MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Reinforcement Learning-based Model Reduction for Partial Differential Equations

Benosman, Mouhacine; Chakrabarty, Ankush; Borggaard, Jeff

TR2020-076 June 24, 2020

Abstract

This paper is dedicated to the problem of stable model reduction for partial differential equations (PDEs). We propose to use proper orthogonal decomposition (POD) method to project the PDE model into a lower dimensional given by an ordinary differential equation (ODE) model. We then stabilize this model, following the closure model approach, by proposing to use reinforcement learning (RL) to learn an optimal closure model term. We analyze the stability of the proposed RL closure model and show its performance on the coupled Burgers equation.

World Congress of the International Federation of Automatic Control (IFAC)

© 2020 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Reinforcement Learning-based Model Reduction for Partial Differential Equations

Mouhacine Benosman *

* *Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA(m_benosman@ieee.org)*

Ankush Chakrabarty **

** *Mitsubishi Electric Research Laboratories*
Cambridge, MA 02139, USA

Jeff Borggaard ***

*** *Interdisciplinary Center for Applied Mathematics, Virginia Tech,*
Blacksburg, VA 24061, USA.

Abstract: This paper is dedicated to the problem of stable model reduction for partial differential equations (PDEs). We propose to use proper orthogonal decomposition (POD) method to project the PDE model into a lower dimensional given by an ordinary differential equation (ODE) model. We then stabilize this model, following the closure model approach, *by proposing to use reinforcement learning (RL) to learn an optimal closure model term.* We analyze the stability of the proposed RL closure model and show its performance on the coupled Burgers equation.

1. INTRODUCTION

Partial differential equations (PDEs) are important mathematical models, which are used to model complex dynamic systems in applied sciences. However, PDEs are infinite dimensional systems, which makes them hard to solve in closed-form, and computationally demanding to solve numerically. For instance, when using finite element method (FEM) discretization, one may end-up with a very large discretization space, which implies large computation time. Because of this complexity, it is often hard to use PDEs to analyze, predict or control these systems in real-time. Instead, one approach that is often used in real applications, is to first reduce the PDE model to an ordinary differential equation (ODE) model, which has a finite dimension and then use this ODE for system identification, estimation and control. This step of converting a PDE to a reduced order model (ROM) ODE, while maintaining a small error between the solutions of both models, is known as stable model reduction.

We address the stable model reduction problem by following the classical closure modeling approach, e.g., [9]. Indeed, closure models are added to the ROM equations to ensure the stability and accuracy of solutions. Closure models have classically been introduced based on physical intuition. Thus, their applicability is limited to those applications where significant research in closure models have been performed. In this work, we propose the use of reinforcement learning (RL) control to constructively design a new stabilizing closure model that is robust to model uncertainties. There are several closure models motivated from physical modeling of fluids, e.g., constant eddy viscosity model, or time and space varying terms, such as Smagorinsky or dynamic subgrid-scale models e.g., [9, 8, 4, 6, 3, 2]. However, there are some conceptual differences with the closure model that we are proposing here. First of all, we propose a closure model that explicitly accounts for model uncertainties in the system. Indeed, we formulate the problem of ROM stabilization at the design step, by considering uncertainties in the ROM model, then using tools borrowed from RL control, we design a closure model which stabilizes the ROM. To our knowledge,*this is the first class of closure model that is designed based on RL control.*

Furthermore, in this work we propose to learn some coefficients of the closure model using a data-driven optimization algorithm. This learning can be used in simulations to find the best closure model by tracking the true behavior of the system. However, an important observation is that *this learning algorithm can be incorporated in real-time simulations*, by feeding realtime measurements from the system into the closure model, and adapting its coefficients. In this way, we always ensure that the ROM is functioning at its optimal performance, regardless of changes or drifts that the system may experience over time. In other words, most closure models typically use static parameters, either chosen by intuition and experience, or are optimally tuned off-line. However, they are unable to auto-tune themselves on-line while the model is being evolved. In this work, the obtained closure model has free parameters that are auto-tuned with a data-driven extremum seeking (ES) algorithm to optimally match the predictions (or measurements) from the PDE model. The idea of using extremum-seeking to auto-tune closure models has been introduced by the authors in [7]. However, the difference with this work lies in the new RL-based stabilizing closure model design, which is then tuned using ES to optimize tracking perfromance.

This paper is organized as follows: Some basic notation and definitions are recalled first. The main idea of this work, namely, the RL-based closure model estimation is then introduced in Section 2, and its auto-tuning using extremum-seeking algorithms is explained in Section 3, finally the performance of the proposed concept of RL-based closure models is demonstrated using the 1D coupled Burgers equation in Section 4.

