Representation of Networks and Systems with Delay: DDEs, DDFs, ODE-PDEs and PIEs

Matthew M. Peet [a],

[a] School for the Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85298 USA.

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. Furthermore, estimation and control of systems in DDE format has proven challenging, despite decades of study. In this paper, we examine alternative representations for 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 examine the coupled ODE-PDE formulation, for which backstepping methods have recently become available. Finally, we consider the algebraic Partial-Integral Equation (PIE) representation, which allows the optimal estimation and control problems to be solved efficiently through the use of recent software packages such as PIETOOLS. In each case, we consider a very general class of delay systems, specifically accounting for all four possible sources of delay - state delay, input delay, output delay, and process delay.

Key words: 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 various ways in which delays can complicate an otherwise straightforward control problem, consider the challenge of controlling a swarm of N UAVs over a wireless network. In this case, UAV i has a state, $x_i \in \mathbb{R}^{n_i}$ which represents displacement from a desired state (the concatenation of all such states is denoted x). There are local sensors on each UAV i, data (y_i) from which are transmitted to a centralized control authority. There is also a input, u, a regulated output, z, and a vector of disturbances, w - including both process and sensor noise. We can model this system as follows.

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

Here

- a_i is the internal dynamics of the UAV i
- a_{ij} is the effect of UAV j on the state of UAV i.
- b_{1i} is the effect of the noise on the motion of UAV i
- b_{2i} is the effect of the central command on UAV i
- c_{2i} is the measurement of the state of UAV i
- d_{21i} is the effect of the noise on the sensor on UAV i
- C_1 gives the weight on states of the fleet of UAVs to minimize in the optimal control problem
- D_{12} gives the weight on actuator commands to minimize in the optimal control problem
- $\hat{\tau}_{ij}$ is the time taken for changes in state of UAV j to affect UAV i
- h_i is the time taken for a command from the central authority to reach UAV i
- $\bar{\tau}_i$ is the time it takes the process disturbance (wind,

Email address: `mpeet@asu.edu` (Matthew M. Peet).

[1] This work was supported by the National Science Foundation under grant No. 1739990.

tracking signal, et c.) to reach UAV i

- $\tilde{\tau}_i$ is the time taken for measurements collected at UAV i to reach the central authority

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 ($\tilde{\tau}_i$).

The delayed network presented here can be modeled as a DDE - a model formulated in Subsection 6.1 as a special case of the more general class of systems described by Eqns. (2)-(4) as presented in Section 2. If we were to consider control of such a network, however, we find that while there are algorithms for control of DDEs [1], these algorithms are computationally complex and are not easily extended to systems with multiple input and output delays (at least when the sensors are corrupted by noise). Furthermore, DDE models cannot be used to represent some important system designs - including a model of feedback described in Subsection 6.4. Finally, considering the problem of dimensionality, we note 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 the low-dimensional nature of the delayed channels is lost.

For all these reasons, we first consider the use of Differential Difference Equations (DDFs) in Section 3 and represent the network in the DDF framework in Subsection 6.2. The DDF formulation allows for the representation of delayed information in heterogeneous low-dimensional channels. Specifically, the infinite-dimensional component of state-space [2,3] 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 [4] 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 6.4 DDFs will allow us to represent the difference equations which arise in some network models.

From the DDF model, in Section 4 we turn to the class of coupled ODE-PDE models. While DDFs are useful for the purposes of representation and simulation, there are very few results on analysis, much less optimal control and estimation which have been developed for such models. By contrast, ODE-PDE models, while equivalent to the DDF model in terms of generality, have a well-understood physical interpretation and are embedded in a field of study for which simulation, analysis and control has been more thoroughly studied. Specifically, backstepping methods have been developed for ODE-PDE models of delay [5,6] and the formulae we present in this section for conversion to the ODE-PDE framework may prove useful if the reader is interested in application or further development of these backstepping methods.

Finally, we consider the Partial Integral Equation (PIE) representation. PIE models are occasionally referred to as integro-differential equations of Barbashin type [7] and have been used since the 1950s to model systems in biology, physics, and continuum mechanics (See chapters 19-20 of [7] for a survey). PIE representations have the form

$$\begin{aligned}\mathcal{T}\dot{\mathbf{x}}(t)+\mathcal{B}_{d1}\dot{w}(t)+\mathcal{B}_{d2}\dot{u}(t)&=\mathcal{A}\mathbf{x}(t)+\mathcal{B}_1w(t)+\mathcal{B}_2u(t)\\ z(t)&=\mathcal{C}_1\mathbf{x}(t)+\mathcal{D}_{11}w(t)+\mathcal{D}_{12}u(t),\\ y(t)&=\mathcal{C}_2\mathbf{x}(t)+\mathcal{D}_{21}w(t)+\mathcal{D}_{22}u(t)\end{aligned}$$

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

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

where

$$\begin{aligned}&\left(\mathcal{P}_{\{R_i\}}\mathbf{\Phi}\right)(s):=\\&R_0(s)\mathbf{\Phi}(s)+\int_{-1}^sR_1(s,\theta)\mathbf{\Phi}(\theta)d\theta+\int_s^0R_2(s,\theta)\mathbf{\Phi}(\theta)d\theta.\end{aligned}$$

PIE representations have the advantage that they are defined by PI operators. Unlike Dirac and differential operators, PI operators are bounded and form an algebra. Furthermore PIE models do not require boundary conditions or continuity constraints [8–10] - simplifying analysis and control problems. Indeed, it has been shown in several recent papers [8–11] that many problems in analysis and optimal estimation and control of coupled ODE-PDE models can be formulated as optimization over the cone of positive PI operators. The further development of efficient software tools for manipulation and optimization of PI operators (See PIETOOLS [12]) means that representation of delay systems using PIEs may allow for the development of new algorithms for control of network models such as in Eqn. (1). Indeed, current versions of PIETOOLS [12] are sufficiently optimized as to allow for analysis and control of systems of up to 50 coupled PDE states on a desktop computer with 64GB RAM.

