2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Gain-Only Neural Operator Approximators of PDE Backstepping Controllers

Rafael Vazquez and Miroslav Krstic

***Abstract*— For the recently introduced deep learning-powered approach to PDE backstepping control, we present an advancement applicable across all the results developed thus far: approximating the control gain function only (a function of one variable), rather than the entire kernel function of the backstepping transformation (a function of two variables). We introduce this idea on a couple benchmark (unstable) PDEs, hyperbolic and parabolic. We alter the approach of quantifying the effect of the approximation error by replacing a backstepping transformation that employs the approximated kernel (suitable for adaptive control) by a transformation that employs the exact kernel (suitable for gain scheduling). A major simplification in the target system arises, with the perturbation due to the approximation shifting from the domain to the boundary condition. This results in a significant difference in the Lyapunov analysis, which nevertheless results in a guarantee of the stability being retained with the simplified approximation approach. The approach of approximating only the control gain function simplifies the operator being approximated and the training of its neural approximation, with an expected reduction in the neural network size. The price for the savings in approximation is paid through a somewhat more intricate Lyapunov analysis, in higher Sobolev spaces for some PDEs, as well as some restrictions on initial conditions that result from higher Sobolev spaces. It is essential to carefully consider the specific requirements and constraints of each problem to determine the most appropriate approach; indeed, recent works have demonstrated the successful application of both full-kernel and gain-only approaches in both adaptive control and gain scheduling contexts.**

I. Introduction

A. Deep learning-powered PDE backstepping

In the field of control of partial differential equations (PDEs), recent advancements have emerged in harnessing deep learning to expedite the computation of gains for model-driven control laws using the backstepping method [4], [19], [29], [40]. These advancements are grounded on a novel neural network breakthrough and its underlying mathematical foundation, termed as DeepONet [27]. This approach can generate close approximations of nonlinear operators, capturing solutions that arise from PDEs defining the gains of controllers. The DeepONet paradigm extends the "universal approximation theorem" for functions [8], [14] to offer a universal approximation for nonlinear operators [6], [23], [24], [26], [27]. For specific PDEs associated with the computation of stabilizing control laws, as the kernel equations arising in the backstepping method, any alteration in the plant parameter functions necessitates finding a new solution.

R. Vazquez is with the Department of Aerospace Engineering, Universidad de Sevilla, Camino de los Descubrimiento s.n., 41092 Sevilla, Spain.
M. Krstic is with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093-0411, USA.

With the deep learning approach, one requires merely a recomputation of the backstepping kernels via a pre-learned DeepONet map, which are then evaluated at certain values to obtain the gain kernels. Thus, this neural architecture embeds the transformation from the plant's functional coefficients to the controller gain kernels, facilitating gain computation as a mere function evaluation, circumventing the need of numerically solving the kernel equations. This methodology for PDE control retains the theoretical assurances of a nominal closed-loop system in its approximate form, and so far it has been developed for both delay-absent hyperbolic [4] and parabolic [19] PDEs, as well as delay-compensating control of hyperbolic partial integro-differential equations (PIDEs) [29] and reaction-diffusion PDEs [40].

B. Contribution of this work

The novelty of this work is proposing a methodology to directly approximate backstepping gains by using neural operators, skipping the need of *approximating* the full kernels as in the previous results. This approach is illustrated with several simple cases: two parabolic cases (respectively, with Dirichlet and Neumann boundary conditions) and a 1-D hyperbolic PIDE equation.

The basic idea is to use the exact, rather than the approximated, kernel in the backstepping transformation and, as a result, to express the discrepancy due to the approximation as a perturbation on the boundary of the target system (produced using the unknown exact backstepping transformation), rather than as a perturbation in the domain of the target system. Our alternative approach leads to a more "unforgiving" perturbation, and induces some additional challenge in the stability analysis, but, as a reward for paying this analytical price, the approximation burden, measured in the training set computation and the size of the neural network, plausibly follows, due to the removal of the key previous restrictions on the approximate kernel.

The size of the perturbation at the target system's boundary is directly controlled by the quality of the approximation, which is in turn enforced by the universal approximation theorem. The requirements that suffice for the approximation to satisfy are established using a robust exponential stability analysis in functional norms that are appropriate for a non-local perturbation that acts at a boundary condition. In the examples tackled in the paper, it is sufficient to use the L^2 norm except in the reaction-diffusion equation with Dirichlet boundary conditions, which requires the use of the H^1 norm.

The results obtained in this work allow for arbitrary decay rates. Interestingly, while in the parabolic case increasing the rates requires both a different gain and a higher approximation quality, in the hyperbolic case the same gain results in an increased decay rate if it is better approximated, due to the finite-time decay of the target system with an exact kernel.