For a vector $q \in \mathbb{R}^n$, the transpose is denoted by q^*. The Euclidean vector norm for $q \in \mathbb{R}^n$ is denoted by $\|\cdot\|$ so that $\|q\| = \sqrt{q^* q}$. $I_{r\times r}$ denotes the $r \times r$ identity matrix (to simplify notations, the dimension might be omitted when clear from the context). We shall abbreviate the time derivative by $\dot{f}(t,x) = \frac{\partial}{\partial t} f(t,x)$, and consider the following Hilbert spaces: $\mathcal{H} = L^2(\Omega)$, $\Omega = (0,1)$, $\mathcal{V} = H^1(\Omega) \subset (\mathcal{H})$ for velocity and $\mathcal{T} = H^1(\Omega) \subset \mathcal{H}$ for temperature. We define the inner product $\langle \cdot,\cdot \rangle_{\mathcal{H}}$ and the associated norm $\|\cdot\|_{\mathcal{H}}$ on $\mathcal{H}$ as $\langle f, g\rangle_{\mathcal{H}} = \int_\Omega f(x)g(x)dx$, for $f, g \in \mathcal{H}$, and $\|f\|^2_{\mathcal{H}} = \int_\Omega |f(x)|^2 dx$. A function $f(t,x)$ is in $L^2([0,t_f];\mathcal{H})$ if for each $0 \le t \le t_f$, $f(t,\cdot) \in \mathcal{H}$, and $\int_0^{t_f} \|f(t,\cdot)\|^2_{\mathcal{H}} dt < \infty$. To generalize the discussion below, we consider the abstract Hilbert space $\mathcal{Z}$, and later specialize to $\mathcal{Z} = \mathcal{V} \oplus \mathcal{T}$ when considering our coupled Burgers equation example.

2. RL-BASED MODEL REDUCTION OF PDES

2.1 Reduced-order PDE Approximation

We consider a stable dynamical system modeled by a nonlinear partial differential equation of the form

$$\dot{z}(t) = \mathcal{F}(z(t)), \qquad z(0) \in \mathcal{Z}, \tag{1}$$

where $\mathcal{Z}$ is an infinite-dimensional Hilbert space. Solutions to this PDE can be approximated in a finite dimensional subspace $\mathcal{Z}^n \subset \mathcal{Z}$, where $\mathcal{Z}^n$ is an n-dimensional finite element subspace of $\mathcal{Z}$, through expensive numerical discretization, which can be impractical for multi-query settings such as analysis and design, and even more so for real-time applications such as prediction and control. In many systems, including fluid flows, solutions of the PDE may be well-approximated using only a few suitable (optimal) basis functions [1].

This gives rise to reduced-order modeling through Galerkin projection, which can be broken down into three main steps: One first discretizes the PDE using a finite, but large, number of basis functions, such as piecewise quadratic (for finite element methods), higher-order polynomials (spectral methods), or splines. In this paper we use the well-established finite element method (FEM). We denote the approximation of the PDE solution by $z_n(t,\cdot) \in \mathcal{Z}^n$.

Secondly, one determines a small set of spatial basis vectors $\phi_i(\cdot) \in \mathcal{Z}^n$, $i = 1,\ldots,r$, $r \ll n$, that approximates the discretized PDE solution with respect to a pre-specified criterion, i.e.,

$$P_n z(t,x) \approx \Phi q(t) = \sum_{i=1}^{r} q_i(t)\phi_i(x). \tag{2}$$

Here, P_n is the projection of $z(t,\cdot)$ onto $\mathcal{Z}^n$, and Φ is a matrix containing the basis vectors $\phi_i(\cdot)$ as column vectors. Note that the dimension n, coming from the high-fidelity discretization of the PDE described above, is generally very large, in contrast to the dimension r of the optimal basis set. Thirdly, a Galerkin projection yields a ROM for the coefficient functions $q(\cdot)$ of the form

$$\dot{q}(t) = F(q(t)), \qquad q(0) \in \mathbb{R}^r. \tag{3}$$

The function $F : \mathbb{R}^r \to \mathbb{R}^r$ is obtained using the weak form of the original PDE and Galerkin projection.

Here we use r POD basis functions [1] to approximate the solution of the PDE, e.g., (1) as follows

$$z_n^{pod}(t,\cdot) = \sum_{i=1}^{r} q_i(t)\phi_i(\cdot) \in \mathcal{Z}^n, \tag{4}$$

where ϕ_i is the ith *POD basis function*, and q_i, $i = 1,...,r$ are the POD projection coefficients. To find the coefficients $q_i(t)$, the (weak form of the) model (1) is projected onto the rth-order POD subspace $\mathcal{Z}^r \subseteq \mathcal{Z}^n \subset \mathcal{Z}$ using a Galerkin projection in $\mathcal{H}$. In particular, both sides of equation (1) are multiplied by the POD basis functions, where $z(t)$ is replaced by $z_n^{pod}(t) \in \mathcal{Z}^n$, and then both sides are integrated over Ω. Using the orthonormality of the POD basis leads to an ODE of the form (3). A projection of the initial condition for $z(0)$ can be used to determine $q(0)$. Note that the Galerkin projection preserves the structure of the nonlinearities of the original PDE.

2.2 Closure Models for ROM Stabilization

We continue to present the problem of stable model reduction in its general form, without specifying a particular type of PDE. However, we now assume an explicit dependence of the general PDE (1) on a single physical parameter μ,

$$\dot{z}(t) = \mathcal{F}(z(t),\mu), \quad z(0) = z_0 \in \mathcal{Z}, \qquad \mu \in \mathbb{R}, \tag{5}$$

as well as

Assumption 1. The solutions of the original PDE model (5) are assumed to be in $L^2([0,\infty);\mathcal{Z})$, $\forall \mu \in \mathbb{R}$.