Finally, we emphasize that this paper does not advocate for any particular time-domain representation (we do not consider here the significant literature on analysis and control in the frequency domain), be it the DDE, DDF, ODE-PDE, or PIE formulation, and does not propose any new algorithms for analysis and control of delay systems per se. Rather, the purpose of this document is to serve as a guide to representation of delay systems in each framework. Specifically, for each representation, we: clearly state the most general form of each representation - allowing for delays in input, output, process and

$$
\begin{aligned}
\dot{x}(t) =& A_0x(t) + \sum_{i=1}^{K} A_i x(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} A_{di}(s)x(t+s)ds + B_1 w(t) + \sum_{i=1}^{N} B_{1i}w(t-\tau_i) \\
&+ \sum_{i=1}^{K}\int_{-\tau_i}^{0} B_{1di}(s)w(t+s)ds + B_2u(t) + \sum_{j=1}^{N} B_{2i}u(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} B_{2di}(s)u(t+s)ds
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
z(t) =& C_{10}x(t) + \sum_{i=1}^{K} C_{1i} x(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{1di}(s)x(t+s)ds + D_{11} w(t) + \sum_{i=1}^{N} D_{11i}w(t-\tau_i) \\
&+ \sum_{i=1}^{K}\int_{-\tau_i}^{0} D_{11di}(s)w(t+s)ds + D_{12}u(t) + \sum_{i=1}^{N} D_{12i}u(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} D_{12di}(s)u(t+s)ds
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
y(t) =& C_{20}x(t) + \sum_{i=1}^{K} C_{2i} x(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{1di}(s)x(t+s)ds + D_{21} w(t) + \sum_{i=1}^{N} D_{21i}w(t-\tau_i) \\
&+ \sum_{i=1}^{K}\int_{-\tau_i}^{0} D_{21di}(s)w(t+s)ds + D_{22}u(t) + \sum_{i=1}^{N} D_{22i}u(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} D_{22di}(s)u(t+s)ds
\end{aligned} \tag{4}
$$

state; provide formulae for conversion between representations; and briefly list advantages and limitations of the representation as applied to network models of the form of Eqn. (1). While subsets of the DDF and ODE-PDE representations of delay systems can be found in the literature [4,13–17], previous models only consider a subset of the possible sources of delay, do not consider the PIE representation, do not discuss the relative advantages of these models as applied to networks, and do not provide formulae for conversion between representation. This guide, then, may be used as a convenient source of information for researchers interested in either selection of a representation or conversion of a representation to an alternative format.

2 The Delay-Differential Equation (DDE) Representation

We begin by defining the signals using the DDE representation:

- 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 $\{\tau_i\}_{i=1}^{K}$. We now propose a set governing equations as defined in Eqns (2)-(4). The dimensions of all matrices in this representation can be inferred from the dimension of the respective state and signals. In Subsection 6.1, the UAV network is formulated in this DDE representation.

2.1 Advantages of the DDE Fromulation

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 applied to 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 Differential-Difference (DDF) Representation

A generalization of the DDE representation is the Differential-Difference (DDF) formulation. Simplified versions of this formulation were previously considered in, e.g. [2,3]. 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 signal r_i may be considered the infinite-dimensional part 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 to the r_i.

The governing equations may now be represented in the more compact form of Eqns. (5).

$$\begin{aligned}
\dot{x}(t) &= A_0x(t) + B_1w(t) + B_2u(t) + B_vv(t) \qquad (5)\\
z(t) &= C_1x(t) + D_{11}w(t) + D_{12}u(t) + D_{1v}v(t)\\
y(t) &= C_2x(t) + D_{21}w(t) + D_{22}u(t) + D_{2v}v(t)\\
r_i(t) &= C_{ri}x(t) + B_{r1i}w(t) + B_{r2i}u(t) + D_{rvi}v(t)\\
v(t) &= \sum_{i=1}^{K} C_{vi}r_i(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{vdi}(s)r_i(t+s)ds
\end{aligned}$$

Although Eqns. (5) are more compact, they are significantly more general than the DDEs in (2)-(4). Specifically, if we define the conversion formula

$$D_{rvi} = 0, \qquad B_v = \begin{bmatrix} I & 0 & 0 \end{bmatrix} \qquad (6)$$

$$C_{ri} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}, \qquad B_{r1i} = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}, \qquad B_{r2i} = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \qquad (7)$$

$$C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix} \qquad (8)$$

$$C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix} \qquad (9)$$

$$D_{1v} = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \qquad D_{2v} = \begin{bmatrix} 0 & 0 & I \end{bmatrix}, \qquad (10)$$

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 Eqns.* (2)-(4). *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.* (6)-(10), *then u, w, x, y, and z also satisfy Eqns.* (5) *with*

$$r_i(t) = \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}.$$

Corollary 2 *Suppose u, w, x, y, r_i and z satisfy Eqns.* (5) *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.* (6)-(10). *Then u, w, x, y, and z also satisfy Eqns.* (2)-(4).

Note that 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_{rv} and C_{vi}, then we have the following set of difference equations

$$r_i(t) = \sum_{j=1}^{K} D_{rv}C_{vj}r_j(t-\tau_j).$$

A more realistic example of difference equations in networks is given in Subsection 6.4, where we provide a model of network control which can be represented in the DDF framework, but not the DDE framework.

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

$$C_{vi} = \tilde{A}_i \qquad \text{and} \qquad C_{ri} = A_i.$$

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.

