Automatica 125 (2021) 109473

Contents lists available at ScienceDirect

Automatica

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

A Partial Integral Equation (PIE) representation of coupled linear PDEs and scalable stability analysis using LMIs ☆

Matthew M. Peet

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

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form 22 September 2020
Accepted 14 December 2020
Available online 20 January 2021

Keywords:
PDEs
PIEs
LMIs
Lyapunov stability

ABSTRACT

We present a new Partial Integral Equation (PIE) representation of Partial Differential Equations (PDEs) in which it is possible to use convex optimization to perform stability analysis with little or no conservatism. The first result gives a standardized representation for coupled linear PDEs in a single spatial variable and shows that any such PDE, suitably well-posed, admits an equivalent PIE representation, defined by the given conversion formulae. This leads to a new prima facie representation of the dynamics without the implicit constraints on system state imposed by boundary conditions. The second result is to show that for systems in this PIE representation, convex optimization may be used to verify stability without discretization. The resulting algorithms are implemented in the Matlab toolbox PIETOOLS, tested on several illustrative examples, compared with previous results, and the code has been posted on Code Ocean. Scalability testing indicates the algorithm can analyze systems of up to 40 coupled PDEs on a desktop computer.

1. Introduction

Partial Differential Equations (PDEs) are used to model systems where the state depends continuously on both time and secondary independent variables. Common examples of such secondary dependence include space; as in flexible structures (Bernoulli–Euler beams) and fluid flow (Navier–Stokes); or maturation, as in cell populations and predator–prey dynamics.

The most common method for computational analysis of PDEs is to project the infinite-dimensional state onto a finite-dimensional vector space using, e.g. Marion and Temam (1989), Ravindran (2000) and Rowley (2005) and to use the extensive literature on control of Ordinary Differential Equations (ODEs) to test stability of, and design controllers for, the resulting finite-dimensional system. However, such discretization approaches are prone to instability (e.g. in the case of hyperbolic balance laws Karafyllis & Krstic, 2019), numerical ill-conditioning, and large-dimensional state–spaces. Furthermore, representation of PDEs using ODEs inevitably neglects higher-order modes, modes which can be inadvertently excited by feedback control via the well-known spillover effect (Balas, 1978).

Work on *computational* methods for analysis and control of PDEs which does not rely on discretization has been more limited. Perhaps the most well-known computational method for stabilization of PDEs without discretization is the backstepping approach to controller synthesis (Aamo, 2013; Krstic & Smyshlyaev, 2008; Smyshlyaev & Krstic, 2005). This approach is not optimization-based, however, and not typically used for stability analysis (An exception being Saba, Argomedo, Auriol, Loreto, & Meglio, 2019). Recently, there has been some work on the use of Linear Matrix Inequalities (LMIs) to find Lyapunov functions for linear and nonlinear PDEs — See Fridman and Orlov (2009), Fridman and Terushkin (2016), Gaye et al. (2013) and Solomon and Fridman (2015). However, because most of these works focus on the nonlinear case, the Lyapunov functions proposed therein are relatively simple and the resulting stability conditions conservative. An extension of the IQC framework to PDEs can be found in Barreau, Scherer, Gouaisbaut, and Seuret (2020).

Numerous analytic (non-computational) methods have been proposed over the years for analysis of PDEs, including the well-developed literature on Semigroup theory (Bastin & Coron, 2016; Bensoussan, Prato, Delfour, & Mitter, 1992; Curtain & Zwart, 1995; Karafyllis & Krstic, 2019; Lasiecka & Triggiani, 2000; Luo, Guo, & Morgül, 2012) and the literature on Port-Hamiltonian systems (Villegas, 2007) for selecting boundary inputs. However, these methods are typically ad-hoc — relying on the expertise of the user to propose and test energy metrics.

Recently, Sum-of-Squares (SOS) has been used for analysis and control of PDEs and examples can be found in Gahlawat and Peet

☆ This work was supported by the National Science Foundation, USA grants # CNS-1739990, CMMI-1935453, and the Office of Naval Research, USA grant #N000014-17-1-2117. The material in this paper was partially presented at the 11th IFAC Workshop on Control of Systems Governed by Partial Differential Equation CDPS, May 20–24, 2019, Oaxaca, Mexico. This paper was recommended for publication in revised form by Associate Editor Thomas Meurer under the direction of Editor Miroslav Krstic.

E-mail address: mpeet@asu.edu.

https://doi.org/10.1016/j.automatica.2020.109473
0005-1098/

(2015, 2016, 2017), Safi, Baudouin, and Seuret (2017) and Ahmadi, Valmorbida, Gayme, and Papachristodoulou (2019), Ahmadi, Valmorbida, and Papachristodoulou (2016), Gahlawat and Valmorbida (2017), Valmorbida, Ahmadi, and Papachristodoulou (2014, 2016). While these SOS-based works are relatively accurate, they: (1) Are mostly limited to scalar PDEs; (2) Suffer from high computational complexity; (3) Are mostly ad-hoc, requiring significant effort to extend the results to new PDEs. For example, these methods have never been able to analyze stability of beam or wave equations. The source of the difficulty in using LMIs and SOS for stability analysis of PDEs is that the solution of a PDE is required to satisfy three sets of constraints: the differential equation; the boundary conditions; and continuity constraints. This is in contrast to ODEs, which are defined by bounded linear operators (matrices) and solutions to which need only satisfy a single differential equation.

The goal of the paper is to create, for PDEs, an equivalent of the LMI framework developed for ODEs. Historically, PDEs (as old as Newton) are defined by two conflicting sets of equations: the PDE itself, which moves the state; and the Boundary Conditions (BCs), which implicitly constrain the motion of the state. We want to unify these conflicting constraints into a new state–space representation of PDEs, defined by an algebra of bounded linear operators, and which directly incorporates: the PDE, the BCs, and the continuity constraints — thereby eliminating issues of well-posedness and obviating the need to account for implicit constraints on the state.

This approach is fundamentally different than previous work using SOS or LMI-based methods. These previous efforts used SOS or positive matrices to propose candidate Lyapunov functions and then attempted to integrate the effect of boundary conditions into the derivative using, e.g. integration by parts. By contrast, our approach is to integrate the effect of boundary conditions into the *dynamics* — thereby obviating the need to account for them in the stability analysis. As a result, our algorithms have no obvious source of conservatism and scale to systems of up to 40 coupled PDEs.

Approach: In this paper, we propose the Partial Integral Equation (PIE) representation of PDEs. PIEs are infinite-dimensional state–space systems of the form

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t),$$
$$\mathbf{x}(0) = \mathbf{x}_0 \in L_2^n[a,b], \tag{1}$$

where the state, $\mathbf{x}(t)$ is in $L_2^n[a,b]$, and the system parameters $(\mathcal{T},\mathcal{A})$ are Partial Integral (3-PI) operators. 3-PI refers to the 3 matrix-valued parameters, $\{N_0, N_1, N_2\}$ which define every such operator $\mathcal{P}_{\{N_i\}} : L_2[a,b] \to L_2[a,b]$ as

$$\left(\mathcal{P}_{\{N_0,N_1,N_2\}}\mathbf{x}\right)(s) := N_0(s)x(s)ds + \int_a^s N_1(s,\theta)x(\theta)d\theta + \int_s^b N_2(s,\theta)x(\theta)d\theta.$$

As shown in Section 4, all 3-PI operators are L_2-bounded and together, they form an algebra (closed under addition, composition, scalar multiplication). Because they are algebraic, 3-PI operators inherit many of the properties of matrices and there is now a Matlab toolbox, PIETOOLS (using SOSTOOLS as a model), which allows for manipulation of 3-PI operators using matrix syntax and which can solve Linear PI Inequality (LPI) constrained optimization problems using the YALMIP syntax for LMIs.

Converting PDE state to PIE state: The first contribution of the paper (extending the results in Peet, 2018a) is to show that the solutions to a broad class of PDEs can be represented using PIEs. For this result, we propose an alternative state–space. To explain this change of state, we consider the following standardized representation of coupled linear PDEs in a single spatial variable, presented in Section 3.

$$\begin{bmatrix}\dot{x}_0(t,s)\\ \dot{x}_1(t,s)\\ \dot{x}_2(t,s)\end{bmatrix} = A_0(s)\underbrace{\begin{bmatrix}x_0(t,s)\\ x_1(t,s)\\ x_2(t,s)\end{bmatrix}}_{\mathbf{x}\in X} + A_1(s)\begin{bmatrix}x_1(t,s)\\ x_2(t,s)\end{bmatrix}_s + A_2(s)\left[x_2(t,s)\right]_{ss}$$

with associated state–space

$$X = \left\{ \begin{bmatrix}x_0\\ x_1\\ x_2\end{bmatrix} \in L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2} \;:\; B\begin{bmatrix}x_1(a)\\ x_1(b)\\ x_2(a)\\ x_2(b)\\ x_{2s}(a)\\ x_{2s}(b)\end{bmatrix} = 0 \right\}.$$

Most 1D PDEs can be formulated using this standardized representation.

For example, consider the damped wave equation

$$\ddot{u}(t,s) = u_{ss}(t,s) - 2a\dot{u}(t,s) - a^2u(t,s), \; s \in [0,1]$$
$$u(t,0) = 0, \qquad u_s(t,1) = -k\dot{u}(t,1).$$

Then, setting $u_1 = \dot{u}$ and $u_2 = u$, we have

$$\begin{bmatrix}\dot{u}_1(t,s)\\ \dot{u}_2(t,s)\end{bmatrix} = \underbrace{\begin{bmatrix}-2a & -a^2\\ 1 & 0\end{bmatrix}}_{A_0}\begin{bmatrix}u_1(t,s)\\ u_2(t,s)\end{bmatrix} + \underbrace{\begin{bmatrix}1\\ 0\end{bmatrix}}_{A_2}u_{2ss}(t,s)$$

with BCs $u_2(0) = 0$, $u_1(0) = 0$, and $u_{2s}(1) = -ku_1(1)$ so that

$$X = \left\{ \begin{bmatrix}u_1\\ u_2\end{bmatrix} \in H_1^1 \times H_2^1 : \underbrace{\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0\\ 0 & k & 0 & 0 & 0 & 1\end{bmatrix}}_{B}\begin{bmatrix}u_1(0)\\ u_1(1)\\ u_2(0)\\ u_2(1)\\ u_{2s}(0)\\ u_{2s}(1)\end{bmatrix} = 0 \right\}.$$

The fundamental state–space A reasonable definition of state is the minimal amount of information needed to forward-propagate the solution. By this measure, and referring to the example above, defining the state of a PDE as $u(t) \in H_1 \times H_2$ is not minimal, as this function contains redundant information regarding the boundary values. We propose, then, that for a PDE defined by A_i and X, the correct definition of state is the so-called *fundamental state*, where for $\mathbf{x} \in X$, we define

$$\mathbf{x}_f = \begin{bmatrix}x_0\\ x_{1s}\\ x_{2ss}\end{bmatrix} = \begin{bmatrix}I & & \\ & \partial_s & \\ & & \partial_s^2\end{bmatrix}\mathbf{x} \in L_2^{n_0+n_1+n_2}.$$

In Section 6, we show that if the PDE is suitably well-posed, there exists a *unitary* 3-PI operator $\mathcal{T} : L_2^{n_0+n_1+n_2} \to X$ such that $\mathbf{x} = \mathcal{T}\mathbf{x}_f$.

Equivalence of PIE and PDE: In Section 7, equipped with the unitary operator, $\mathcal{T}$, we propose a 3-PI operator $\mathcal{A}$ such that for any solution $\mathbf{x}_f(t) \in L_2$ of the PIE in Eq. (1), $\mathbf{x}(t) = \mathcal{T}\mathbf{x}_f(t)$ satisfies the PDE defined by A_i, X and that, conversely, for any solution of the PDE, $\mathbf{x}$, the fundamental state $\mathbf{x}_f(t) = \mathrm{diag}(I, \partial_s, \partial_s^2)\mathbf{x}(t)$ satisfies the PIE. We further show that exponential stability of the PIE and PDE are equivalent.

