Minimal Differential Difference Realizations of Delay Differential, Differential Difference, and Neutral Delay Systems

Matthew M. Peet

***Abstract*— Delay-Differential Equations (DDEs) are often used to represent control of and over large networks. However, the presence of delay makes the problems of analysis and control of such networks challenging. Recently, Differential Difference Equations (DDFs) have been proposed as a modelling framework which allows us to more efficiently represent the low-dimensional nature of delayed channels in a network or large-scale delayed system. Unfortunately, however, the standard conversion formulae from DDE to DDF do not account for this low-dimensional structure - hence any efficient DDF representation of a large delayed network or system must be hand-crafted. In this paper, we propose an algorithm for constructing DDF realizations of both DDE and DDF systems wherein the dimension of the delayed channels has been minimized. Furthermore, we provide a convenient PIETOOLS implementation of these algorithms and show that the algorithm significantly reduces the complexity of the model for several illustrative examples, including Neutral Delay Systems (NDSs).**

***Index Terms*— Computational methods, Delay systems, Distributed parameter systems, Modeling, Network analysis and control**

I. Introduction

ACCURATE models of control of and over networks invariably include communication delay. Sources of delay include: state delay; input delay; process delay; and output delay. Furthermore, as the number of agents in the network increases, the number of delays increases proportionally. Unfortunately, the presence of delay complicates the problems of analysis and control of these networks. Furthermore, although delay is generally considered undesirable, and network designs often minimize the number of delayed channels, the low-dimensional delay structure of the network is typically lost when solving analysis and control problems. For example, even using such simple tools as a Padé reduction of the delayed system to an ODE, unless the structure of the delayed channels is exploited efficiently, optimal control of a simple network of 5 states and 10 delays will become intractable. This is because standard Delay-Differential Equation (DDE) models (Eqn. (2)) and tools do not account for the low-dimensional structure of the delay channels.

For example, consider a relatively simple model of network control with delay, such as might be used to represent a fleet of UAVs:

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

In this model, there are N^2+2N delayed channels, each of dimension $\mathbb{R}^{Nn_i+p+m}$. Thus for a simple 5 agent network, each with 1 input, 1 output, and 4 states, there are $35\cdot 22 = 770$ infinite-dimensional states. Even assuming we use a 6th order Pade approximation, that yields 4620 ODE states - too large for many linear optimal control algorithms.

By contrast, if we are able to represent the network as a Differential-Difference (DDF) Equation (See Eqn. (3)), then there are still N^2+2N delayed states, however, the dimensions of these states are heterogeneous, meaning the total number of infinite-dimensional states is $25\cdot 4 + 10 = 110$ - a number which significantly reduces the complexity of the problem. Note, however, that this DDF realization was hand-crafted and may not be minimal in any sense.

In addition to the problem of minimal DDF realization of DDEs, we are occasionally confronted with the problem of minimal DDF realization of DDFs. Inefficient DDF representations often arise from naive conversion of DDEs (as presented in Subsection II-C) and and Neutral Delay Systems (NDSs) (as presented in Subsection II-D) to DDFs. Specifically, if the DDF representation of these DDE and NDS systems is not chosen carefully, such a representation may contain unnecessary infinite-dimensional channels - creating the complexity problems described above.

In this paper, we propose an algorithm for constructing minimal DDF realizations of DDE systems. As illustrated in the Examples in Section VI, the existence of such minimal realizations can dramatically reduce the computational complexity of analysis and control problems for delayed networks. In addition, we extend this result to an algorithm for minimal DDF realizations of DDFs - thus also solving the problem of inefficient DDF representation of NDSs.

a) Overview: Having motivated the problem of minimal DDF realization of both DDEs and DDFs, we provide some background on the use of DDFs in analysis and control of systems with delay.

We first note that the DDF is not new. DDF models have been proposed in, e.g. [1]–[7], and equivalence between certain DDE/DDF representations have been studied in [8], [9]. More recently, a comprehensive treatment of the relationships between the solutions of DDEs, DDFs, NDSs, ODE-PDEs and

M. Peet is with the School for the Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85298 USA. e-mail: mpeet@asu.edu

This work was supported by the National Science Foundation under grants No. CNS-1739990 and CMMI-1931270.

