Online identification and control of PDEs via Reinforcement Learning methods

Alessandro Alla
work in collaboration with A. Pacifico, A. Pesare and M. Palladino

Università Ca'Foscari Venezia

Nonlinear PDEs: theory, numerics and applications
a conference in memory of Maurizio

In memory of Maurizio

Story of our lives

I'll be back in 10 minutes. If not, read again. WARNING: Convergence not guaranteed!!!

Figure: October 2019. Selfie with Maurizio at Forte de Copacabana.

Figure: October 2019. Dinner at Braseiro da Gavea.

Figure: June 2022. Maurizio playing bocce

Figure: September 2022. Photo from Cetraro.

Maurizio's mottos

- Do not panic
- Go on
- Calma
- Eccoci (dopo almeno 30 minuti di ritardo)
- Non ci siamo
- Abbiamo ancora 5 minuti
- Anche questa è fatta
- Va bene, bye bye

- Do not panic
- Go on
- Calm down
- Here, we are (after a delay of 30 minutes)
- We are not there yet
- We still have 5 minutes
- And also this one is done
- Ok, bye bye

Problem setting

State equation

$$\begin{cases} y_t(t,\xi) = \sum_{j=1}^{n} \mu_j F_j(y(t,\xi), y_\xi(t,\xi), y_{\xi\xi}(t,\xi), y_{\xi\xi\xi}(t,\xi), \dots) + B(\xi)u(t), & t \in [0,\infty), \xi \in (a,b), \\ y(0,\xi) = y_0(\xi), & \xi \in [a,b], \\ y(t,a) = 0, \; y(t,b) = 0, & t \in [0,\infty) \end{cases}$$

Cost functional

$$J(u;\mu) = \int_0^\infty \|y(t,\cdot;\mu)\|^2_{L^2(a,b)} + R\|u(t;\mu)\|^2 \, dt, \qquad R > 0$$

Control problem

$\min_{u \in U} J(u;\mu)$ such that y solves the state equation and μ is the parameter to identify

Assumptions of our problem

- $y : [0, \infty] \times \mathbb{R} \to \mathbb{R}$, $\mu_j \in \mathbb{R}$, $u(t) : [0, \infty) \to \mathbb{R}^m$ and $B(\xi) : [a, b] \to \in \mathbb{R}^{1\times m}$
- the model is the sum of simple monomial bases functions F_j of y and its derivatives
- the functions F_j's may be thought as a **library** with terms that has to be selected by the coefficients μ_j's
- the system is fully identified by the knowledge of the coefficient $\mu = (\mu_1, \ldots, \mu_n) \in \mathbb{R}^n$ which is considered **unknown** in the present work
- we set zero Dirichlet boundary conditions (without loss of generality)

Notation

For a given parameter configuration $\widetilde{\mu} \in \mathbb{R}^n$, we will refer to $y(t, \cdot; \widetilde{\mu})$ as the solution of the state equation with $\mu = \widetilde{\mu}$ and to $u(t; \widetilde{\mu})$ as the control computed using the state equation with $\mu = \widetilde{\mu}$

Outline

1. Building blocks: Bayesian linear regression and State Dependent Riccati equation
   - Bayesian linear regression
   - State Dependent Riccati Equation (SDRE)
2. Identification and control through RL methods
3. Numerical experiments

Outline

1. Building blocks: Bayesian linear regression and State Dependent Riccati equation
   - Bayesian linear regression
   - State Dependent Riccati Equation (SDRE)
2. Identification and control through RL methods
3. Numerical experiments

Linear regression (LR)

In LR, we consider *data* in the form of input-output pairs

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1,\dots,d}$$

and we suppose that the output variable $y_i \in \mathbb{R}$ can be expressed approximately as a linear function of the input variable $x_i \in \mathbb{R}^n$, i.e.