A disadvantage of the DDF formulation is that fewer computational and analysis tools are available for systems in this representation. This is partially due to the fact that the class of systems is larger than the DDE and thus the tools must be more general. However, we do note that versions of both the Lyapunov-Krasovskii [2] and Lyapunov-Razumikhin [18] stability tests have been formulated in the DDF framework.

4 The Coupled ODE-PDE Representation

Before proceeding to the PIE representation, we consider the coupled ODE-PDE representation. Although widely recognized as a physical interpretation of delay systems [17], ODE-PDE models have not typically been used in analysis and control of systems with delay. Recently, however, backstepping methods originally developed for control of PDE models have been extended to systems with delay - See [5,6]. For this reason, we present here a general form of ODE-PDE model and define the process of conversion from DDFs.

Note that the ODE-PDE representations of delay systems as presented here are equivalent to the class of DDFs. Since we have shown that DDEs are a special case of DDFs, we present the ODE-PDE form as the logical alternative to the DDF and do not bother to convert directly between DDE and ODE-PDE. 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.

$$\begin{aligned}
\dot{x}(t) &= A_0x(t) + B_1w(t) + B_2u(t) + B_vv(t) \qquad (11)\\
z(t) &= C_1x(t) + D_{11}w(t) + D_{12}u(t) + D_{1v}v(t)\\
y(t) &= C_2x(t) + D_{21}w(t) + D_{22}u(t) + D_{2v}v(t)\\
\dot{\phi}_i(t,s) &= \frac{1}{\tau_i}\phi_{i,s}(t,s)\\
\phi_i(t,0) &= C_{ri}x(t) + B_{r1i}w(t) + B_{r2i}u(t) + D_{rvi}v(t)\\
v(t) &= \sum_{i=1}^{K} C_{vi}\phi_i(t,-1) + \sum_{i=1}^{K}\int_{-1}^{0}\tau_iC_{vdi}(\tau_is)\phi_i(t,s)ds
\end{aligned}$$

In Eqns. (11), 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. (11) and Eqns. (5) are equivalent, as in the following lemma.

Lemma 3 *Suppose u, w, x, r_i, v, y, and z satisfy Eqns.* (5)*. Then u, w, x, v, y, and z also satisfy Eqns.* (11) *with*

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

Similarly, if u, w, x, v, y, ϕ_i and z satisfy Eqns. (11)*, then u, w, x, v, y, and z satisfy Eqns.* (5) *with $r_i(t) = \phi_i(t,0)$.*

4.1 Advantages of the ODE-PDE Representation

The ODE-PDE representation may help us understand the dimension of the state. As a generalization of the differential-difference formulation, if the dimension of r_i is $r(t) \in \mathbb{R}^{p_i}$, then the dimension of the PDE state is $\phi(t,s) \in \mathbb{R}^{\sum_i p_i}$.

In addition, by scaling the pipes and ignoring the distributed delay, the ODE-PDE representation isolates the effect of the delay parameters to a single term - $\dot{\phi}_i(t,s) = \frac{1}{\tau_i}\phi_{i,s}(t,s)$. This feature makes it easier to understand the effects of uncertainty and time-variation in the delay parameter.

Finally, we mention that the ODE-PDE form is the native representation used for recently developed backstepping methods for systems with delay, such as proposed in [5,6] and use of the conversion formulae provided may allow these methods to be applied to solve a larger class of systems - including difference equations.

5 The Partial Integral Equation (PIE) Representation

Recall that a Partial Integral Equation (PIE) has the form

$$\begin{aligned}
\mathcal{T}\dot{\mathbf{x}}(t) + \mathcal{B}_{T_1}\dot{w}(t) + \mathcal{B}_{T_2}\dot{u}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1w(t) + \mathcal{B}_2u(t)\\
z(t) &= \mathcal{C}_1\mathbf{x}(t) + \mathcal{D}_{11}w(t) + \mathcal{D}_{12}u(t),\\
y(t) &= \mathcal{C}_2\mathbf{x}(t) + \mathcal{D}_{21}w(t) + \mathcal{D}_{22}u(t),
\end{aligned} \qquad (12)$$

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. (12) also define solutions to Eqns. (5) and Eqns. (11). Specifically, let us define

$$\mathcal{A} := \mathcal{P}\begin{bmatrix} \mathbf{A}_0, & \mathbf{A}\\ 0, & \{I_\tau, 0, 0\}\end{bmatrix}, \qquad \mathcal{T} := \mathcal{P}\begin{bmatrix} I, & 0\\ \mathbf{T}_0, & \{0, \mathbf{T}_a, \mathbf{T}_b\}\end{bmatrix} \qquad (13)$$

$$\mathcal{B}_1 := \mathcal{P}\begin{bmatrix} \mathbf{B}_1, & \emptyset\\ 0, & \{\emptyset\}\end{bmatrix}, \qquad \mathcal{B}_2 := \mathcal{P}\begin{bmatrix} \mathbf{B}_2, & \emptyset\\ 0, & \{\emptyset\}\end{bmatrix}, \qquad (14)$$

$$\mathcal{B}_{T_1} := \mathcal{P}\begin{bmatrix} 0, & \emptyset\\ \mathbf{T}_1, & \{\emptyset\}\end{bmatrix}, \qquad \mathcal{B}_{T_2} := \mathcal{P}\begin{bmatrix} 0, & \emptyset\\ \mathbf{T}_2, & \{\emptyset\}\end{bmatrix} \qquad (15)$$

$$\mathcal{C}_1 := \mathcal{P}\begin{bmatrix} \mathbf{C}_{10}, & \mathbf{C}_{11}\\ \emptyset, & \{\emptyset\}\end{bmatrix}, \quad \mathcal{C}_2 := \mathcal{P}\begin{bmatrix} \mathbf{C}_{20}, & \mathbf{C}_{21}\\ \emptyset, & \{\emptyset\}\end{bmatrix}, \qquad (16)$$

$$\mathcal{D}_{ij} := \mathcal{P}\begin{bmatrix} \mathbf{D}_{ij}, & \emptyset\\ \emptyset, & \{\emptyset\}\end{bmatrix} \qquad (17)$$

where the matrices are as defined in Eqns. (18)-(26).

