Automatica 127 (2021) 109508

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Representation of networks and systems with delay: DDEs, DDFs, ODE–PDEs and PIEs[☆]

Matthew M. Peet

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

ARTICLE INFO

Article history:
Received 9 October 2019
Received in revised form 20 August 2020
Accepted 18 December 2020
Available online 24 February 2021

Keywords:
Delay
PDEs
Networked control

ABSTRACT

Delay-Differential Equations (DDEs) are the most common representation for systems with delay. However, the DDE representation is limited. In network models with delay, the delayed channels are low-dimensional and accounting for this heterogeneity is not possible 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. We then apply these representations to 3 archetypical network models.

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 ways in which delays can complicate an otherwise straightforward control problem, consider control of a swarm of N UAVs over a wireless network. In this case, each UAV, i, has a state, $x_i(t) \in \mathbb{R}^{n_i}$ which may represent, e.g. displacement (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 a centralized vector of inputs, u, a regulated vector of outputs, z, and a vector of disturbances, w — including both process and sensor noise. We model this system as follows.

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

- 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 disturbance to 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 disturbance to 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, tracking signal, et c.) to reach UAV i

☆ This work was supported by the National Science Foundation under grants No. 1739990 and 1935453. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Nikolaos Bekiaris-Liberis under the direction of Editor Miroslav Krstic.

E-mail address: mpeet@asu.edu.

https://doi.org/10.1016/j.automatica.2021.109508

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

This UAV network is modeled as a DDE — a structure formulated in Eq. (2) in Section 2. If we consider control of such a network, however, we find that while there are algorithms for control of DDEs (See Peet, 2020a), 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 inefficiency of the algorithms, but rather by the failure to account for the low dimensional nature of the delayed channels. Specifically, we note that in the UAV model, 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 UAV network as a DDE (a formulation given in Section 6.1), then the low-dimensional nature of the delayed channels is lost. Furthermore, DDEs cannot represent some important system designs — including a model of feedback described in Section 7.

For these reasons, in Section 3, we consider the use of Differential Difference Equations (DDFs). The DDF can be used to model both DDEs and neutral-type systems, while also allowing for the assignment of delayed information to heterogeneous low-dimensional channels. Specifically, the infinite-dimensional component of state–space (as defined in Gu, 2010; Pepe, Jiang & Fridman, 2008) of the UAV network in the DDF framework is $\prod_i L_2[-\tau_i, 0]^{n_i}$ as opposed to $\prod_i L_2[-\tau_i, 0]^{\sum n_i}$ using a DDE. In addition, DDFs allow us to represent difference equations which arise in some network models — See Section 7.

From the DDF model we turn to coupled ODE–PDE models in Section 4. ODE–PDEs can be used to model a variety of systems. However, for the particular class of ODE–PDEs we use in Section 4, the solutions to the ODE–PDE are equivalent to those of the DDF (as defined in Section 3). Backstepping methods have been developed for ODE–PDE models of delay (e.g. Krstic & Smyshlyaev, 2008; Zhu, Su, & Krstic, 2015) and the formulae we present for conversion of DDFs to ODE–PDEs may prove useful if the reader is interested in application or further development of these backstepping methods.

Next, in Section 5, we consider Partial Integral Equations (PIEs) (Appell, Kalitvin, & Zabrejko, 2000). PIEs are a generalization of integro-differential equations of Barbashin type which have been used since the 1950s to model systems in biology, physics, and continuum mechanics (See chapters 19–20 of Appell et al., 2000 for a survey). PIEs and ODE–PDEs define an equivalent set of solutions and in this section, we provide formulae for conversion of DDEs and DDFs to PIEs. PIE models 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, PIE models do not require boundary conditions or continuity constraints — simplifying analysis and optimal control problems. Indeed, it has been recently shown in Das, Shivakumar, Weiland, and Peet (2019) and Shivakumar, Das, Weiland, and Peet (2019) that many problems in analysis, optimal estimation and control of ODE–PDE models can be formulated as optimization over the cone of positive PI operators. In Section 8, we show that the PIE formulation allows for H_∞-optimal control of a 40 user, 80-state, 40-delay, 40-input, 40-disturbance network model of temperature control.

Finally, we emphasize that this paper does not advocate for any particular time-domain representation (we do not consider the 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: state the most general form of the representation — allowing for delays in input, output, process and state; define a notion of solution; provide formulae for conversion between representations under which solutions are equivalent; and briefly list advantages and limitations of the representation as applied to network models of the form of Eq. (1). As discussed in the conclusions, these results can be used to establish notions of stability which are equivalent in all representations and to allow for conversion of optimal controllers and estimators between representations.

While subsets of the DDF and ODE–PDE representations of delay systems can be found in the literature (Bensoussan, Da Prato, Delfour, & Mitter, 1993; Gu, 2010; Gu, Kharitonov, & Chen, 2003; Mazenc, Ito, & Pepe, 2013; Niculescu, 2001; Pepe, Jiang & Fridman, 2008; Pepe, Karafyllis & Jiang, 2008), and some of these equivalences are known (Karafyllis & Krstic, 2014; Richard, 2003), previous works do not: consider all input–output signals and sources of delay; include PIEs; compare the relative advantages of the models as applied to networks; or provide formulae for conversion between representations. 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. For convenience and comparison, all representations are listed in Box I. All conversion formulae are listed in Boxes II and III. Finally, note that all proofs have been omitted, but are included in the extended version of this paper on Arxiv (Peet, 2020b).