$$y_i \approx x_i^T \theta, \qquad \text{for} \quad i = 1, \dots, d$$

We look for a parameter $\theta \in \mathbb{R}^n$ such that we minimize the sum of squared residuals

$$E(\theta) = \sum_{i=1}^{d} |y_i - x_i^T \theta|^2$$

The *LS solution* can be computed analytically and is given by

$$\theta_{LS} = (X^T X)^{-1} X^T Y$$

where we collected all the observed inputs in a matrix $X \in \mathbb{R}^{d \times n}$ and all the observed outputs in a vector $Y \in \mathbb{R}^d$

Bayesian Linear Regression (BLR)

BLR is a probabilistic method for solving the classical LR problem.
In BLR, the deviation of the data from the linear model can be described by a Gaussian noise

$$y_i = x_i^T \theta + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

where $\theta \in \mathbb{R}^n$ is an unknown parameter to be determined and $\sigma > 0$ known.

The available information on the parameter θ is included in the model through the definition of a *prior distribution*, e.g. $\theta \sim \mathcal{N}(m_0, \Sigma_0)$

$$\bar{\theta}_{BLR} = \left(\frac{1}{\sigma^2} X^T X + \Sigma_0^{-1}\right)^{-1} \left(\frac{1}{\sigma^2} X^T Y + \Sigma_0^{-1} m_0\right)$$

- BLR provides a quantification of the uncertainty of this estimate
- the estimate $\bar{\theta}_{BLR}$ converges to the LS solution, when the noise variance σ goes to 0

Control Problem

$$\dot{x}(t) = A(x(t))x(t) + B(x(t))u(t), \quad t \in (0, \infty),$$

$$\min_{u(\cdot)\in U} J(u(\cdot)) := \int_0^\infty \left(\|x(t)\|_Q^2 + \|u(t)\|_R^2 \right) dt$$

Assumptions

- $x(t) : [0, \infty] \to \mathbb{R}^d, \qquad A(x) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$
- $u(\cdot) \in \mathcal{U} := L^\infty(\mathbb{R}_+; \mathbb{R}^m)$
- $\|x\|_Q^2 := x^\top Qx$ with $Q \in \mathbb{R}^{d\times d}$, $Q \succ 0$, $\quad \|u\|_R^2 = u^\top Ru$ with $R \in \mathbb{R}^{m\times m}$, $R \succ 0$
- This formulation of the $\mathcal{H}_2$ synthesis corresponds to the asymptotic stabilization of nonlinear dynamics towards the origin

State-Dependent Riccati Equation (SDRE)

LQR

$A(x) = A$ with $A \in \mathbb{R}^{d\times d}$ and $B(x) = B \in \mathbb{R}^{d\times m}$, $V(x) = x^\top \Pi x$, with $\Pi \in \mathbb{R}^{d\times d}$ positive definite, and HJB becomes an Algebraic Riccati Equation (ARE) for Π

$$A^\top \Pi + \Pi A - \Pi B R^{-1} B^\top \Pi + Q = 0$$
$$u(x) := -R^{-1} B^\top \Pi x$$

SDRE

$$\dot{x} = A(x)x + B(x)u(t)$$

$$A^\top(x)\Pi(x) + \Pi(x)A(x) - \Pi(x)B(x)R^{-1}B^\top(x)\Pi(x) + Q = 0$$

Solving this equation leads to a state-dependent Riccati operator $\Pi(x)$, with nonlinear feedback law

$$u(x) := -R^{-1}B^\top(x)\Pi(x)x$$

State-Dependent Riccati Equation (SDRE)

Remarks

- SDRE can be interpreted as a MPC loop where at a given instant, the dynamics $(A(x), B(x))$ are frozen at the current state and an LQR feedback is numerically approximated
- Even if this solution is computed for every state x, the closed-loop differs from the optimal feedback obtained from solving HJB, as the SDRE approach assumes the value function is always **locally approximated** as $V(x) \approx x^\top \Pi(x) x$
- It is possible to show local asymptotic stability for the SDRE feedback (for ODEs)