Lemma 4 *Suppose u, w, x, ϕ_i, v, y, and z satisfy Eqns.* (11)*. Then u, w, y, and z also satisfy Eqns.* (12)

$$\mathbf{T}_0 = \begin{bmatrix} C_{r1} + D_{rv1}C_{vx} \\ \vdots \\ C_{rK} + D_{rvK}C_{vx} \end{bmatrix}, \qquad \mathbf{T}_1 = \begin{bmatrix} B_{r11} + D_{rv1}D_{vw} \\ \vdots \\ B_{r1K} + D_{rvK}D_{vw} \end{bmatrix}, \qquad \mathbf{T}_2 = \begin{bmatrix} B_{r21} + D_{rv1}D_{vu} \\ \vdots \\ B_{r2K} + D_{rvK}D_{vu} \end{bmatrix} \tag{18}$$

$$\mathbf{T}_a(s) = \begin{bmatrix} D_{rv1}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s) \end{bmatrix} \\ \vdots \\ D_{rvK}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s) \end{bmatrix} \end{bmatrix}, \qquad \mathbf{T}_b = -I + \mathbf{T}_a(s), \quad I_\tau = \begin{bmatrix} \frac{1}{\tau_1}I & & \\ & \ddots & \\ & & \frac{1}{\tau_K}I \end{bmatrix}, \tag{19}$$

$$\mathbf{A}_0 = A_0 + B_vC_{vx}, \quad \mathbf{A}(s) = B_v\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s) \end{bmatrix}, \quad \mathbf{B}_1 = B_1 + B_vD_{vw}, \quad \mathbf{B}_2 = B_2 + B_vD_{vu}, \tag{20}$$

$$\mathbf{C}_{10} = C_1 + D_{1v}C_{vx}, \qquad \mathbf{C}_{11} = D_{1v}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s) \end{bmatrix}, \tag{21}$$

$$\mathbf{C}_{20} = C_2 + D_{2v}C_{vx}, \qquad \mathbf{C}_{21} = D_{2v}\begin{bmatrix} C_{I1}(s) & \cdots & C_{IK}(s) \end{bmatrix}, \tag{22}$$

$$\mathbf{D}_{11} = (D_{11} + D_{1v}D_{vw}), \quad \mathbf{D}_{12} = (D_{12} + D_{1v}D_{vu}), \quad \mathbf{D}_{21} = (D_{21} + D_{2v}D_{vw}), \quad \mathbf{D}_{22} = (D_{22} + D_{2v}D_{vu}) \tag{23}$$

$$\hat{C}_{vi} = C_{vi} + \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s)ds, \qquad D_I = \left(I - \left(\sum_{i=1}^{K} \hat{C}_{vi}D_{rvi}\right)\right)^{-1} \tag{24}$$

$$C_{vx} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}C_{ri}\right), \qquad D_{vw} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}B_{r1i}\right), \qquad D_{vu} = D_I\left(\sum_{i=1}^{K} \hat{C}_{vi}B_{r2i}\right) \tag{25}$$

$$C_{Ii}(s) = -D_I\left(C_{vi} + \tau_i\int_{-1}^{s} C_{vdi}(\tau_i\eta)d\eta\right) \tag{26}$$

---

with $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ as defined in (13)-(17) *and*

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

Corollary 5 *Suppose u, w, y, $\mathbf{x}$ and z satisfy Eqns.* (12) *with $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}$ as defined in* (13)-(17)*. Then u, w, y, and z satisfy Eqns.* (11) *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. To obtain the PIE representation, we eliminate ϕ_i, $\phi_i(0)$, and $\phi_i(-1)$ from Eqns. (11). This is done using the relationship

$$\phi_i(t,s) = \phi_i(t,0) - \int_{s}^{0} \phi_{i,s}(t,\eta)d\eta. \tag{27}$$

Suppose u, w, x, ϕ_i, v, y, and z satisfy Eqns. (11). The main challenge is to solve for v in terms of x, w, u and $\phi_{i,s}$. Specifically, we seek to show that

$$\begin{aligned} v(t) =& C_{vx}x(t) + D_{vw}w(t) + D_{vu}u(t) \\ &+ \int_{-1}^{0}\sum_{i=1}^{K} C_{Ii}(s)\phi_{i,s}(t,s)ds. \end{aligned}$$

Recall from Eqns. (11) that

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

and

$$\phi_i(t,0) = C_{ri}x(t) + B_{r1i}w(t) + B_{r2i}u(t) + D_{rvi}v(t).$$

From Eqn. (27),

$$\phi_i(t,-1) = \phi_i(t,0) - \int_{-1}^{0} \phi_{i,s}(t,\eta)d\eta$$

and hence

$$
\begin{aligned}
v(t) &= \sum_{i=1}^{K} C_{vi} \left(\phi_i(t,0) - \int_{-1}^{0} \phi_{i,s}(t,\eta)d\eta \right) \\
&\quad + \sum_{i=1}^{K} \int_{-1}^{0} \tau_i C_{vdi}(\tau_i s) \left(\phi_i(t,0) - \int_{s}^{0} \phi_{i,s}(t,\eta)d\eta \right) ds \\
&= \left(\sum_{i=1}^{K} \hat{C}_{vi} C_{ri} \right) x(t) + \left(\sum_{i=1}^{K} \hat{C}_{vi} B_{r1i} \right) w(t) \\
&\quad + \left(\sum_{i=1}^{K} \hat{C}_{vi} B_{r2i} \right) u(t) + \left(\sum_{i=1}^{K} \hat{C}_{vi} D_{rvi} \right) v(t) \\
&\quad - \left(\int_{-1}^{0} \sum_{i=1}^{K} C_{vi} \phi_{i,s}(t,\eta) d\eta \right) \\
&\quad - \left(\sum_{i=1}^{K} \tau_i \int_{-1}^{0} \int_{s}^{0} C_{vdi}(\tau_i s) \phi_{i,s}(t,\eta) d\eta ds \right).
\end{aligned}
$$