A Linear PI Inequality (LPI) for stability: Aside from having a minimal state–space, the advantage of the PIE representation of a PDE is computational. Recall that our goal is to create a framework akin to LMIs which can be used to study PDEs. Because PIEs do not have BCs, all information needed to define the solution is contained in the fundamental state, $\mathbf{x}_f \in L_2$. Therefore, unlike PDEs, where the effect of the BCs needs to be "brought in" using integration by parts, Poincare inequalities, etc., PIEs are a prima facie representation of the dynamics. This allows us to pose the stability test for PIEs as a convex optimization problem of the following form:

$$\text{Find}_{\mathcal{P}=\mathcal{P}_{\{N_i\}}} : \mathcal{P} \geq \epsilon I,$$
$$\mathcal{T}^*\mathcal{P}\mathcal{A} + \mathcal{A}^*\mathcal{P}\mathcal{T} \leq -\delta\mathcal{T}^*\mathcal{T}.$$

This LPI, then, is a straightforward generalization of the Lyapunov LMI: $P > 0$, $A^TP + PA < 0$. Furthermore, again motivated by the efficient Matlab parser YALMIP (Lofberg, 2004), we have recently developed the Matlab toolbox PIETOOLS (Shivakumar, Das, & Peet, 2020) allows us to manipulate 3-PI operators using matrix syntax and solve Linear PI Inequality (LPI) constrained optimization problems. Thus, if $\{\mathcal{T}, \mathcal{A}\}$ are as defined in Section 7, then our stability test reduces to 3 lines of Matlab code

```
[X,P] = poslpivar(X,n,I,d);
D = -epneg*T'*T-A'*P*T-T'*P*A;
X = lpi_ineq(X,D);
```

where the functions `poslpivar` and `lpi_ineq` are defined in Section 9.3 and in Shivakumar et al. (2020). We emphasize that this code applies to any PDE in standardized format and since there is no need to bring in the boundary conditions, there is no obvious source of conservatism. Specifically, in Section 11, we apply the algorithm to beam and wave equations for which there are no previous LMI-based stability conditions. Furthermore, the lack of conservatism is verified in Section 10 by comparing against known stability margins taken from the literature.

Finally, computational complexity depends on the degree of the polynomial parameters in the 3-PI variable $\mathcal{P}$ (corresponding to the complexity of the candidate Lyapunov function). However, most problems only require very simple Lyapunov functions. In this case, the scalability of the method is comparable to the complexity of discretization-based analysis. Specifically, if we choose the polynomial parameters to have degree 2, then the algorithm can analyze stability of more than 40 coupled PDEs on a desktop computer.

In the following two illustrations, we attempt to further introduce our BC free PIE representation by: (1) showing that the BCs *cannot* be ignored by convex optimization algorithms and (2) showing that the PDE can be reformulated in a way where the BCs *can* be ignored.

1.1. BCs cannot be ignored

The goal of the paper is to find a representation of PDEs which can be interpreted as we would interpret an ODE — without implicit constraints on the solution imposed by BCs or continuity. Such a representation, then, allows us to use convex optimization tools to analyze the system and, in particular, to prove stability. In this context, let us consider what happens when we try to use convex optimization to study the PDE, and in particular, let us illustrate what happens when that optimization tool treats the PDE like an ODE — i.e. *without* considering continuity constraints or boundary conditions. To this end, suppose we are given a PDE, parameterized by A_0, A_1, A_2 as follows.

$$\dot{\mathbf{x}}(t,s) = A_0(s)\mathbf{x}(t,s) + A_1(s)\mathbf{x}_s(t,s) + A_2(s)\mathbf{x}_{ss}(t,s)$$

An obvious class of candidate Lyapunov functions for this system is parameterized by M as

$$V(\mathbf{x}) = \left\langle \mathbf{x}, \mathcal{P}_{\{M,0,0\}}\mathbf{x}\right\rangle_{L_2} = \int_a^b \mathbf{x}(s)^T M(s)\mathbf{x}(s)ds.$$

As would be the case for an ODE, $V(\mathbf{x}) \geq \epsilon \left\|\mathbf{x}\right\|^2$ if $M(s) \geq \epsilon I$ for all s and some $\epsilon > 0$ — a constraint which is easy to enforce using, e.g. SOS. However, if we now take the derivative of this candidate Lyapunov function we obtain

$$\dot{V}(\mathbf{x}) = \int_a^b \begin{bmatrix}\mathbf{x}(s)\\ \mathbf{x}_s(s)\\ \mathbf{x}_{ss}(s)\end{bmatrix}^T D(s) \begin{bmatrix}\mathbf{x}(s)\\ \mathbf{x}_s(s)\\ \mathbf{x}_{ss}(s)\end{bmatrix} ds$$

$$D(s) := \begin{bmatrix} A_0(s)^T M(s) + M(s)A_0(s) & M(s)A_1(s) & M(s)A_2(s)\\ A_1(s)^T M(s) & 0 & 0\\ A_2(s)^T M(s) & 0 & 0\end{bmatrix}.$$

Now, if we were to propose a convex optimization algorithm which treats the system like an ODE (without considering the BCs and continuity constraints), we would constrain $D(s) \leq 0$ and this would imply stability. Unfortunately, however, $D(s) \not\leq 0$ for ANY choice of $M, A_1, A_2 \neq 0$. The problem, of course, is that the differentiation operator branches $\mathbf{x}$ into $\mathbf{x}_s$ and $\mathbf{x}_{ss}$, neither of which are independent of $\mathbf{x}$. Moreover, the information which determines the relationship between $\mathbf{x}$, $\mathbf{x}_s$ and $\mathbf{x}_{ss}$ is not embedded in the differential equation. Rather, this information is implicit in the BCs and continuity constraints. At this point, of course, an intelligent user would "bring in" the boundary and continuity properties to obtain a new stability condition using, e.g. integration by parts or Stokes Theorem. However, such secondary steps are not easily incorporated in a convex optimization algorithm. The goal of this paper, then, is to find a prima facie representation of the PDE with no implicit constraints on continuity and BCs, and wherein we may develop a computational framework which mirrors that of LMIs for ODEs.

1.2. BCs can be moved into the dynamics

Having argued that BCs cannot be ignored, let us show that the BCs can be brought into the dynamics in a way which may be more suitable for tools based on convex optimization. Specifically, in this subsection, we consider an example of how the incorporation of BCs and continuity can take a system (wherein the dynamics do not explicitly depend on the partial derivatives), and transform it into a system with explicit dependence on the partial derivatives and inputs. Consider the following non-partial-differential, yet distributed-parameter system.

$$\dot{\mathbf{u}}(t,s) = \mathbf{u}(t,s), \;\; \mathbf{u}(t,0) = w_1(t), \;\; \mathbf{u}_s(t,0) = w_2(t).$$

To allow for the specified BCs, we restrict continuity of $\mathbf{u}$ as $\mathbf{u}(t) \in H_2^1$. Note that if we ignore the BCs and continuity constraints, the system does not appear to be a PDE since the dynamics are identical at every point in the domain. However, if we now combine the fundamental theorem of calculus with integration by parts, we obtain a very different, yet equivalent, set of dynamics — dynamics with explicit dependence on the partial derivatives.

$$\dot{\mathbf{u}}(t,s) = s w_1(t) + w_2(t) + \int_0^s (s-\eta)\mathbf{u}_{\eta\eta}(t,\eta)d\eta$$

This formulation of the same system directly incorporates BCs and continuity into the dynamics — which are now expressed using the partial derivative $\mathbf{u}_{ss}$. In addition, while the original formulation was spatially decoupled, with w_1, w_2 only acting at the boundary, the new formulation shows that the exogenous inputs w_1, w_2 are felt instantaneously at every point in the domain. In contrast to the original representation of the dynamics, the integral formulation of the same system is more suitable for convex optimization since the effect of the BCs and continuity is explicitly included in the dynamics (and hence the BCs and continuity can now be ignored by the algorithm).

The first goal of this paper, then, is: (1) to show that a broad class of PDEs can be reformulated in a way which specifies precisely how the BCs affect the dynamics (the PIE representation) and (2) to provide universal formulae for constructing such a representation.

2. Notation

We define $L_2[a,b]^n$ to be the space of $\mathbb{R}^n$-valued Lebesgue integrable functions defined on $[a,b]$ and equipped with the standard inner product. We use $W^{k,p}[a,b]^n$ to denote the Sobolev subspace of $L_p[a,b]^n$ defined as $\{u \in L_p[a,b]^n \;:\; \frac{\partial^q}{\partial x^q} u \in L_p \text{ for all } q \le k\}$. $H_k[a,b] := W^{k,2}[a,b]$ and $H_k^n[a,b] = \prod_{i=1}^n H_k[a,b]$. For efficiency, we typically omit the domain, so that, e.g. $H_k^n := H_k^n[a,b]$ unless otherwise stated. $I_n \in \mathbb{R}^{n\times n}$ and $0_{n_1\times n_2} \in \mathbb{R}^{n_1\times n_2}$ are used to denote the identity and zero matrices and the subscripts are omitted when the dimension of the matrices is clear from context. $\mathbf{I}$ denotes the indicator function $\mathbf{I} : \mathbb{R} \to \{0,1\}$, defined as

$$\mathbf{I}(s) = \begin{cases} 1, & \text{if } s > 0 \\ 0, & \text{otherwise.} \end{cases}$$

3. A standardized PDE representation

The two primary contributions of this paper are: a formula for conversion of PDEs to PIEs; and an LPI framework for Lyapunov stability analysis of PIEs (Section 8). The significance of the latter result clearly depends on the set of PDEs which can be converted to PIEs. In this section, we propose a standardized framework for representation of PDEs. In Section 7, we will show that for any such PDE, there exists a PIE for which a solution to the PIE yields a solution to the PDE and vice-versa. The class of PDEs considered here is not exhaustive, however. That is, there exist PDEs not listed here which may be converted to PIEs. Furthermore, there exist PIEs which do not have a coupled PDE representation.

We consider coupled PDEs of the form

$$\begin{bmatrix}\dot{x}_0(t,s)\\ \dot{x}_1(t,s)\\ \dot{x}_2(t,s)\end{bmatrix} = A_0(s)\begin{bmatrix}x_0(t,s)\\ x_1(t,s)\\ x_2(t,s)\end{bmatrix} + A_1(s)\begin{bmatrix}\partial_s x_1(t,s)\\ \partial_s x_2(t,s)\end{bmatrix} + A_2(s)\left[\partial_s^2 x_2(t,s)\right] \tag{2}$$

and with solution restricted to the domain

$$X := \left\{ \begin{bmatrix}x_0\\ x_1\\ x_2\end{bmatrix} \in L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2} \;:\; B\begin{bmatrix}x_1(a)\\ x_1(b)\\ x_2(a)\\ x_2(b)\\ x_{2s}(a)\\ x_{2s}(b)\end{bmatrix} = 0 \right\} \tag{3}$$

where

$$B\begin{bmatrix} I_{n_1} 0 & 0 & \\ I_{n_1} & 0 & 0\\ 0 & I_{n_2} & 0\\ 0 & I_{n_2} & (b-a)I_{n_2}\\ 0 & 0 & I_{n_2}\\ 0 & 0 & I_{n_2}\end{bmatrix} \quad \text{is invertible.} \tag{4}$$

Specifically, given $\mathbf{x}_0 \in X$, we say that $\mathbf{x}$ satisfies the PDE defined by $\{A_i, X\}$ if $\mathbf{x}$ is Fréchet differentiable, $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}(t) \in X$ and Eq. (2) is satisfied for all $t \ge 0$.

3.1. A guide to partition of states

The partition of states in Eq. (2) is not overly restrictive and there is no special structure to this generator. The partition is purely organizational and defined by the domain restriction $\mathbf{x} \in L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2}$. States with no restrictions on continuity are assigned to be an element of the vector $x_0(t) \in L_2[a,b]^{n_0}$. If a state has no continuity properties, then these states cannot be differentiated and it is not possible to assign boundary conditions, as the limits $x_0(a), x_0(b)$ do not exist. States which are continuous, but not continuously differentiable are assigned to $x_1(t) \in H_1^{n_1}$. The continuity property of these states allow for boundary conditions, as $x_1(a), x_1(b)$ exist. However, since $\partial_s x_1 \in L_2[a,b]^{n_1}$ is not continuous, we cannot assign boundary conditions which involve $x_{1s}(a)$ or $x_{1s}(b)$, as these limits do not exist. Finally, states which are required to be continuously differentiable are assigned to $x_2(t) \in H_2^{n_2}$ and admit boundary conditions involving $x_{2s}(a)$ or $x_{2s}(b)$ and second-order spatial derivatives, $\partial_s^2 x_2$. This standardized representation specifically excludes states in H_k^n where $k > 2$. Although the results of the paper can be extended to such systems, such an extension is not considered here.

3.2. A guide to boundary conditions

In this subsection, we propose restrictions on the matrix B which are equivalent to Eq. (4). Specifically, we require that the row rank of B must be $n_1 + 2n_2$ and that B contains no boundary conditions of a given prohibited form. Note that the rank condition on B is not overly restrictive as, to the best of our knowledge, it is a necessary condition for existence of a unique solution for any PDE in standardized form.

3.2.1. Prohibited boundary conditions

A necessary and sufficient condition for B to satisfy Eq. (4) is for $B \in \mathbb{R}^{(n_1+2n_2)\times(2n_1+4n_2)}$ to have row rank $n_1 + 2n_2$ and to define no boundary conditions consisting of a linear combination of $x_1(a) - x_1(b) = 0$, $x_2(a) + (b-a)x_{2s}(a) - x_2(b) = 0$, or $x_{2s}(a) - x_{2s}(b) = 0$.