We further assume that the parameter μ is critical for the stability and accuracy of the model, i.e., changing the parameter can either make the model unstable, or lead to inaccurate predictions. Since we are interested in fluid dynamics problems, we can consider μ as a viscosity coefficient. The corresponding reduced-order POD model takes the form (3) and (4):

$$\dot{q}(t) = F(q(t),\mu). \tag{6}$$

The issue with this Galerkin POD-ROM (denoted POD-ROM-G) is that the norm of q, and hence z_n^{pod}, might become unbounded at a finite time, even if the solution of (5) is bounded.

The main idea behind the closure modeling approach is to introduce a penalty term $H(\cdot)$ which is added to the original POD-ROM-G, as follows

$$\dot{q}(t) = F(q(t),\mu) + H(q(t)). \tag{7}$$

The term $H(\cdot)$ is chosen depending on the structure of $F(\cdot,\cdot)$ to stabilize the solutions of (7).

2.3 Main Result: RL-based Closure Model

Here we introduce the main result of this work, namely using RL to compute a closure term H that is robust to model uncertainties. We first rewrite the right-hand side of the ROM model (6) to isolate the linear viscous term as follows,

$$F(q(t),\mu) = \widetilde{F}(q(t)) + \mu\, Dq(t), \tag{8}$$

where $D \in \mathbb{R}^{r\times r}$ represents a constant, symmetric negative definite matrix, and the function $\widetilde{F}(\cdot)$ represents the rest of the ROM model, i.e., the part without damping [1].

To follow the framework of [10], we discretize out model (7), (8), for example by using a simple first order Euler approximation, as follows

$$q(k+1) = (I_{r\times r} + h_t\mu\, D)q(k) + h_t\widetilde{F}(q(k)) + h_t H(q(k)), \tag{9}$$

where $k = 0, 1, ...,$ and $h_t > 0$ represents the integration time-step.

Next, we assume that $\widetilde{F}(\cdot)$ satisfies

Assumption 2. (Lipschitz continuity of $\widetilde{F}$). The nonlinearity $\widetilde{F}$ is Lipschitz continuous in the domain $\mathbb{D}_q \in \mathbb{R}^r$. That is,

$$\|\widetilde{F}(q_1) - \widetilde{F}(q_2)\| \le \mathfrak{L}^*_\phi\|q_1 - q_2\| \tag{10}$$

for any $q_1, q_2 \in \mathbb{D}_q$. Also, $\widetilde{F}(0) = 0$.

Remark 1. We underline here that we do not need the exact knowledge of the nonlinear term $\widetilde{F}$ to design our RL-based closure model. Indeed, we only need to know an estimate of its Lipschitz constant $\mathfrak{L}^*_\phi$. This estimate can be obtained for instance by using the data-driven algorithm proposed in [10]. In that sense, the proposed RL-based closure model stabilization is robust w.r.t. the uncertainties of the nonlinear term $\widetilde{F}$.

The main idea that we are proposing here is to consider the closure model function $H(q(t))$ as a virtual controller, which we then propose to compute using RL control, more specifically, we will use adaptive dynamic programing (ADP) to learn the best closure model.

Let us first recall the basic formulation of ADP. Given a control policy $u(q)$, we define an infinite horizon cost functional given an initial state $q_0 \in \mathbb{R}^r$ as

$$\mathcal{J}(q(0), u) = \sum_{t=0}^{\infty} \gamma^t\, \mathcal{U}(q(k), u(q(k))), \tag{11}$$

where, $\gamma \in (0, 1]$ in (11) is a forgetting/discount factor, $\mathcal{U}$ is a function with non-negative range, $\mathcal{U}(0, 0) = 0$, and $\{q(k)\}$ denotes the sequence of states generated by the closed loop system

$$q(k+1) = Aq(k) + Bu(q(k)) + \phi\left(C_q q(k)\right), \tag{12}$$

where, in our case, we define the terms to be

$$\begin{aligned} A = I_{r\times r} + h_t\mu D,\ B = h_t\, I_{r\times r},\\ C_q = I_{r\times r},\ \phi = h_t\widetilde{F}(q(t)),\ u(q) = H(q). \end{aligned} \tag{13}$$

Before formally stating our objective, we need to introduce the following definition.

Definition 1. A continuous control policy $u(\cdot) : \mathbb{R}^r \to \mathbb{R}^r$ is *admissible* on $X \subset \mathbb{R}^r$ if it stabilizes the closed loop system (12) on X and $\mathcal{J}(q(0), u)$ is finite for any $q(0) \in X$.

We want to design an optimal control policy that achieves the optimal cost

$$\mathcal{J}_\infty(q(0)) = \inf_{u\in\mathfrak{U}} \mathcal{J}\big(q(0), u\big), \tag{14}$$

for any $q(0) \in \mathbb{R}^r$. Here, $\mathfrak{U}$ denotes the set of all admissible control policies. In other words, we wish to compute an optimal control policy

$$u_\infty = \arg\inf_{u\in\mathfrak{U}} \mathcal{J}\big(q(0), u\big). \tag{15}$$