Eliminating v from the RHS, we obtain

$$
\begin{aligned}
v(t) =& C_{vx}x(t) + D_{vw}w(t) + D_{vu}u(t) \\
&- D_I \left(\int_{-1}^{0} \sum_{i=1}^{K} C_{vi} \phi_{i,s}(t,\eta) d\eta \right) \\
&- D_I \left(\sum_{i=1}^{K} \tau_i \int_{-1}^{0} \int_{s}^{0} C_{vdi}(\tau_i s) \phi_{i,s}(t,\eta) d\eta ds \right).
\end{aligned}
$$

Using the identity

$$
\int_{-1}^{0} \int_{s}^{0} f(s,\eta) d\eta ds = \int_{-1}^{0} \int_{-1}^{s} f(\eta,s) d\eta ds,
$$

we obtain

$$
\begin{aligned}
v(t) &= C_{vx}x(t) + D_{vw}w(t) + D_{vu}u(t) \\
&\quad - D_I \left(\int_{-1}^{0} \sum_{i=1}^{K} C_{vi} \phi_{i,s}(t,s) ds \right) \\
&\quad - D_I \left(\sum_{i=1}^{K} \int_{-1}^{0} \left(\int_{-1}^{s} \tau_i C_{vdi}(\tau_i \eta) d\eta \right) \phi_{i,s}(t,s) ds \right) \\
&= C_{vx}x(t) + D_{vw}w(t) + D_{vu}u(t) \\
&\quad - D_I \left(\int_{-1}^{0} \sum_{i=1}^{K} \left(C_{vi} + \tau_i \int_{-1}^{s} C_{vdi}(\tau_i \eta) d\eta \right) \phi_{i,s}(t,s) ds \right) \\
&= C_{vx}x(t) + D_{vw}w(t) + D_{vu}u(t) \\
&\quad + \int_{-1}^{0} \sum_{i=1}^{K} C_{Ii}(s) \phi_{i,s}(t,s) ds.
\end{aligned}
$$

The rest of the proof is straightforward. Plugging this expression for $v(t)$ into Eqns. (11), we obtain

$$
\begin{aligned}
z(t) &= \mathbf{C}_{10}x(t) + \int_{-1}^{0} \mathbf{C}_{11}(s)\mathbf{\Phi}(t,\eta)d\eta + \mathbf{D}_{11}w(t) + \mathbf{D}_{12}u(t) \\
&= \mathcal{C}_1 \mathbf{x}(t) + \mathcal{D}_{11}w(t) + \mathcal{D}_{12}u(t) \\
y(t) &= \mathbf{C}_{20}x(t) + \int_{-1}^{0} \mathbf{C}_{21}(s)\mathbf{\Phi}(t,\eta)d\eta + \mathbf{D}_{21}w(t) + \mathbf{D}_{22}u(t) \\
&= \mathcal{C}_2 \mathbf{x}(t) + \mathcal{D}_{21}w(t) + \mathcal{D}_{22}u(t)
\end{aligned}
$$

where

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

Likewise,

$$
\begin{aligned}
&\begin{bmatrix} \dot{x}(t) \\ \dot{\phi}_1(t,s) \\ \vdots \\ \dot{\phi}_K(t,s) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_0 x(t) + \int_{-1}^{0} \mathbf{A}(\eta)\mathbf{\Phi}(t,\eta)d\eta \\ I_\tau \mathbf{\Phi}(t,s) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ 0 \end{bmatrix} w(t) + \begin{bmatrix} \mathbf{B}_2 \\ 0 \end{bmatrix} u(t) \\
&= \mathcal{A}\mathbf{x}(t) + \mathcal{B}_1 w(t) + \mathcal{B}_2 u(t).
\end{aligned}
$$

Finally, we observe that

$$
\begin{aligned}
\phi_i(t,s) &= C_{ri}x(t) + B_{r1i}w(t) + B_{r2i}u(t) + D_{rvi}v(t) \\
&\qquad - \int_{s}^{0} \phi_{i,s}(t,\eta)d\eta \\
&= (C_{ri} + D_{rvi}C_{vx})x(t) + (B_{r1i} + D_{rvi}D_{vw})w(t) \\
&\qquad + (B_{r2i} + D_{rvi}D_{vu})u(t) \\
&\quad - \int_{s}^{0} \phi_{i,s}(t,\eta)d\eta + \left(\int_{-1}^{0} \sum_{j=1}^{K} D_{rvi} C_{Ij}(s) \phi_{j,s}(t,s) ds \right).
\end{aligned}
$$

Hence

$$
\begin{aligned}
\begin{bmatrix} \phi_1(t,\cdot) \\ \vdots \\ \phi_K(t,\cdot) \end{bmatrix} &= \mathbf{T}_0 x(t) + \mathbf{T}_1 w(t) + \mathbf{T}_2 u(t) \\
&\quad + \int_{-1}^{s} \mathbf{T}_a(\eta)\mathbf{\Phi}(t,\eta)d\eta + \int_{s}^{0} \mathbf{T}_b(\eta)\mathbf{\Phi}(t,\eta)d\eta
\end{aligned}
$$

and

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

Finally, we differentiate and combine these expressions to obtain

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

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 use of 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 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 subalgebra of PI operators parameterized by polynomials, then the software package PIETOOLS [12] allows for: manipulation of PI operators as a class object; declaration of PI operator variables; enforcement of PI operators 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 [12] or any of the recent papers on analysis and control in the PIE framework [19,8,10,20,11,9]. 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. (12) 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 hance $\dot{w}$ 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.