PIEs was published in [10], which considered a large class of input-output systems and provided naive conversion formulae between representations (some of which are referenced in this paper). However, use of the naive conversion formulae in [10] often results in high-dimensional delay channels - significantly complicating analysis and control problems based on this representation. The goal of this paper, then, is to improve the results in [10] by providing algorithms for the efficient construction of *minimal* DDF representations - thereby allowing for more efficient representation in the DDF framework and hence more effective use of the analysis and control algorithms designed for the DDF class of systems.

At this point, we should probably define more carefully what we mean by minimal DDF representations. We mostly use the equivalent term "minimal realization" (as opposed to representation) and this usage is intended to reflect a generalization of the concept of minimal realization of state-space ODE systems, wherein unobservable and uncontrollable states can be eliminated without changing the input-output properties of the system. A survey of minimal realization of state-space ODEs can be found in [11]. In this context, the purpose of this paper is to show how unobservable/uncontrollable infinite-dimensional channels can be eliminated from the DDE and DDF representations without altering the input-output properties of the system in any way. Note that this formulation of the problem is distinct from the field of model reduction, wherein the reduced states affect the input-output properties - but are deemed to be insignificant in some sense.

Having concluded the introduction, we now briefly overview the organization of the paper. In Section II we define the DDE and DDF representations and recall some naive conversion formulae previously proposed in [10]. In Section III we use the Singular Value Decomposition (SVD) to identify unused infinite-dimensional subspaces in the DDE and propose a new, equivalent DDF realization which does not include these subspaces. In Section IV, we extend this result to eliminate unused subspaces in DDF representations. Next, in Section V, we present a user-friendly PIETOOLS implementation of these algorithms. In Sections VI and VII we then apply the algorithms to network problems and illustrative examples - demonstrating dramatic reductions in computational complexity.

A. Notation

Shorthand notation includes the Hilbert spaces $L_2^m[X] := L_2(X;\mathbb{R}^m)$ and $W_2^m[X] := W^{1,2}(X;\mathbb{R}^m) = H^1(X;\mathbb{R}^m) = \{x : x, \dot{x} \in L_2^m[X]\}$. We use L_2^m and W_2^m when domains are clear from context. $I_n \in \mathbb{S}^n$ denotes the identity matrix. $0_{n\times m} \in \mathbb{R}^{n\times m}$ is the matrix of zeros. In both cases, n and m are omitted when dimensions are clear from context. For a natural number, $K \in \mathbb{N}$, we adopt the index shorthand notation where $i \in [K]$ denotes $i = 1, \cdots, K$.

II. The DDE and DDF representations

In this section, we recall the form of a DDE and DDF. We then briefly restate a prior result showing how a DDE may be formulated as a DDF. We also recall the formulae for conversion of a Neutral Delay System (NDS) to a DDF.

A. The Delay Differential Equation (DDE) Model

The general form of a DDE is modeled as

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

where $0 < \tau_1 < \cdots < \tau_K$. For given $u \in W^{1,2}[0,\infty]^p$, $w \in W^{1,2}[0,\infty]^m$ (with $u(s) = 0$ and $w(s) = 0$ for $s \le 0$) 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 $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ 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 Eqns. (2) are satisfied for all $t \ge 0$. Note that under these conditions, existence of a continuously differentiable solution x is guaranteed as in, e.g. Thm. 3.3 of Chapter 3 in [12] (See also Thm. 1.1 of Chapter 6 in [13]).

B. The Differential Difference (DDF) Model

The general form of a DDF is given as follows.

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

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

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

for $i \in [K]$, 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 $i \in [K]$, and $v : [0,\infty] \to \mathbb{R}^{n_v}$ satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ 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 $i \in [K]$, and Eqns. (3) are satisfied for all $t \ge 0$. In this manuscript, we assume the C_{vdi} are bounded. Under these conditions and definitions, existence of a solution x, r_i, v with r_i continuously differentiable follows from [14] p. 226; or [12], Thms. 3.1 and 5.4.