Lemma 1. *Suppose $B \in \mathbb{R}^{(n_1+2n_2)\times(2n_1+4n_2)}$. Define*

$$T^\perp := \begin{bmatrix} I_{n_1} & -I_{n_1} & 0 & 0 & 0 & 0\\ 0 & 0 & I_{n_2} & -I_{n_2} & I_{n_2}(b-a) & 0\\ 0 & 0 & 0 & 0 & I_{n_2} & -I_{n_2}\end{bmatrix}.$$

Eq. (4) is satisfied if and only if $B \in \mathbb{R}^{(n_1+2n_2)\times(2n_1+4n_2)}$, has row rank $n_1 + 2n_2$ and the row space of B and the row space of $T^\perp$ has trivial intersection.

Proof. Let the T matrix from Eq. (4) be as defined in Eq. (8). Now suppose BT is invertible. Since $T \in \mathbb{R}^{(2n_1+4n_2)\times(n_1+2n_2)}$, we require $B \in \mathbb{R}^{(n_1+2n_2)\times(2n_1+4n_2)}$ in order for BT to exist and be square. Since $BT \in \mathbb{R}^{(n_1+2n_2)\times(n_1+2n_2)}$, B must also have row rank $n_1 + 2n_2$. Now, since T has column rank $n_1 + 2n_2$, its row rank is also $n_1 + 2n_2$. Now $T^\perp$ has row rank $n_1 + 2n_2$ and $T^\perp T = 0$. Thus the row space of $T^\perp$ lies in $Im(T)^\perp$ and since $Im(T)^\perp$ is of dimension $n_1 + 2n_2$, the row space of $T^\perp$ is $Im(T)^\perp$. Therefore $x^T BT = 0$ for some $x \ne 0$, if and only if the intersection of the row space of B and that of $T^\perp$ is non-trivial. This establishes necessity. For sufficiency, we assume $B \in \mathbb{R}^{(n_1+2n_2)\times(2n_1+4n_2)}$ and has row rank $n_1 + 2n_2$. As shown, the row space of B has trivial intersection with $Im(T)^\perp$. Again, we have that $x^T BT = 0$ implies $x = 0$, from which we conclude invertibility. □

Note 1. The restriction on prohibited boundary condition is subtle. For example, $x_2(a) = x_2(b)$ is permitted, except if combined with $x_{2s}(a) = 0$ (combining with $x_{2s}(b) = 0$ is still OK). Meanwhile, $x_1(a) = x_1(b)$ and $x_{2s}(a) = x_{2s}(b)$ are never OK. Additionally, $x_{2s}(a) = 0$ is OK, unless combined with $x_2(a) = x_2(b)$ or $x_{2s}(b) = 0$. Of course, the most reliable way to check if certain boundary conditions are permitted is to simply construct B and check the rank of BT. The PIETOOLS implementation described in Section 9.3 will do this automatically and generate an error if BT is not invertible.

3.2.2. A note on necessity of Eq. (4)

Boundary conditions of the form $x_1(a) = x_1(b)$ are periodic and imply

$$\int_a^b x_{1s}(s)ds = 0.$$

Likewise $x_{2s}(a) = x_{2s}(b)$ implies

$$\int_a^b x_{2ss}(s)ds = 0,$$

and $x_2(a) + (b-a)x_{2s}(a) = x_2(b)$ implies

$$\int_a^b \int_a^s x_{2\eta\eta}(\eta)d\eta ds = 0.$$

In this way, the prohibited BCs represent integral constraints on the respective PIE (fundamental) states, $x_{1s} \in L_2$ and $x_{2ss} \in L_2$, meaning these PIE states are not minimal (dynamics expressed using these states have implicit constraints). One option in these cases may be to redefine the PIE states modulo an integral constraint, however this extension is left for future work.

3.3. Euler–Bernoulli beam example

In order to better understand how to write a PDE in the standardized PDE form of Eq. (2), let us consider the cantilevered Euler–Bernoulli beam:

$$\ddot{u}(t,s) = -cu_{ssss}(t,s), \qquad \text{where}$$
$$u(0) = u_s(0) = u_{ss}(L) = u_{sss}(L) = 0.$$

We wish to construct a standardized PDE representation of this classic diffusive model. Following the approach in, e.g. Villegas (2007) (from which we also get the Timoshenko beam model in Section 11), we first introduce the augmented state $u_1 = \dot{u}$ - a choice which leads to "natural" BCs for which the system is well-posed (Luo et al., 2012). This choice also eliminates the second order time-derivative, ü. Since $u \in H_4$, we eliminate the fourth-order spatial derivative by creating the augmented state $u_2 = u_{ss}$. Taking the time-derivative of these states, we obtain

$$\dot{u}_1 = \ddot{u} = -cu_{ssss} = -cu_{2ss}$$
$$\dot{u}_2 = \partial_t \partial_s^2 u = \partial_s^2 \dot{u} = u_{1ss}.$$

These equations are now in the standardized form

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0 & -c \\ 1 & 0 \end{bmatrix}}_{A_2} \mathbf{x}_{ss}(t)$$

where $A_0 = A_1 = 0$, $n_2 = 2$, and $n_0 = n_1 = 0$. We now examine the boundary conditions using these states:

$$u_{ss}(L) = u_2(L) = 0 \quad \text{and} \quad u_{sss}(L) = u_{2s}(L) = 0.$$

These boundary conditions are insufficient, as the resulting rank is 2. However, we may include the "natural" BCs by differentiating in time to obtain

$$\dot{u}(0) = u_1(0) = 0 \quad \text{and} \quad \dot{u}_s(0) = u_{1s}(0) = 0.$$

We now have 4 boundary conditions, which we use to construct the B matrix as

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{B} \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_1(L) \\ u_2(L) \\ u_{1s}(0) \\ u_{2s}(0) \\ u_{1s}(L) \\ u_{2s}(L) \end{bmatrix} = 0.$$

The B matrix is now of rank $4 = n_1 + 2n_2$ and satisfies Eq. (4).

Now, if u satisfies the E-B beam equation for some initial condition, we have that $u_1 = \dot{u}$, $u_2 = u_{ss}$ satisfy the standardized PDE model. However, conversely, if u_1, u_2 satisfy the standardized PDE for some initial condition, then in order to construct a solution to the original PDE, we must integrate u_1 in time. However, this requires knowledge of $u(0)$. Thus we find that some information on the system solution has been lost in the standardized representation. Note, however, that we could retain this information by including a third state, $u_3 = u$, so that $\dot{u}_3 = u_1$ and then the solutions would be equivalent.

3.4. Exponential stability of coupled PDE systems

In this subsection, we define two notions of exponential stability with respect to the standardized PDE representation — stability in the X norm and stability in the L_2 norm. In Section 5, we will define the notion of exponential stability for PIEs. In Section 8, we will show that exponential stability of a PIE representation of a standardized PDE is equivalent to exponential stability of the original standardized PDE in the X norm.

Definition 2. We say the PDE defined by $\{A_i, X\}$ is exponentially stable in X if there exist constants $K, \gamma > 0$ such that for any $\mathbf{x}_0 \in X$, any solution $\mathbf{x}$ of the PDE defined by $\{A_i, X\}$ satisfies

$$\|\mathbf{x}(t)\|_{L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2}} \le K \|\mathbf{x}_0\|_{L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2}} e^{-\gamma t}.$$

Definition 3. We say the PDE defined by $\{A_i, X\}$ is exponentially stable in L_2 if there exist constants $K, \gamma > 0$ such that for any $\mathbf{x}_0 \in X$, any solution $\mathbf{x}$ of the PDE defined by $\{A_i, X\}$ satisfies

$$\|\mathbf{x}(t)\|_{L_2^{n_0+n_1+n_2}} \le K \|\mathbf{x}_0\|_{L_2^{n_0+n_1+n_2}} e^{-\gamma t}.$$

Note 2. Exponential stability in X implies exponential stability in L_2, since $\|\mathbf{x}\|_{L_2} \le \|\mathbf{x}\|_{H_k}$ for any $\mathbf{x} \in H_k$ and $k \ge 0$. Furthermore, our definitions of exponential stability imply that all states in $\mathbf{x}$ must be exponentially decreasing in the given norm. Because not all standardized PDE representations of a given scalar high-order PDE necessarily use the same set of first-order states (See, e.g. the E-B beam example), this raises the possibility one standardized PDE representation may be exponentially stable, while another may not.

Note that in the case where the X or L_2 stability definition holds with $\gamma = 0$, we say that the system is Lyapunov stable or neutrally stable.

4. 3-PI operators form an algebra

In Section 7, we will show how to construct an equivalent PIE representation of any PDE in the standardized form described in Section 3. PIEs, as will be defined in Section 5, have the advantage that they are parameterized by the class of 3-PI operators, which are bounded on L_2 and form an algebra. Furthermore, candidate Lyapunov functions can be parameterized using 3-PI operators. The algebraic nature of 3-PI operators significantly simplifies the problem of analysis and control of PIEs.

Formally, we say that an operator $\mathcal{P}$ is 3-PI if there exist 3 bounded matrix-valued functions $N_0 : [a,b] \to \mathbb{R}^{n \times n}$, $N_1 : [a,b]^2 \to \mathbb{R}^{n \times n}$, and $N_2 : [a,b]^2 \to \mathbb{R}^{n \times n}$ such that

$$(\mathcal{P}\mathbf{x})(s) := \left(\mathcal{P}_{\{N_0,N_1,N_2\}}\mathbf{x}\right)(s) := N_0(s)x(s) + \int_a^s N_1(s,\theta)x(\theta)d\theta + \int_s^b N_2(s,\theta)x(\theta)d\theta,$$

where N_0 defines a multiplier operator and N_1, N_2 define the kernel of an integral operator.

For given N_0, N_1, N_2, we use $\mathcal{P}_{\{N_0,N_1,N_2\}} : L_2^n \to L_2^n$ to denote the corresponding PI operator. When clear from context, we use the shorthand notation $\mathcal{P}_{\{N_i\}}$ to indicate $\mathcal{P}_{\{N_0,N_1,N_2\}}$.

One may interpret 3-PI operators to be an extension of matrices, wherein N_0 defines the diagonal of the matrix, N_1 contains the sub-diagonal terms, and N_2 contains the terms above the diagonal.

In the following subsections, we show that this class of bounded linear operators is closed under composition and adjoint (closure under addition and scalar multiplication follows immediately from addition and scalar multiplication of parameters). Furthermore, these results show that if we define the set of 3-PI operators with polynomial parameters N_0, N_1, and N_2, then this set forms a sub-algebra.

4.1. Composition of 3-PI operators

In this subsection, we derive an analytic formula for the composition 3-PI operators. Specifically, we have the following.

Lemma 4. *For any bounded functions $B_0, N_0 : [a, b] \to \mathbb{R}^{n\times n}$, $B_1, B_2, N_1, N_2 : [a, b]^2 \to \mathbb{R}^{n\times n}$, we have*

$$\mathcal{P}_{\{R_i\}} = \mathcal{P}_{\{B_i\}}\mathcal{P}_{\{N_i\}}$$

where

$$\begin{aligned}
R_0(s) &= B_0(s)N_0(s), \qquad (5)\\
R_1(s,\theta) &= B_0(s)N_1(s,\theta) + B_1(s,\theta)N_0(\theta)\\
&\quad + \int_a^\theta B_1(s,\xi)N_2(\xi,\theta)d\xi + \int_\theta^s B_1(s,\xi)N_1(\xi,\theta)d\xi\\
&\quad + \int_s^b B_2(s,\xi)N_1(\xi,\theta)d\xi,\\
R_2(s,\theta) &= B_0(s)N_2(s,\theta) + B_2(s,\theta)N_0(\theta)\\
&\quad + \int_a^s B_1(s,\xi)N_2(\xi,\theta)d\xi + \int_s^\theta B_2(s,\xi)N_2(\xi,\theta)d\xi\\
&\quad + \int_\theta^b B_2(s,\xi)N_1(\xi,\theta)d\xi.
\end{aligned}$$

Proof. See the extended paper in Peet (2018b) for a proof.

This lemma proves that the class of 3-PI operators is closed under composition.

Corollary 5. *Suppose that $\{B_i\}$ and $\{N_i\}$ are matrices of polynomials. Then if $\mathcal{P}_{\{R_i\}} = \mathcal{P}_{\{B_i\}}\mathcal{P}_{\{N_i\}}$, $\{R_i\}$ are matrices of polynomials.*

Proof. The algebra of polynomials is closed under multiplication and integration. Therefore, the proof follows from the expressions for $\{R_i\}$ given in Lemma 4. □

This corollary implies that the subset of 3-PI operators with polynomial parameters is likewise closed under composition and therefore forms a subalgebra.

4.2. The adjoint of 3-PI operators

Next, we give a formula for the adjoint of a 3-PI operator.