Given that the backstepping approach has become an ubiquitous technique in PDE control, advances in approximation of gains with learning approaches, such as the one we propose, can potentially be extended to many families of systems. Indeed, backstepping was originally introduced for feedback controls in one-dimensional reaction-diffusion PDEs [18], but it has since expanded to multi-dimensional applications [37] and has been employed in diverse systems like flow control [35], [38], heat loops [36], and more [1], [5], [7], [12], [17], [20], [28], [30], [34]. Since the approach presented in this paper is based on reusing both the target system and Lyapunov function used in the original design, the promise for extension of the method is remarkable, with some caveats, as explained next.

C. Comparison of approaches

The methodology of this paper contrasts with the full-kernel approximation strategies delineated in [4], [19], [29], [40] in several key aspects, each approach presenting unique advantages and challenges. In what follows the approach in the present paper will be referred to as the "gain-only approach", whereas the later will be referred to as the "full-kernel approach".

A salient advantage of the gain-only approach lies in its focus on approximating an inherently 1-dimensional gain kernel. This specificity paves the way for a more tractable loss function during training, streamlining the neural network design by reducing the number of hyperparameters. Such an approach not only necessitates a diminished training set size, attributed to the output function's singular argument, but also should lead to a marked reduction in the required training duration. Additionally, the analytical calculations needed to be performed to derive the "perturbed" target system are more straightforward, as backstepping transformation that employs the exact kernel circumvents the intricacies of derivatives or traces of the approximated kernels. The gain-only methodology also accommodates a broader range of coefficient smoothness, facilitating the inclusion of non-differentiable functions, even though functions should be at least Lipschitz continous for the universal approximation theorem to hold. In comparison, the full-kernel method entails more demanding requirements for the plant coefficients.

However, the simplicity of the gain-only approach is not without its drawbacks. The Dirichlet parabolic scenarios under this approach demand an H^1 analysis, inherently more challenging than the L^2 analysis, and calls for smoother initial conditions alongside compatibility conditions. Moreover, in all cases the need of bounds for inverse kernels may pose a challenge. Although they can be surmised indirectly from the direct kernel's bounds, the results are often conservatively skewed.

In instances where kernel approximation is crucial to design considerations, such as in adaptive control [31], one might expect that only the full-kernel approach would be applicable. However, recent works have successfully applied both full-kernel and gain-only approaches in this context. In [21], the authors first demonstrate the applicability of the full-kernel approach in adaptive control, where the design of the update law requires a target system based on the known approximate kernel. Additionally, they present an alternative approach using a passive identifier, which allows the gain-only approach to be used in adaptive control without requiring the approximation of the kernel's derivative.

Gain scheduling (GS) is another control technique that can benefit from kernel approximation. GS adapts the controller gains based on the current operating conditions of the system, allowing for improved performance over a wide range of operating points. In the context of PDEs, GS involves treating the plant coefficients as quasi-constant and updating the controller gains accordingly. The gain-only approach has been successfully applied to PDE GS in [22]. In this work, the target system under GS involves nonlinear perturbations resulting from both the quasi-constant treatment of plant coefficients and the approximation of the kernel. The gain-only approach simplifies the GS design by removing the perturbations due to kernel approximation while still handling the perturbations arising from the quasi-constant treatment of plant coefficients, demonstrating its effectiveness in GS applications.

While the gain-only approach provides several substantial improvements, it remains crucial to fully understand the trade-offs in specific contexts, as demonstrated by these recent works [21], [22]. The choice between full-kernel and gain-only approaches depends on the specific requirements and constraints of the problem at hand, and researchers should carefully consider the implications of each approach when designing their control strategies.

D. Generating the training set

It must be noted that both methods face an equal challenge in the generation of the training set, given that the controller gains (functions of a single argument, or as we may call them, "1D functions") are always produced as traces of the full backstepping kernels. Thus the numerical computation of the full 2D backstepping kernel cannot be skipped in the training of the 1D control gain functions.

The backstepping kernel equations are (typically) linear hyperbolic PDEs on a specific triangular domain, described by Goursat [13], with unique boundary conditions. The topic of numerical solution of Goursat PDEs for PDE backstepping has not been extensively addressed in the literature. Hints on numerical algorithms are scattered across various sources [2], [3], [11], [15], [16], [18], [36]. Advanced methods for Goursat problems [9], however, have not been utilized for backstepping kernel equations, and adapting these techniques can be complex, especially when dealing with discontinuities. A new rather general method based on power series approximations has been recently developed [33], and its extension to MATLAB [25] looks promising as a tool to generate training sets.

E. Structure of this paper

The gain approximation approach proposed in this paper is introduced through several examples, each of which finishes with a Theorem giving the conditions under which the feedback law using the gain approximated by a neural operator can provide exponential stability. We start with Section II with the easiest possible example, a 1-D hyperbolic PIDE plant. Next, in Section III, two reaction-diffusion cases are considered, namely the Dirichlet case and the Neumann case, which are treated in parallel. We then finish in Section IV with some concluding remarks. Note that proofs are skipped due to page limitation; they can be consulted in the ArXiv preprint of this paper [39].