C. A Naive Conversion from DDE to DDF

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

$$\begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I, \quad C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix},$$
$$C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix}, \; D_{rvi} = 0,$$
$$\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = I, \tag{4}$$

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 Eqns. (4). Given u, w, x_0, the functions x, y, and z satisfy the DDE defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ if and only if x, y, z, and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ where

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

See [10] for a proof.

D. A Naive Conversion from NDS to DDF

Like DDEs, Neutral Delay Systems (NDSs) can also be reformulated in the DDF representation. Consider the following general form of NDS.

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_{10} & D_{11} & D_{12} \\ C_{20} & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} \tag{5}$$
$$+ \sum_{i=1}^{K} \begin{bmatrix} A_i & B_{1i} & B_{2i} & E_i \\ C_{1i} & D_{11i} & D_{12i} & E_{1i} \\ C_{2i} & D_{21i} & D_{22i} & E_{2i} \end{bmatrix} \begin{bmatrix} x(t-\tau_i) \\ w(t-\tau_i) \\ u(t-\tau_i) \\ \dot{x}(t-\tau_i) \end{bmatrix}$$
$$+ \sum_{i=1}^{K} \int_{-\tau_i}^{0} \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) & E_{di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) & E_{1di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) & E_{2di}(s) \end{bmatrix} \begin{bmatrix} x(t+s) \\ w(t+s) \\ u(t+s) \\ \dot{x}(t+s) \end{bmatrix} ds$$

The definition of solution can be found in [10] and imposes a continuity constraint on the initial condition to ensure the solution is continuously differentiable. Similar to the process for a DDE, a NDS can be represented as a DDF using the following definitions.

$$D_{rvi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I & 0 & 0 \end{bmatrix}, \quad C_{vi} = \begin{bmatrix} A_i & B_{1i} & B_{2i} & E_i \\ C_{1i} & D_{11i} & D_{12i} & E_{1i} \\ C_{2i} & D_{21i} & D_{22i} & E_{2i} \end{bmatrix},$$
$$C_{vdi}(s) = \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) & E_{di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) & E_{1di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) & E_{2di}(s) \end{bmatrix},$$
$$\begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I, \; \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ A_0 & B_1 & B_2 \end{bmatrix} \tag{6}$$

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 Eqns. (6). Given u, w, x_0, the functions x, y, and z satisfy the NDS defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ if and only if x, y, z, v and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ where

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

See [10] for a proof.

III. Minimal DDF Realizations of DDEs

The conversion formulae in Eqns. (4) yield a representation wherein all states are delayed and hence this representation does not account for the fact that in many large-scale DDEs only relatively low-dimensional subsets of the state experience delay. For such systems, therefore, the conversion from DDE to DDF must be carefully selected by the modeler to account for this structure. In the following theorem we provide a class of equivalent DDF realizations of the DDE which can be combined with a Singular Value Decomposition (SVD) to eliminate unused delay channels.

Theorem 3: Define

$$P_i := \begin{bmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{bmatrix},$$
$$P_{di}(s) := \begin{bmatrix} A_{di}(s) & B_{1di}(s) & B_{2di}(s) \\ C_{1di}(s) & D_{11di}(s) & D_{12di}(s) \\ C_{2di}(s) & D_{21di}(s) & D_{22di}(s) \end{bmatrix}$$

and let $\hat{P}_{di}$ satisfy $P_{di}(s) = Z(s)\hat{P}_{di}$ for $s \in [-\tau_i, 0]$, $i \in [K]$ and for some $Z(s)$. Suppose $U_i \in \mathbb{R}^{2(n+q+r)\times p_i}, V_i \in \mathbb{R}^{(n+m+p)\times p_i}$ satisfy

$$U_i V_i^T := \begin{bmatrix} P_i \\ \hat{P}_{di} \end{bmatrix} \qquad i \in [K].$$

Let (for $i \in [K]$)

$$\begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} \end{bmatrix} = V_i^T, \quad \begin{bmatrix} C_{vi} \\ C_{vdi} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & Z(s) \end{bmatrix} U_i,$$
$$\begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} = I, \quad \text{and} \quad D_{rvi} = 0.$$

Then, given u, w, x_0, the functions x, y, and z satisfy the DDE defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$ if and only if x, y, z, v and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, C_{ri}, C_{vi}, C_{vdi}\}$ where

$$r_i(t) = V_i^T \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}, \quad r_{i0} = V_i^T \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} \qquad i \in [K].$$

Note that if $Z(s)$ is a monomial basis then $\hat{P}_{di}$ is uniquely defined.