Notation. I_n is the identity matrix in $\mathbb{R}^{n\times n}$, e_i is the ith canonical unit vector, $\mathbf{1}_n$ is the dimension n vector of all ones. $0_{n,m}$ is the zero matrix of dimension $\mathbb{R}^{n\times m}$, $0_n := 0_{n,n}$, and $W^{n,2}[X]$ is the nth-order Sobolev subspace of $L_2[X]$.

2. The DDE representation

We begin by defining the signals in the Delay-Differential Equation (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 $\{\tau_i\}_{i=1}^K$ with $0 < \tau_1 < \cdots < \tau_K$. For given $u \in L_2^p$, $w \in L_2^m$, and initial condition $x_0 \in W^{1,2}[-\tau_K, 0]^n$, we say that $x : [-\tau_K, \infty] \to \mathbb{R}^n$, $z : [0, \infty] \to \mathbb{R}^q$, and $y : [0, \infty] \to \mathbb{R}^r$ satisfy the DDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ if x is differentiable on $[0, \infty]$ (from the right at $t = 0$), $x(s) = x_0(s)$ for $s \in [-\tau_K, 0]$, and Eqs. (2) are satisfied for all $t \geq 0$. If any $B_{1i}, D_{11i}, D_{21i} \neq 0$, we require $w \in W^{1,2}[0, \infty]^m$ and $w(s) = 0$ for $s \leq 0$. If any $B_{2i}, D_{12i}, D_{22i} \neq 0$, we require $u \in W^{1,2}[0, \infty]^p$ and $u(s) = 0$ for $s \leq 0$.

Under the conditions stated above, existence of a classical continuously differentiable solution x is guaranteed as in, e.g. Thm. 3.3 of Chapter 3 in Kolmanovskii and Myshkis (1999) (See also Thm. 1.1 of Chapter 6 in Hale, 1971). Note that the dimensions

The Class of Delay-Differential Equations (DDEs):

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

The Class of Differential-Difference Equations (DDFs):

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\\ r_i(t)\end{bmatrix}=\begin{bmatrix}A_0 & B_1 & B_2\\ C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\\ C_{ri} & B_{r1i} & B_{r2i}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix}+\begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\\ D_{rvi}\end{bmatrix}v(t),$$
$$v(t)=\sum_{i=1}^{K}C_{vi}r_i(t-\tau_i)+\sum_{i=1}^{K}\int_{-\tau_i}^{0}C_{vdi}(s)r_i(t+s)ds. \tag{3}$$

The Class of Neutral-Type Systems (NDS):

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\end{bmatrix}=\begin{bmatrix}A_0 & B_1 & B_2\\ C_{10} & D_{11} & D_{12}\\ C_{20} & D_{21} & D_{22}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix}+\sum_{i=1}^{K}\begin{bmatrix}A_i & B_{1i} & B_{2i} & E_i\\ C_{1i} & D_{11i} & D_{12i} & E_{1i}\\ C_{2i} & D_{21i} & D_{22i} & E_{2i}\end{bmatrix}\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\\ \dot{x}(t-\tau_i)\end{bmatrix}$$
$$+\sum_{i=1}^{K}\int_{-\tau_i}^{0}\begin{bmatrix}A_{di}(s) & B_{1di}(s) & B_{2di}(s) & E_{di}(s)\\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) & E_{1di}(s)\\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) & E_{2di}(s)\end{bmatrix}\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\\ \dot{x}(t+s)\end{bmatrix}ds \tag{4}$$

The Class of ODE–PDE Systems:

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\\ y(t)\\ \phi_i(t,0)\end{bmatrix}=\begin{bmatrix}A_0 & B_1 & B_2\\ C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\\ C_{ri} & B_{r1i} & B_{r2i}\end{bmatrix}\begin{bmatrix}x(t)\\ w(t)\\ u(t)\end{bmatrix}+\begin{bmatrix}B_v\\ D_{1v}\\ D_{2v}\\ D_{rvi}\end{bmatrix}v(t)$$
$$\dot{\phi}_i(t,s)=\frac{1}{\tau_i}\phi_{i,s}(t,s),\qquad v(t)=\sum_{i=1}^{K}C_{vi}\phi_i(t,-1)+\sum_{i=1}^{K}\int_{-1}^{0}\tau_iC_{vdi}(\tau_is)\phi_i(t,s)ds \tag{5}$$

The Class of Partial Integral Equation (PIE) Systems:

$$\begin{aligned}\mathcal{T}\dot{\mathbf{x}}(t)+\mathcal{B}_{T_1}\dot{w}(t)+\mathcal{B}_{T_2}\dot{u}(t)&=\mathcal{A}\mathbf{x}(t)+\mathcal{B}_1w(t)+\mathcal{B}_2u(t),\\ z(t)&=\mathcal{C}_1\mathbf{x}(t)+\mathcal{D}_{11}w(t)+\mathcal{D}_{12}u(t),\\ y(t)&=\mathcal{C}_2\mathbf{x}(t)+\mathcal{D}_{21}w(t)+\mathcal{D}_{22}u(t).\end{aligned} \tag{6}$$

Box I. Formulation of the DDE, DDF, NDS, ODE–PDE, and PIE Representations of Systems with Delay.

of all matrices in this representation can be inferred from the dimension of the respective state and signals.