II. 1-D HYPERBOLIC PIDE

Consider the plant

$$u_t = u_x + g(x)u(0,t) + \int_0^x f(x,y)u(y)dy, \quad (1)$$

$$u(1,t) = U(t). \quad (2)$$

where $u(x,t)$ is the state, $U(t)$ the actuation, $x \in [0,1]$, $t > 0$, for $f \in \mathscr{C}^0(\mathscr{T})$ and $g \in \mathscr{C}^0([0,1])$, where $\mathscr{T} = \{(x,y) : 0 \le y \le x \le 1\}$.

A. Backstepping feedback law design for 1-D hyperbolic PDEs

The backstepping method is based on the use of a direct/inverse backstepping transformation pair

$$w(x,t) = u(x,t) - \int_0^x K(x,\xi)u(\xi,t)d\xi, \quad (3)$$

$$u(x,t) = w(x,t) + \int_0^x L(x,\xi)w(\xi,t)d\xi, \quad (4)$$

where $K(x,\xi)$ and $L(x,\xi)$, are, respectively, the direct and inverse backstepping kernels, that verify hyperbolic PDEs (the kernel equations) [20] involving the coefficients f and g of the system, namely

$$K_x + K_\xi = \int_\xi^x K(x,s)f(s,\xi)ds - f(x,\xi), \quad (5)$$

$$K(x,0) = \int_0^x K(x,s)g(s)ds - g(x), \quad (6)$$

in the domain $\mathscr{T}$, and a very similar equation for $L(x,\xi)$.

Applying (3) to (1)–(2), with the kernel satisfying (5)–(6), one can find that the PDE verified by the new $w(x,t)$ variable, the target system, is

$$w_t = w_x, \quad (7)$$

$$w(1,t) = U(t) - \int_0^1 K(1,\xi)u(\xi,t)d\xi. \quad (8)$$

Thus, defining the feedback gain in terms of the backstepping kernel as

$$K_1(\xi) = K(1,\xi)\,, \quad (9)$$

and using the control law

$$U(t) = \int_0^1 K_1(\xi)u(\xi,t)d\xi\,, \quad (10)$$

one achieves a homogeneous boundary condition $w(1,t) = 0$ in (8) and, consequently, the target system becomes exponentially stable (in fact convergent to zero in finite time 1 due to the unity transport speed). Since the backstepping transformation is invertible, this implies exponential stability in the original plant coordinates $u(x,t)$, see [20] for details, where the following result, later important to our work, is stated and proven.

Theorem 1: Consider the equations verified by $K(x,\xi)$ (given by (5)–(6)) and $L(x,\xi)$ (given in [20]) in the domain $\mathscr{T}$ with $f \in \mathscr{C}^0(\mathscr{T})$ and $g \in \mathscr{C}^0([0,1])$. Then, there exists a unique solution $K, L \in \mathscr{C}^1(\mathscr{T})$, and denoting $\bar{g} = \|g\|_\infty = \max_{x\in[0,1]}|g(x)|$, $\bar{f} = \|f\|_\infty = \max_{(x,y)\in\mathscr{T}}|f(x,y)|$ and $\|K\|_\infty = \max_{(x,\xi)\in\mathscr{T}}|K(x,\xi)|$ (and similarly for L), one has

$$\|L\|_\infty, \|K\|_\infty \le \left(\bar{f} + \bar{g}\right)\mathrm{e}^{\bar{f}+\bar{g}} \quad (11)$$

B. Accuracy of approximation of backstepping 1-D hyperbolic gain operators with DeepONet

The main idea of this work (compared with the previous results [4], [19], [29], [40]) is to directly approximate the gain operator $K_1(\xi)$ by DeepONet as $\hat{K}_1(\xi)$ and thus apply an approximate feedback law $U(t) = \int_0^1 \hat{K}_1(\xi)u(\xi,t)d\xi$.

Let the operator $\mathscr{K}_1 : \mathscr{C}^0(\mathscr{T}) \times \mathscr{C}^0([0,1]) \to \mathscr{C}^1([0,1])$ be given by

$$K_1(x) =: \mathscr{K}_1(f,g)(x) \quad (12)$$

By applying the DeepONet universal approximation Theorem (see [10, Theorem 2.1]), we get the following key result for the approximation of the backstepping kernel gain (the $\mathscr{K}_1$ operator) by a DeepONet (see [4] for the exact definition of a neural operator). The proof of continuity and Lipschitzness is obtained by mimicking the successive approximation calculation in the proof of Theorem 1.

Theorem 2: For all B_f, $B_g > 0$ and $\epsilon > 0$, there exists a continuous and Lispschitz neural operator $\hat{\mathscr{K}}_1$ such that, for all $x \in [0,1]$,

$$\left|\mathscr{K}_1(f,g)(x) - \hat{\mathscr{K}}_1(f,g)(x)\right| < \epsilon \quad (13)$$

holds for all Lipschitz f and g with the properties that $\|f\|_\infty \le B_f$ and $\|g\|_\infty \le B_g$.