Proof: To simplify the proof, we define the Dirac operators $\Delta_i : W^{1,2}[-\tau_i, 0]^n \to \mathbb{R}^n$ by $\Delta_i \mathbf{x} := \mathbf{x}(-\tau_i)$ and the integral operators $\mathbf{I}_{Z,i} : L_2[-\tau_i]^n \to \mathbb{R}^n$ by

$$\mathbf{I}_{Z,i}\mathbf{x}(s) := \int_{-\tau_i}^{0} Z(s)\mathbf{x}(s)ds.$$

Now suppose that x, y, and z satisfy the DDE defined by $\{A_i, B_i, C_i, D_{ij}, \cdots\}$. For any $t \geq 0$, denote the functions: $x_{t,i}(s) := x(t+s)$; $w_{t,i}(s) := w(t+s)$; and $u_{t,i}(s) := u(t+s)$

for $s \in [-\tau_i, 0]$, $i \in [K]$. Then by definition of solution, we have

$$\begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \in W^{1,2}[-\tau_i, 0]^{n+m+p}.$$

Now, if the r_i are as defined above, we have for $i \in [K]$

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

and hence

$$r_i(t + \cdot) = V_i^T \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \in W^{1,2}[-\tau_i, 0]^{p_i}.$$

Furthermore,

$$\begin{aligned}
\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} &= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K P_i \begin{bmatrix} x(t-\tau_i) \\ w(t-\tau_i) \\ u(t-\tau_i) \end{bmatrix} \\
&\quad + \sum_{i=1}^K \int_{-\tau_i}^0 P_{di}(s) \begin{bmatrix} x(t+s) \\ w(t+s) \\ u(t+s) \end{bmatrix} ds \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K \Delta_i P_i \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} + \sum_{i=1}^K \mathbf{I}_{Z,i} \hat{P}_{di} \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K \left(\Delta_i P_i + \mathbf{I}_{Z,i} \hat{P}_{di} \right) \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K \begin{bmatrix} \Delta_i & \mathbf{I}_{Z,i} \end{bmatrix} \begin{bmatrix} P_i \\ \hat{P}_{di} \end{bmatrix} \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K \begin{bmatrix} \Delta_i & \mathbf{I}_{Z,i} \end{bmatrix} U_i V_i^T \begin{bmatrix} x_{t,i} \\ w_{t,i} \\ u_{t,i} \end{bmatrix} \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \sum_{i=1}^K C_{vi} r_i(t-\tau_i) + \sum_{i=1}^K \int_{-\tau_i}^0 C_{vdi}(s) r_i(t+s) ds \\
&= P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + v(t) = P_0 \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} B_v \\ D_{1v} \\ D_{2v} \end{bmatrix} v(t)
\end{aligned}$$

as desired. The sewing condition is satisfied since $D_{rvi} = 0$. Thus we conclude the that x, y, z, v and r_i satisfy the DDF defined by $\{A_i, B_i, C_i, D_{ij}, C_{ri}, C_{vi}, C_{vdi}\}$. The steps can be reversed to prove the converse - as illustrated in the proof of Theorem 4. ■

The matrices U_i and V_i parameterize the set of equivalent DDF realizations. In the following subsection, we show how U_i and V_i may be selected to minimize the dimension of r_i - the infinite-dimensional delayed channels.