SDRE

Algorithm 1: SDRE-MPC loop

Require: $\{t_0, t_1, \ldots\}$, model, R, Q,
1: **for** $k = 0, 1, \ldots$ **do**
2: Compute $\Pi(x(t_k))$ from SDRE
3: Set $K(x(t_k)) := R^{-1}B^\top(x(t_k))\Pi(x(t_k))$
4: Set $u(t_k) := -K(x(t_k))x(t_k)$
5: Integrate system dynamics to $x(t_{k+1})$
6: **end for**

Warning

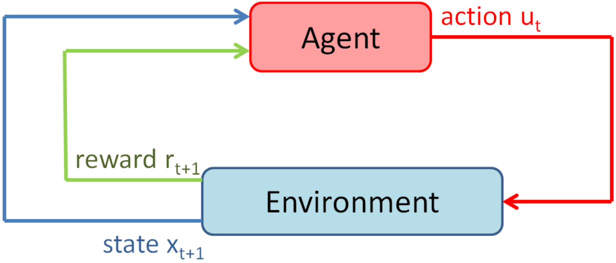
This algorithm requires an high rate of calls to an ARE solver. This is a demanding computational task for the type of large-scale dynamics arising in optimal control of PDEs

Reference

A., D. Kalise, V. Simoncini. *State-dependent Riccati equation feedback stabilization for nonlinear PDEs*, ACOM, 2023

Outline

Reinforcement Learning (RL)



Optimal Control

$$\begin{cases} \min_{u \in U} J(u) = \int_0^\infty \|x(s)\|_Q + \|u(s)\|_R \, ds \\ \dot{x}(t) = A(x(t))x(t) + B(x(t))u(t)) \\ x(0) = x_0 \end{cases}$$

Reinforcement Learning (RL)

Reinforcement Learning	**Optimal Control**
Agent	Controller
State	State
Action	Control
Reward	(opposite of) Cost
Environment	Controlled System

Both RL and OC

- are sequential decision problems
- try to optimize not only immediate rewards but also future ones

RL deals with control problems in which the dynamics of the system is (partially) uncertain but observable

State equation

$$\begin{cases} y_t(t,\xi) = \sum_{j=1}^{n} \mu_j F_j(y(t,\xi), y_\xi(t,\xi), y_{\xi\xi}(t,\xi), y_{\xi\xi\xi}(t,\xi), \dots) + B(\xi)u(t), & t \in [0,\infty), \xi \in (a,b), \\ y(0,\xi) = y_0(\xi), & \xi \in [a,b], \\ y(t,a) = 0, \ \ y(t,b) = 0, & t \in [0,\infty) \end{cases}$$

FD discretization

- $x(t) : [0,\infty) \to \mathbb{R}^d$ with $x_i(t) \approx y(t,\xi_i)$ for $i = 1, \dots, d$
- $A(x) = \sum_{j=1}^{n} \mu_j A_j(x)$ and $F_j \approx A_j$ with $A_j(x) : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ for $j = 1, \dots, n$
- **the coefficients μ_j unknown**
- the terms $A_j(x)$ given as the library
- the discretizazion of the cost corresponds to the choice $Q = \Delta\xi I_d$ with $\Delta\xi > 0$ being the spatial step size and I_d the $d \times d$ identity matrix

Goal

- control an unknown problem
- discover on the fly the problem we are controlling

RL assumption

Let μ^* be the true parameter configuration. **The dynamics generated by this true model configuration μ^* is always observable as a black box.**
This is a typical assumption in the Reinforcement Learning setting, where an agent can take actions and observe how the environment responds to them

Notation

- $x(t; u(t; \widetilde{\mu}), \mu^*)$: trajectory computed with control $u(t; \widetilde{\mu})$ related to the true model μ^*
- $x^i(u(t_i; \widetilde{\mu}), \mu^*) = x(t_i; u(t_i; \widetilde{\mu}), \mu^*)$: solution at discrete time t_i we will identify