C. Stabilization of 1-D hyperbolic equations under DeepONet gain feedback

The following theorem states our main results regarding the stabilization properties of the backstepping design when the feedback gain is approximated by a DeepONet.

Theorem 3: Let $B_f > 0$, $B_g > 0$ and $c > 0$ be arbitrarily large and consider the system (1)–(2) for any $f \in \mathscr{C}^0(\mathscr{T})$ and $g \in \mathscr{C}^0([0,1])$, both Lipschitz functions, which satisfy $\|g\|_\infty \le B_g$ and $\|f\|_\infty \le B_f$. The feedback

$$U(t) = \int_0^1 \hat{K}_1(\xi)u(\xi,t)d\xi \quad (14)$$

with all NO gain kernels $\hat{K}_1 = \hat{\mathscr{K}}_1(f,g)$ of any approximation accuracy

$$0 < \epsilon < \epsilon^*(B_f, B_g, c) := \frac{\sqrt{\frac{c}{2\mathrm{e}^c}}}{1 + (B_f + B_g)\,\mathrm{e}^{B_f+B_g}} \quad (15)$$

in relation to the exact backstepping kernel gain $K_1 = \mathscr{K}_1(f,g)$ ensures that the closed-loop system satisfies the following L^2 exponential stability bound for some $M > 0$ with decay rate given by $c/8$:

$$\|u(\cdot,t)\|_{L^2} \le M\mathrm{e}^{-\frac{c}{8}(t-t_0)}\|u(\cdot,t_0)\|_{L^2} \quad (16)$$

Proof: Let $B_f > 0$, $B_g > 0$ and $c > 0$ be arbitrarily large. Considering the use of the feedback law $U(t) = \int_0^1 \hat{K}_1(\xi)u(\xi,t)d\xi$ in (8) we reach

$$w_t = w_x, \quad (17)$$

$$w(1,t) = \int_0^1 \left(\hat{K}_1(\xi) - K_1(\xi)\right)u(\xi,t)d\xi, \quad (18)$$

Now, using the exact inverse backstepping transformation,

$$w_t = w_x, \tag{19}$$

$$w(1,t) = \int_0^1 \left(\hat{K}_1(\xi) - K_1(\xi)\right) \left[w(\xi,t) + \int_0^\xi L(\xi,s)w(s,t)ds\right] d\xi, \tag{20}$$

and switching the order of integration in the second part of the integral, and calling

$$G(\xi) = -\tilde{K}_1(\xi) - \int_\xi^1 L(s,\xi)\tilde{K}_1(s)ds\,, \tag{21}$$

where $\tilde{K}_1(\xi) := K_1(\xi) - \hat{K}_1(\xi)$. We reach:

$$w_t = w_x, \tag{22}$$

$$w(1,t) = \int_0^1 G(\xi)w(\xi,t), \tag{23}$$

From Theorem 2, given B_g and B_f, and for any $\epsilon > 0$ such that $\epsilon < \epsilon^*$ with ϵ^* given in the Theorem statement, there exists a neural operator that ensures that

$$|\tilde{K}_1(\xi)| < \epsilon\,, \quad \forall \xi \in [0,1] \tag{24}$$

and therefore $|G(\xi)| \le \epsilon(1+\|L\|_\infty)\, \forall \xi \in [0,1]$ where $\|L\|_\infty$ also depends on B_f and B_g as stated in Theorem 1. Now define

$$V = \int_0^1 \mathrm{e}^{cx} w^2(x,t)dx \tag{25}$$

One then obtains

$$\dot{V} = 2\int_0^1 \mathrm{e}^{cx} w(x,t)w_x(x,t)dx \tag{26}$$

Note that

$$\int_0^1 \mathrm{e}^{cx} w(x,t)w_x(x,t)dx = -\frac{c}{2}\int_0^1 \mathrm{e}^{cx} w^2(x,t) + \frac{\mathrm{e}^c}{2} w^2(1,t) - \frac{1}{2}w^2(0,t)$$

Thus, defining $\|G\|_\infty = \max_{\xi\in[0,1]} |G(\xi)|$,

$$\begin{aligned} \dot{V} &\le -\frac{c}{2}\int_0^1 \mathrm{e}^{cx} w^2(x,t) + \frac{\mathrm{e}^c}{2}\left(\int_0^1 G(\xi)w(\xi,t)d\xi\right)^2 \\ &\le -\frac{c}{2}\int_0^1 \mathrm{e}^{cx} w^2(x,t) + \frac{\mathrm{e}^c\|G\|_\infty^2}{2}\int_0^1 w^2(\xi,t)d\xi \\ &\le -\left(\frac{c}{2} - \frac{\mathrm{e}^c\|G\|_\infty^2}{2}\right)V \end{aligned} \tag{27}$$

and since

$$\begin{aligned} \|G\|_\infty &\le \epsilon(1+\|L\|_\infty) \\ &< \epsilon^*\left(1 + (B_f+B_g)\mathrm{e}^{B_f+B_g}\right) \quad (28) \\ &\le \sqrt{\frac{c\mathrm{e}^{-c}}{2}} \quad (29) \end{aligned}$$

and thus we reach $\dot{V} \le -\frac{c}{4}V$ and the proof follows by the equivalence of V to the square of the L^2 norm of w and the use of the direct and inverse backstepping transformations, and their bounds, to express the obtained result in terms of the L^2 of u using the bounds of Theorem 1, see e.g. [4]. ■