Directly constructing such an optimal controller is very challenging for general nonlinear systems with high state dimension. Therefore, we shall use adaptive/approximate dynamic programming (ADP): a class of iterative, data-driven algorithms that generate a convergent sequence of control policies whose limit is provably the optimal control policy $u_\infty(q)$. Recall the optimal value function given by (14) and the optimal control policy (15). From the Bellman optimality principle, we know that the discrete-time Hamilton-Jacobi-Bellman equations are given by

$$J_\infty(q(k)) = \inf_{u\in\mathfrak{U}} \left(\mathcal{U}(q(k), u(q(k))) + \gamma J_\infty(q(k+1))\right), \tag{16}$$

$$u_\infty(q(k)) = \arg\inf_{u\in\mathfrak{U}} \left(\mathcal{U}(q(k), u(q(k))) + \gamma J_\infty(q(k+1))\right), \tag{17}$$

where $J_\infty(q(k))$ is the optimal value function and $u_\infty(q(k))$ is the optimal control policy. The key operations in ADP methods involve setting an admissible control policy $u_0(x)$ and then iterating the policy evaluation step

$$\mathcal{J}_{I+1}(q(k)) = \mathcal{U}\big(q(k), u_I(q(k))\big) + \gamma\mathcal{J}_{I+1}(q(k+1)) \tag{18a}$$

and the policy improvement step

$$u_{I+1}(q(k)) = \arg\min_{u(\cdot)} \Big(\mathcal{U}\Big(q(k), u(q(k))\Big) + \gamma\mathcal{J}_{I+1}(q(k+1))\Big), \tag{18b}$$

$I = 0, 1, ...,$ until convergence.

Next, we recall the following definition.

Definition 2. The equilibrium point $q = 0$ of the closed-loop system (12) is globally exponentially stable with a decay rate α if there exist scalars $C_0 > 0$ and $\alpha \in (0, 1)$ such that $\|q(k)\| \le C_0\alpha^{(k-k_0)}\|q(0)\|$ for any $q(0) \in \mathbb{R}^r$.

The following design theorem provides a method to construct an initial linear stabilizing policy $u_0(x) = K_0x$ such that the origin is a GES equilibrium state of the closed-loop system (12).

Theorem 1. Suppose that Assumptions 1–2 hold, and that there exist matrices $P = P^\top \succ 0 \in \mathbb{R}^{n_x\times n_x}$, $K_0 \in \mathbb{R}^{n_u\times n_x}$, and scalars $\alpha \in (0, 1)$, $\nu > 0$ such that

$$\Psi + \Gamma^\top\mathcal{M}\Gamma \preceq 0, \tag{19}$$

where

$$\Psi = \begin{bmatrix} (A+BK_0)^\top P(A+BK_0) - \alpha^2 P & \star \\ P(A+BK_0) & P \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} C_q & 0 \\ 0 & I \end{bmatrix},\ \text{and}\ \mathcal{M} = \begin{bmatrix} \nu^{-1}(\mathfrak{L}^*_\phi)^2 I & 0 \\ 0 & -\nu^{-1}I \end{bmatrix}.$$

Then the equilibrium $x = 0$ of the closed-loop system (12) is GES with decay rate α.

Proof 1. Refer to [10].

Note that we do not need to know $\phi(\cdot)$ to satisfy conditions (19), which makes the proposed closure model robust to model uncertainties (see Remark 1).

We shall now provide LMI-based conditions for computing the initial control policy K_0, the initial domain of attraction P and ν via convex programming.

Theorem 2. Fix $\alpha \in (0, 1)$ and $\mathfrak{L}^*_\phi$ as defined in Assumption 2. If there exist matrices $S = S^\top \succ 0$, Y, and a scalar $\nu > 0$ such that the LMI conditions

$$\begin{bmatrix} -\alpha^2 S & \star & \star & \star \\ 0 & -\nu I & \star & \star \\ AS + BY & S & -S & \star \\ \mathfrak{L}^*_\phi C_q S & 0 & 0 & -\nu I \end{bmatrix} \preceq 0 \tag{20}$$

---

[1] We can extend the results to the case with nonlinear damping terms in $\tilde{F}$, as long as, we can still impose similar (uniform w.r.t. μ) Lipschitz condition on $\tilde{F}$)

are satisfied, then the matrices $K_0 = YS^{-1}$, $P = S^{-1}$ and scalar ν satisfy the conditions (19) with the same α and $\mathfrak{L}^*_\phi$.

Proof 2. Refer to [10].

We can now state the following admissibility Corollary (see [10]).

Corollary 1. Let

$$\mathcal{U}(q(k),u(k)) = q(k)^\top Q q(k) + u(k)^\top R u(k) \tag{21}$$

for some matrices $Q = Q^\top \succeq 0$ and $R = R^\top \succ 0$. Then the initial control policy $u(0) = K_0 q$ obtained by solving (20) is an admissible control policy on $\mathbb{R}^r$.

Now that we know $u(0) = K_0 q$ is an admissible control policy, we are ready to proceed with the policy iteration steps (18). Typically, an analytical form of $\mathcal{J}_I$ is not known *a priori*, so we resort to a shallow neural approximator/truncated basis expansion for fitting this function, assuming $\mathcal{J}_I$ is smooth for every $I \in \mathbb{N} \cup \{\infty\}$. Concretely, we represent the value function and cost functions as:

$$\mathcal{J}_I(q) := \omega_I^\top \psi(q) \tag{22}$$

where $\psi(\cdot) : \mathbb{R}^r \to \mathbb{R}^{n_0}$ denotes the vector of differentiable basis functions (equivalently, hidden layer neuron activations) and $\omega \in \mathbb{R}^{n_0}$ is the corresponding column vector of basis coefficients (equivalently, neural weights).