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. However, the DDE 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 is the Differential-Difference (DDF) formulation. In addition to the signals included in the DDE, the DDF adds the following.

- The items stored in the signal $r_i(t) \in \mathbb{R}^{p_i}$ are the parts of x, w, u, v which are delayed by amount τ_i. The r_i are 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 into the r_i.

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

For given $u \in L_2^p$, $w \in L_2^m$, and initial conditions $x_0 \in \mathbb{R}^n$, $r_{i0} \in W^{1,2}[-\tau_i, 0]^{p_i}$ satisfying the "sewing condition"

$$r_{i0}(0) = C_{ri}x_0 + D_{rvi}\left(\sum_{i=1}^{K} C_{vi}r_{i0}(-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{vdi}(s)r_{i0}(s)ds\right)$$

for $i = 1, \ldots, K$, we say that $x : [0, \infty] \to \mathbb{R}^n$, $z : [0, \infty] \to \mathbb{R}^q$, $y : [0, \infty] \to \mathbb{R}^r$, $r_i : [-\tau_i, \infty] \to \mathbb{R}^{p_i}$ for $i = 1, \ldots, K$, and $v : [0, \infty] \to \mathbb{R}^{n_v}$ satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ if x is differentiable on $[0, \infty]$, $r_i(s) = r_{i0}(s)$ for $s \in [-\tau_i, 0]$, $r_i(t + \cdot) \in W^{1,2}[-\tau_i, 0]$ for $i = 1, \ldots, K$, and Eqs. (3) are satisfied for all $t \geq 0$. In this manuscript, we assume the C_{vdi} are bounded and in the case where $B_{r1i} \neq 0$ or $B_{r2i} \neq 0$, we require $w \in W^{1,2}[0, \infty]^m$ with $w(s) = 0$ for $s \leq 0$ or $u \in W^{1,2}[0, \infty]^p$ with $u(s) = 0$ for $s \leq 0$, respectively.

Under the conditions stated above, existence of a classical solution x, r_i, v is guaranteed as in Hale (1971), Chapter 9, Thm. 1.1. Furthermore, the "sewing condition" and constraints on w and u ensure the solution r_i is continuously differentiable as in Gil' (2012) p. 226; or Kolmanovskii and Myshkis (1999), Thms. 3.1 and 5.4. Note also that the condition $r_i(t + \cdot) \in W^{1,2}$ may be relaxed to continuity as treated in Henry (1974).

3.1. DDEs are a special case of DDFs

Although Eqs. (3) are more compact, they are more general than the DDEs in (2). Specifically, if we use the conversion formula defined in Eq. (8), then the solution to the DDF is also a solution to the DDE and vice-versa.

Lemma 1. *Suppose that C_{vi}, C_{vdi}, C_{ri}, B_{r1i}, B_{r1i}, D_{rvi}, B_v, D_{1v}, and D_{2v} are as defined in Eqs. (8). Given u, w, x_0, the functions x, y, and z satisfy the DDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ if and only if x, y, z, and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ where*

$$r_i(t) = \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}, \quad r_{i0} = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} \qquad i = 1, \ldots, K.$$

3.2. Neutral-delay systems (NDSs) are a special case of DDFs

DDFs are a natural extension of NDSs, which have the general form of Eq. (4) where for simplicity, we assume $x(t), w(t), u(t) = 0$ for all $t \leq 0$. The conversion from NDS to DDF is given in Eq. (9).

Lemma 2. *Suppose that C_{vi}, C_{vdi}, C_{ri}, B_{r1i}, B_{r2i}, D_{rvi}, B_v, D_{1v}, and D_{2v} are as defined in Eqs. (9). Given u, w, the functions x, y, and z satisfy the NDS defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ if and only if x, y, z, v and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ where $r_{i0} = 0$ and*

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

and

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

3.3. Advantages of the DDF representation

The first advantage of the DDF is that it may include difference equations. To illustrate, suppose we set all matrices to zero except D_{rvi} and C_{vi}. Then we have the following set of Difference Equations (DEs)

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

Another example of DEs can be found in Section 7, where we provide a model of network control which can be represented as a DDF, but not a DDE. A related advantage of the DDF is the ability of DDFs to generate discontinuous solutions if the "sewing condition" on initial conditions is relaxed. This ability is not inherited using our formulation of ODE–PDE or PIE.

The second advantage of the DDF 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 Section 6.2 and the temperature control network in Section 8.

A disadvantage of the DDF is that fewer tools are available for analysis and control of DDFs. This is partially because 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 & Chen, 1998) stability tests have been formulated in the DDF framework.

4. The coupled ODE-PDE representation

We next consider the coupled ODE–PDE representation. Widely recognized as a physical interpretation of delay systems (Hale, 1971; Richard, 2003), ODE–PDE representations allow us to use backstepping methods originally developed for control of PDE models and which have recently been extended to systems with delay — See Karafyllis and Krstic (2014), Krstic and Smyshlyaev (2008) and Zhu et al. (2015). The particular class of ODE–PDE systems, as given in Eq. (5), is equivalent to the class of DDFs. Since we have shown that DDEs are a special case of DDFs, we present only the conversion between DDF and ODE–PDE. Such conversion is trivial, however, as all matrices in the following ODE–PDE model are the same ones used to define the DDF.