III. Reaction-Diffusion equation

Consider the plant

$$u_t = u_{xx} + \lambda(x)u, \tag{30}$$

where $u(x,t)$ is the state, $x \in (0,1)$, $t > 0$, for $\lambda \in \mathscr{C}^0([0,1])$, with two possible boundary conditions (Dirichlet and Neumann cases). In the Dirichlet case we have

$$u(0,t) = 0, \quad u(1,t) = U(t), \tag{31}$$

with $U(t)$ being the actuation, whereas in the Neumann case we have

$$u_x(0,t) = 0, \quad u_x(1,t) = U(t). \tag{32}$$

A. Backstepping feedback law design for reaction-diffusion equations

As in the hyperbolic 1-D case, we employ a direct/inverse backstepping transformation pair defined exactly as (3)–(4). In this case, choosing some value of $c \ge 0$, the kernel equations verified by the direct transformation kernel [32] involving the coefficient λ of the system is as follows

$$K_{xx}(x,\xi) - K_{\xi\xi}(x,\xi) = (\lambda(\xi)+c)K(x,\xi), \tag{33}$$

$$K(x,x) = -\frac{1}{2}\int_0^x (\lambda(s)+c)\,ds, \tag{34}$$

in the domain $\mathscr{T}$, with the additional boundary condition $K(x,0) = 0$ in the Dirichlet case and $K_\xi(x,0) = 0$ in the Neumann case. A very similar equation is satisfied by $L(x,\xi)$.

1) Dirichlet case: Applying (3) to (30), (31), with the kernel satisfying (33)–(34) and $K(x,0) = 0$ condition, one can find that the PDE verified by the new $w(x,t)$ variable, the target system, is

$$w_t = w_{xx} - cw, \tag{35}$$

$$w(0,t) = 0, \tag{36}$$

$$w(1,t) = U(t) - \int_0^1 K(1,\xi)u(\xi,t)d\xi, \tag{37}$$

Thus, defining the feedback gain in terms of the backstepping kernel as in the hyperbolic 1-D case, $K_1^D(\xi) = K(1,\xi)$, and using the control law

$$U(t) = \int_0^1 K_1^D(\xi)u(\xi,t)d\xi\,, \tag{38}$$

one achieves a homogeneous boundary condition $w(1,t) = 0$ in (37) and, consequently, the target system becomes exponentially stable. Since the backstepping transformation is invertible, this implies exponential stability in the original plant coordinates $u(x,t)$, see [32] for details.

2) Neumann case: As in the Dirichlet case, applying (3) to (30), (32), with the kernel satisfying (33)–(34) and $K_\xi(x,0) = 0$ condition, one can find that the PDE verified by the new $w(x,t)$ variable, the target system, is

$$w_t = w_{xx} - cw, \tag{39}$$

$$w_x(0,t) = 0, \tag{40}$$

$$w_x(1,t) = -qw(1,t) + U(t) - (K(1,1)+q)u(1,t) - \int_0^1 \left(K_x(1,\xi) - qK(1,\xi)\right)u(\xi,t)d\xi, \tag{41}$$

where the term $-qw(1,t)$, where $q>0$ can take any value, adds some slight extra complications compared with the Dirichlet case. It has been added due to the reaction-diffusion equation with Neumann boundary conditions possessing a zero eigenvalue. By using $U(t) = (K(1,1)+q)u(1,t) + \int_0^1 (K_x(1,\xi) - qK(1,\xi))\, u(\xi,t)d\xi$, the target system becomes a stable reaction-diffusion equation if $q>0$.

Noting that $K(1,1)$ is directly obtained from (34), denote then the gain as $K_1^N(\xi) = K_x(1,\xi) - qK(1,\xi)$, and using the control law

$$U(t) = (K(1,1)+q)u(1,t) + \int_0^1 K_1^N(\xi)u(\xi,t)d\xi, \quad (42)$$

one achieves a boundary condition $w(1,t) = -qw(1,t)$ in (41) and, consequently, the target system becomes exponentially stable, see [32].