5.2 *Direct Conversion Between DDE and PIE Representations*

In this subsection, we bypass the DDF and give a formula for direct conversion between DDE and PIE representations. This formula is given in Eqns. (28)-(37). We note, however, that most large network models have significant structure in the form of low-dimensional delay channels. This structure can be modeled in the DDF representation (but not in the DDE representation), as illustrated in Section 6 for our network model.

6 Modeling of a Network of UAVs

To illustrate some of the differences between the DDE, DDF, ODE-PDE 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 complexity and generality of ODE-PDE and PIE models is inherited from the DDF representation and the conversion to these representations 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 $x_i \in \mathbb{R}^n$ and the outputs are all dimension $y_i(t) \in \mathbb{R}^r$. In this case, we re-write the network model in Eqns. (1) as

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

6.1 *DDE Representation*

To put this network in the DDE representation, we use the model in Eqns. (2)-(4) where $K = 3N$ while C_{10} and

$$
I_\tau = \begin{bmatrix} \frac{1}{\tau_1} I & & \\ & \ddots & \\ & & \frac{1}{\tau_K} I \end{bmatrix}, \qquad \mathbf{T}_0 = \begin{bmatrix} I \\ \vdots \\ I \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{T}_1 = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}, \qquad \mathbf{T}_2 = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \tag{28}
$$

$$
\mathbf{A}_0 = A_0 + \sum_{i=1}^{K} A_i + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i A_{di}(\tau_i s), \qquad \mathbf{A}(s) = -\begin{bmatrix} X_{A1}(s) & \cdots & X_{AK}(s) \end{bmatrix} \tag{29}
$$

$$
\mathbf{C}_{11} = -\begin{bmatrix} X_{C11}(s) & \cdots & X_{C1K}(s) \end{bmatrix}, \qquad \mathbf{C}_{21} = -\begin{bmatrix} X_{C21}(s) & \cdots & X_{C2K}(s) \end{bmatrix}, \tag{30}
$$

$$
X_{Ai}(s) = \begin{bmatrix} A_i & B_{1i} & B_{2i} \end{bmatrix} + \tau_i \int_{-1}^{s} \begin{bmatrix} A_{di}(\tau_i \eta) & B_{1di}(\tau_i \eta) & B_{2di}(\tau_i \eta) \end{bmatrix} d\eta, \tag{31}
$$

$$
X_{C1i}(s) = \begin{bmatrix} C_{1i} & D_{11i} & D_{12i} \end{bmatrix} + \tau_i \int_{-1}^{s} \begin{bmatrix} C_{1di}(\tau_i \eta) & D_{11di}(\tau_i \eta) & D_{12di}(\tau_i \eta) \end{bmatrix} d\eta, \tag{32}
$$

$$
X_{C2i}(s) = \begin{bmatrix} C_{2i} & D_{21i} & D_{22i} \end{bmatrix} + \tau_i \int_{-1}^{s} \begin{bmatrix} C_{2di}(\tau_i \eta) & D_{21di}(\tau_i \eta) & D_{22di}(\tau_i \eta) \end{bmatrix} d\eta, \tag{33}
$$

$$
\mathbf{B}_1 = B_1 + \sum_{i=1}^{K} B_{1i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i B_{1di}(\tau_i s), \qquad \mathbf{B}_2 = B_2 + \sum_{i=1}^{K} B_{2i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i B_{2di}(\tau_i s), \tag{34}
$$

$$
\mathbf{C}_{10} = C_1 + \sum_{i=1}^{K} C_{1i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i C_{1di}(\tau_i s), \qquad \mathbf{C}_{20} = C_2 + \sum_{i=1}^{K} C_{2i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i C_{2di}(\tau_i s), \tag{35}
$$

$$
\mathbf{D}_{11} = D_{11} + \sum_{i=1}^{K} D_{11i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i D_{11di}(\tau_i s) ds, \qquad \mathbf{D}_{12} = D_{12} + \sum_{i=1}^{K} D_{12i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i D_{12di}(\tau_i s) ds \tag{36}
$$

$$
\mathbf{D}_{21} = D_{21} + \sum_{i=1}^{K} D_{21i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i D_{21di}(\tau_i s) ds, \qquad \mathbf{D}_{22} = D_{22} + \sum_{i=1}^{K} D_{22i} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i D_{22di}(\tau_i s) ds \tag{37}
$$

D_{12} are unchanged. First, define

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

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

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

All undefined matrices in Eqns. (2)-(4) are 0. Thus, for example, $B_{1,(N+1)} = 0$, $B_{2,N} = 0$, $C_{2,2N} = 0$ and

$$
B_{11} = \begin{bmatrix} b_{11} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, B_{12} = \begin{bmatrix} 0 \\ b_{12} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, B_{1N} = \begin{bmatrix} 0 \\ \vdots \\ b_{1N} \end{bmatrix}
$$

$$
B_{2,N+1} = \begin{bmatrix} b_{2,1} \\ \vdots \\ 0 \end{bmatrix}, B_{2,2N} = \begin{bmatrix} 0 \\ \vdots \\ b_{2,N} \end{bmatrix},
$$

$$
C_{2,2N+1} = \begin{bmatrix} c_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, D_{21,2N+1} = \begin{bmatrix} d_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ et c.}
$$

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 channel of size $3N(nN + m + p)$.