For given $u \in L_2^p$, $w \in L_2^m$, and initial conditions $x_0 \in \mathbb{R}^n$, $\phi_{i0} \in W^{1,2}[-1, 0]^{p_i}$ satisfying the "sewing condition"

$$\phi_{i0}(0) = C_{ri}x_0 \qquad (7)$$
$$+ D_{rvi}\left(\sum_{i=1}^{K} C_{vi}\phi_{i0}(-1) + \sum_{i=1}^{K}\int_{-1}^{0}\tau_i C_{vdi}(\tau_i s)\phi_{i0}(s)ds\right)$$

for $i = 1, \ldots, K$, we say that $x : [0, \infty] \to \mathbb{R}^n$, $z : [0, \infty] \to \mathbb{R}^q$, $y : [0, \infty] \to \mathbb{R}^r$, $\phi_i(t) \in W^{1,2}[-1, 0]^{p_i}$ for $i = 1, \ldots, K$, and $v : [0, \infty] \to \mathbb{R}^{n_v}$ satisfy the ODE–PDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ if x is differentiable and ϕ_i is Fréchet differentiable on $[0, \infty]$, $x(0) = x_0$, $\phi_i(0, s) = \phi_{i0}(s)$ for $s \in [-1, 0]$ for $i = 1, \ldots, K$, and Eqs. (5) are satisfied for all $t \geq 0$. As for the DDF, if $B_{r1i} \neq 0$ or $B_{r2i} \neq 0$, we require $w \in W^{1,2}$ or $u \in W^{1,2}$, respectively.

In Eqs. (5), the infinite-dimensional part of the state is ϕ_i — which represents a pipe through which information is flowing. Our formulation is somewhat atypical 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. Solutions to Eqs. (5) and (3) are equivalent, as in the following lemma.

Lemma 3. *Suppose for given u, w, r_{i0}, that x, r_i, v, y, and z satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$. Then for u, w, $\phi_{i0}(s) = r_{i0}(\tau_i s)$, we have that x, v, y, and z also satisfy the ODE–PDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ with $\phi_i(t,s) = r_i(t+\tau_i s)$. Similarly, for given u, w, ϕ_{i0}, if x, v, y, ϕ_i and z satisfy the ODE–PDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$, then x, v, y, and z satisfy the DDF with $r_i(t) = \phi_i(t,0)$ and $r_{i0}(s) = \phi_{i0}(s/\tau_i)$.*

4.1. Advantages of the ODE-PDE representation

In the ODE–PDE representation, the infinite-dimensional part of the state is $\boldsymbol{\phi}(t) \in W^{1,2}[-1,0]^{\sum_i p_i}$. Significantly, 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. Additionally, the ODE–PDE is the native representation used for recently developed backstepping methods for systems with delay, such as proposed in Krstic and Smyshlyaev (2008), Karafyllis and Krstic (2014) and Zhu et al. (2015) 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 PIE representation

A Partial Integral Equation (PIE) has the form of Eq. (6), 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

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

For given $u \in L_2^p$, $w \in L_2^m$, and initial conditions $\mathbf{x}_0 \in \mathbb{R}^n \times L_2[-1,0]^p$, we say that $\mathbf{x}(t) \in \mathbb{R}^n \times L_2[-1,0]^p$, $z : [0,\infty] \to \mathbb{R}^q$, $y : [0,\infty] \to \mathbb{R}^r$ satisfy the PIE defined by $\{\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}\}$ if $\mathbf{x}$ is Fréchet differentiable on $[0,\infty]$, $\mathbf{x}(0) = \mathbf{x}_0$ and Eqs. (6) are satisfied for all $t \geq 0$. As for the ODE–PDE, if $\mathcal{B}_{T_1} \neq 0$ or $\mathcal{B}_{T_2} \neq 0$ we require $w \in W^{1,2}$ or $u \in W^{1,2}$, with $w(0) = 0$ or $u(0) = 0$, respectively.

此刻 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 DDF or ODE–PDE representation, it is relatively straightforward to convert to a PIE by defining the operators $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}$ for which solutions to Eqs. (6) also define solutions to Eqs. (3) (DDF) and Eqs. (5) (ODE-PDE). Specifically, let us define $\{\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}\}$ as in Eqs. (10) where the required matrices are as defined in Eqs. (11). Then we have the following.

Lemma 4. *Given u, w, and x_0, ϕ_{i0} satisfying the "Sewing Condition (7)", Suppose x, ϕ_i, v, y, and z satisfy the ODE–PDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$. Then y and z also satisfy the PIE defined by $\{\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}\}$ with $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}$ as defined in Eqs. (10) and*

$$\mathbf{x}(t) := \begin{bmatrix} x(t) \\ \partial_s\phi_1(t,\cdot) \\ \vdots \\ \partial_s\phi_K(t,\cdot)\end{bmatrix} \quad \mathbf{x}_0 := \begin{bmatrix} x_0 \\ \partial_s\phi_{10} \\ \vdots \\ \partial_s\phi_{K0}\end{bmatrix}.$$

Furthermore, for given u, w, $\mathbf{x}_0 \in \mathbb{R}^n \times L_2[-1,0]^p$, if y, z and $\mathbf{x}$ satisfy the PIE defined by $\{\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}\}$, then x, ϕ_i, v, y, and z satisfy the ODE–PDE defined by $\{A_i, B_i, C_i, D_{ij}, \ldots\}$ where

$$\begin{bmatrix} x(t) \\ \phi_1(t,\cdot) \\ \vdots \\ \phi_K(t,\cdot)\end{bmatrix} = \mathcal{T}\mathbf{x}(t) + \mathcal{B}_{T1}w(t) + \mathcal{B}_{T2}u(t), \quad \begin{bmatrix} x_0 \\ \phi_{10} \\ \vdots \\ \phi_{K0}\end{bmatrix} = \mathcal{T}\mathbf{x}_0.$$

Note that while solutions of the ODE–PDE are equivalent to those of the PIE, some notions of stability of such solutions may not be.