3) Kernel bounds for Dirichlet and Neumann cases: In both the Dirichlet and Neumann cases, the next result follows [32]

Theorem 4: Consider the equations verified by $K(x,\xi)$ (given by (33)–(34) and $K(x,0)=0$ in the Dirichlet case or $K_\xi(x,0)=0$ in the Neumann case) and $L(x,\xi)$ (given in [32]) in the domain $\mathscr{T}$ with $\lambda \in \mathscr{C}^0([0,1])$ and $c>0$. Then, there exists a unique solution $K, L \in \mathscr{C}^1(\mathscr{T})$, and denoting $\bar{\lambda} = \|\lambda\|_\infty = \max_{x\in[0,1]} |\lambda(x)|$, one has

$$\|L\|_\infty, \|K\|_\infty \leq \left(c+\bar{\lambda}\right) \mathrm{e}^{2\left(c+\bar{\lambda}\right)} \quad \text{(Dirichlet)} \quad (43)$$

$$\|L\|_\infty, \|K\|_\infty \leq 2\left(c+\bar{\lambda}\right) \mathrm{e}^{4\left(c+\bar{\lambda}\right)} \quad \text{(Neumann)} \quad (44)$$

B. Accuracy of approximation of backstepping reaction-diffusion gain operators with DeepONet

As in Section II-B, we approximate (for both Dirichlet and Neumann cases) the gain operator $K_1(\xi)$ by DeepONet as $\hat{K}_1(\xi)$ as defined for each case. Let the operators $\mathscr{K}_1^D : \mathscr{C}^0([0,1]) \times \mathbb{R}^+ \to \mathscr{C}^1([0,1])$ and $\mathscr{K}_1^N : \mathscr{C}^0([0,1]) \times \mathbb{R}^+ \times \mathbb{R}^+ \to \mathscr{C}^1([0,1])$ be given by

$$K_1^D(x) =: \mathscr{K}_1^D(\lambda,c)(x), \quad K_1^N(x) =: \mathscr{K}_1^N(\lambda,c,q)(x) \quad (45)$$

As in Theorem 2 for the hyperbolic case, we get the following key result for the approximation of the reaction-diffusion backstepping kernel gains by a DeepONet, which we state simultaneously both for the $\mathscr{K}_1^D$ and $\mathscr{K}_1^N$ operators (Dirichlet and Neumann cases).

Theorem 5: For all B_λ, $c>0$ and $\epsilon>0$ (and $q>0$ in the Neumann case), there exists a continuous and Lispschitz neural operator $\hat{\mathscr{K}}_1^D$ (resp. $\hat{\mathscr{K}}_1^N$ in the Neumann case) such that, for all $x\in[0,1]$, the following holds for all Lipschitz λ with the property that $\|\lambda\|_\infty \leq B_\lambda$:

$$\left|\mathscr{K}_1^D(\lambda,c)(x) - \hat{\mathscr{K}}_1^D(\lambda,c)(x)\right| < \epsilon \quad (46)$$

in the Dirichlet case or

$$\left|\mathscr{K}_1^N(\lambda,c,q)(x) - \hat{\mathscr{K}}_1^N(\lambda,c,q)(x)\right| < \epsilon \quad (47)$$

in the Neumann case.

C. Stabilization of reaction-difussion equations under DeepONet gain feedback

Next, we state our main stability results when the backstepping feedback gain is approximated by a DeepONet, both in the Dirichlet and Neumann cases, which are given separately due to substantial differences.

1) Dirichlet case: In the Dirichlet case, one obtains an H^1 stabilization result, as given next.

Theorem 6: Let $B_\lambda > 0$ and $c \geq 0$ be arbitrarily large and consider the system (30)–(31) for any $\lambda \in \mathscr{C}^0([0,1])$ a Lipschitz function which satisfies $\|\lambda\|_\infty \leq B_\lambda$. The feedback

$$U(t) = \int_0^1 \hat{K}_1(\xi)u(\xi,t)d\xi \quad (48)$$

with all NO gain kernels $\hat{K}_1^D = \hat{\mathscr{K}}_1^D(c,\lambda)$ of any approximation accuracy

$$0 < \epsilon < \epsilon^*(B_\lambda, c) := \frac{1}{\sqrt{20}\left(1 + (c+B_\lambda)\,\mathrm{e}^{2(c+B_\lambda)}\right)} \quad (49)$$

in relation to the exact backstepping kernel gain $K_1^D = \mathscr{K}_1^D(c,\lambda)$ ensures that the closed-loop system satisfies the following H^1 exponential stability bound with arbitrary decay rate:

$$\|u(\cdot,t)\|_{H^1} \leq M \mathrm{e}^{-\left(c+\frac{1}{12}\right)(t-t_0)} \|u(\cdot,t_0)\|_{H^1} \quad (50)$$

See the ArXiv preprint of this paper [39] for the proof.

2) Neumann case: In the Neumann case, one obtains an L^2 stabilization result, as given next.