Lemma 6. *For any bounded functions $N_0 : [a, b] \to \mathbb{R}^{n\times n}$, $N_1, N_2 : [a, b]^2 \to \mathbb{R}^{n\times n}$ and any $\mathbf{x}, \mathbf{y} \in L_2^n[a, b]$, we have*

$$\left\langle \mathcal{P}_{\{N_i\}}\mathbf{x}, \mathbf{y}\right\rangle_{L_2} = \left\langle \mathbf{x}, \mathcal{P}_{\{\hat{N}_i\}}\mathbf{y}\right\rangle_{L_2}$$

where

$$\begin{aligned}
\hat{N}_0(s) &= N_0(s)^T, \quad \hat{N}_1(s,\eta) = N_2(\eta,s)^T,\\
\hat{N}_2(s,\eta) &= N_1(\eta,s)^T. \qquad (6)
\end{aligned}$$

Proof. See the extended paper in Peet (2018b) for a proof.

The following Corollary follows immediately from Lemma 6.

Corollary 7. *Suppose that $\{N_i\}$ are matrices of polynomials. Then, using the adjoint with respect to L_2, if $\mathcal{P}_{\{\hat{N}_i\}} = \mathcal{P}^*_{\{N_i\}}$, $\{\hat{N}_i\}$ are matrices of polynomials.*

5. Partial Integral Equations (PIEs)

In this section, we give the autonomous form of a Partial Integral Equation (PIE) and define notions of solution and exponential stability. Specifically, for given 3-PI operators $\mathcal{A}, \mathcal{T}$, we say, for an initial condition, $\mathbf{x}_0 \in L_2^n$, that $\mathbf{x} : \mathbb{R}^+ \to L_2^n$ satisfies the PIE defined by $\{\mathcal{A}, \mathcal{T}\}$ if $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}$ is Frechét differentiable for all $t \ge 0$ and

$$\mathcal{T}\dot{\mathbf{x}}(t) = \mathcal{A}x(t) \qquad (7)$$

for all $t \ge 0$.

Note that not all PIEs are well-posed in the sense of Hadamard and at present we propose no direct conditions on the PIE which allows us to conclude well-posedness. However, we will show in Section 7 that if a well-posed PDE is in standardized form (satisfying Eq. (4)), and the PIE is generated from that standardized PDE using the formulae in Section 7, then the PIE is well-posed.

5.1. Exponential stability of PIEs

Having defined PIE's, we now define the notion of exponential stability we will use.

Definition 8. We say the PIE defined by the 3-PI operators $\{\mathcal{A}, \mathcal{T}\}$ is exponentially stable if there exist constants K and $\gamma > 0$ such that for $\mathbf{x}(0) \in L_2^n$, any solution $\mathbf{x}$ satisfies

$$\|\mathbf{x}(t)\|_{L_2} \le K\,\|\mathbf{x}(0)\|_{L_2}\, e^{-\gamma t}.$$

In the case where the exponential stability definition holds with $\gamma = 0$, we say the PIE is stable in the sense of Lyapunov or neutrally stable.

6. A unitary map from X to L_2

In this section, we show equivalence between the Hilbert space $L_2^{n_0+n_1+n_2}$ and the space

$$X = \left\{ \begin{bmatrix} x_0\\ x_1\\ x_2 \end{bmatrix} \in L_2^{n_0} \times H_1^{n_1} \times H_2^{n_2} \;:\; B\begin{bmatrix} x_1(a)\\ x_1(b)\\ x_2(a)\\ x_2(b)\\ x_{2s}(a)\\ x_{2s}(b)\end{bmatrix} = 0 \right\}$$

where B satisfies Eq. (4) and X is equipped with the inner product

$$\langle \mathbf{x}, \mathbf{y}\rangle_X = \langle x_0, y_0\rangle_{L_2} + \langle \partial_s x_1, \partial_s y_1\rangle_{L_2} + \left\langle \partial_s^2 x_2, \partial_s^2 y_2\right\rangle_{L_2}.$$

Specifically, in this section, we

- Construct a unitary map $\mathcal{T} : L_2^{n_0+n_1+n_2} \to X$ where $\mathcal{T}$ is a 3-PI operator.
- Show $\langle\cdot,\cdot\rangle_X$ is an inner product and X is Hilbert with this inner product.

- Show that for $\mathbf{x} \in X$, the norm $\|\cdot\|_X$ is equivalent to the norm $\|\cdot\|_{L_2\times H_1\times H_2}$ where recall

$$\|\mathbf{x}\|_{L_2\times H^1\times H^2} = \|x_0\|_{L_2} + \|x_1\|_{H_1} + \|x_2\|_{H_2}.$$

6.1. The unitary map, $\mathcal{T}$

In this subsection, we define the 3-PI operator $\mathcal{T} = \mathcal{P}_{\{G_i\}}$ such that if

$$\mathbf{x} \in X \qquad \text{and} \qquad \hat{\mathbf{x}} \in L_2^{n_0+n_1+n_2}$$

then

$$\mathbf{x} = \mathcal{T}\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x} \qquad \text{and} \qquad \hat{\mathbf{x}} = \begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathcal{T}\hat{\mathbf{x}}.$$

First, we first show that a PDE state $\mathbf{x} \in H_2$ can be represented using the PIE state, $\partial_s^2\mathbf{x} \in L_2$ and a set of 'core' boundary conditions $(x(a), x_s(a))$.

Lemma 9. *Suppose that $\mathbf{x} \in H_2^n[a,b]$. Then for any $s \in [a,b]$,*

$$x(s) = x(a) + \int_a^s x_s(\eta)d\eta$$
$$x_s(s) = x_s(a) + \int_a^s x_{ss}(\eta)d\eta$$
$$x(s) = x(a) + x_s(a)(s-a) + \int_a^s (s-\eta)x_{ss}(\eta)d\eta.$$

Proof. The first two identities are the fundamental theorem of calculus. The third identity is a repeated application of the fundamental theorem of calculus, combined with a change of variables. That is, for any $s \in [a,b]$,

$$\begin{aligned} x(s) &= x(a) + \int_a^s x_s(\eta)d\eta \\ &= x(a) + \int_a^s x_s(a)ds + \int_a^s\int_a^\eta x_{ss}(\zeta)d\zeta d\eta. \end{aligned}$$

Examining the 3rd term, where recall $\mathbf{I}(s)$ is the indicator function,

$$\begin{aligned} &\int_a^s\int_a^\eta x_{ss}(\zeta)d\zeta d\eta = \int_a^b\int_a^b \mathbf{I}(s-\eta)\mathbf{I}(\eta-\zeta)x_{ss}(\zeta)d\zeta d\eta \\ &= \int_a^b\left(\int_a^b \mathbf{I}(s-\eta)\mathbf{I}(\eta-\zeta)d\eta\right)x_{ss}(\zeta)d\zeta \\ &= \int_a^b \mathbf{I}(s-\zeta)\left(\int_s^\zeta d\eta\right)x_{ss}(\zeta)d\zeta = \int_a^s (s-\zeta)\,x_{ss}(\zeta)d\zeta \end{aligned}$$

which is the desired result. □

As an obvious corollary, we have

$$x(b) = x(a) + \int_a^b x_s(\eta)d\eta$$
$$x_s(b) = x_s(a) + \int_a^b x_{ss}(\eta)d\eta$$
$$x(b) = x(a) + x_s(a)(b-a) + \int_a^b (b-\eta)x_{ss}(\eta)d\eta.$$

The implication is that any boundary value can be expressed using two other boundary identities. In the standardized PDE representation, we have a generic set of boundary conditions defined by the matrix B. In the following theorem, we generalize Lemma 9 in order to express the PDE state $\mathbf{x} \in X$ in terms of the PIE state, $\hat{\mathbf{x}} \in L_2^{n_0+n_1+n_2}$, and generalized BCs (which are equal to zero). This allows us to define the map $\mathcal{T}$.

Theorem 10. *Let $\mathcal{T} = \mathcal{P}_{\{G_0,G_1,G_2\}}$ with G_i as defined in Eqs. (8), which can be found in Box I. Then for any $\mathbf{x} \in X$,*

$$\mathbf{x} = \mathcal{T}\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}.$$

Furthermore, for any $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in L_2^{n_0+n_1+n_2}$, $\mathcal{T}\hat{\mathbf{x}}, \mathcal{T}\hat{\mathbf{y}} \in X$ and $\langle\mathcal{T}\hat{\mathbf{x}}, \mathcal{T}\hat{\mathbf{y}}\rangle_X = \langle\hat{\mathbf{x}}, \hat{\mathbf{y}}\rangle_{L_2}$.

Proof. See the extended paper in Peet (2018b) for a proof.

Corollary 11. *Let $\mathcal{H} = \mathcal{P}_{\{G_3,G_4,G_5\}}$ with G_i as defined in Eqs. (8). Then for any $\mathbf{x} \in X$,*

$$\begin{bmatrix} 0 & \partial_s & 0 \\ 0 & 0 & \partial_s \end{bmatrix}\mathbf{x} = \mathcal{H}\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}.$$

Proof. By Theorem 10,

$$\mathbf{x} = \mathcal{T}\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}.$$

Now, for any $\hat{\mathbf{x}} \in L_2^{n_0+n_1+n_2}$, it can be readily verified through differentiation that

$$\begin{bmatrix} 0 & \partial_s & 0 \\ 0 & 0 & \partial_s \end{bmatrix}\mathcal{T}\hat{\mathbf{x}} = \mathcal{H}\hat{\mathbf{x}}$$

which completes the proof. □

Corollary 12. *The operator $\mathcal{T} = \mathcal{P}_{\{G_0,G_1,G_2\}} : L_2^{n_0+n_1+n_2} \to X$ is unitary.*

Proof. Theorem 10 shows that for any $\mathbf{x} \in X$, there exists some $\hat{\mathbf{x}} \in L_2$ such that $\mathbf{x} = \mathcal{T}\hat{\mathbf{x}}$ (surjective). Furthermore, for any $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in L_2$, $\langle\mathcal{T}\hat{\mathbf{x}}, \mathcal{T}\hat{\mathbf{y}}\rangle_X = \langle\hat{\mathbf{x}}, \hat{\mathbf{y}}\rangle_{L_2}$, which concludes the proof. □

Because $L_2^{n_0+n_1+n_2}$ is a Hilbert space and $\mathcal{T}$ is unitary, Corollary 12 implies X is a Hilbert space.

6.2. Equivalence of norms

In this subsection, we briefly show that the norms $\|\cdot\|_X$ and $\|\cdot\|_{L_2\times H_1\times H_2}$ are equivalent.

Lemma 13. *For any $\mathbf{x} \in X$, $\|\mathbf{x}\|_X \le \|\mathbf{x}\|_{L_2\times H_1\times H_2}$ and there exists a constant $c > 0$ such that $\|\mathbf{x}\|_{L_2\times H_1\times H_2} \le c\,\|\mathbf{x}\|_X$.*

Proof. First, we note that

$$\begin{aligned} \|\mathbf{x}\|_{L_2\times H_1\times H_2} &= \left\|\begin{bmatrix} 0 \\ x_1 \\ x_2 \end{bmatrix}\right\|_{L_2} + \left\|\begin{bmatrix} 0 \\ 0 \\ x_{2s} \end{bmatrix}\right\|_{L_2} + \left\|\begin{bmatrix} x_0 \\ x_{1s} \\ x_{2ss} \end{bmatrix}\right\|_{L_2} \\ &= \left\|\begin{bmatrix} 0 \\ x_1 \\ x_2 \end{bmatrix}\right\|_{L_2} + \left\|\begin{bmatrix} 0 \\ 0 \\ x_{2s} \end{bmatrix}\right\|_{L_2} + \|\mathbf{x}\|_X \end{aligned}$$

and hence $\|\mathbf{x}\|_X \le \|\mathbf{x}\|_{L_2\times H_1\times H_2}$. Now, since $G_i, H_i \in L_\infty[a,b]$, there exist $c_1, c_2 > 0$ such that

$$\begin{aligned} \left\|\begin{bmatrix} 0 \\ x_1 \\ x_2 \end{bmatrix}\right\|_{L_2} &\le \|\mathbf{x}\|_{L_2} = \left\|\mathcal{T}\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}\right\|_{L_2} \\ &\le c_1\left\|\begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}\right\|_{L_2} = c_1\,\|\mathbf{x}\|_X \end{aligned}$$

$$G_0(s)=\begin{bmatrix}I_{n_0}&0&0\\0&0&0\\0&0&0\end{bmatrix},\qquad G_1(s,\theta)=\begin{bmatrix}0&0&0\\0&I_{n_1}&0\\0&0&(s-\theta)I_{n_2}\end{bmatrix}+G_2(s,\theta),\qquad G_2(s,\theta)=-K(s)(BT)^{-1}BQ(s,\theta),$$
$$G_3(s)=\begin{bmatrix}0&I_{n_1}&0\\0&0&0\end{bmatrix},\qquad G_4(s,\theta)=\begin{bmatrix}0&0&0\\0&0&I_{n_2}\end{bmatrix}+G_5(s,\theta),\qquad G_5(s,\theta)=-V(BT)^{-1}BQ(s,\theta),$$
$$T=\begin{bmatrix}I_{n_1}&0&0\\I_{n_1}&0&0\\0&I_{n_2}&0\\0&I_{n_2}&(b-a)I_{n_2}\\0&0&I_{n_2}\\0&0&I_{n_2}\end{bmatrix},\quad Q(s,\theta)=\begin{bmatrix}0&0&0\\0&I_{n_1}&0\\0&0&0\\0&0&(b-\theta)I_{n_2}\\0&0&0\\0&0&I_{n_2}\end{bmatrix},\quad K(s)=\begin{bmatrix}0&0&0\\I_{n_1}&0&0\\0&I_{n_2}&(s-a)I_{n_2}\end{bmatrix},\quad V=\begin{bmatrix}0&0&0\\0&0&I_{n_2}\end{bmatrix}. \tag{8}$$