6.2 DDF Representation

To put the network model in the DDF representation, we retain the matrix A_0 from the DDE model in Subsection 6.1, set $C_1 = C_{10}$ and leave D_{12} unchanged. Note that we do not use the naive DDE-DDF conversion formulae in Eqns. (6)-(10) as these formulae do not leverage the low-dimensional nature of the delayed channels. Our first step is to define the vector $r(t)$ 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

$$\begin{aligned}
&B_{r1,i} = b_{1i},\\
&B_{r1,2N+i} = d_{21i},\\
&B_{r2,N+i} = b_{2i},\\
&C_{r,2N+i} = c_{2i}.
\end{aligned}$$

In this case, we presume the individual 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. Indeed, using our definitions, 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 finite-dimensional output $v(t)$ by defining C_{vi} blockwise for $i = 1, \cdots, 3N$ as

$$[C_{vi}]_j = \begin{cases} I & i = j \\ 0 & \text{otherwise}, \end{cases}$$

so that

$$C_{v1} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, C_{v2} = \begin{bmatrix} 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, C_{v,3N} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix}$$

which yields

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

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

$$\begin{aligned}
B_v &= \begin{bmatrix} I & \cdots & I & I & \cdots & I & 0 \end{bmatrix},\\
D_{2v} &= \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & I \end{bmatrix},
\end{aligned}$$

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 [1], 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 we suppose $n = r = 1$, 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 would be further reduced if each UAV has its own input and disturbance (a likely scenario).

To further illustrate the importance of converting efficiently, we note that if we had included the N^2 state delays from the original model, the dimension of r in the DDF formulation would be $(2n + r)N + nN^2$. This may seem large, but if again $n = r = 1$, we would still be able to control 6 or 7 UAVs (for $N = 7$, the dimension is 70). By contrast, if we had used the naive conversion between DDE and DDF, this dimension would be $(nN + m + p + r)(N^2 + 3N)$ - meaning we would only be able to control 2 UAVs using a single input and disturbance (for $N = 3$ the dimension is 108).

6.4 A Network Model which is a DDF, but not a DDE

In this subsection, we present a network model which can be represented using DDFs, ODE-PDEs, and PIEs, but not using DDEs. These models arise from the use of static feedback - i.e. $u(t) = Fy(t)$ where $y(t)$ is the concatenated vector of outputs from the UAVs. Note that y may include measurement of all states (the static state feedback problem). In this example, let us ignore output, process and state delay, but retain input delay and add a term which models the impact of actuator input $u(t)$ on the sensors as

$$y_i(t) = c_{2i}x_i(t) + d_{21i}w(t) + d_{22i}u(t - \tau_i).$$

Let A_0, C_1, D_{12}, B_{2i}, C_{vi} be as defined in Subsection 6.2 and define

$$B_1 = \begin{bmatrix} b_{11} \\ \vdots \\ b_{1N} \end{bmatrix}, \quad D_{21} = \begin{bmatrix} d_{21,1} \\ \vdots \\ d_{21,N} \end{bmatrix}$$

$$C_2 = \mathrm{diag}(c_{2,1}, \cdots, c_{2,N}), \quad [D_{22i}]_i = d_{22i}.$$

Aggregating the measurements, we have

$$y(t) = C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} u(t - \tau_i).$$

Now, substituting $u(t) = Fy(t)$ into the sensed output term, we obtain solutions of the form

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + \sum_i B_{2i} F y(t - \tau_i) \\
z(t) &= C_1 x(t) + D_{12} F y(t) \\
y(t) &= C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} F y(t - \tau_i).
\end{aligned} \tag{38}$$

Clearly, there is no DDE model with solutions which satisfy Eqns. (38) due to the recursion in the output. However, these solutions can be constructed using the DDF (and consequently the ODE-PDE and PIE frameworks). To construct such a model, we define the following terms.

$$\begin{aligned}
&\tilde{D}_{12} = D_{12} F D_{21}, \quad \tilde{D}_{22} = 0, \quad \tilde{C}_1 = C_1 + D_{12} F C_2 \\
&C_{ri} = F C_2, \qquad B_{r1i} = F D_{21}, \qquad [D_{rvi}]_i = F D_{22i} \\
&B_v = \begin{bmatrix} B_{21} & \cdots & B_{2N} \end{bmatrix}, \quad [C_{vi}]_j = \begin{cases} I & i = j \\ 0 & \text{otherwise,} \end{cases} \\
&D_{1v} = D_{12} F D_{2v}, \quad D_{2v} = \begin{bmatrix} D_{22,1} & \cdots & D_{22,N} \end{bmatrix}
\end{aligned} \tag{39}$$

Lemma 6 *suppose r_i, v, y, x, and z satisfy*

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + B_v v(t) \\
z(t) &= \tilde{C}_1 x(t) + \tilde{D}_{11} w(t) + \tilde{D}_{12} u(t) + D_{1v} v(t) \\
y(t) &= C_2 x(t) + D_{21} w(t) + D_{2v} v(t) \\
r_i(t) &= C_{ri} x(t) + B_{r1i} w(t) + D_{rvi} v(t) \\
v(t) &= \sum_{i=1}^{K} C_{vi} r_i(t - \tau_i)
\end{aligned}$$

with B_v $\tilde{D}_{12}$, $\tilde{C}_1$, D_{1v}, D_{2v}, C_{ri}, B_{r1i}, D_{rvi}, and C_{vi} as defined in Eqns. (39)*. Then x, z and y also satisfy Eqns.* (38)*.*