Theorem 7: Let $B_\lambda > 0$, $q > 1$ and $c \geq 0$ be arbitrarily large and consider the system (30)–(32) for any $\lambda \in \mathscr{C}^0([0,1])$ a Lipschitz function which satisfies $\|\lambda\|_\infty \leq B_\lambda$. The feedback

$$U(t) = (K(1,1)+q)u(1,t) + \int_0^1 \hat{K}_1^N(\xi)u(\xi,t)d\xi \quad (51)$$

with all NO gain kernels $\hat{K}_1^N = \hat{\mathscr{K}}_1^N(c,\lambda,q)$ of any approximation accuracy

$$0 < \epsilon < \epsilon^*(B_\lambda, c, q) := \sqrt{\frac{q-1}{2}}\, \frac{1}{\left(1 + 2\,(c+B_\lambda)\,\mathrm{e}^{4(c+B_\lambda)}\right)} \quad (52)$$

in relation to the exact backstepping kernel gain $K_1^N = \mathscr{K}_1^N(c,\lambda,q)$ ensures the closed-loop system satisfies the following L^2 exponential stability bound with arbitrary decay:

$$\|u(\cdot,t)\|_{L^2} \leq M \mathrm{e}^{-\left(c+\frac{1}{8}\right)(t-t_0)} \|u(\cdot,t_0)\|_{L^2} \quad (53)$$

See the ArXiv preprint of this paper [39] for the proof.

IV. Conclusion

This work delved into the use of the DeepONet framework for computing gain kernels arising in the backstepping method for control of partial differential equations. A novel methodology to directly approximate backstepping gains using neural operators was introduced and validated across multiple case studies, including hyperbolic and parabolic plants. The efficacy of this method was critically examined in comparison to the previous approach fully approximating the backstepping kernel, highlighting advantages, disadvantages and inherent challenges.

A natural progression for this research is to extend the methodology to encompass other system configurations, such as hyperbolic coupled $n+m$, or parabolic coupled systems, or higher-dimensional geometries like the n-dimensional ball. Although foundational challenges presented by discontinuous kernels in hyperbolic and parabolic designs, as well as the complexities of hyperspherical harmonics in the n-dimensional ball setting should be tackled, the groundwork established in this study suggests that these extensions can be addressed methodically. The potential adaptability to coupled designs is particularly interesting. Managing the discontinuities in the gains that stem from the piecewise-only continuous kernels will be crucial. This may necessitate segmenting the kernels into multiple partitions for individual approximation, but does not preclude the application of the method.

Another interesting extension lies in the realm of developing observer gains. In this context, in-domain perturbations show up, as opposed to the boundary perturbations that appeared in thiw work. Nevertheless, one can expect similar complexities to those addressed herein, having to use similar Sobolev spaces and Lyapunov functionals and thus obtaining similar results regarding observer convergence.

ACKNOWLEDGMENTS

R. Vazquez was supported by grant TED2021-132099B-C33 funded by MICIU/ AEI/ 10.13039 /501100011033 and by "European Union NextGenerationEU/PRTR."

REFERENCES

[1] G. Andrade, R. Vazquez, and D. Pagano, "Backstepping stabilization of a linearized ODE-PDE Rijke tube model," *Automatica*, vol. 96, pp. 98–109, 2018.

[2] H. Anfinsen and O. Aamo, *Adaptive control of hyperbolic PDEs*. Springer, 2019.

[3] J. Auriol and D. Bresch-Pietri, "Robust state-feedback stabilization of an underactuated network of interconnected $n+m$ hyperbolic PDE systems," *Automatica*, vol. 136, p. 110040, 2022.

[4] L. Bhan, Y. Shi, and M. Krstic, "Neural operators for bypassing gain and control computations in PDE backstepping," *arXiv preprint arXiv:2302.14265*, 2023.

[5] G. Chen, R. Vazquez, and M. Krstic, "Rapid stabilization of Timoshenko beam by PDE backstepping," *IEEE Transactions on Automatic Control*, 2023.

[6] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.

[7] J.-M. Coron, R. Vazquez, M. Krstic, and G. Bastin, "Local exponential H^2 stabilization of a 2×2 quasilinear hyperbolic system using backstepping," *SIAM J. Control Optim.*, vol. 51, pp. 2005–2035, 2013.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[9] J. Day, "A Runge-Kutta method for the numerical solution of the Goursat problem in hyperbolic partial differential equations," *The Computer Journal*, vol. 9, no. 1, pp. 81–83, 1966.

[10] B. Deng, Y. Shin, L. Lu, Z. Zhang, and G. E. Karniadakis, "Approximation rates of deeponets for learning operators arising from advection–diffusion equations," *Neural Networks*, vol. 153, pp. 411–426, 2022.

[11] J. Deutscher and J. Gabriel, "A backstepping approach to output regulation for coupled linear wave-ODE systems," *Automatica*, vol. 123, p. 109338, 2021.