A. Eliminating Unused DDE Channels via the SVD

The matrices P_i and $\hat{P}_{di}(s)$ represent all possible ways in which a delayed channel can be used. However, these matrices are not full column rank and the nullspace of $\begin{bmatrix} P_i \\ \hat{P}_{di} \end{bmatrix}$ represents the subspace of unused information in each delay channel. To remove such unused subspaces, therefore, for each delay, i, we may perform an SVD of the form

$$\begin{bmatrix} P_i \\ \hat{P}_{di} \end{bmatrix} = U \Sigma V^T$$

where $U \in \mathbb{R}^{2(n+q+r) \times 2(n+q+r)}$ and $V \in \mathbb{R}^{(n+p+m) \times (n+p+m)}$ are unitary, $\Sigma \in \mathbb{R}^{2(n+q+r) \times n+p+m}$ is rectangular and diagonal. If Σ has p_i non-zero singular values, we may therefore construct an equivalent realization

$$\begin{bmatrix} P_i \\ \hat{P}_{di} \end{bmatrix} = U_i V_i^T$$

where $U_i \in \mathbb{R}^{2(n+q+r) \times p_i}$ is the first p_i columns of $U\Sigma$ and $V_i \in \mathbb{R}^{(n+p+m) \times p_i}$ is the first p_i columns of V. Clearly, by Theorem 3, the resulting dimension of the infinite-dimensional channel i will be p_i.

IV. Minimal DDF Realization of DDFs

As illustrated using the naive conversion of NDS to DDF in Subsection II-D, there may be cases where we would like to identify a minimal DDF realization of a given DDF. In this case, we may extend the result in Theorem 3 to minimal DDF realization of a given DDF.

Theorem 4: Define

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

and let $\hat{T}_{di}$ satisfy $T_{di}(s) = Z(s)\hat{T}_{di}$ for $s \in [-\tau_i, 0]$, $i \in [K]$ and for some $Z(s)$. Suppose $U_i \in \mathbb{R}^{2n_v \times \tilde{p}_i}$ and $V_i \in \mathbb{R}^{n+m+p+n_v \times \tilde{p}_i}$ satisfy

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

Let, for $i \in [K]$,

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

Then, given u, w, x_0, the functions x, y, z, r_i, v satisfy the DDF defined by $\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, B_{r1i}, B_{r2i}, C_{ri}, C_{vi}, C_{vdi}, D_{rvi}\}$ if and only if x, y, z, v and $\tilde{r}_i$ satisfy the DDF defined by $\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, \tilde{B}_{r1i}, \tilde{B}_{r2i}, \tilde{C}_{ri}, \tilde{C}_{vi}, \tilde{C}_{vdi}, \tilde{D}_{rvi}\}$ where

$$\tilde{r}_i(t) = V_i^T \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t) \end{bmatrix}, \quad r_{i0} = V_i^T \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} \qquad i \in [K].$$

Proof: Let us prove necessity. Suppose that x, y, z, v and $\tilde{r}_i$ satisfy the DDF defined by $\{A_0, B_i, C_i, D_{ij}, B_v, D_{iv}, \tilde{B}_{r1i}, \tilde{B}_{r2i}, \tilde{C}_{ri}, \tilde{C}_{vi}, \tilde{C}_{vdi}, \tilde{D}_{rvi}\}$. Let

$$r_i(t) = \begin{bmatrix} C_{ri} & B_{r1i} & B_{r2i} & D_{rvi} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t) \end{bmatrix} \qquad i \in [K].$$