5.1. Advantages of the PIE representation

Like the DDF and ODE–PDE, PIEs can be used to represent low-dimensional delay channels. An additional advantage is the lack of boundary conditions or the 'sewing' constraint on the initial condition in, e.g. Eq. (7). This is significant in that the implicit dynamics in an ODE–PDE imposed by boundary conditions on ϕ_i complicate stability and optimal control problems. By contrast, in PIEs, the infinite-dimension part of the state is $\partial_s\phi_i$ which is in L_2 but is otherwise unconstrained. Furthermore, PIEs are defined using the algebra of Partial Integral (PI) operators. The algebraic nature of PI operators implies that most tools developed for matrices can be extended to PIEs — including the LMI framework. Specifically, the LMIs for H_∞-optimal observer and controller synthesis have been extended to PIEs, as can be found in Shivakumar, Das, Weiland and Peet (2020) and Wu, Shivakumar, Peet, and Hua (2020), respectively. We refer to Linear PI Inequalities (LPIs) as this extension of the LMI framework and a Matlab toolbox for solving LPIs can be found in Shivakumar, Das and Peet (2020). An example of these synthesis results can be found in Section 8.

5.2. Conversion from DDE to PIE

In this subsection, we bypass the DDF and give a formula for direct conversion between the DDE and PIE representations. This formula is given in Eqs. (12).

6. Modeling of a network of UAVs

To compare the DDE, DDF, ODE–PDE and PIE representations, we return to control of a network of UAVs. In this section, we focus on the DDE and DDF representations, as conversion from DDF to ODE–PDE or PIE is straightforward using the formulae provided. For simplicity, we eliminate the state delays $\hat{\tau}_{ij}$ governing interactions between UAVs (we will consider state delays in Section 8) and map the process, input, and output delays to a common set of delays, $\{\tau_j\}_{j=1}^{3N}$ where the index for the process delay for UAV i is as $\tau_i = \bar{\tau}_i$, the index for input delay for UAV i is as $\tau_{N+i} = h_i$, and the index of the output delay from UAV i is as $\tau_{2N+i} = \tilde{\tau}_i$. The process noise is dimension $w(t) \in \mathbb{R}^m$, the common 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 model in Eqs. (1) as

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

Conversion Formula from DDE to DDF:

$$\begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I_{n+q+r}, \; C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix}, \; C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix}, \; D_{rvi} = 0, \; \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = I_{n+m+p}. \tag{8}$$

Conversion Formula from NDS to DDF:

$$D_{rvi} = \begin{bmatrix} 0_n & 0 & 0 \\ 0 & 0_{m,q} & 0 \\ 0 & 0 & 0_{p,r} \\ I_n & 0 & 0 \end{bmatrix}, \qquad \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = \begin{bmatrix} I_n & 0 & 0 \\ 0 & I_m & 0 \\ 0 & 0 & I_p \\ A_0 & B_1 & B_2 \end{bmatrix}, \qquad \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I_{n+q+r},$$

$$C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} & E_i \\ C_{1i} & D_{11i} & D_{12i} & E_{1i} \\ C_{2i} & D_{21i} & D_{22i} & E_{2i} \end{bmatrix}, \quad C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) & E_{di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) & E_{1di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) & E_{2di}(s) \end{bmatrix}. \tag{9}$$

Conversion Formula from ODE–PDE or DDF to PIE:

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

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

where

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

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

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

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

Box II. Conversion formulae from DDE to DDF, NDS to DDF, and DDF/ODE–PDE to PIE.

Conversion Formula from DDE to PIE: $\mathcal{T}, \mathcal{A}, \mathcal{B}_i, \mathcal{C}_i, \mathcal{D}_{ij}, \mathcal{B}_{T_i}$ are as defined in Eq. (10) where now

$$I_\tau = \begin{bmatrix} \frac{1}{\tau_1} I_{n+m+p} & & \\ & \ddots & \\ & & \frac{1}{\tau_K} I_{n+m+p} \end{bmatrix}, \; \mathbf{T}_0 = \begin{bmatrix} \begin{bmatrix} I_n & 0_{n,m} & 0_{n,p} \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} I_n & 0_{n,m} & 0_{n,p} \end{bmatrix}^T \end{bmatrix}, \; \mathbf{T}_1 = \begin{bmatrix} \begin{bmatrix} 0_{m,n} & I_m & 0_{m,p} \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} 0_{m,n} & I_m & 0_{m,p} \end{bmatrix}^T \end{bmatrix}, \; \mathbf{T}_2 = \begin{bmatrix} \begin{bmatrix} 0_{p,n} & 0_{p,m} & I_p \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} 0_{p,n} & 0_{p,m} & I_p \end{bmatrix}^T \end{bmatrix},$$