[12] F. Di Meglio, R. Vazquez, and M. Krstic, "Stabilization of a system of $n+1$ coupled first-order hyperbolic linear PDEs with a single boundary input," *IEEE Trans. Aut. Contr.*, vol. 58, pp. 3097–3111, 2013.

[13] R. Holten, "Generalized Goursat problem," *Pacific Journal of Mathematics*, vol. 12, pp. 207–224, 1962.

[14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[15] L. Jadachowsi, T. Meurer, and A. Kugi, "An efficient implementation of backstepping observers for time-varying parabolic PDEs," *IFAC Proceedings Volumes*, vol. 45, pp. 798–803, 2012.

[16] S. Kerschbaum and J. Deutscher, "Backstepping control of coupled linear parabolic PDEs with space and time dependent coefficients," *IEEE Trans. Aut. Contr.*, vol. 65, no. 7, pp. 3060–3067, 2019.

[17] M. Krstic, *Delay Compensation for nonlinear, Adaptive, and PDE Systems*. Birkhauser, 2009.

[18] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs*. SIAM, 2008.

[19] M. Krstic, L. Bhan, and Y. Shi, "Neural operators of backstepping controller and observer gain functions for reaction-diffusion PDEs," *arXiv preprint arXiv:2303.10506*, 2023.

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

[21] M. Lamarque, L. Bhan, Y. Shi, and M. Krstic, "Adaptive neural-operator backstepping control of a benchmark hyperbolic PDE," *arXiv preprint arXiv:2401.07862*, 2024.

[22] M. Lamarque, L. Bhan, R. Vazquez, and M. Krstic, "Gain scheduling with a neural operator for a transport PDE with nonlinear recirculation," *arXiv preprint arXiv:2401.02511*, 2024.

[23] Z. Li, N. Kovachki, K. Azizzadenesheli, *et al.*, "Neural operator: Graph kernel network for partial differential equations," *arXiv preprint*, vol. 2003.03485, 2020.

[24] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint*, vol. 2010.08895, 2020.

[25] X. Lin, R. Vazquez, and M. Krstic, "Towards a MATLAB toolbox to compute backstepping kernels using the power series method," *arXiv preprint arXiv:2403.16070*, 2024.

[26] L. Lu, P. Jin, and G. E. Karniadakis, "Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," *arXiv preprint*, vol. 1910.03193, 2019.

[27] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via Deeponet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[28] J. Qi, R. Vazquez, and M. Krstic, "Multi-agent deployment in 3-D via PDE control," *IEEE Transactions on Automatic Control*, vol. 60, pp. 891–906, 2015.

[29] J. Qi, J. Zhang, and M. Krstic, "Neural operators for delay-compensating control of hyperbolic PIDEs," *arXiv preprint arXiv:2307.11436*, 2023.

[30] A. Smyshlyaev, E. Cerpa, and M. Krstic, "Boundary stabilization of a 1-D wave equation with in-domain antidamping," *SIAM J. Control Optim.*, vol. 48, pp. 4014–4031, 2010.

[31] A. Smyshlyaev and M. Krstic, *Adaptive Control of Parabolic PDEs*. Princeton University Press, 2010.

[32] ——, "Closed-form boundary state feedbacks for a class of 1-D partial integro-differential equations," *IEEE Transactions on Automatic control*, vol. 49, no. 12, pp. 2185–2202, 2004.

[33] R. Vazquez, G. Chen, J. Qiao, and M. Krstic, "The power series method to compute backstepping kernel gains: theory and practice," in *2023 Conference on Decision and Control*, 2023.

[34] R. Vazquez and M. Krstic, "Control of 1-D parabolic PDEs with Volterra nonlinearities — Part I: Design," *Automatica*, vol. 44, pp. 2778–2790, 2008.

[35] ——, *Control of Turbulent and Magnetohydrodynamic Channel Flow*. Birkhauser, 2008.

[36] ——, "Boundary observer for output-feedback stabilization of thermal convection loop," *IEEE Trans. Control Syst. Technol.*, vol. 18, pp. 789–797, 2010.

[37] ——, "Boundary control of reaction-diffusion PDEs on balls in spaces of arbitrary dimensions," *ESAIM:Control Optim. Calc. Var.*, vol. 22, no. 4, pp. 1078–1096, 2016.

[38] R. Vazquez, E. Trelat, and J.-M. Coron, "Control for fast and stable laminar-to-high-Reynolds-numbers transfer in a 2-D Navier-Stokes channel flow," *Disc. Cont. Dyn. Syst. Ser. B*, vol. 10, pp. 925–956, 2008.

[39] R. Vazquez and M. Krstic, "Gain-only neural operator approximators of PDE backstepping controllers," *arXiv preprint arXiv:2403.19344*, 2024.

[40] S. Wang, M. Diagne, and M. Krstić, "Deep learning of delay-compensated backstepping for reaction-diffusion PDEs," *arXiv preprint arXiv:2308.10501*, 2023.