Then

$$
\begin{aligned}
v(t) &= \sum_{i=1}^{K} \tilde{C}_{vi}\tilde{r}_i(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} \tilde{C}_{vdi}(s)\tilde{r}_i(t+s)ds. \\
&= \sum_{i=1}^{K} \tilde{C}_{vi}\begin{bmatrix}\tilde{C}_{ri} & \tilde{B}_{r1i} & \tilde{B}_{r2i} & \tilde{D}_{rvi}\end{bmatrix}\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\\ v(t-\tau_i)\end{bmatrix} \\
&+ \sum_{i=1}^{K}\int_{-\tau_i}^{0} \tilde{C}_{vdi}(s)\begin{bmatrix}\tilde{C}_{ri} & \tilde{B}_{r1i} & \tilde{B}_{r2i} & \tilde{D}_{rvi}\end{bmatrix}\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\\ v(t+s)\end{bmatrix} ds. \\
&= \sum_{i=1}^{K} \tilde{C}_{vi}V_i^T\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\\ v(t-\tau_i)\end{bmatrix} + \sum_{i=1}^{K}\int_{-\tau_i}^{0} \tilde{C}_{vdi}(s)V_i^T\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\\ v(t+s)\end{bmatrix} ds. \\
&= \sum_{i=1}^{K} \begin{bmatrix}\Delta_i & \mathbf{I}_{Z,i}\end{bmatrix} U_i V_i^T \begin{bmatrix}x_{t,i}\\ w_{t,i}\\ u_{t,i}\\ v_{t,i}\end{bmatrix} \\
&= \sum_{i=1}^{K} \begin{bmatrix}\Delta_i & \mathbf{I}_{Z,i}\end{bmatrix}\begin{bmatrix}T_i\\ \hat{T}_{di}\end{bmatrix}\begin{bmatrix}x_{t,i}\\ w_{t,i}\\ u_{t,i}\\ v_{t,i}\end{bmatrix} = \sum_{i=1}^{K}\left(\Delta_i T_i + \mathbf{I}_{Z,i}\hat{T}_{di}\right)\begin{bmatrix}x_{t,i}\\ w_{t,i}\\ u_{t,i}\\ v_{t,i}\end{bmatrix} \\
&= \sum_{i=1}^{K} C_{vi}\begin{bmatrix}C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix}\begin{bmatrix}x(t-\tau_i)\\ w(t-\tau_i)\\ u(t-\tau_i)\\ v(t-\tau_i)\end{bmatrix} \\
&+ \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{vdi}(s)\begin{bmatrix}C_{ri} & B_{r1i} & B_{r2i} & D_{rvi}\end{bmatrix}\begin{bmatrix}x(t+s)\\ w(t+s)\\ u(t+s)\\ v(t+s)\end{bmatrix} ds \\
&= \sum_{i=1}^{K} C_{vi}r_i(t-\tau_i) + \sum_{i=1}^{K}\int_{-\tau_i}^{0} C_{vdi}(s)r_i(t+s)ds.
\end{aligned}
$$

Thus we conclude

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

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

as desired. Furthermore, the sewing constraint is satisfied. These steps can be reversed to obtain sufficiency, as in the proof of Theorem 3. ■

A. Eliminating Unused DDF Channels via the SVD

As in Subsection III-A, the matrices T_i and $\hat{T}_{di}(s)$ represent the ways a delayed channel can be used. To remove unused subspaces from the DDF, therefore, for each delay, i, we again perform an SVD

$$
\begin{bmatrix}T_i\\ \hat{T}_{di}\end{bmatrix} = U\Sigma V^T
$$

If Σ has $\tilde{p}_i$ non-zero singular values, we again construct the equivalent DDF realization

$$
\begin{bmatrix}T_i\\ \hat{T}_{di}\end{bmatrix} = U_i V_i^T
$$

where U_i is the first $\tilde{p}_i$ columns of $U\Sigma$ and V_i is the first $\tilde{p}_i$ columns of V. From Theorem 4, the resulting dimension of the infinite-dimensional channel i is $\tilde{p}_i$.

V. PIETOOLS Implementation

PIETOOLS is a robust and easy-to-use toolbox for converting DDEs, DDFs, and ODE-PDEs into Partial Integral Equations (PIEs). The toolbox also includes an algorithm and interface for solving Linear Partial Integral Inequalities (LPIs). For systems in PIE format, LPIs have been proposed to solve various analysis and optimal control problems - e.g. [15]–[17]. For our purposes, however, we focus on the interface for input of DDE and DDF systems. Specifically, the `PIETOOLS_DDE.m` and `PIETOOLS_DDF.m` interfaces. These interfaces do not require the user to declare all elements of the DDE or DDF representations - only those parts which are non-zero. Exploiting these interfaces, we have created an additional functionality in PIETOOLS 2020a which allows the user to convert an existing DDE or DDF representation to an equivalent minimal DDF representation. This functionality may be accessed from the `PIETOOLS_DDE.m` and `PIETOOLS_DDF.m` interfaces after declaring and initializing the DDE or DDF using the command `minimize_PIETOOLS_DDE` or `minimize_PIETOOLS_DDF`. These commands convert the given DDE or DDF to a minimal DDF realization, which can then be accessed directly or converted to a PIE for interface with one of the LPI tools. For convenience, we also include a NDS to DDF converter, which can be called using `convert_PIETOOLS_NDS2DDF`. Documentation for PIETOOLS 2020a can be found in [18] and in the PIETOOLS user manual, available at [19], or in the headers of the converter scripts.

