y_i(t)\in\mathbb{R}^r$. In this case, we re-write the network as in Eqns. (1):

$$
\begin{aligned}
\dot{x}_i(t) &= a_i x_i(t)+\sum\nolimits_{j=1}^{N}a_{ij}x_j(t)\\
&\quad + b_{1i}w(t-\tau_i)+b_{2i}u(t-\tau_{N+i})\\
z(t) &= C_1x(t)+D_{12}u(t)\\
y_i(t) &= c_{2i}x_i(t-\tau_{2N+i})+d_{21i}w(t-\tau_{2N+i}).
\end{aligned}
$$

6.1 DDE Representation

To model this network as a DDE, we consider Eqn. (2) where $K=3N$ for a given C_{10} and D_{12}. First, we define A_0 blockwise as

$$
[A_0]_{ij}=\begin{cases}a_i, & i=j\\ a_{ij} & \text{otherwise}\end{cases}
$$

and define the following matrices blockwise for $i = 1,\cdots,N$ as

$$
\begin{aligned}
B_{1,i}&=e_i\otimes b_{1i}, \quad B_{2,N+i}=e_i\otimes b_{2i},\\
C_{2,2N+i}&=e_i\otimes c_{2i}, \quad D_{21,2N+i}=e_i\otimes d_{2i}.
\end{aligned}
$$

All other undefined matrices in Eqn. (2) are 0. The DDE representation of the network has the obvious disadvantage that there are $3N$ delays and each delayed channel contains all states and inputs - yielding an aggregate delayed channel of size $\mathbb{R}^{3N(nN+m+p)}$.

6.2 DDF Representation

To efficiently model the network model as a DDF, we retain the matrix A_0 from the DDE model in Subsec. 6.1, set $C_1=C_{10}$ and leave D_{12} unchanged. First, we define the vectors $r_i(t)$ and $v(t)$ using B_{r1i}, B_{r2i},C_{ri}, C_{vi}, B_v, and B_{2v} (all other matrices are 0). The first 3 sets of matrices are defined for $i=1,\cdots,N$ as $B_{r1,i}=b_{1i}$, $B_{r1,2N+i}=d_{21i}$, $B_{r2,N+i}=b_{2i}$, and $C_{r,2N+i}=c_{2i}$. We presume the UAV state dimensions (n) are less than the size of the aggregate input (m) and disturbance vectors (p) (i.e. $n<m$ and $n<p$). In this case it is preferable to delay only the part of the input and disturbance signals which affects each UAV. We now have the following definition for r_i for $i=1,\cdots,3N$.

$r_i(t) =$

$$\begin{cases} b_{1i}w(t) & i \in [1, N] \\ b_{2,i-N}u(t) & i \in [N+1, 2N] \\ c_{2,i-2N}x_{i-2N}(t) + d_{21,i-2N}w(t) & i \in [2N+1, 3N]. \end{cases}$$

Next, we construct output $v(t)$ by defining C_{vi} for $i = 1, \cdots, 3N$ as $C_{vi} = e_i \otimes I_{p_i}$ which yields

$$v(t) = \begin{bmatrix} r_1(t-\tau_1)^T & \cdots & r_{3N}(t-\tau_{3N})^T \end{bmatrix}^T.$$

Finally, we feed $v(t)$ back into the dynamics using

$$B_v = [I \; \cdots \; I \; I \; \cdots \; I \; 0], \; D_{2v} = [0 \; \cdots \; 0 \; 0 \; \cdots \; 0 \; I],$$

which recovers the network model.

6.3 Complexity Analysis

Notice that in the DDF model, each delay increases the size of $r(t)$. Specifically: each process delay add n states; each input delay adds n states; and each output delay adds r states. The resulting size of the infinite-dimensional part of the state is then $(2n+r)N$. Assuming that optimal control and estimation problems are tractable when the number of infinite-dimensional states is less than 50, we may infer something about the relative merits of the DDF vs. DDE representations for control purposes. First, we note that if we had used the naive conversion in Section 3, this dimension would be much larger - $(m+p+r)(3N)$ where recall we assume $m, p > n$. This type of representation would then reduce the number of controllable UAVs by at least 1/3 and probably much more. Second, if $n = r = 1$, then it is possible to control 17 UAVs. However, if we had used the naive representation or the DDE formulation (and assuming only a single shared disturbance and input), we would only be able to control at most 5 or 6 UAVs. This number would be further reduced if each UAV has its own input and disturbance (a likely scenario).