$$\mathbf{T}_a = 0_{(n+m+p)K}, \quad \mathbf{T}_b = -I_{(n+m+p)K},$$

$$\begin{bmatrix} \mathbf{A}(s) \\ \mathbf{C}_{11}(s) \\ \mathbf{C}_{21}(s) \end{bmatrix} = -\begin{bmatrix} X_1(s) & \cdots & X_K(s) \end{bmatrix}, \qquad X_i(s) = \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix} + \tau_i \int_{-1}^{s} \begin{bmatrix} A_{di}(\tau_i\eta) & B_{1di}(\tau_i\eta) & B_{2di}(\tau_i\eta) \\ C_{1di}(\tau_i\eta) & D_{11di}(\tau_i\eta) & D_{12di}(\tau_i\eta) \\ C_{2di}(\tau_i\eta) & D_{21di}(\tau_i\eta) & D_{22di}(\tau_i\eta) \end{bmatrix} d\eta,$$

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_{10} & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_{20} & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_{10} & D_{11} & D_{12} \\ C_{20} & D_{21} & D_{22} \end{bmatrix} + \sum_{i=1}^{K} \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix} + \int_{-1}^{0} \sum_{i=1}^{K} \tau_i \begin{bmatrix} A_{di}(\tau_i s) & B_{1di}(\tau_i s) & B_{2di}(\tau_i s) \\ C_{1di}(\tau_i s) & D_{11di}(\tau_i s) & D_{12di}(\tau_i s) \\ C_{2di}(\tau_i s) & D_{21di}(\tau_i s) & D_{22di}(\tau_i s) \end{bmatrix} ds. \tag{12}$$

Box III. Direct conversion formula from DDE to PIE, bypassing the DDF.

6.1. The DDE representation

To model this network as a DDE, we consider Eq. (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 $i = 1, \dots, N$ as

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

All other undefined matrices in Eq. (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. The DDF representation

To efficiently model the network model as a DDF, we retain the matrix A_0 from the DDE model in Section 6.1, set $C_1 = C_{10}$ and leave D_{12} unchanged. Our first step is to 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 $i = 1, \dots, N$ 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 $i = 1, \dots, 3N$.

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

Next, we construct output $v(t)$ by defining C_{vi} for $i = 1, \dots, 3N$ 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

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

which recovers the network model.

6.3. Complexity of DDEs vs. DDFs

In the DDF model, the infinite-dimensional state is r_i. In our DDF formulation of the UAV model: each process delay adds n states; each input delay adds n states; and each output delay adds r states to this vector. The aggregated infinite-dimensional state is then L_2^($\sum_{p_i} = (2n + r)N$). Assuming that optimal control and estimation problems are tractable when the number of infinite-dimensional states is less than 50 (Peet, 2020a), and if we suppose $n = r = 1$, then it is possible to control 17 UAVs. By contrast, in the DDE model of our UAVs, the infinite-dimensional state is $L_2^{3N(m+p+r)}$ (meaning we can control at most 5 or 6 UAVs).

7. A network 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 Section 6.2 and define

$$B_1 = \begin{bmatrix} b_{11} \\ \vdots \\ b_{1N} \end{bmatrix}, \quad D_{21} = \begin{bmatrix} d_{21,1} \\ \vdots \\ d_{21,N} \end{bmatrix}, \quad C_2 = \begin{bmatrix} c_{2,1} & & \\ & \ddots & \\ & & c_{2,N} \end{bmatrix},$$

$$D_{22i} = e_i \otimes d_{22i}.$$

Aggregating the measurements, we have

$$y(t) = C_2x(t) + D_{21}w(t) + \sum_{i=1}^N D_{22i}u(t-\tau_i).$$

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

$$\dot{x}(t) = A_0x(t) + B_1w(t) + \sum_{i=1}^N B_{2i}Fy(t-\tau_i),$$
$$z(t) = C_1x(t) + D_{12}Fy(t),$$
$$y(t) = C_2x(t) + D_{21}w(t) + \sum_{i=1}^N D_{22i}Fy(t-\tau_i). \tag{13}$$

Clearly, there is no DDE model with solutions which satisfy Eqs. (13) due to the recursion in the output (Henry, 1974). However (assuming appropriate initial conditions), 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.

$$\tilde{D}_{12} = D_{12}FD_{21}, \quad \tilde{D}_{22} = 0, \quad \tilde{C}_1 = C_1 + D_{12}FC_2$$
$$C_{ri} = FC_2, \quad B_{r1i} = FD_{21}, \quad [D_{rvi}]_i = FD_{22i}$$
$$B_v = \begin{bmatrix} B_{21} & \cdots & B_{2N} \end{bmatrix}, \quad C_{vi} = e_i \otimes I$$
$$D_{1v} = D_{12}FD_{2v}, \quad D_{2v} = \begin{bmatrix} D_{22,1} & \cdots & D_{22,N} \end{bmatrix} \tag{14}$$

Lemma 5. *For given r_{i0}, x_0, suppose r_i, v, y, x, and z satisfy the DDF defined by*

$$\{A_0, B_1, B_v, \tilde{C}_1, \tilde{D}_{12}, D_{1v}, D_{2v}, C_{ri}, B_{r1i}, D_{rvi}, C_{vi}\}$$

given by Eqs. (14). Then x, z and y also satisfy Eqs. (13).