It is not always clear how to initialize the weights of the neural approximators (22). We propose initializing the weights as follows. Since our initial Lyapunov function is quadratic, we include the quadratic terms of the components of x to be in the basis $\psi(q)$. Then we can express the initial Lyapunov function $q^\top P q$ obtained by solving (20) with appropriate weights in the $\psi(q)$, respectively, setting all other weights to be zero. With the approximator initialized as above, the policy evaluation step (18a) is replaced by

$$\omega_{I+1}^\top\big(\psi(q(k)) - \gamma\psi(q(k+1))\big) = \mathcal{U}\left(q(k), u_I(q(k))\right), \tag{23a}$$

from which one can solve for ω_{I+1} recursively via

$$\omega_{I+1} = \omega_I - \eta_I \varphi_I \left(\omega_I^\top \varphi_I - \mathcal{U}\left(q(k), u_I(q(k))\right)\right),$$

where η_I is a learning rate parameter that is usually selected to be an element from the sequence $\{\eta_I\} \to 0$ as $I \to \infty$, and $\varphi_I = \psi(q(k)) - \gamma\psi(q(k+1))$. Subsequently, the policy improvement step (18b) is replaced by

$$u_{I+1} = \arg\min_{u(\cdot)} \big(\mathcal{U}\left(q(k), u(q(k))\right) + \gamma \omega_{I+1}^\top \psi(q(k+1))\big).$$

This minimization problem is typically non-convex and therefore, challenging to solve to optimality. In some specific cases, one of which is that the cost function is quadratic as described in (21), the policy improvement step becomes considerably simpler to execute, namely

$$u_{I+1}(q) = -\frac{\gamma}{2} R^{-1} B^\top \nabla\psi(q)^\top \omega_{I+1}. \tag{23b}$$

This can be evaluated as R and B are known, and ψ is differentiable and chosen by the user, so $\nabla\psi$ is computable.

Since we prove that u_0 is an admissible control policy, we can use arguments identical to [11] [Theorem 3.2 and Theorem 4.1] to claim that if the optimal value function and the optimal control policy are dense in the space of functions induced by the basis function expansions (22), then the weights of the neural approximator employed in the PI steps (23) converges to the optimal weights; that is, the optimal value function $\mathcal{J}_\infty$ and the optimal control policy u_∞ are achieved asymptotically. We now present our main result.

Theorem 3. (RL-based stabilizing closure model) Consider the PDE (5) under Assumption 1, together with its ROM model

$$\dot{q}(t) = \widetilde{F}(q(t)) + \mu\, Dq(t) + H(q(t)), \tag{24}$$

where $\widetilde{F}(\cdot)$ satisfies Assumption 2, $D \in \mathbb{R}^{r\times r}$ is symmetric negative definite, and $\mu > 0$ is the nominal value of the viscosity coefficient in (5). Then, the nonlinear closure model $H(q)$ computed using the RL controller (23a), (23b), where $u_0(q) = K_0 q$, K_0 obtained by the SDP (20), practically stabilizes the solutions of the ROM (24) to an ϵ-neighborhood of the origin.

Proof: Due to space limitation the proof has been removed, but will be included in a longer version of this work. □

3. EXTREMUM-SEEKING BASED CLOSURE MODEL AUTO-TUNING

ES-based closure model auto-tuning has many advantages. First of all, the closure models can be valid for longer time intervals when compared to standard closure models with constant coefficients that are identified off-line over a (fixed) finite time interval. Secondly, the optimality of the closure model ensures that the ROM obtains the most accuracy for a given low-dimensional basis, leading to the smallest possible ROM for a given application.

We begin by defining a suitable learning cost function for the ES algorithm. The goals of the learning is to ensure that the solutions of the ROM (6) are close to those of the approximation $z_n(t,\cdot)$ to the original PDE (5).

We first introduce some tuning coefficients in the ROM model (24), as follows

$$\dot{q}(t) = \widetilde{F}(q(t)) + (\mu + \mu_e)\, Dq(t) + \mu_{nl} H(q(t)), \tag{25}$$

where $\mu_e > 0$, and $\mu_{nl} > 0$ are two positive tuning parameters. We then define the learning cost as a positive definite function of the norm of the error between the numerical solutions of (5) and the ROM (25),

$$\begin{aligned} Q(\widehat{\boldsymbol{\mu}}) &= \widetilde{H}(e_z(t,\widehat{\boldsymbol{\mu}})), \\ e_z(t,\widehat{\boldsymbol{\mu}}) &= z_n^{pod}(t,x;\widehat{\boldsymbol{\mu}}) - z_n(t,x;\boldsymbol{\mu}), \end{aligned} \tag{26}$$

where $\widehat{\boldsymbol{\mu}} = [\widehat{\mu}_e, \widehat{\mu}_{nl}]^* \in \mathbb{R}^2$ denotes the learned parameters, and $\widetilde{H}(\cdot)$ is a positive definite function of e_z. Note that the error e_z could be computed off-line using solutions of the ROM (4), (7) and exact (e.g., FEM-based) solutions of the PDE (5). The error could be also computed on-line where the $z_n^{pod}(t,x;\widehat{\boldsymbol{\mu}})$ is obtained from solving the ROM model (4), (7) on-line, and the $z_n(t,x;\boldsymbol{\mu})$ are real measurements of the system at selected spatial locations $\{x_i\}$. The latter approach would circumvent the FEM model, and directly operate on the system, making the reduced-order model more consistent with respect to the operating plant.