How we learn μ through RL

- The observation of the true trajectory might provides or approximates the solution of the original controlled problem for a given input $u(t, \widetilde{\mu})$
- We do not compute the whole trajectory since we aim to discover and control **on the fly** updating the parameter configuration at each time instance
- We drop the dependence in what follows, e.g. $x^i := x^i(u(t_i; \widetilde{\mu}), \mu^*)$

Implicit scheme with explicit gain matrix K^i

$$\frac{x^{i+1} - x^i}{\Delta t} \approx \sum_{j=1}^{n} \mu_j A_j(x^{i+1}) x^{i+1} - BK^i x^{i+1}, \quad i = 0, 1, \dots$$

How we learn μ through RL

LS problem

$$\frac{x^{i+1}-x^i}{\Delta t} + BK^i x^{i+1} \approx \sum_{j=1}^{n} \mu_j A_j(x^{i+1})x^{i+1} \Longrightarrow Y^i \approx X^i \mu^i \qquad i = 1,2,3,\dots$$

- $X^i := [A_1(x^{i+1})x^{i+1}, \dots A_n(x^{i+1})x^{i+1}] \in \mathbb{R}^{d\times n}$
- $Y^i := \frac{x^{i+1}-x^i}{\Delta t} + BK^i x^{i+1} \in \mathbb{R}^d$ and $\mu^i \in \mathbb{R}^n$

Remarks

- μ^i is computed using BLR every time iteration
- we learn the system on the fly using the data provided by the control problem
- **Stopping criteria:** $\|\mu^{i+1} - \mu^i\| \leq tol_\mu$ with $tol_\mu > 0$ being the threshold since we look for a constant configuration

Algorithm 2: Online RL algorithm

Require: $\{t_0, t_1, \dots\}$, model $\{A_j(x)\}_{j=1}^{n}$, B, R, Q, $\widetilde{\mu}_0$, tol_μ, flag $= 0$,

for $i = 0, 1, \dots$ **do**

  Obtain $\Pi(x(t_i); \widetilde{\mu}^i)$ from ARE with $\widetilde{\mu}^i$

  Set $K(x(t_i); \widetilde{\mu}^i) := R^{-1}B^\top(x(t_i))\Pi(x(t_i); \widetilde{\mu}^i)$

  Set $u(t_i; \widetilde{\mu}^i) := -K(x(t_i); \widetilde{\mu}^i)x(t_i)$ (or $u(t_i; \widetilde{\mu}^i) := -K(x(t_i); \widetilde{\mu}^i)x(t_{i+1})$)

  Observe the trajectories $x^{i+1}(u(t_i; \tilde{\mu}^i), \mu^*)$

  if flag $==0$ **then**

    Update $\widetilde{\mu}^i$ using BLR

    if $\|\widetilde{\mu}^i - \widetilde{\mu}^{i-1}\|_\infty < tol_\mu$ **then**

      flag $= 1$

      $\widetilde{\mu} := \widetilde{\mu}^i$

    end if

  else

    $\widetilde{\mu}^i = \widetilde{\mu}$

  end if

end for

Outline

Numerical experiments

$$
\begin{cases}
y_t(t,\xi) = \mu_1 y_{\xi\xi}(t,\xi) + \mu_2 y_\xi(t,\xi) + \mu_3 y(t,\xi) + \mu_4 y^2(t,\xi) \\
\qquad + \mu_5 y^3(t,\xi) + \mu_6 y(t,\xi) y_\xi(t,\xi) + \mu_7 y_{\xi\xi\xi}(t,\xi) + Bu(t) & t \in [0, t_{end}], \xi \in (0,1) \\
y(0,\xi) = y_0(\xi) & \xi \in [0,1] \\
y(t,0) = 0,\ y(t,1) = 0 \ \ \mathrm{OR} \ \ y_\xi(t,0) = 0,\ y_\xi(t,1) = 0 & t \in [0, t_{end}]
\end{cases}
$$

The term $A(x)$ will be given by

$$
A(x) = \mu_1 \Delta_d + \mu_2 T + \mu_3 I_d + \mu_4 diag(x) + \mu_5 diag(x \circ x) + \mu_6 \tilde{D}(x) + \mu_7 M
$$