Box I.

and

$$\left\|\begin{bmatrix}0\\0\\x_{2s}\end{bmatrix}\right\|_{L_2}\le\left\|\begin{bmatrix}0\\x_{1s}\\x_{2s}\end{bmatrix}\right\|_{L_2}=\left\|\mathcal{H}\begin{bmatrix}I&&\\&\partial_s&\\&&\partial_s^2\end{bmatrix}\mathbf{x}\right\|_{L_2}$$
$$\le c_2\left\|\begin{bmatrix}I&&\\&\partial_s&\\&&\partial_s^2\end{bmatrix}\mathbf{x}\right\|_{L_2}=c_2\left\|\mathbf{x}\right\|_X.$$

Therefore, we conclude that

$$\|\mathbf{x}\|_{L_2\times H_1\times H_2}\le(1+c_1+c_2)\|\mathbf{x}\|_X$$

as desired. □

This result shows that for PDE systems in standardized form, stability in $\|\cdot\|_X$ and $\|\cdot\|_{L_2\times H_1\times H_2}$ are equivalent.

7. Converting PDEs to PIEs

In this section, we show that for any PDE in standardized form, there exists a PIE for which any solution to the PDE defines a solution to the PIE and any solution to the PIE defines a solution to the PDE. We further show that this result implies that exponential stability of the PIE is equivalent to exponential stability of the PDE in X.

7.1. Equivalence of solutions for PDEs and PIEs

Now that we have the unitary 3-PI operator $\mathcal{T}:=\mathcal{P}_{\{G_0,G_1,G_2\}}$ where

$$\mathbf{x}=\mathcal{T}\begin{bmatrix}I&&\\&\partial_s&\\&&\partial_s^2\end{bmatrix}\mathbf{x}$$

for any $\mathbf{x}\in X$, conversion of the PDE to a PIE (Eq. (7)) is direct.

Lemma 14. *Given $\hat{\mathbf{x}}_0(t)\in L_2^{n_0+n_1+n_2}$, the function $\hat{\mathbf{x}}(t)\in L_2^{n_0+n_1+n_2}$ satisfies the PIE defined by $\{\mathcal{T},\mathcal{A}\}$ if and only if for $\mathbf{x}_0=\mathcal{T}\hat{\mathbf{x}}_0$, the function $\mathbf{x}(t)=\mathcal{T}\hat{\mathbf{x}}(t)$ satisfies the PDE defined by $\{A_i,X\}$ where*

$$\mathcal{T}:=\mathcal{P}_{\{G_0,G_1,G_2\}},\qquad \mathcal{A}:=\mathcal{P}_{\{H_i\}}$$
$$H_0(s)=A_0(s)G_0(s)+A_1(s)G_3(s)+A_{20}(s)$$
$$H_1(s,\theta)=A_0(s)G_1(s,\theta)+A_1(s)G_4(s,\theta),$$
$$H_2(s,\theta)=A_0(s)G_2(s,\theta)+A_1(s)G_5(s,\theta),$$
$$A_{20}(s)=\begin{bmatrix}0&0&A_2(s)\end{bmatrix} \tag{9}$$

where the G_i are as defined in Eqs. (8).

Proof. Define $\mathcal{H}:=\mathcal{P}_{\{G_3,G_4,G_5\}}$ and

$$\mathcal{D}_1:=\begin{bmatrix}I&&\\&\partial_s&\\&&\partial_s^2\end{bmatrix},\quad \mathcal{D}_2:=\begin{bmatrix}0&\partial_s&0\\0&0&\partial_s\end{bmatrix}.$$

Suppose $\mathbf{x}$ satisfies the PDE and $\hat{\mathbf{x}}(t)=\mathcal{D}_1\mathbf{x}(t)$. By Theorem 10 and Lemma 4 and the definition of the G_i, we have

$$\begin{aligned}\dot{\mathbf{x}}(t)&=\mathcal{P}_{\{A_0,0,0\}}\mathbf{x}(t)+\mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_2\mathbf{x}(t)+\mathcal{P}_{\{A_{20},0,0\}}\mathcal{D}_1\mathbf{x}(t)\\&=\mathcal{P}_{\{A_0,0,0\}}\mathcal{T}\hat{\mathbf{x}}(t)+\mathcal{P}_{\{A_1,0,0\}}\mathcal{H}\hat{\mathbf{x}}(t)+\mathcal{P}_{\{A_{20},0,0\}}\hat{\mathbf{x}}(t)\\&=\mathcal{P}_{\{A_0G_0,A_0G_1,A_0G_2\}}\hat{\mathbf{x}}(t)\\&\quad+\mathcal{P}_{\{A_1G_3,A_1G_4,A_1G_5\}}\hat{\mathbf{x}}(t)+\mathcal{P}_{\{A_{20},0,0\}}\hat{\mathbf{x}}(t)\\&=\mathcal{P}_{\{H_0,H_1,H_2\}}\hat{\mathbf{x}}(t)=\mathcal{A}\hat{\mathbf{x}}(t).\end{aligned}$$

Finally, $\dot{\mathbf{x}}(t)=\mathcal{T}\dot{\hat{\mathbf{x}}}(t)$ and $\hat{\mathbf{x}}(0)=\mathcal{D}_1\mathbf{x}(0)=\mathcal{D}_1\mathbf{x}_0=\mathcal{D}_1\mathcal{T}\hat{\mathbf{x}}_0=\hat{\mathbf{x}}_0$.

Conversely, suppose $\hat{\mathbf{x}}(t)$ solves the PIE. Define $\mathbf{x}(t)=\mathcal{T}\hat{\mathbf{x}}(t)$. Then by Theorem 10, $\mathbf{x}(t)\in X$ and

$$\begin{aligned}\dot{\mathbf{x}}(t)&=\mathcal{T}\dot{\hat{\mathbf{x}}}(t)=\mathcal{A}\hat{\mathbf{x}}(t)\\&=\mathcal{P}_{\{A_0,0,0\}}\mathcal{T}\hat{\mathbf{x}}(t)+\mathcal{P}_{\{A_1,0,0\}}\mathcal{H}\hat{\mathbf{x}}(t)+\mathcal{P}_{\{A_{20},0,0\}}\hat{\mathbf{x}}(t)\\&=\mathcal{P}_{\{A_0,0,0\}}\mathbf{x}(t)+\mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_2\mathbf{x}(t)+\mathcal{P}_{\{A_{20},0,0\}}\mathcal{D}_1\mathbf{x}(t)\end{aligned}$$

as desired. Furthermore, $\mathbf{x}(0)=\mathcal{T}\hat{\mathbf{x}}(0)=\mathcal{T}\hat{\mathbf{x}}_0=\mathbf{x}_0$. □

Note 3. While the conversion formulae in Eqs. (8) are relatively complex, this is because they encompass a very large class of PDEs and must account for every case. Individual PIE representations of specific PDEs, by contrast are typically rather simple. In the following subsection, we demonstrate one such representation.

7.2. PIE representation of the E-B beam

To illustrate the PIE representation, we again consider the Euler–Bernoulli beam model, using the standardized PDE representation of Section 3.3. Applying the formulae in Eqs. (8), we obtain the PIE $\{\mathcal{T},\mathcal{A}\}$ where

$$\mathcal{T}:=\mathcal{P}_{\{N_i\}},\qquad \mathcal{A}:=\mathcal{P}_{\{R_i\}} \tag{10}$$
$$N_0=0,\qquad N_1=\begin{bmatrix}s-\theta&0\\0&0\end{bmatrix},\qquad N_2=\begin{bmatrix}0&0\\0&\theta-s\end{bmatrix},$$
$$R_0=\begin{bmatrix}0&-c\\1&0\end{bmatrix},\qquad R_1=0,\qquad R_2=0.$$

7.3. Stability equivalence for PDEs and PIEs

The following result uses the unitary property of the state transformation, $\mathcal{T}$, to show equivalence between stability of PDEs and PIEs in a certain sense.

Lemma 15. *The PDE defined by $\{A_i, X\}$ is exponentially stable in X with constants $K, \gamma > 0$ if and only if the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$, where $\{\mathcal{T}, \mathcal{A}\}$ are as defined in Eq. (9), is exponentially stable with constants $K, \gamma > 0$.*

Proof. Suppose the PDE defined by $\{A_i, X\}$ is exponentially stable with constants $K, \gamma > 0$. Then for any $\mathbf{x}_0 \in X$, any solution $\mathbf{x}$ of the PDE defined by $\{A_i, X\}$ satisfies $\|\mathbf{x}(t)\|_X \le K\|\mathbf{x}_0\|_X e^{-\gamma t}$. Now for $\hat{\mathbf{x}}_0 \in L_2^{n_0+n_1+n_2}$, let $\hat{\mathbf{x}}$ be a solution of the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$. Define $\mathbf{x}_0 := \mathcal{T}\hat{\mathbf{x}}_0 \in X$ and $\mathbf{x}(t) := \mathcal{T}\hat{\mathbf{x}}(t)$. Then by Lemma 14, $\mathbf{x}(t)$ satisfies the PDE defined by $\{A_i, X\}$ with initial condition $\mathbf{x}_0$. Therefore, by Theorem 10,

$$\begin{aligned}\left\|\hat{\mathbf{x}}(t)\right\|_{L_2} &= \left\|\mathcal{T}\hat{\mathbf{x}}(t)\right\|_X = \|\mathbf{x}(t)\|_X \\ &\le K\|\mathbf{x}_0\|_X e^{-\gamma t} = K\left\|\mathcal{T}\hat{\mathbf{x}}_0\right\|_X e^{-\gamma t} \\ &= K\left\|\hat{\mathbf{x}}_0\right\|_{L_2} e^{-\gamma t}.\end{aligned}$$

Conversely, suppose the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$ is exponentially stable with constants $K, \gamma > 0$. Then for any $\hat{\mathbf{x}}_0 \in L_2$, any solution $\hat{\mathbf{x}}$ of the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$ satisfies $\left\|\hat{\mathbf{x}}(t)\right\|_{L_2} \le K\left\|\hat{\mathbf{x}}_0\right\|_{L_2} e^{-\gamma t}$. Now for $\mathbf{x}_0 \in X$, let $\mathbf{x}$ be a solution of the PDE defined by $\{A_i, X\}$. Define

$$\hat{\mathbf{x}}_0 := \begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}_0 \in L_2, \;\; \hat{\mathbf{x}}(t) := \begin{bmatrix} I & & \\ & \partial_s & \\ & & \partial_s^2 \end{bmatrix}\mathbf{x}(t) \in L_2.$$

Then by Lemma 14, $\mathbf{x}(t) = \mathcal{T}\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{x}}(t)$ satisfies the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$ with initial condition $\hat{\mathbf{x}}_0$. Therefore, by Theorem 10,

$$\begin{aligned}\|\mathbf{x}(t)\|_X &= \left\|\mathcal{T}\hat{\mathbf{x}}(t)\right\|_X = \left\|\hat{\mathbf{x}}(t)\right\|_{L_2} \\ &\le K\left\|\hat{\mathbf{x}}_0\right\|_{L_2} e^{-\gamma t} = K\left\|\mathcal{T}\hat{\mathbf{x}}_0\right\|_X e^{-\gamma t} \\ &= K\|\mathbf{x}_0\|_X e^{-\gamma t}. \quad \square\end{aligned}$$

Having shown that stability of a PIE is equivalent to that of a PDE in a precisely defined sense, we now proceed to define a Linear PI Inequality (LPI), whose feasibility guarantees exponential stability of a PDE in standardized form.

8. Lyapunov stability as an LPI

Using the 3-PI algebra, we may now succinctly represent our Lyapunov stability conditions. The procedure is relatively straightforward.