To derive formal convergence results, we introduce some classical assumptions on the learning cost function.

Assumption 3. The cost function $Q(\cdot)$ in (26) has a local minimum at $\widehat{\boldsymbol{\mu}} = \boldsymbol{\mu}^{\rm opt}$.

Assumption 4. The cost function $Q(\cdot)$ in (26) is analytic and its variation with respect to $\boldsymbol{\mu}$ is bounded in the neighborhood of $\boldsymbol{\mu}^{\rm opt}$, i.e., $\|\nabla_{\boldsymbol{\mu}} Q(\widetilde{\boldsymbol{\mu}})\| \leq \xi_2$, $\xi_2 > 0$, for all $\widetilde{\boldsymbol{\mu}} \in \mathcal{N}(\boldsymbol{\mu}^{\rm opt})$, where $\mathcal{N}(\boldsymbol{\mu}^{\rm opt})$ denotes a compact neighborhood of $\boldsymbol{\mu}^{\rm opt}$.

Under these assumptions the following lemma holds.

Lemma 1. Consider the PDE (5) under Assumption 1, together with its ROM model (25). Furthermore, suppose the closure model coefficients $\widehat{\boldsymbol{\mu}} = [\mu_e,\ \mu_{nl}]^*$ are tuned using the ES algorithm

$$\dot{y}_1(t) = a_1 \sin\left(\omega_1 t + \frac{\pi}{2}\right) Q, \dot{y}_2(t) = a_2 \sin\left(\omega_2 t + \frac{\pi}{2}\right) Q,$$
$$\widehat{\mu}_e(t) = y_1 + a_1 \sin\left(\omega_1 t - \frac{\pi}{2}\right), \widehat{\mu}_{nl}(t) = y_2 + a_2 \sin\left(\omega_2 t - \frac{\pi}{2}\right) \tag{27}$$

where $\omega_{\max} = \max(\omega_1, \omega_2) > \omega^{\text{opt}}$, ω^{opt} large enough, and $Q(\cdot)$ is given by (26). Let $e_{\boldsymbol{\mu}}(t) := [{\boldsymbol{\mu}_e}^{\text{opt}} - \widehat{\boldsymbol{\mu}}_e(t), {\boldsymbol{\mu}_{nl}}^{\text{opt}} - \widehat{\boldsymbol{\mu}}_{nl}(t)]^*$ be the error between the current tuned values, and the optimal values $\boldsymbol{\mu}_e^{\text{opt}},\ \boldsymbol{\mu}_{nl}^{\text{opt}}$. Then, under Assumptions 3, and 4, the norm of the distance to the optimal values admits the following bound

$$\|e_{\boldsymbol{\mu}}(t)\| \leq \frac{\xi_1}{\omega_{\max}} + \sqrt{a_1^2 + a_2^2},\ t \to \infty, \tag{28}$$

where $a_1,\ a_2 > 0$, $\xi_1 > 0$, and the learning cost function approaches its optimal value within the following upper-bound

$$\|Q(\widehat{\boldsymbol{\mu}}) - Q(\boldsymbol{\mu}^{\text{opt}})\| \leq \xi_2(\frac{\xi_1}{\omega} + \sqrt{a_1^2 + a_2^2}), \tag{29}$$

as $t \to \infty$, where $\xi_2 = \max_{\boldsymbol{\mu} \in \mathcal{N}(\boldsymbol{\mu}^{\text{opt}})} \|\nabla_{\boldsymbol{\mu}} Q(\boldsymbol{\mu})\|$.

Proof 3. Refer to [7].

4. THE CASE OF THE BURGERS EQUATION

As an example application of our approach, we consider the coupled Burgers equation of the form

$$\begin{cases} \dfrac{\partial w(t,x)}{\partial t} + w(t,x)\dfrac{w(t,x)}{\partial x} = \mu \dfrac{\partial^2 w(t,x)}{\partial x^2} - \kappa T(t,x) \\ \dfrac{\partial T(t,x)}{\partial t} + w(t,x)\dfrac{\partial T(t,x)}{\partial x} = c\dfrac{\partial^2 T(t,x)}{\partial x^2} + f(t,x), \end{cases} \tag{30}$$

where $T(\cdot,\cdot)$ represents the temperature, $w(\cdot,\cdot)$ represents the velocity field, κ is the coefficient of the thermal expansion, c the heat diffusion coefficient, μ the viscosity (inverse of the Reynolds number Re), $x \in [0,1]$ is the one dimensional space variable, $t > 0$, and $f(\cdot,\cdot)$ is the external forcing term such that $f \in L^2((0,\infty), X)$, $X = L^2([0,1])$. The boundary conditions are imposed as: $w(t,0) = w_L$, $\frac{\partial w(t,1)}{\partial x} = w_R, T(t,0) = T_L$, $\quad T(t,1) = T_R$, where w_L, w_R, T_L, T_R are positive constants, and L and R denote left and right boundary, respectively. The initial conditions are imposed as: $w(0,x) = w_0(x) \in L^2([0,1]), T(0,x) = T_0(x) \in L^2([0,1])$, and are specified below. Following a Galerkin projection onto the subspace spanned by the POD basis functions, the coupled Burgers equation is reduced to a POD ROM with the following structure (e.g., see [9])