VI. Application to DDE Network Examples

In this section, we apply the DDE to minimal DDF realization algorithm (as defined in Section III) to 2 network examples and determine the associated total dimension of the infinite-dimensional state as compared with the naive conversion formulae in Subsection II-C. The total is defined as the sum of the dimensions of every delayed channel: $d = \sum_i p_i$. The PIETOOLS functionality is used for implementation of the algorithm, as described in Section V. Computation times (IPM step in Sedumi) for solving the H_∞-optimal control problem are also listed for minimal realizations. For the non-minimal realizations, memory requirements exceed 128GB RAM and hence no computation times are available.

a) Example 1: In this example, we propose a UAV equivalent of a chain of n masses connected by springs and dampers, where the spring and damping action is delayed. In addition, the first mass is connected to a static leader and control inputs occur only at the first mass. The sensed output is the position of the final mass and the regulated output is the position of the final mass with a weighted control effort.

$$
\begin{aligned}
\dot{x}_1(t) &= \begin{bmatrix}0 & 1\\ -k & -b\end{bmatrix}(x_1(t) + x_1(t-\tau_1)) + \begin{bmatrix}0 & 0\\ k & b\end{bmatrix} x_2(t-\tau_2) + u(t) \\
\dot{x}_n(t) &= \begin{bmatrix}0 & 1\\ -k & -b\end{bmatrix} + \begin{bmatrix}0 & 0\\ k & b\end{bmatrix} x_{n-1}(t-\tau_n) + w(t) \\
\dot{x}_i(t) &= \begin{bmatrix}0 & 1\\ -2k & -2b\end{bmatrix} + \begin{bmatrix}0 & 0\\ k & b\end{bmatrix}(x_{i-1}(t-\tau_i) + x_{i+1}(t-\tau_{i+1})) \\
y(t) &= x_n(t), \qquad z(t) = x_n(t) + .1u(t)
\end{aligned}
$$

where $k = 1$, $b = .2$, and $\tau_i = h \cdot i$ with $h = .2$. Note for $n = 5$, the system is unstable for $h > .09$. The total dimension of the delayed channels with and without minimal realizations are listed in Table I. We also include associated computation times for solving the H_∞-optimal state feedback problem.

b) Example 2: Example 2 is a network of showering users. For brevity we refer to [10] for the definition of this model. As for Example 1, the results are listed in Table I.

VII. Application to NDS Examples

We now repeat the analysis in Section VI, but applied to minimal DDF realizations of NDSs. Specifically, we first construct the naive DDF representation proposed in Subsection II-D and then apply the algorithm defined in Section IV to find a minimal DDF representation of this DDF. We apply this to 2 NDSs found in the literature and again compare the associated total dimension of the infinite-dimensional state with the dimension of the DDF from the naive conversion formulae in Subsection II-D. Again, the PIETOOLS functionality is used, as described in Section V and computation times for solving the stability analysis problem are also listed for both minimal and non-minimal realizations.

a) Example 3: This example is taken from [20]. Total dimensions and computation times (stable for $\tau_1 < 2.04$, $\tau_2 = 3\tau_1$) are listed in Table I.