Theorem 16. *Suppose there exist $\alpha, \delta > 0$, $N_0 : [a, b] \to \mathbb{R}^{n\times n}$, $N_1, N_2 : [a, b]^2 \to \mathbb{R}^{n\times n}$ such that for $\mathcal{P} := \mathcal{P}_{\{N_0,N_1,N_2\}}$, $\mathcal{P} = \mathcal{P}^* \ge \alpha I$ and*

$$\mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A} \le -\delta\mathcal{T}^*\mathcal{T}$$

where $\mathcal{T}$ and $\mathcal{A}$ are as defined in Eq. (9). Then any solution, $\mathbf{x}(t)$ of the PDE defined by $\{A_i, X\}$ satisfies

$$\|\mathbf{x}(t)\|_{L_2} \le \frac{\zeta}{\alpha}\|\mathbf{x}(0)\|_{L_2}^2 e^{-\delta/\zeta t}.$$

where $\zeta = \|\mathcal{P}\|_{\mathcal{L}(L_2)}$.

Proof. Suppose $\hat{\mathbf{x}}$ solves the PIE defined by $\{\mathcal{T}, \mathcal{A}\}$ for some $\hat{\mathbf{x}}_0$. Consider the candidate Lyapunov function defined as

$$V(\hat{\mathbf{x}}) = \left\langle\hat{\mathbf{x}}(t), \mathcal{T}^*\mathcal{P}\mathcal{T}\hat{\mathbf{x}}(t)\right\rangle_{L_2} \ge \alpha\left\|\mathcal{T}\hat{\mathbf{x}}\right\|_{L_2}^2.$$

The derivative of V along solution $\hat{\mathbf{x}}$ is

$$\begin{aligned}\dot V(\hat{\mathbf{x}}(t)) &= \left\langle\mathcal{T}\dot{\hat{\mathbf{x}}}(t), \mathcal{P}\mathcal{T}\hat{\mathbf{x}}(t)\right\rangle_{L_2} + \left\langle\hat{\mathbf{x}}(t), \mathcal{P}\mathcal{T}\dot{\hat{\mathbf{x}}}(t)\right\rangle_{L_2} \\ &= \left\langle\mathcal{A}\hat{\mathbf{x}}(t), \mathcal{P}\mathcal{T}\hat{\mathbf{x}}(t)\right\rangle_{L_2} + \left\langle\mathcal{T}\hat{\mathbf{x}}(t), \mathcal{P}\mathcal{A}\hat{\mathbf{x}}(t)\right\rangle_{L_2} \\ &= \left\langle\hat{\mathbf{x}}(t), \left(\mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A}\right)\hat{\mathbf{x}}(t)\right\rangle_{L_2} \\ &\le -\delta\left\|\mathcal{T}\hat{\mathbf{x}}(t)\right\|_{L_2}^2.\end{aligned}$$

Recall $\|\mathcal{P}\|_{\mathcal{L}(L_2)} = \zeta$. Then by a standard application of Gronwall–Bellman, we have

$$\left\|\mathcal{T}\hat{\mathbf{x}}(t)\right\|_{L_2}^2 \le \frac{\zeta}{\alpha}\left\|\mathcal{T}\hat{\mathbf{x}}_0\right\|_{L_2}^2 e^{-\delta/\zeta t}.$$

Now for any solution, $\mathbf{x}$ of the PDE defined by $\{A_i, X\}$ with initial condition $\mathbf{x}_0$, we have $\mathbf{x}(t) = \mathcal{T}\hat{\mathbf{x}}(t)$ where $\hat{\mathbf{x}}$ is a solution of the PIE with initial condition $\hat{\mathbf{x}}_0$ where $\mathbf{x}_0 = \mathcal{T}\hat{\mathbf{x}}_0$. Thus

$$\|\mathbf{x}(t)\|_{L_2}^2 \le \frac{\zeta}{\alpha}\|\mathbf{x}_0\|_{L_2}^2 e^{-\delta/\zeta t}. \quad \square$$

Note 4. Theorem 16 proves exponential stability of the PDE with respect to the L_2 norm and not the X-norm. While it is possible to formulate a test for stability in the X-norm, this would differ from most existing results and the literature and hence is omitted. Note, however, that for any $\mathbf{x} \in L_2$, $\mathcal{T}\mathbf{x} = 0$ if and only if $\mathbf{x} = 0$ (modulo a set of zero measure).

Note 5. Theorem 16 is equivalent to the Lyapunov inequality for PDEs with the restriction that the Lyapunov operator be a PI operator. This, in turn, may be interpreted as a dissipativity condition on the generator. Such conditions are sometimes enforced using multiplier approaches as in, e.g. Luo et al. (2012), and have been shown to be necessary and sufficient for stability of infinite-dimensional systems, as in Curtain and Zwart (1995) and Datko (1970). Note that the constraint that the operator $\mathcal{P}$ be self-adjoint is not conservative as any Lyapunov function defined by a non-self-adjoint operator admits a representation using a self-adjoint operator.

Theorem 16 poses a convex optimization problem, whose feasibility implies stability of solutions of a coupled linear PDE. We refer to such optimization problems as Linear PI Inequalities (LPIs). Solving an LPI requires parameterizing the 3-PI operator $\mathcal{P}$ using polynomials and enforcing the inequalities using LMIs. In the following section, we briefly introduce a method of enforcing positivity of a 3-PI operator using LMI constraints.

9. Solving the stability LPI via PIETOOLS

In Section 8, we formulated the question of Lyapunov stability as an LPI. In this section, we will we propose a general form of LPI and show how these convex optimization problems can be solved under the assumption that all 3-PI operators are parameterized by polynomials.

For given 3-PI operators $\{\mathcal{E}_{ij}, \mathcal{F}_{ij}, \mathcal{G}_i\}$ and linear operator $\mathcal{L}$, a Linear PI Inequality (LPI) is a convex optimization of the form

$$\begin{aligned}&\min_{N_{0i},N_{1i},N_{2i}} \; \mathcal{L}(\{N_{ij}\}) \qquad (11)\\ &\sum_{j=1}^{K}\mathcal{E}_{ij}^*\mathcal{P}_{\{N_{1i},N_{2i},N_{3i}\}}\mathcal{F}_{ij} + \mathcal{G}_i \ge 0 \qquad i = 1,\ldots,L.\end{aligned}$$

LPIs of the form of Eq. (11) can be solved directly using PIETOOLS (Shivakumar et al., 2020). Composition and adjoint are algebraic operations on the 3-PI parameters and are computed using the formulae in Section 4. Positivity is enforced using an LMI constraint as described in the following subsection.

9.1. Enforcing positivity of 3-PI operators

In this subsection, for a given self-adjoint 3-PI operator with polynomial parameters ($\{N_i\}$), we have given an LMI constraint on the coefficients of the polynomials $\{N_i\}$ which enforces a constraint of the form $\mathcal{P}_{\{N_i\}} \geq 0$. Specifically, the following proposition (a slight modification of the result in Peet, 2019) gives necessary and sufficient conditions for a 3-PI operator to have a 3-PI square root.

Proposition 17. *For any bounded functions $Z(s)$, $Z(s,\theta)$, and g, where g is scalar and $g(s) \geq 0$ for all $s \in [a,b]$ and*

$$\begin{aligned}
N_0(s) &= g(s)Z(s)^T P_{11} Z(s),\\
N_1(s,\theta) &= g(s)Z(s)^T P_{12} Z(s,\theta) + g(\theta)Z(\theta,s)^T P_{31} Z(\theta)\\
&\quad + \int_a^\theta g(\nu)Z(\nu,s)^T P_{33} Z(\nu,\theta)d\nu\\
&\quad\quad + \int_\theta^s g(\nu)Z(\nu,s)^T P_{32} Z(\nu,\theta)d\nu\\
&\quad\quad\quad + \int_s^L g(\nu)Z(\nu,s)^T P_{22} Z(\nu,\theta)d\nu,\\
N_2(s,\theta) &= g(s)Z(s)^T P_{13} Z(s,\theta) + g(\theta)Z(\theta,s)^T P_{21} Z(\theta)\\
&\quad + \int_a^s g(\nu)Z(\nu,s)^T P_{33} Z(\nu,\theta)d\nu\\
&\quad\quad + \int_s^\theta g(\nu)Z(\nu,s)^T P_{23} Z(\nu,\theta)d\nu\\
&\quad\quad\quad + \int_\theta^L g(\nu)Z(\nu,s)^T P_{22} Z(\nu,\theta)d\nu,
\end{aligned}$$

where

$$P = P^T = \begin{bmatrix} P_{11} & P_{12} & P_{13}\\ P_{21} & P_{22} & P_{23}\\ P_{31} & P_{32} & P_{33}\end{bmatrix} \geq 0,$$

we have $\mathcal{P}^*_{\{N_i\}} = \mathcal{P}_{\{N_i\}}$ and $\left\langle \mathbf{x}, \mathcal{P}_{\{N_i\}}\mathbf{x}\right\rangle_{L_2} \geq 0$ for all $\mathbf{x} \in L_2[a,b]$.

Proof. It is relatively easy to show that the N_i satisfy Eq. (6) with $\{\hat{N}_i\} = \{N_i\}$. Therefore, by Lemma 6 $\mathcal{P}_{\{N_i\}}$ is self adjoint. Now define the operator

$$(\mathcal{Z}\mathbf{x})(s) = \begin{bmatrix} \sqrt{g(s)}Z(s)\mathbf{x}(s)\\ \int_a^s \sqrt{g(s)}Z(s,\theta)\mathbf{x}(\theta)d\theta\\ \int_s^b \sqrt{g(s)}Z(s,\theta)\mathbf{x}(\theta)d\theta \end{bmatrix}.$$

Then by expanding out the composition formulae, we find $\mathcal{P}_{\{N_i\}} = \mathcal{Z}^* P \mathcal{Z}$ and since $P \geq 0$, $P = (P^{\frac{1}{2}})^T P^{\frac{1}{2}}$ for some $P^{\frac{1}{2}}$. Thus

$$\begin{aligned}
\left\langle \mathbf{x}, \mathcal{P}_{\{N_i\}}\mathbf{x}\right\rangle_{L_2} &= \langle \mathcal{Z}\mathbf{x}, P\mathcal{Z}\mathbf{x}\rangle_{L_2}\\
&= \left\langle P^{\frac{1}{2}}\mathcal{Z}\mathbf{x}, P^{\frac{1}{2}}\mathcal{Z}\mathbf{x}\right\rangle_{L_2} \geq 0. \quad \square
\end{aligned}$$

Note that Proposition 17 does not ensure that $\mathcal{P}_{\{N_i\}}$ is coercive. To obtain a coercive operator, one must add a coercive term of the form $\mathcal{P}_{\{\alpha I,0,0\}}$ for some $\alpha > 0$.

When we desire the $\{N_i\}$ to be polynomial, we may choose Z to be the vector of monomials of bounded degree, d. For $g(s) = 1$, the operators are positive on any domain. However, for $g(s) = (s-a)(b-s)$ the operator is only positive on the given domain $[a,b]$. For the most accurate results, we combine both choices of g. For notational convenience, we define the set of functions which satisfy Proposition 17 in this way. Specifically, we denote $Z_d(x)$ as the matrix whose rows are a vector monomial basis for the vector-valued polynomials of degree d or less and define the cone of positive operators with polynomial multipliers and kernels associated with degree d as

$\Omega_d := \{\mathcal{P}_{\{N_i\}} + \mathcal{P}_{\{M_i\}}$: $\{N_i\}$ and $\{M_i\}$ satisfy the conditions of Proposition 17 with $Z = Z_d$ and where $g(s) = 1$ and $g(s) = (s-a)(b-s)$, resp.$\}$

The dimension of the matrices M_i and N_i should be clear from context. The constraint $\mathcal{P}_{\{R_i\}} \in \Omega_d$ is then an LMI constraint on the coefficients of the polynomials $\{R_i\}$ and guarantees that $P_{\{R_i\}} \geq 0$. A Matlab toolbox (PIETOOLS) for setting up and solving LPIs based on Proposition 17 has recently been proposed and is discussed in Section 9.3.

9.2. The degree-bounded stability test

By restricting the degree of the polynomial parameters, $\{N_i\}$, we obtain a PIETOOLS-based LMI which enforces the LPI conditions of Theorem 16.

Theorem 18. *For any $d \in \mathbb{N}$, suppose there exist $\alpha, \delta > 0$, and polynomials $N_0 : [a,b] \to \mathbb{R}^{n\times n}$, $N_1, N_2 : [a,b]^2 \to \mathbb{R}^{n\times n}$ such that*

$$\mathcal{P} := \mathcal{P}_{\{N_0-\alpha I, N_1, N_2\}} \in \Omega_d$$

and

$$-\delta\mathcal{T}^*\mathcal{T} - \mathcal{A}^*\mathcal{P}\mathcal{T} - \mathcal{T}^*\mathcal{P}\mathcal{A} \in \Omega_d$$

where $\mathcal{T}$ and $\mathcal{A}$ are as defined in Eq. (9). Then any solution the PDE defined by $\{A_i, X\}$ is exponentially stable in L_2.