Time integration

Controlled trajectories are integrated in time using an implicit Euler method, which is accelerated using a **Jacobian–Free Newton Krylov** method using 10^{-5} as threshold for the stopping criteria of the method and less of 500 iterations

Test 1: Allen-Cahn

$$\begin{cases} y_t(t,\xi) = y_{\xi\xi}(t,\xi) + 11(y(t,\xi) - y^3(t,\xi)) + u(t), & t \in (0,0.5],\ x \in (0,1), \\ y(0,\xi) = 0.2\sin(\pi\xi), & x \in (0,1), \\ y(t,0) = 0,\ \ y(t,1) = 0, & t \in [0,0.5] \end{cases}$$

Parameters

- $d = 101$
- $\mu_1 = 1, \mu_3 = 11, \mu_5 = -11$
- $\Delta\xi = 0.01 = \Delta t$
- B vector is given by a vector of ones

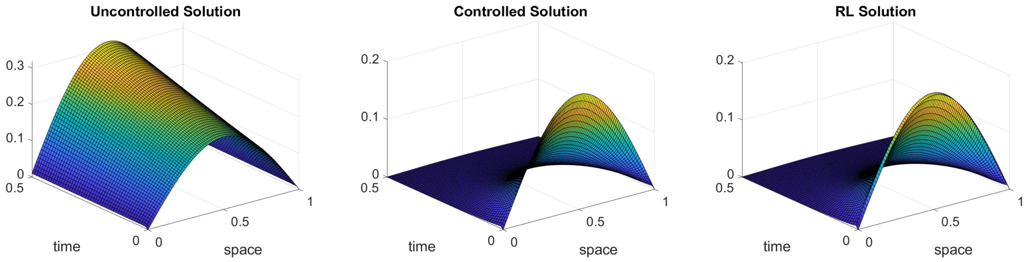
Reference

N. Altmüller, L. Grüne, K. Worthmann. *Receding horizon optimal control for the wave equation*, CDC, 2010

Test 1: Allen-Cahn

True μ	1	0	11	0	-11	0	0
Reconstr. μ	0.9992	-0.0017	11.0008	-0.0653	-10.8232	0.0431	0

Table: Reconstructed parameter configuration



Test 1: Allen-Cahn

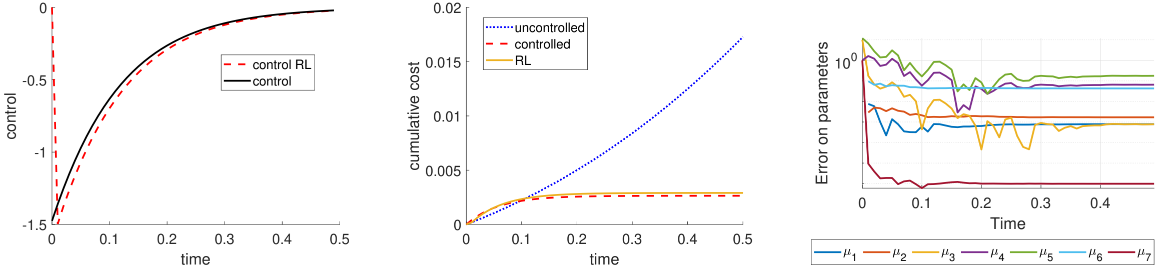


Figure: On the left the comparison between the control found using knowledge of the true μ and the control found by the RL algorithm is shown. In the middle, the cumulative cost. On the right, the error on the parameter estimation at each time.