7. CONTROL OF A LARGE NETWORK

7.1 The Temperature Network Model

To illustrate the computational advantages of the DDF and PIE frameworks, we use the scalable network model in Peet (2020a). This is a problem in hotel management with a centralized hot-water source and multiple showering customers. Specifically, consider a user attempting to achieve a desired shower temperature by adjusting a hot-water tap. The model assumes user i will adjust their tap position $(T_{1i}(t))$ at a rate proportional to the difference between current temperature $(T_{2i}(t))$ and desired temperature $(w_i(t))$ and with constant of proportionality α_i. When multiple users are present and available hot water pressure is finite, the actions of each user will affect the temperature of all other users. This is modeled using γ_{ij}, which represents the fractional reduction of user i's hot water pressure caused by an increase in hot water consumption by user j. There is also a large transport delay caused by flow of hot water from the source to the showerhead of user i, τ_i. Next, we add a centralized tracking control system which senses both tap position and water temperature. However, this controller can not sense the desired water temperatures, $w_i(t)$ - which is thus modeled as a disturbance. The regulated output is sum of the tap actions of all users: T_{1i} and the sum of the centralized interventions, u_i. Obviously, we do not wish to regulate actual water temperature, T_{2i} as this would result in cold showers.

$$\begin{aligned} \dot{T}_{1i}(t) &= T_{2i}(t) - w_i(t) \qquad (9) \\ \dot{T}_{2i}(t) &= -\alpha_i \left(T_{2i}(t-\tau_i) - w_i(t)\right) \\ &\quad + \sum\nolimits_{j \neq i} \gamma_{ij}\alpha_j \left(T_{2j}(t-\tau_j) - w_j(t)\right) + u_i(t) \\ z(t) &= \left[\sum\nolimits_{i=1}^{N} T_{1i}(t) \qquad .1\sum\nolimits_{i=1}^{N} u_i(t)\right]^T. \end{aligned}$$

For a scalable instance of this problem with N users, we may choose $\alpha_i = 1$, $\gamma_{ij} = 1/N$, $\tau_i = i$, and $w_i(t) = N$. We find that for these values, the optimal closed-loop gain from disturbance to regulated output remains in the range of $.35 - .4$, irrespective of the number of users.

DDE Formulation of the problem Aggregating these dynamics as in Eqn. (2), we have the state vector

$$x(t) = [T_{11}(t) \; \cdots \; T_{1N}(t) \; T_{21}(t) \; \cdots \; T_{2N}(t)]^T$$

and the defining matrices as follows.

$$\begin{aligned} A_0 &= \begin{bmatrix} 0_{N\times N} & I \\ 0_{N\times N} & 0_{N\times N} \end{bmatrix}, \quad A_i = \begin{bmatrix} 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \hat{A}_i \end{bmatrix} \\ \hat{A}_i &= \Gamma * \operatorname{diag}(e_i) = \Gamma * \operatorname{diag}([0_{1\times i-1} \; 1 \; 0_{1\times N-i}]) \\ B_1 &= \begin{bmatrix} -I_N \\ -\Gamma \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0_{N\times N} \\ I_N \end{bmatrix} \\ [\Gamma]_{ij} &= \begin{cases} \gamma_{ij}\alpha_j & i \neq j \\ -\alpha_i & i = j \end{cases} \qquad i, j = 1, \cdots, N \\ C_1 &= \begin{bmatrix} \mathbf{1}_N^T & 0_{1\times N} \\ 0_{1\times N} & 0_{1\times N} \end{bmatrix}, \; D_{11} = [0_{2\times N}], \; D_{12} = \begin{bmatrix} 0_{1\times N} \\ .1\mathbf{1}_N^T \end{bmatrix} \end{aligned}$$

where $\mathbf{1}_N$ is the length-N vector of all ones. In this formulation, we have $2N$ states, N disturbances, N inputs, 2 regulated outputs and N delays. In Peet (2020a), this meant we were limited to approximately 4 users - implying $2N * N = 32$ infinite-dimensional channels. As will be seen, in the DDF formulation, these algorithms are able to handle optimal control with 40 users.