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

b) Example 4: This example problem recently appeared in [21]. Total dimensions and computation times (stable for $\tau_1 < .603$) are listed in Table I.

$$\begin{aligned} \dot{x}(t) = &\begin{bmatrix} -2 & .2 & -.3 & 0 & -.4 \\ .2 & -3.8 & 0 & .7 & 0 \\ .8 & 0 & -1.6 & 0 & 0 \\ 0 & .8 & -.6 & -2 & .3 \\ -1 & -.1 & -1.5 & 0 & -1.8 \end{bmatrix} x(t) \\ &+ \begin{bmatrix} -2.2 & 0 & 0 & 1 & 0 \\ 1.6 & -2.2 & 1.6 & 0 & 0 \\ -0.2 & -0.2 & -0.2 & -0.2 & -0.2 \\ 0 & 0.4 & -1.4 & -3.4 & 1 \\ -0.2 & 0.4 & -0.1 & -1.1 & -3.3 \end{bmatrix} x(t-\tau) \\ &+ \begin{bmatrix} 0.40888 & 0.00888 & 0.20888 & -0.09112 & -0.29112 \\ 0 & 0.2 & 0 & 0 & 0.6 \\ -0.1 & -0.4 & 0 & -0.8 & 0 \\ 0 & 0 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & -0.1 \end{bmatrix} \dot{x}(t-\tau) \end{aligned}$$

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

TABLE I

The total dimension of delayed channels $\sum_i p_i$ and computation times for nominal and minimal realizations. Computation times are H_∞-control for Exs. 1 and 2 and stability analysis for Exs. 3 and 4.

VIII. Conclusion

In this paper, we have presented an algorithm for constructing minimal DDF realizations of both DDFs and DDEs, along with an efficient PIETOOLS implementation of these algorithms. The significance of these results lies in the ability to rapidly generate efficient representations of large networks and delayed systems without the hand-crafting associated with application-specific identification of low-dimensional delayed channels. These efficient representations often result in dramatic reductions of the computational complexity of algorithms for simulation, analysis and control of DDEs and DDFs. Specifically, the minimal DDF realizations may ultimately be used for reduction to lower-dimensional ODEs using, e.g. Padé type approximation, or may be used directly for analysis and control via such infinite-dimensional techniques as the Partial Integral Equation (PIE) framework.

References

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

[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] F. Mazenc, H. Ito, and P. Pepe, "Construction of lyapunov functionals for coupled differential and continuous time difference equations," in *Proceedings of the IEEE Conference on Decision and Control*, 2013.

[5] P. Pepe, I. Karafyllis, and Z.-P. Jiang, "On the Liapunov–Krasovskii methodology for the ISS of systems described by coupled delay differential and difference equations," *Automatica*, vol. 44, no. 9, 2008.

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

[7] 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.

[8] I. Karafyllis and M. Krstic, "On the relation of delay equations to first-order hyperbolic partial differential equations," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 20, no. 3, 2014.

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

[10] M. Peet, "Representation of networks and systems with delay: DDEs, DDFs, ODE-PDEs and PIEs," *Automatica*, 2020, to Appear.

[11] B. D. Schutter, "Minimal state-space realization in linear system theory: an overview," *Journal of computational and applied mathematics*, vol. 121, no. 1-2, pp. 331–354, 2000.

[12] V. Kolmanovskii and A. Myshkis, *Introduction to the theory and applications of functional differential equations*. Kluwer Academic Publishers, 1999.

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

[14] M. Gil', *Stability of finite and infinite dimensional systems*. Kluwer, 2012.

[15] S. Shivakumar, A. Das, S. Weiland, and M. Peet, "Duality and H_∞-optimal control of ODE-PDE systems," in *Proceedings of the IEEE Conference on Decision and Control*, 2020, arXiv:2004.03638.

[16] 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.

[17] 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.

[18] 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.

[19] M. Peet, S. Shivakumar, and A. Das, "PIETOOLS," https://control.asu.edu/pietools, July 2020.

[20] N. Zhao, X. Zhang, Y. Xue, and P. Shi, "Necessary conditions for exponential stability of linear neutral type systems with multiple time delays," *Journal of the Franklin Institute*, vol. 355, no. 1, pp. 458–473, 2018.

[21] A. Ramirez, D. Breda, and R. Sipahi, "A scalable approach to compute delay margin of a class of neutral-type time delay systems," *SIAM Journal on Control and Optimization*, to Appear.