8. Optimal control of a large network

To illustrate the computational advantages of the DDFs, ODE–PDEs, and PIEs for controller synthesis problems, we consider the scalable network model with state-delay for centralized control of water temperature for multiple showering customers as defined in Peet (2020a). If T_{1i} is the tap position and T_{2i} is the temperature for user i, then the dynamics of this model are given by

$$\dot{T}_{1i}(t) = T_{2i}(t) - w_i(t), \tag{15}$$
$$\dot{T}_{2i}(t) = -\alpha_i\left(T_{2i}(t-\tau_i) - w_i(t)\right) + \sum_{j\neq i}^N \gamma_{ij}\alpha_j\left(T_{2j}(t-\tau_j) - w_j(t)\right) + u_i(t),$$
$$z(t) = \begin{bmatrix} \sum_{i=1}^N T_{1i}(t) & .1\sum_{i=1}^N u_i(t) \end{bmatrix}^T.$$

For N users, we choose $\alpha_i = 1$, $\gamma_{ij} = 1/N$, $\tau_i = i$, and $w_i(t) = N$.

8.0.1. DDE formulation of the network

We first form the aggregate state vector as

$$x(t)=\begin{bmatrix}T_{11}(t) & \cdots & T_{1N}(t) & T_{21}(t) & \cdots & T_{2N}(t)\end{bmatrix}^T$$

and define the DDE model using

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

In this formulation, we have $n=2N$ states, $m=N$ disturbances, $p=N$ inputs, $q=2$ regulated outputs and $K=N$ delays ($\tau_{ij}=\tau_j$). Using the SOS-based H_∞-optimal controller synthesis algorithm for DDEs as presented in Peet (2020a), we were able to design controllers for $N=4$ users. This corresponds to an infinite-dimensional channel of size $L_2^{nK=32}$.

8.0.2. DDF formulation of the network

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

$$r_i(t)=\begin{bmatrix}0_{1\times N+i-1} & 1 & 0_{1\times N-i}\end{bmatrix}x(t)=T_{2i}(t).$$

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

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

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

$$v(t)=\begin{bmatrix}T_{21}(t-\tau_1) & \cdots & T_{2N}(t-\tau_N)\end{bmatrix}^T.$$

This is accomplished by defining

$$C_{vi}=e_i=\begin{bmatrix}0_{1\times i-1} & 1 & 0_{1\times N-i}\end{bmatrix}^T,\qquad C_{vdi}=0_{2\times N}.$$

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.

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

In the DDF formulation, we have $n=2N$ states, $m=N$ disturbances, $p=N$ inputs, $q=2$ regulated outputs, $K=N$ delays ($\tau_{ij}=\tau_j$), and K delay channels, each of dimension L_2^1.

8.1. H_∞-optimal control using PIETOOLS 2020a

For H_∞-optimal controller synthesis, we used the DDF to PIE converter `convert_PIETOOLS_DDF` and H_∞-optimal synthesis option in the PIETOOLS 2020a Matlab toolbox, as described in Shivakumar, Das and Peet (2020) and available online at Peet, Shivakumar, and Das (2020). The DDF system input format for this toolbox is described in detail in the user manual (Peet et al., 2020), as is the converter and controller synthesis feature. In this toolbox, the extreme performance option was selected to decrease computation times and reduce memory usage. The H_∞-optimal controller synthesis feature in PIETOOLS solves the optimal control problem for a PIE and is based on the result in Shivakumar, Das, Weiland and Peet (2020). The numerical test was performed on a desktop computer with 128GB RAM and a 3 GHz intel processor. CPU seconds is as listed for the interior-point calculations determined by Sedumi. The computation times, indexed by number of users, are listed in Table 1. In all cases, the achieved closed-loop H_∞-norm was in the interval [.3, 3]. Practically, we observe that the controller synthesis problem is tractable up to 40 users — a significant improvement from the 4 users in Peet (2020a). Note that 40 users corresponds to an aggregated infinite-dimensional channel of size $L_2^{\sum_i p_i=N=40}$. Also recall that for 40 users, we have 80 states, 40 inputs, 40 disturbances and 40 delays.

Table 1
IPM CPU sec vs. # of states (N) for H_∞-optimal control of Eq. (15).

N→	1	3	5	10	20	30	40
CPU sec	.48	.638	2.42	94.7	5455	35k	157k

Note that the PIETOOLS 2020a toolbox does not require use of the PIE formulation and will convert a DDE to a DDF, if desired. There is also a feature for constructing minimal DDF representations of DDEs — which can be very useful for solving large network problems. The conversion from a NDS to DDF is also included in the PIETOOLS library `examples_DDF_library_PIETOOLS.m`.

9. 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 from DDE to DDF is not advised if the delayed channels are low-dimensional (although PIETOOLS 2020a includes a feature for constructing minimal DDF representations of DDEs). Using the given formulae and definitions of solution, we show that the set of solutions for the DDF, ODE–PDE, and PIE are equivalent. These results imply that if there is a valid conversion formula, many solutions to the H_∞-optimal control and estimation problems can be converted between representations by applying this formula to the closed-loop system. However, this only works if optimality is defined in terms of the finite-dimensional vectors, x_0, u, w, x, y, z. This is because any input–output pair $(u, w, x_0) \mapsto (y, z, x)$ which defines a solution to one representation also defines a solution for every other representation for which there is a valid conversion formula. Likewise, stability of the representations is equivalent as long as the stability definition only involves the finite-dimensional vectors, x_0, x, u, w, y, z.