DDF Formulation of the problem In the DDF formulation, $x(t)$ is unchanged. However, we now define the delayed channels as

$$r_i(t) = [0_{1\times N+i-1} \; 1 \; 0_{1\times N-i}]\, x(t) = T_{2i}(t).$$

This is done by defining C_{ri}, B_{r1i}, B_{r2i} and D_{rvi} as

$$\begin{aligned} C_{ri} &= [0_{1\times N+i-1} \; 1 \; 0_{1\times N-i}] \\ B_{r1i} &= 0_{1\times N} \qquad B_{r2i} = 0_{1\times N} \qquad D_{rvi} = 0_{1\times N}. \end{aligned}$$

We would like the output of the delayed channels to be the delayed states as

$$v(t) = [T_{21}(t-\tau_1) \; \cdots \; T_{2N}(t-\tau_N)]^T.$$

This is accomplished by defining

$$C_{vi} = e_i = [0_{1\times i-1} \; 1 \; 0_{1\times N-i}]^T, \qquad C_{vdi} = 0_{2\times N}.$$

Finally, we retain $A_0, B_1, B_2, C_1, C_2, D_{11}, D_{12}$ from the DDE formulation, and use B_v and D_{1v} to model how the delayed terms affect the state dynamics and output signal.

$$B_v = \begin{bmatrix} 0_{N\times N} \\ \Gamma \end{bmatrix}, \qquad D_{1v} = 0$$

7.2 Implementation and Numerical Results

For this analysis, we used the PIETOOLS_DDF toolbox, which may be executed directly on CodeOcean at Peet (2020c). This toolbox converts a DDF representation to a PIE representation and may also be used to convert a DDE to a DDF, if desired. Once in PIE formulation, H_∞-optimal controller synthesis is performed using the

# of users	1	3	5	10	20	30	40
IPM CPU sec	.48	.638	2.42	94.7	5455	35620	157200

Table 1. CPU sec indexed by # of users (N)

PIETOOLS toolbox as described in Shivakumar et al. (2020a). The Linear Operator Inequality (LOI) for H_∞-optimal controller synthesis for PIEs is as given in Shivakumar et al. (2020b). This LOI is defined as follows.

Theorem 6. (Shivakumar et al. (2020b)). Suppose $\mathcal{B}_{T1} = \mathcal{B}_{T2} = 0$ and there exist bounded linear operators $\mathcal{P} : L_2^n[a,b] \to L_2^n[a,b]$ and $\mathcal{Z} : L_2^n[a,b] \to \mathbb{R}$, such that $\mathcal{P}$ is coercive and

$$\begin{bmatrix} -\gamma I & \mathcal{D}_{11}^* & (\mathcal{P}\mathcal{C}_1^* + \mathcal{Z}^*\mathcal{D}_{12}^*) \\ (\cdot)^* & -\gamma I & \mathcal{B}_1 \\ (\cdot)^* & (\cdot)^* & (\cdot)^* + \mathcal{T}(\mathcal{A}\mathcal{P} + \mathcal{B}_2\mathcal{Z})^* \end{bmatrix} \prec 0. \qquad (10)$$

Then $\mathcal{P}^{-1}$ is a bounded and coercive linear operator on L_2 and if $u(t) = \mathcal{Z}\mathcal{P}^{-1}\mathbf{x}(t)$, for any $w \in L_2$, any solution of (6) satisfies $\|z\|_{L_2} \le \gamma \|w\|_{L_2}$.

All numerical tests were performed on a desktop computer with 128GB RAM and a 3 GHz intel processor. CPU seconds indicates time for the interior-point calculations determined by the SDP solver Sedumi. In this run, the PIETOOLS extreme performance option was used to decrease computation times and reduce memory usage. The computation times, indexed by number of users, are listed in Table 1. Numerically, we observe that the controller synthesis problem is tractable up to 40 users. Recall that for 40 users, we have 80 states, 40 inputs, 40 disturbances and 40 delays. In Peet (2020a), for the same problem with the DDE framework, control was memory limited to 4 users.