$$\begin{aligned} \begin{pmatrix} \dot{q}_w \\ \dot{q}_T \end{pmatrix} &= B_1 + \mu B_2 + \mu\ D\ q + \tilde{D}q + Cqq^T, \\ w_n^{pod}(x,t) &= w_{av}(x) + \sum_{i=1}^{i=r} \phi_{wi}(x) q_{wi}(t), \\ T_n^{pod}(x,t) &= T_{av}(x) + \sum_{i=1}^{i=r} \phi_{Ti}(x) q_{Ti}(t), \end{aligned} \tag{31}$$

where matrix B_1 is due to the projection of the forcing term f, matrix B_2 is due to the projection of the boundary conditions, matrix D is due to the projection of the viscosity damping term $\mu\frac{\partial^2 w(t,x)}{\partial x^2}$, matrix $\tilde{D}$ is due to the projection of the thermal coupling and the heat diffusion terms $-\kappa T(t,x)$, $c\frac{\partial^2 T(t,x)}{\partial x^2}$, and the matrix C is due to the projection of the gradient-based terms $w\frac{w(t,x)}{\partial x}$, and $w\frac{\partial T(t,x)}{\partial x}$. The notations $\phi_{wi}(x)$, $q_{wi}(t)$ $(i = 1, ..., r_w)$, $\phi_{Ti}(x)$, $q_{Ti}(t)$ $(i = 1, ..., r_T)$, stand for the space basis functions and the time projection coordinates, for the velocity and the temperature, respectively. The terms $w_{av}(x)$, $T_{av}(x)$ represent the mean values (over time) of w and T, respectively.

We test the stabilization performance of our RL-based closure model. by considering the coupled Burgers equation (30), with the parameters $Re = 1000$, $\kappa = 5 \times 10^{-4}$, $c = 1 \times 10^{-2}$, the trivial boundary conditions $w_L = w_R = 0$, $T_L = T_R = 0$, a simulation time-length $t_f = 1s$ and zero forcing, $f = 0$. We use 10 POD modes for both variables (temperature and velocity). For the choice of the initial conditions, we follow [9], where the simplified Burgers' equation has been used in the context of POD ROM stabilization. Indeed, in [9] the authors propose two types of initial conditions for the velocity variable, which led to instability of the nominal POD ROM, i.e., the basic Galerkin POD ROM (POD ROM-G) without any closure model. Accordingly, we choose the following initial conditions:

$$w(x,0) = \begin{cases} 1, \text{ if } x \in\ [0,\ 0.5] \\ 0, \text{ if } x \in\ ]0.5,\ 1], \end{cases} T(x,0) = \begin{cases} 1, \text{ if } x \in\ [0,\ 0.5] \\ 0, \text{ if } x \in\ ]0.5,\ 1], \end{cases} \tag{32}$$

We report in Figure 1 the solutions [2] of the POD ROM-G (without closure model). We can see clearly in this figure that the POD ROM-G solution is unstable, with a clear blowup of the velocity profile. We compute the RL-based closure model from (23a), and (23b). We then run the ROM-CL which is the ROM with the closure model H computed from (23a), (23b), and report the corresponding solutions in Figure 2. We can see clearly that the closure model stabilizes the ROM as expected. However, although we recover the stability of the original PDE, after adding the RL-based closure term, the performance of the ROM-CL model is rather average, as one can observe from the error profiles in Figure 3. To improve the ROM-CL model performance in reproducing the true temperature and velocity distributions, we add an auto-tuning layer to the ROM-CL model, by using an auto-tuning extremum seeking algorithm, as explained in Section 3.

We implement the ROM-CL (25), where here again the closure term is given by the RL-controller (23a), (23b). The coefficients of the closure model are tuned using the first Euler discretization of (27), where the learning cost is defined as

$$Q(\boldsymbol{\mu}) = \int_0^{t_f} \langle e_T, e_T \rangle_{\mathcal{H}} dt + \int_0^{t_f} \langle e_{\mathbf{v}}, e_{\mathbf{v}} \rangle_{\mathcal{H}} dt. \tag{33}$$

$e_T = P_r T_n - T_n^{pod}$, $e_{\mathbf{v}} = P_r \mathbf{v}_n - \mathbf{v}_n^{pod}$ define the errors between the projection of the true model solution onto the POD space $\mathcal{Z}^r$ and the POD-ROM solution for temperature and velocity, respectively. We select the following ES coefficients: $a_1 = 8.10^{-5}$ $[-]$, $\omega_1 = 10$ $[\frac{\text{rad}}{\text{sec}}]$, $a_2 = 8.10^{-5}$ $[-]$, $\omega_2 = 12$ $[\frac{\text{rad}}{\text{sec}}]$.

We report in Figure 3 the ES learning cost over the learning iterations, where we see an improvement of the overall tracking cost function. The associated tuning coefficients estimation is shown in Figure 3. Finally, the performance of the ROM-CL after tuning is shown in Figures 4 , and 4, where we see a large decrease of the tracking errors,

---

[2] Due to space limitations we only report the velocity profile.

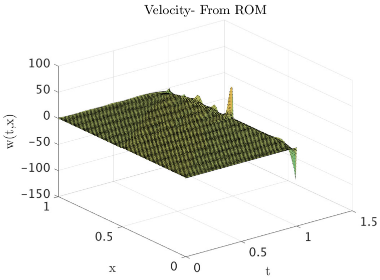

Fig. 1. Closure-model-free POD ROM solution of (30).