PROOF. Suppose r_i, v, y, x satisfy Eqns. (5). Then

$$\begin{aligned}
y(t) &= C_2 x(t) + D_{21} w(t) + D_{2v} v(t) \\
&= C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} v_i(t)
\end{aligned}$$

and hence

$$\begin{aligned}
r_i(t) &= C_{ri} x(t) + B_{r1i} w(t) + D_{rvi} v(t) \\
&= F C_2 x(t) + F D_{21} w(t) + \sum_{i=1}^{N} F D_{22i} v_i(t) \\
&= F \left(C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} v_i(t) \right) = F y(t).
\end{aligned}$$

Next,

$$v_i(t) = r_i(t - \tau_i) = F y(t - \tau_i)$$

and we conclude

$$y(t) = C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} F y(t - \tau_i).$$

Similarly

$$\begin{aligned}
\dot{x}(t) &= A_0 x(t) + B_1 w(t) + B_v v(t) \\
&= A_0 x(t) + B_1 w(t) + \sum_{i=1}^{N} B_{2i} v_i(t) \\
&= A_0 x(t) + B_1 w(t) + \sum_{i=1}^{N} B_{2i} F y(t - \tau_i)
\end{aligned}$$

and finally,

$$\begin{aligned}
&z(t) = \tilde{C}_1 x(t) + \tilde{D}_{12} w(t) + D_{1v} v(t) \\
&= C_1 x(t) + D_{12} F \left(C_2 x(t) + D_{21} w(t) + \sum_{i=1}^{N} D_{22i} v_i(t) \right) \\
&= C_1 x(t) + D_{12} F y(t)
\end{aligned}$$

as desired.

Note that if $u(t) \in \mathbb{R}^p$, the dimension of the infinite-dimensional state is $\mathbb{R}^{pN}$

7 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, although direct conversion between DDE and DDF is not advised if the delayed channels are low-dimensional. These formulae are meant to provide a convenient reference for researchers interested in exploring alternative representations. We have shown using an example of a network of UAVs that some networks cannot be modeled in the DDE formulation and that careful choice of representation can significantly reduce the complexity of the underlying analysis and control problems. Specifically, we have shown that formulation in the DDF/ODE-PDE/PIE framework allows for control of up to 17 UAVs on a desktop computer with 64GB RAM, while formulation in the DDE framework (or inefficient conversion to the DDF framework) only allows for control of 5 or 6 UAVs. Finally, the inclusion of the PIE representation in this work is intended to facilitate the exploitation of new and emerging algorithms for manipulation and optimization of PI operators using toolboxes such as PIETOOLS [11].

References

[1] M. Peet, "A convex solution of the H_∞-optimal controller synthesis problem for multi-delay systems," *SIAM Journal on Control and Optimization*, 2019, submitted.

[2] K. Gu, "Stability problem of systems with multiple delay channels," *Automatica*, vol. 46, no. 4, pp. 743–751, 2010.

[3] P. Pepe, Z.-P. Jiang, and E. Fridman, "A new Lyapunov–Krasovskii methodology for coupled delay differential and difference equations," *International journal of control*, vol. 81, no. 1, pp. 107–115, 2008.

[4] A. Bensoussan, G. Da Prato, M. Delfour, and S. Mitter, *Representation and control of infinite dimensional systems.* Birkhäuser Boston, 1993, vol. 1.

[5] M. Krstic and A. Smyshlyaev, "Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays," *Systems & Control Letters*, vol. 57, no. 9, pp. 750–758, 2008.

[6] Y. Zhu, H. Su, and M. Krstic, "Adaptive backstepping control of uncertain linear systems under unknown actuator delay," *Automatica*, vol. 54, pp. 256–265, 2015.

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

[8] S. Shivakumar, A. Das, S. Weiland, and M. Peet, "A generalized lmi formulation for input-output analysis of linear systems of odes coupled with pdes," in *Proceedings of the IEEE Conference on Decision and Control*, 2019.

[9] M. Peet., "A new mathematical framework for representation and analysis of coupled pdes," *IEEE Transactions on Automatic Control*, submitted.

[10] A. Das, S. Shivakumar, S. Weiland, and M. Peet, "h_∞ optimal estimation for linear coupled PDE systems," in *Proceedings of the IEEE Conference on Decision and Control*, 2019.

[11] S. Wu and M. Peet, "H_∞-optimal estimation in the PIE framework for systems with multiple delays and sensor noise," in *Proceedings of the American Control Conference*, 2020.

[12] S. Shivakumar, A. Das, and M. Peet, "Pietools: A matlab toolbox for manipulation and optimization of partial integral operators," in *Proceedings of the American Control Conference*, 2020.

[13] J. Hale, "Functional differential equations," in *Analytic theory of differential equations.* Springer, 1971, pp. 9–22.

[14] R. Curtain and H. Zwart, *An introduction to infinite-dimensional linear systems theory.* Springer-Verlag, 1995, vol. 138.

[15] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems.* Birkhauser, 2003.

[16] S.-I. Niculescu, *Delay Effects on Stability: A Robust Control Approach*, ser. Lecture Notes in Control and Information Science. Springer-Verlag, May 2001, vol. 269.

[17] J.-P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, pp. 1667–1694, 2003.

[18] S. Zhang and M.-P. Chen, "A new Razumikhin theorem for delay difference equations," *Computers & Mathematics with Applications*, vol. 36, no. 10-12, pp. 405–412, 1998.

[19] S. Shivakumar and M. Peet, "Computing input-ouput properties of coupled PDE systems," in *Proceedings of the American Control Conference*, 2019.

[20] S. Wu, C.-C. Hua, and M. Peet, "Estimator-based output-feedback stabilization of linear multi-delay systems using SOS," in *Proceedings of the IEEE Conference on Decision and Control*, 2019.