Test 2: Viscous Burgers

$$\begin{cases} y_t(t,\xi) = 0.01 y_{\xi\xi}(t,\xi) + y(t,\xi) y_x(t,\xi) + B(\xi)u(t), & t \in [0,1],\ \xi \in (-1.5,1.5), \\ y(0,\xi) = \sin(\pi x)\chi_{[0,1]}(\xi), & \xi \in (-1.5,1.5), \\ y(t,-1.5) = 0,\ y(t,1.5) = 0, & t \in [0,1] \end{cases}$$

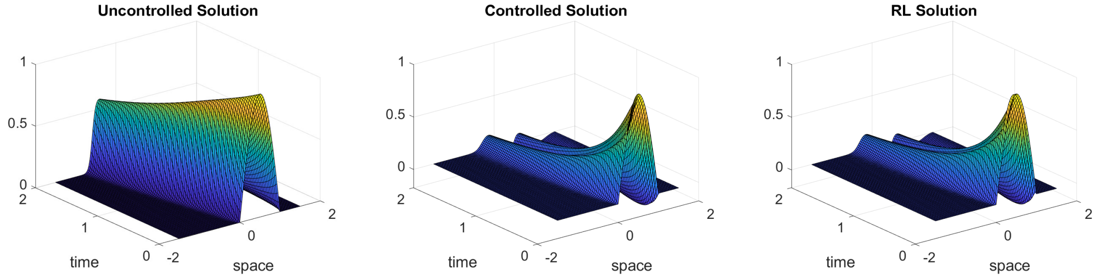
Parameters

- $d = 121$
- $\Delta\xi = 0.025 = \Delta t$
- $\mu_1 = 0.01, \mu_6 = 1$
- $B(\xi) = \begin{pmatrix} \chi_{[0.25,0.5]}(\xi) & 0 \\ 0 & \chi_{[0.75,1]}(\xi) \end{pmatrix}$
- $u(t) \in \mathbb{R}^2$

Test 2: Viscous Burgers

True μ	0.01	0	0	0	0	1	0
Reconstr. μ	0.0096	0	-0.0008	0.002	-0.001	0.9999	0

Table: Reconstructed parameter configuration



Test 2: Viscous Burgers

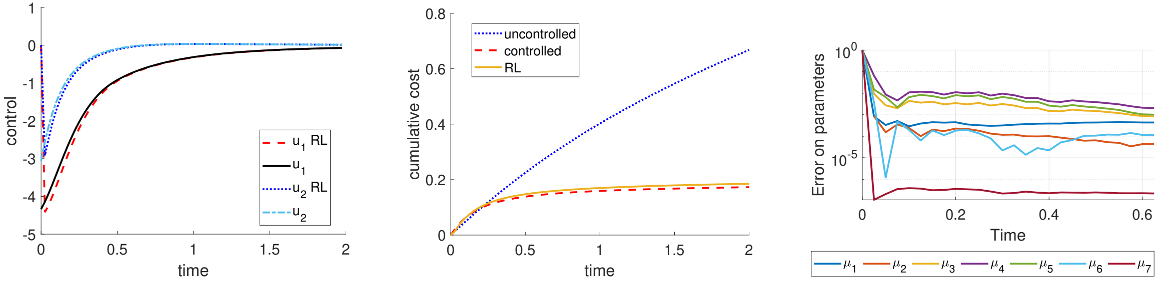


Figure: The left plot shows the comparison between each of the two components of the control found using knowledge of the true μ and the control found by the RL algorithm. The middle plot shows the cumulative cost. The right plot shows the error on the parameter estimation at each time until the update stop.

Test 2: Viscous Burgers – Black box

Full library

True μ	0.01	0	0	0	0	1	0
Reconstr. μ	0.0096	-0.0002	0.0008	0.0004	0.1762	1.021	0

Library without μ_5 (the y^3 term)