Note that as mentioned in the previous subsection, the constraint $\in \Omega_d$ is an LMI constraint.

9.3. PIETOOLS implementation

In this subsection, we give sample code using the PIETOOLS 2020 toolbox which verifies that the conditions of Theorem 18 are satisfied.

A detailed manual for the PIETOOLS 2020 toolbox can be found in Shivakumar et al. (2020). This toolbox allows for declaration and manipulation of 3-PI operators and 3-PI decision variables and enforcement of LPI constraints. PIETOOLS uses aspects of the SOSTOOLS LMI conversion process and pvar polynomial objects as defined in MULTIPOLY. PIETOOLS defines the `opvar` class of PI operators and overloads the multiplication (∗), addition (+) and adjoint (’) operations using the formulae in Lemmas 4 and 6. Concatenation, and scalar multiplication are likewise defined so that 3-PI operators can be treated in a similar manner to matrices.

To facilitate implementation of the conditions of Theorem 18, we have created the script `PIETOOLS_PDE`, which is distributed with the PIETOOLS 2020 toolbox. To use this script *only* requires the user to define the standardized form of the PDE — as illustrated in Step (3) below. Specifically, the user must define `n0,n1,n2,A0,A1,A2,B`, although `A2` may be omitted if `n2=0`. The user specifies that a stability test is desired by setting `stability=1` and can specify the desired accuracy through `settings_PIETOOLS` scripts, although by default we use the `settings_PIETOOLS_light` script, which corresponds to $d = 1$. An overview of the steps included in the `PIETOOLS_PDE` script is provided below along with a brief description of each step.

(1) Define independent polynomial variables. These are the spatial variables in the PDE.

```
pvar s,th;
```

(2) Initialize an optimization problem structure, X.

```
X = sosprogram([s,th]);
```

(3) Define the standardized PDE representation and use the provided script (convert_PIETOOLS_pde) to construct $\mathcal{T}$ and $\mathcal{A}$ using the formulae in Lemma 14.

```
stability=1;
n1=..;n2=..;n3=..;
A0=..;A1=..;A2=..;B=..;
convert_PIETOOLS_pde;
```

(4) Declare the positive 3-PI operator $\mathcal{P}$ and add inequality constraints. Proposition 17 is used to construct $\mathcal{P}$ and enforce the constraint $\mathcal{A}^*\mathcal{P}\mathcal{T} + \mathcal{T}^*\mathcal{P}\mathcal{A} \leq -\delta\mathcal{T}^*\mathcal{T}$ in Theorem 16. Transpose and composition in the term $\mathcal{A}^*\mathcal{P}\mathcal{T}$ are performed using the formulae in Lemmas 6 and 4. If feasible, these steps thus enforce the conditions in Theorem 16 where δ is determined by the user choice of epneg (default value is $\delta = 0$). In this code, n is state dimension, I is the interval $[a, b]$, and d is the degree of the polynomial parameters in $\mathcal{P}$. This step is executed automatically (using executive_PIETOOLS_stability) if the user has declared the option stability=1.

```
[X,P] = poslpivar(X,n,I,d);
D = -epneg*T'*T-A'*P*T-T'*P*A;
X = lpi_ineq(X,D);
```

(5) Call the SDP solver.

```
X = sossolve(X);
```

(6) Get the solution. P_s is the 3-PI operator, $\mathcal{P}$.

```
P_s = sosgetsol_opvar(X,P);
```

We conclude that if PIETOOLS finds the LPI is feasible for $\delta = 0$, the system is stable as per Theorem 16. If $\delta > 0$, we conclude exponential stability. Note that the degree, d, enters at Step (4) and is defined in the settings script, which defaults to settings_PIETOOLS_light ($d = 1$). If higher degree is required, the setting may be changed manually or using the settings_PIETOOLS_heavy ($d = 2$) script. Instructions for declaring the PDE in form of Eq. (2) are included in the header to PIETOOLS_PDE.

10. Numerical tests of accuracy and scalability

In this section, we examine the accuracy and computational complexity (scalability) of the proposed stability analysis algorithm by applying Theorem 18 to several well-studied and relatively trivial test cases. The algorithms are implemented using the PIETOOLS toolbox described in the previous section, and use the settings_PIETOOLS_light ($d = 1$) option. All computation times are listed for an Intel i7-6950x processor with 64 GB RAM and only account for time taken to solve the resulting LMI using Sedumi, excluding time taken for problem setup and polynomial manipulations. In cases where the limiting value of a parameter is listed for which the system is stable, the limiting value was determined using a bisection on that parameter.

Example 1. We begin with several variations of the diffusion equation. The first is adapted from Valmorbida et al. (2014),

$$\dot{x}(t,s) = \lambda x(t,s) + x_{ss}(t,s)$$

where $x(0) = x(1) = 0$ and which is known to be stable if and only if $\lambda < \pi^2 = 9.869604\cdots$. For the choice of $d = 1$ in Theorem 18, the algorithm is able to prove stability for $\lambda \leq 9.8696$ with a computation time of .54 s.

Example 2. The second example from Valmorbida et al. (2016) is the same, but changes the boundary conditions to $x(0) = 0$ and $x_s(1) = 0$ and is unstable for $\lambda > 2.467$. For $d = 1$, the algorithm is able to prove stability for $\lambda \leq 2.467$ with identical computation time.

Example 3. The third example from Gahlawat and Peet (2017) is not homogeneous

$$\begin{aligned}\dot{x}(t,s) &= (-.5s^3 + 1.3s^2 - 1.5s + .7 + \lambda)x(t,s)\\ &\quad + (3s^2 - 2s)x_s(t,s) + (s^3 - s^2 + 2)x_{ss}(t,s)\end{aligned}$$

where $x(0) = 0$ and $x_s(1) = 0$ and was estimated numerically to be unstable for $\lambda > 4.65$. For $d = 1$, the algorithm is able to prove stability for $\lambda \leq 4.65$ with similar computation time (compare to $\lambda = 4.62$ in Gahlawat & Peet, 2017).

Example 4. In this example from Valmorbida et al. (2016), we have

$$\dot{x}(t,s) = \begin{bmatrix} 0 & 0 & 0\\ s & 0 & 0\\ s^2 & -s^3 & 0\end{bmatrix} x(t,s) + R^{-1}x_{ss}(t,s)$$

with $x(0) = 0$ and $x_s(1) = 0$. In this case, using $d = 1$, we were able to prove stability for any tested value of R (vs. $R \leq 21$ in Valmorbida et al., 2016) with a computation time of 4.06 s. No upper limit was found.

Example 5. For our last numerical comparison, we consider some of the recent literature on coupled linear hyperbolic systems (Diagne, Bastin, & Coron, 2012; Lamare, Girard, & Prieur, 2016; Saba et al., 2019), often representing conservation or balance laws. Although there are several variations of the problem formulation, we consider the recent work of Saba et al. (2019), as it seems to contain the most accurate results. Consider

$$\dot{x}(t,s) = \underbrace{\begin{bmatrix} 0 & \sigma_1\\ \sigma_2 & 0\end{bmatrix}}_{A_0} x(t,s) + \underbrace{\begin{bmatrix} -\frac{1}{r_1} & 0\\ 0 & \frac{1}{r_2}\end{bmatrix}}_{A_1} x_s(t,s)$$

with boundary conditions $x_1(0) = qx_2(0)$ and $x_2(1) = \rho x_1(1)$. In this case, we have

$$B = \begin{bmatrix} 1 & -q & 0 & 0\\ 0 & 0 & -\rho & 1\end{bmatrix}.$$

Using $d = 1$, $r_1 = 0$, $r_2 = 1.1$, $\sigma_1 = 1$, $q = 1.2$, by gridding the parameters σ_2 and ρ, we are able to verify stability for all stable parameter values indicated in Figure 5 in Saba et al. (2019). For example, at $\rho = -.4$, we were able to prove stability for $\sigma_2 \leq 1.048$.

Example 6 (*Scalability*)**.** Finally, we explore computational complexity using a simple n-dimensional diffusion equation

$$\dot{x}(t,s) = x(t,s) + x_{ss}(t,s)$$

where $x(t,s) \in \mathbb{R}^n$. We then evaluate the computation time to perform the feasibility test for different size problems, from $n = 1$ to $n = 40$, choosing $d = 1$ – See Table 1. Note that no factors other than d influence computation time and the result is always stability.

11. Illustrations of beam and wave equations

In Section 10, we demonstrated that the proposed stability test has no obvious conservatism by finding parameter values corresponding to the stability limit for several well-studied examples.

Table 1
Number of PDEs vs. computation time for stability test.

n	1	5	10	20	30	40
s	.504	1.907	71.63	2706	23 920	103 700

However, the representation of these PDEs in the generalized PDE form of Eq. (2) was straightforward. In this section, we provide some guidance on how the user might identify the A_i and B matrices in Eq. (2) for several less obvious examples — focusing on four well-known wave and beam examples. The beam examples are particularly interesting in that (to the best of our knowledge) they have not previously been analyzed using LMI-based methods. As we proceed, we call particular attention to the following two questions.

- What are the states?
- What are the boundary conditions?

Choice of state: Prior to the introduction of state–space, ODEs would often be represented using scalar equations. For example, the spring-mass:

$$m\ddot{x}(t) = -c\dot{x}(t) - kx(t) + F(t)$$

is a scalar ODE. To represent this in the vector-valued state–space framework, we use $x_1 = x$ and define an auxiliary state $x_2 = \dot{x}$. Similarly, PDEs are often represented as scalar equations using higher-order time derivatives (e.g. The wave equation is $\ddot{w} = w_{xx}$). The standardized PDE representation in Eq. (2), however, uses only first-order time derivatives. Furthermore, as discussed in Section 3.3, the use of the standardized representation occasionally involves loss of some state information and may affect the question of stability. Specifically, the exponential stability criterion in Theorem 18 implies all states decay exponentially. For example, If a PDE is L_2-stable in u, but not u_s, then if u_s is included in the standardized representation, the PIE stability analysis will not be able to verify stability.

Boundary conditions: Identification of a sufficient number of boundary conditions in the universal framework is particularly important. For the B matrix to have sufficient rank, the solution must be uniquely defined (which may prohibit periodic boundary conditions). One consideration to be aware of is that when we introduce additional variables to eliminate higher-order time-derivatives, these new variables must also have associated boundary conditions. This is typically solved by differentiating the original boundary conditions in time to obtain boundary conditions for the new variables.

In the following examples, we illustrate the process of choosing state and constructing the A_i and B matrices.

11.1. Beam equation examples

We first consider both the Euler–Bernoulli (E-B) and Timoshenko (T) beam equations. This case is particularly interesting, as the E-B model is fundamentally diffusive and the T model has hyperbolic character. Furthermore, both these models are known to be energy-conserving (Luo et al., 2012), meaning that they are stable, but not exponentially stable.

Euler–Bernoulli: In this first case, we simply recall our formulation of the cantilevered E-B beam from Section 3.3:

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0 & -c \\ 1 & 0 \end{bmatrix}}_{A_2} \mathbf{x}_{ss}(t)$$

where $A_0 = A_1 = 0$, $n_2 = 2$, and $n_0 = n_1 = 0$. The boundary conditions take the form

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{B} \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_1(L) \\ u_2(L) \\ u_{1s}(0) \\ u_{2s}(0) \\ u_{1s}(L) \\ u_{2s}(L) \end{bmatrix} = 0.$$

Entering $\{A_i, B\}$ into the script `PIETOOLS_PDE`, and applying the results of Section 9.3 for `epneg=0`, we conclude the E-B beam is stable (using $\delta = 0$ in Theorem 18) for any tested value of $c > 0$. However, when `epneg=.0001`, we have $\delta > 0$, and PIETOOLS is unable to find a Lyapunov function, indicating this formulation is not exponentially stable (as expected).