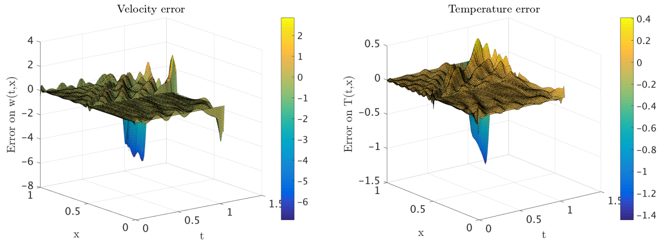

Fig. 2. ROM-CL error profiles of (30), with reinforcement learning.

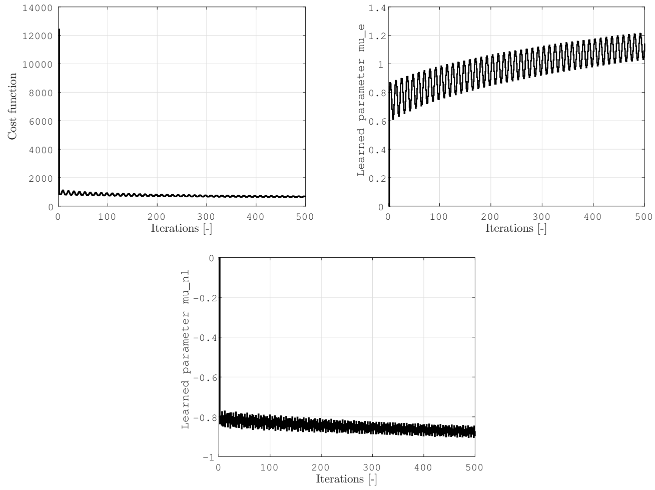

Fig. 3. (Left) ROM-CL learning cost. (Mid) ROM-CL μ_e coefficient tuning. (Right) ROM-CL coefficients tuning.

comparatively to the errors obtained with the ROM-CL without ES tuning.

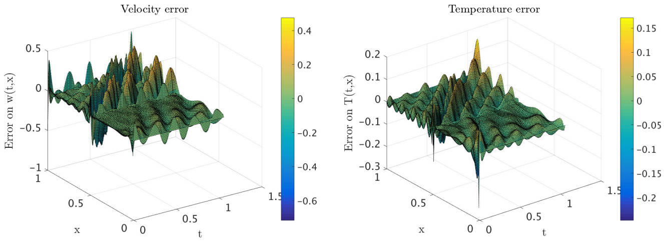

Fig. 4. ROM-CL error profiles of (30) (with auto-tuning).

5. CONCLUSION

In this paper we have focused on the problem of model reduction of infinite dimension systems modeled by partial differential equations. We have proposed to use reinforcement learning (RL) control to design stabilizing closure models for reduced order models. The obtained stabilizing closure models are robust to model uncertainties, which makes them appealing for real-life applications, like for instance fluid dynamics modeling applications. To further improve the validity of the reduced order models, we added a tuning layer to the proposed RL-based closure models, by tuning (possibly online) some of their free coefficients using an extremum seeking algorithm.

REFERENCES

[1] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry.* Cambridge University Press, 1998.

[2] M. Couplet, C. Basdevant, and P. Sagaut, “Calibrated reduced-order POD-Galerkin system for fluid flow modelling,” *Journal of Computational Physics*, vol. 207, no. 1, pp. 192–220, 2005.

[3] V. L. Kalb and A. E. Deane, “An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models,” *Physics of Fluids*, vol. 19, no. 5, p. 054106, 2007.

[4] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, “Goal-oriented, model-constrained optimization for reduction of large-scale systems,” *Journal of Computational Physics*, vol. 224, no. 2, pp. 880–896, 2007.

[5] M. Ilak, S. Bagheri, L. Brandt, C. W. Rowley, and D. S. Henningson, “Model reduction of the nonlinear complex Ginzburg-Landau equation,” *SIAM Journal on Applied Dynamical Systems*, vol. 9, no. 4, pp. 1284–1302, 2010.

[6] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, and M. Barone, “Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment,” *Computer Methods in Applied Mechanics and Engineering*, vol. 272, pp. 251–270, 2014.

[7] M. Benosman, J Borggaard, O. San, and B. Kramer “Learning-based Robust Stabilization for Reduced-Order Models of 2D and 3D Boussinesq Equations,” in *Applied Mathematical Modelling*, Vol. 49, pp. 162–181, 2016.

[8] M. Balajewicz, E. Dowell, and B. Noack, “Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation,” *Journal of Fluid Mechanics*, vol. 729, no. 1, pp. 285–308, 2013.

[9] O. San and T. Iliescu, “Proper orthogonal decomposition closure models for fluid flows: Burgers equation,” *International Journal of Numerical Analyis and Modeling*, vol. 1, no. 1, pp. 1–18, 2013.

[10] A. Chakrabarty, D. K. Jha and Y. Wang, “Data-Driven Control Policies for Partially Known Systems via Kernelized Lipschitz Learning,” in *Proc. of the 2019 American Control Conference (ACC)*, Philadelphia, PA, USA, pp. 4192-4197, July 2019.

[11] D. Liu and Q. Wei, “Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, 3, pp. 621–634, 2014.