True μ	0.01	0	0	0	–	1	0
Reconstr. μ	0.0101	-0.0003	0.0007	0.0116	–	1.0199	0

Library with the right terms

True μ	0.01	–	–	–	–	1	–
Reconstr. μ	0.0099	–	–	–	–	1.0564	–

Test 2: Viscous Burgers – Black box

	full	NO μ_5	Only (μ_1, μ_6)
$\frac{\|sol\_RL - sol\_RL\_bb\|_2}{\|sol\_RL\|_2}$	0.017	0.017	0.017
$\frac{\|sol\_c - sol\_RL\_bb\|_2}{\|sol\_c\|_2}$	0.021	0.021	0.021
$\frac{\|sol\_c - sol\_RL\|_2}{\|sol\_c\|_2}$	0.021	0.021	0.021
$\frac{\|sol\_c - sol\_c\_bb\|_2}{\|sol\_c\|_2}$	0.016	0.016	0.016

Table: Comparison between RL solutions with and without black box

Comments

- With or without black box same order of the error
- Values found in the reconstructed model do not modify the quality of the results

Test 3: Korteweg-De Vries

$$
\begin{cases}
y_t(t,\xi) = \frac{1}{2} y_{\xi\xi}(t,\xi) + 6y(t,\xi)y_\xi(t,\xi) \\
\qquad - y_{\xi\xi\xi}(t,\xi) + \chi_{[1,4]}(\xi)u(t), & t \in [0,2],\ x \in (-10,7), \\
y(0,\xi) = \chi_{[0,6]}(\xi)\left(\cos\left(\frac{\pi}{3}(\xi-3)\right)+1\right) & \xi \in (-10,7), \\
y(t,-10) = 0,\ y(t,7) = 0, & t \in [0,2]
\end{cases}
$$

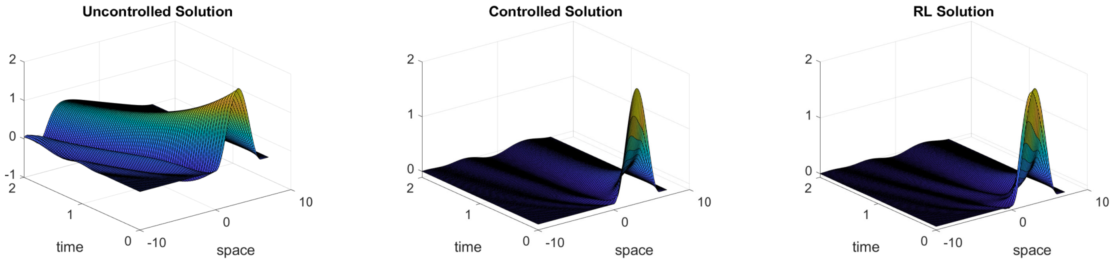
Parameters

- $d = 171$
- $\mu_1 = 0.5,\ \mu_6 = 6,\ \mu_7 = -1$
- $\Delta\xi = 0.1, \Delta t = 0.025$

Test 3: Korteweg-De Vries

True μ	0.5	0	0	0	0	6	-1
Reconstr. μ	0.4931	0.0012	0.0004	0.001	-0.0016	5.9943	-0.9999

Table: Reconstructed parameter configuration



Test 3: Korteweg-De Vries

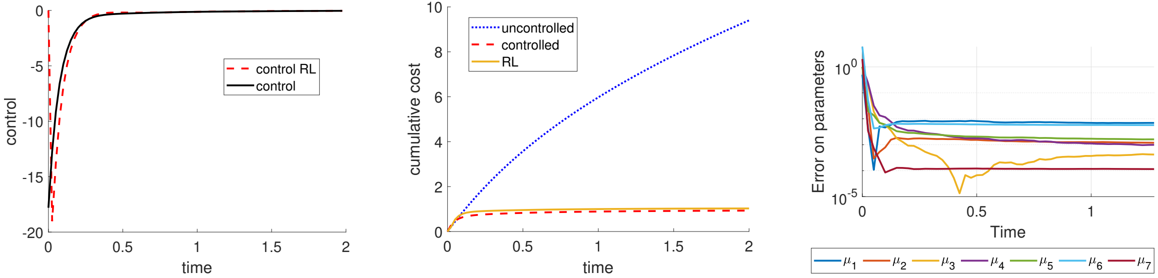
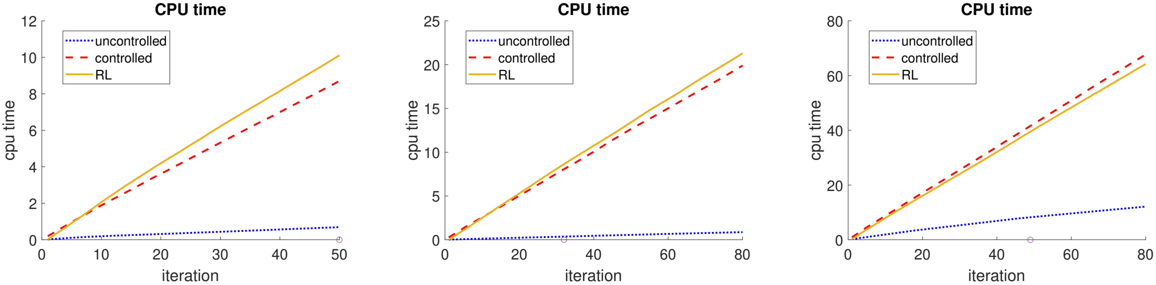