The results and formulae in this paper are meant to provide a convenient reference for researchers interested in exploring alternative representations of delay systems. A summary of the representations and conversion formulae is given in Table 2, along with examples of simulation tools and controller synthesis results. 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. Finally, we have shown that H_∞-optimal control in the DDF/ODE–PDE/PIE framework allows up to 40 agents, while formulation in the DDE framework only allows for control of 4 agents.

Table 2
Conversion formulae (DDF,PDE,PIE), simulation tools (Sim), controller design tools (H_∞), and model definitions (Model) for each class of systems (PDE→ODE-PDE).

Need→	DDF	PDE	PIE	Sim	H_∞	Model
DDE	(8)	(8)+(5)	(12)	Bellen and Zennaro (2013)	Peet (2020a)	(2)
Neut.	(9)	(9)+(5)	(9)+(11)	Bellen and Zennaro (2013)	Xu, Lam, Yang, and Verriest (2003)	(4)
DDF	X	(5)	(11)	–	–	(3)
PDE	X	X	Shivakumar et al. (2019)	Wouwer, Saucez, and Vilas (2014)	Krstic and Smyshlyaev (2008)	(5)
PIE	X	X	X	–	Shivakumar, Das, Weiland and Peet (2020)	(6)

References

Appell, J., Kalitvin, A., & Zabrejko, P. (2000). *Partial integral operators and integro-differential equations: pure and applied mathematics*. CRC Press.

Bellen, A., & Zennaro, M. (2013). *Numerical methods for delay differential equations*. Oxford University Press.

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

Das, A., Shivakumar, S., Weiland, S., & Peet, M. (2019). H_∞ optimal estimation for linear coupled PDE systems. In *Proc. of the IEEE Conf. on Decision and Control* (pp. 262–267).

Gil', M. (2012). *Stability of finite and infinite dimensional systems*. Kluwer.

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

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

Hale, J. (1971). Functional differential equations. In *Analytic Theory of Differential Equations* (pp. 9–22). Springer.

Henry, D. (1974). Linear autonomous neutral functional differential equations. *Journal of Differential Equations*, *15*(1), 106–128.

Karafyllis, I., & Krstic, M. (2014). On the relation of delay equations to first-order hyperbolic partial differential equations. *ESAIM. Control, Optimisation and Calculus of Variations*, *20*(3).

Kolmanovskii, V., & Myshkis, A. (1999). *Introduction to the theory and applications of functional differential equations*. Kluwer Academic Publishers.

Krstic, M., & 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.

Mazenc, F., Ito, H., & Pepe, P. (2013). Construction of Lyapunov functionals for coupled differential and continuous time difference equations.

Niculescu, S.-I. (2001). *Lecture notes in control and information science*: *vol. 269*, *Delay effects on stability: a robust control approach*. 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., Shivakumar, S., & Das, A. (2020). PIETOOLS. https://control.asu.edu/pietools.

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

Pepe, P., Karafyllis, I., & Jiang, Z.-P. (2008). On the liapunov–krasovskii methodology for the ISS of systems described by coupled delay differential and difference equations. *Automatica*, *44*(9).

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

Shivakumar, S., Das, A., & Peet, M. (2020). PIETOOLS: A Matlab® toolbox for manipulation and optimization of partial integral operators. In *Proc. of the American Control Conf.* (pp. 2667–2672).

Shivakumar, S., Das, A., Weiland, S., & Peet, M. (2019). A generalized LMI formulation for input-output analysis of linear systems of ODEs coupled with PDEs. In *Proc. of the IEEE Conf. on Decision and Control* (pp. 280–285).

Shivakumar, S., Das, A., Weiland, S., & Peet, M. (2020). Duality and H_∞-optimal control of ODE-PDE systems. In *Proc. of the IEEE Conf. on Decision and Control* (pp. 5689–5696). arXiv:2004.03638.

Wouwer, A., Saucez, P., & Vilas, C. (2014). *Simulation of ODE/PDE models with MATLAB®, OCTAVE and SCILAB*. Springer.

Wu, S., Shivakumar, S., Peet, M., & Hua, C.-C. (2020). H_∞-optimal observer design for linear systems with delays in states, outputs and disturbances. In *Proc. of the IEEE Conf. on Decision and Control* (pp. 983–988). arXiv:2004.04482.

Xu, S., Lam, J., Yang, C., & Verriest, E. (2003). An LMI approach to guaranteed cost control for uncertain linear neutral delay systems. *International Journal of Robust and Nonlinear Control*, *13*(1).

Zhang, S., & 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., & Krstic, M. (2015). Adaptive backstepping control of uncertain linear systems under unknown actuator delay. *Automatica*, *54*, 256–265.

Matthew M. Peet is an Associate Professor of Aerospace Engineering at Arizona State University. He received B.Sc. degrees in Physics and in Aerospace Engineering from the University of Texas at Austin in 1999 and the M.Sc. and Ph.D. degrees in Aeronautics and Astronautics from Stanford University in 2001 and 2006. He was a Postdoctoral Fellow at INRIA from 2006–2008. From 2008–2012 he was an Assistant Professor of Aerospace Engineering at the Illinois Institute of Technology. He has been with Arizona State University since 2012.