Timoshenko beam We now consider the Timoshenko beam model where, for simplicity, we set $\rho = E = I = \kappa = G = 1$:

$$\begin{aligned} \ddot{w} &= \partial_s(w_s - \phi) &&= -\phi_s + w_{ss} \\ \ddot{\phi} &= \phi_{ss} + (w_s - \phi) &&= -\phi + w_s + \phi_{ss} \end{aligned}$$

with boundary conditions of the form

$$\begin{aligned} &\phi(0) = 0, && w(0) = 0, \\ &\phi_s(L) = 0, && w_s(L) - \phi(L) = 0. \end{aligned}$$

Our first step is to eliminate the second-order time-derivatives, and hence we choose $u_1 = \dot{w}$ and $u_3 = \dot{\phi}$. Using the boundary conditions as a guide, we choose the remaining states as $u_2 = w_s - \phi$ and $u_4 = \phi_s$. Note that this choice of states is a natural set of coordinates as the Timoschenko beam is known to be energy conserving with respect to these states (Luo et al., 2012). In summary, we have

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \dot{w} \\ w_s - \phi \\ \dot{\phi} \\ \phi_s \end{bmatrix}.$$

This gives us 4 first order boundary conditions

$$u_1(0) = 0, \quad u_3(0) = 0, \quad u_4(L) = 0, \quad u_2(L) = 0.$$

Reconstructing the dynamics, we now have

$$\begin{aligned} \dot{u}_1 &= u_{2s}, & \dot{u}_2 &= u_{1s} - u_3 \\ \dot{u}_3 &= u_{4s} + u_2, & \dot{u}_4 &= u_{3s}. \end{aligned}$$

Expressing this in our standard form we have the purely hyperbolic construction

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_0} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{A_1} \begin{bmatrix} u_{1s} \\ u_{2s} \\ u_{3s} \\ u_{4s} \end{bmatrix}$$

where $A_2 = [\,]$ and $n_0 = n_2 = 0$ and $n_1 = 4$. The B matrix is then

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{B} \begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \\ u_4(0) \\ u_1(L) \\ u_2(L) \\ u_3(L) \\ u_4(L) \end{bmatrix} = 0$$

where B has row rank $n_1 = 4$ and satisfies Eq. (4). The script `PIETOOLS_PDE` indicates this system is stable (using $\delta = 0$ in Theorem 18 and `epneg=0` in PIETOOLS). However, when $\delta > 0$, the code is unable to find a Lyapunov function, indicating this formulation is not exponentially stable (as expected).

11.2. Wave equation with boundary feedback examples

In this subsection, we consider wave equations attached at one end and free at the other with damping at the free end. This is a well-studied problem for which numerous stability results are available in the literature (Chen, 1979; Datko, Lagnese, & Polis, 1986). The simplest formulation is

$$\begin{aligned}&\ddot{u}(t,s)=u_{ss}(t,s)\\&u(t,0)=0 \qquad u_s(t,L)=-k\dot{u}(t,L).\end{aligned}$$

As with the beam examples, this has a purely hyperbolic formulation. Guided by the boundary conditions, we choose

$$u_1(t,s)=\dot{u}(t,s), \qquad u_2(t,s)=u_s(t,s).$$

This yields

$$\begin{bmatrix}\dot{u}_1\\ \dot{u}_2\end{bmatrix}=\underbrace{\begin{bmatrix}0&1\\1&0\end{bmatrix}}_{A_1}\begin{bmatrix}u_{1s}\\u_{2s}\end{bmatrix}$$

where $A_0=0$, $A_2=[\,]$, $n_1=n_2=0$ and $n_1=2$. The boundary conditions are now

$$\underbrace{\begin{bmatrix}0&1&0&0\\0&0&k&1\end{bmatrix}}_{B}\begin{bmatrix}u(0)\\u(L)\end{bmatrix}=0.$$

This formulation is computed to be exponentially stable (using $\delta=.1$ in Theorem 16 or `epneg=.1` in PIETOOLS) in the given state u_t, u_s for any tested value of $k>0$. We now consider a variation on this formulation.

Diffusive formulation As a variation, we consider a non-diffusive formulation from Chen (1979) which was shown to be asymptotically stable in the state u for $a^2+k^2>0$.

$$\begin{aligned}&\ddot{u}(t,s)=u_{ss}(t,s)-2a\dot{u}(t,s)-a^2u(t,s),\ s\in[0,1]\\&u(t,0)=0, \qquad u_s(t,1)=-k\dot{u}(t,1)\end{aligned}$$

In this case, we are forced to choose the variables $u_1=u_t$ and $u_2=u$ yielding the diffusive formulation

$$\begin{bmatrix}\dot{u}_1\\ \dot{u}_2\end{bmatrix}=\underbrace{\begin{bmatrix}-2a&-a^2\\1&0\end{bmatrix}}_{A_0}\begin{bmatrix}u_1\\u_2\end{bmatrix}+\underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{A_2}u_{2ss}$$

where $A_1=0$, $n_0=0$, $n_1=1$, and $n_2=1$. Note in this case that the boundary conditions on u_1 force us to consider this a hyperbolic state and the boundary conditions on u_2 make this a diffusive state! These boundary conditions are now expressed as

$$\begin{bmatrix}0&0&1&0&0&0\\1&0&0&0&0&0\\0&k&0&0&0&1\end{bmatrix}\begin{bmatrix}u_1(0)\\u_1(L)\\u_2(0)\\u_2(L)\\u_{2s}(0)\\u_{2s}(L)\end{bmatrix}=0.$$

Computation indicates this model is neutrally stable, but not exponentially stable in the given state — a result confirmed in Chen (1979) and Datko et al. (1986).

12. Conclusion

In this paper, we have shown how to use LMIs to accurately test stability of a large class of coupled linear PDEs. To achieve this result, we have defined a unitary state transformation which allows us to convert well-posed coupled linear PDEs – defined on state $\mathbf{x}_p\in X$, with associated boundary conditions and continuity constraints – to Partial-Integral Equations (PIEs) with state $\mathbf{x}_f\in L_2$ — a formulation which is defined using the algebra of 3-PI partial-integral operators and which does not require boundary conditions or continuity constraints on $\mathbf{x}_f$. We have shown that stability of PDEs can be reformulated as a Linear PI Inequality (LPI) expressed using 3-PI operators and operator positivity constraints. We have shown how to parameterize 3-PI operators using polynomials and how to enforce positivity of 3-PI operators using LMI constraints on the coefficients of these polynomials. We have used the Matlab toolbox PIETOOLS to solve the resulting LPIs and applied the results to a variety of numerical examples. The numerical results indicate little or no conservatism in the resulting stability conditions to several significant figures even for low polynomial degree. By conversion of LMIs developed for ODEs to LPIs, it is possible that these results can be extended to: PDEs with uncertainty; H_∞-gain analysis of PDEs; H_∞-optimal observer synthesis for PDEs; and H_∞-optimal control of PDEs. In addition, it is possible that the framework may be extended to multiple spatial dimensions using the multivariate representation proposed in Peet (2009).

References

Aamo, O. (2013). Disturbance rejection in 2 x 2 linear hyperbolic systems. *IEEE Transactions on Automatic Control*, *58*(5), 1095–1106.
Ahmadi, M., Valmorbida, G., Gayme, D., & Papachristodoulou, A. (2019). A framework for input-output analysis of wall-bounded shear flows. *Journal of Fluid Mechanics*, *873*, 742–785.
Ahmadi, M., Valmorbida, G., & Papachristodoulou, A. (2016). Dissipation inequalities for the analysis of a class of PDEs. *Automatica*, *66*, 163–171.
Balas, M. (1978). Active control of flexible systems. *Journal of Optimization Theory and Applications*, *25*(3), 415–436.
Barreau, M., Scherer, C., Gouaisbaut, F., & Seuret, A. (2020). Integral quadratic constraints on linear infinite-dimensional systems for robust stability analysis. arXiv preprint arXiv:2003.06283.
Bastin, G., & Coron, J.-M. (2016). *Stability and boundary stabilization of 1-D hyperbolic systems, Vol. 88*. Springer.
Bensoussan, A., Prato, G. D., Delfour, M. C., & Mitter, S. K. (1992). *Representation and control of infinite dimensional systems volume I*. Birkhäuser.
Chen, G. (1979). Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain. *Journal de Matheématiques Pures et Appliquées*, *58*, 249–273.
Curtain, R., & Zwart, H. (1995). *An introduction to infinite-dimensional linear systems theory*. Springer-Verlag.
Datko, R. (1970). Extending a theorem of A. M. Liapunov to Hilbert space. *Journal of Mathematical Analysis and Applications*, *32*(3), 610–616.
Datko, R., Lagnese, J., & Polis, M. (1986). An example on the effect of time delays in boundary feedback stabilization of wave equations. *SIAM Journal on Control and Optimization*, *24*(1), 152–156.
Diagne, A., Bastin, G., & Coron, J.-M. (2012). Lyapunov exponential stability of 1-D linear hyperbolic systems of balance laws. *Automatica*, *48*(1), 109–114.
Fridman, E., & Orlov, Y. (2009). An LMI approach to H_∞ boundary control of semilinear parabolic and hyperbolic systems. *Automatica*, *45*(9), 2060–2066.
Fridman, E., & Terushkin, M. (2016). New stability and exact observability conditions for semilinear wave equations. *Automatica*, *63*, 1–10.
Gahlawat, A., & Peet, M. (2015). Output feedback control of inhomogeneous parabolic PDEs with point actuation and point measurement using SOS and semi-separable kernels. In *Proceedings of the IEEE conference on decision and control*.
Gahlawat, A., & Peet, M. (2016). Optimal state feedback boundary control of parabolic PDEs using SOS polynomials. In *Proceedings of the American control conference*.
Gahlawat, A., & Peet, M. (2017). A convex sum-of-squares approach to analysis, state feedback and output feedback control of parabolic PDEs. *IEEE Transactions on Automatic Control*, *62*(4), 1636–1651.
Gahlawat, A., & Valmorbida, G. (2017). A semi-definite programming approach to stability analysis of linear partial differential equations. In *Proceedings of the IEEE conference on decision and control* (pp. 1882–1887).
Gaye, O., Autrique, L., Orlov, Y., Moulay, E., Brémond, S., & Nouailletas, R. (2013). H_∞ stabilization of the current profile in tokamak plasmas via an LMI approach. *Automatica*, *49*(9), 2795–2804.
Karafyllis, I., & Krstic, M. (2019). *Input-to-state stability for PDEs*. Springer.
Krstic, M., & Smyshlyaev, A. (2008). *Boundary control of PDEs: A course on backstepping designs, Vol. 16*. SIAM.
Lamare, P.-O., Girard, A., & Prieur, C. (2016). An optimisation approach for stability analysis and controller synthesis of linear hyperbolic systems. *ESAIM. Control, Optimisation and Calculus of Variations*, *22*(4), 1236–1263.
Lasiecka, I., & Triggiani, R. (2000). *Control theory for partial differential equations: Volume 1, Abstract parabolic systems: Continuous and approximation theories*. Cambridge University Press.

Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in MATLAB. In *Computer aided control systems design, 2004 IEEE international symposium on* (pp. 284–289).

Luo, Z.-H., Guo, B.-Z., & Morgül, O. (2012). *Stability and stabilization of infinite dimensional systems with applications*. Springer Science & Business Media.

Marion, M., & Temam, R. (1989). Nonlinear Galerkin methods. *SIAM Journal on Numerical Analysis*, *26*(5), 1139–1157.

Peet, M. M. (2009). Exponentially stable nonlinear systems have polynomial Lyapunov functions on bounded regions. *IEEE Transactions on Automatic Control*, *52*(5).

Peet, M. (2018a). A new state-space representation for coupled PDEs and scalable Lyapunov stability analysis in the SOS framework. In *Proceedings of the IEEE conference on decision and control*.

Peet, M. (2018b). *A partial integral equation (PIE) representation of coupled linear pdes and scalable stability analysis using LMIs*: *Tech. Rep.*, arXiv.org, http://arxiv.org/abs/1812.06794.

Peet, M. (2019). A dual to Lyapanov's second method for linear systems with multiple delays and implementation using SOS. *IEEE Transactions on Automatic Control*, *64*(3), 944–959.

Ravindran, S. (2000). A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, *34*(5), 425–448.

Rowley, C. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, *15*(03), 997–1013.

Saba, D., Argomedo, F., Auriol, J., Loreto, M. D., & Meglio, F. D. (2019). Stability analysis for a class of linear 2x2 hyperbolic PDEs using a backstepping transform. *IEEE Transactions on Automatic Control*.

Safi, M., Baudouin, L., & Seuret, A. (2017). Tractable sufficient stability conditions for a system coupling linear transport and differential equations. *Systems & Control Letters*, *110*, 1–8.

Shivakumar, S., Das, A., & Peet, M. (2020). PIETOOLS: a Matlab toolbox for manipulation and optimization of partial integral operators. In *Proceedings of the American control conference*. http://control.asu.edu/pietools.

Smyshlyaev, A., & Krstic, M. (2005). Backstepping observers for a class of parabolic PDEs. *Systems & Control Letters*, *54*(7), 613–625.

Solomon, O., & Fridman, E. (2015). Stability and passivity analysis of semilinear diffusion PDEs with time-delays. *International Journal of Control*, *88*(1), 180–192.

Valmorbida, G., Ahmadi, M., & Papachristodoulou, A. (2014). Semi-definite programming and functional inequalities for distributed parameter systems. In *Proceedings of the IEEE conference on decision and control* (pp. 4304–4309).

Valmorbida, G., Ahmadi, M., & Papachristodoulou, A. (2016). Stability analysis for a class of partial differential equations via semidefinite programming. *IEEE Transactions on Automatic Control*, *61*(6), 1649–1654.

Villegas, J. (2007). *A port-Hamiltonian approach to distributed parameter systems*. (Ph.D. dissertation).

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.