8. CONCLUSION

This paper summarizes four possible representations for systems with delay: the Delay-Differential Equation (DDE) form; the Differential-Difference (DDF) form; the ODE-PDE form; and the Partial-Integral Equation (PIE) form. Formulae are given for conversion between these representations. We have shown that some networks cannot be modeled in the DDE formulation and, using an example of a network of UAVs, that careful choice of representation can significantly reduce the complexity of the underlying analysis and control problems. Using a scalable network model of temperature control, we have shown that formulation in the DDF/PIE framework allows for optimal control of up to 40 users with an aggregate 80 states, 40 inputs, 40 disturbances and 40 delays on a desktop computer with 128GB RAM, while formulation in the DDE framework (or inefficient conversion to the DDF framework) only allows for control of 4 users.

REFERENCES

Appell, J., Kalitvin, A., and Zabrejko, P. (2000). *Partial Integral Operators and Integro-Differential Equations: Pure and Applied Mathematics*. CRC Press.

Bensoussan, A., Da Prato, G., Delfour, M., and Mitter, S. (1993). *Representation and control of infinite dimensional systems*, volume 1. Birkhäuser Boston.

Curtain, R. and Zwart, H. (1995). *An introduction to infinite-dimensional linear systems theory*, volume 138. Springer-Verlag.

Gu, K. (2010). Stability problem of systems with multiple delay channels. *Automatica*, 46(4), 743–751.

Gu, K., Kharitonov, V.L., and Chen, J. (2003). *Stability of Time-Delay Systems*. Birkhauser.

Hale, J. (1971). Functional differential equations. In *Analytic theory of differential equations*, 9–22. Springer.

Krstic, M. and Smyshlyaev, A. (2008). Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays. *Systems & Control Letters*, 57(9), 750–758.

Niculescu, S.I. (2001). *Delay Effects on Stability: A Robust Control Approach*, volume 269 of *Lecture Notes in Control and Information Science*. Springer-Verlag.

Peet, M. (2020a). A convex solution of the H_∞-optimal controller synthesis problem for multi-delay systems. *SIAM Journal on Control and Optimization*, 58(3), 1547–1578.

Peet, M. (2020b). Representation of networks and systems with delay: DDEs, DDFs, ODE-PDEs and PIEs. Technical report, arXiv.org. https://arxiv.org/abs/1910.03881.

Peet., M. (2021). A partial integral equation representation of coupled linear PDEs and scalable stability analysis using LMIs. *Automatica*, 125.

Peet, M.M. (2020c). PIETOOLS_DDF. https://www.codeocean.com/capsule/2383614. doi:10.24433/CO.2383614.v2.

Pepe, P., Jiang, Z.P., and Fridman, E. (2008). A new Lyapunov–Krasovskii methodology for coupled delay differential and difference equations. *International journal of control*, 81(1), 107–115.

Richard, J.P. (2003). Time-delay systems: An overview of some recent advances and open problems. *Automatica*, 39, 1667–1694.

Shivakumar, S., Das, A., and Peet, M. (2020a). PIETOOLS: a Matlab toolbox for manipulation and optimization of partial integral operators. In *Proceedings of the American Control Conference*, 2667–2672.

Shivakumar, S., Das, A., Weiland, S., and Peet, M. (2020b). Duality and H_∞-optimal control of ODE-PDE systems. In *Proceedings of the IEEE Conference on Decision and Control*, 5689–5696.

Shivakumar, S., Jagt, D., Das, A., and Peet, M. (2021). *PIETOOLS 2021b: User Manual.* arXiv: 2101.02050.

Wu, S., Hua, C.C., and Peet, M. (2019). Estimator-based output-feedback stabilization of linear multi-delay systems using SOS. In *Proceedings of the IEEE Conference on Decision and Control*, 983–988.

Zhang, S. and Chen, M.P. (1998). A new Razumikhin theorem for delay difference equations. *Computers & Mathematics with Applications*, 36(10-12), 405–412.

Zhu, Y., Su, H., and Krstic, M. (2015). Adaptive backstepping control of uncertain linear systems under unknown actuator delay. *Automatica*, 54, 256–265.