Figure: On the left the comparison between the control found using knowledge of the true μ and the control found by the RL algorithm is shown. In the middle the cumulative cost. On the right, the error on the parameter estimation at each time until the update stop.

CPU times

	uncontrolled	SDRE	RL controlled
Test 1 ($d = 101$)	0.69s	8.7s	10.1s
Test 2 ($d = 121$)	0.87s	19.9s	21.3s
Test 3 ($d = 171$)	12.1s	67.7s	64.2s

Table: CPU times of the three presented tests. The times have been computed as the arithmetic mean of the time required to complete 50 algorithm's executions



Convergence to the PDE

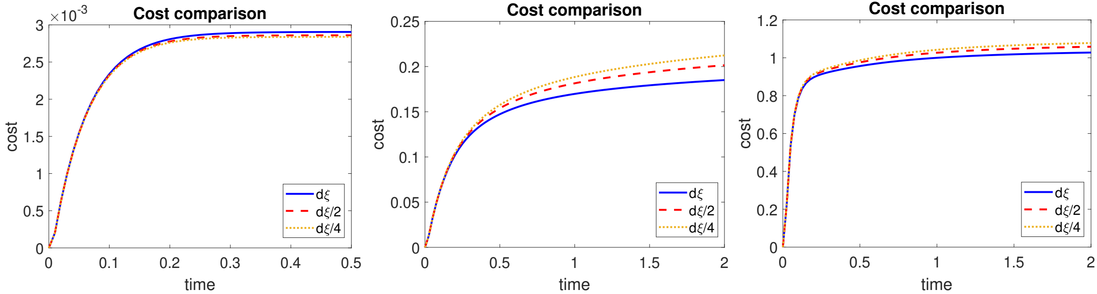


Figure: Comparison of the cost functionals for Test 1 (left), Test 2(middle), Test 3(right)

Conclusions and future works

Conclusions

- We have presented a new algorithm that identifies and controls an unknown PDE
- The method relies on RL assumptions where the model can be osbervable
- Numerical experiments have shown the convergence of our method

Future works

- Large-scale problems
- Theoretical convergence results

References

- A. Alla, D. Kalise, V. Simoncini, *State-dependent Riccati equation feedback stabilization for nonlinear PDEs*, 2023
- A. Alla, A. Pacifico, M. Palladino, A. Pesare, *Online identification and control of PDEs via Reinforcement Learning methods*, in preparation, 2023.
- H.T. Banks, B.M. Lewis, H.T. Tran, *Nonlinear feedback controllers and compensators a state-dependent Riccati equation approach*, 2007
- S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, 2016
- R. Murray, M. Palladino, *A model for system uncertainty in reinforcement learning*, 2018
- A. Pacifico, A. Pesare, M. Falcone, *A New Algorithm for the LQR Problem with Partially Unknown Dynamics*, 2022
- A. Pesare, M. Palladino, M. Falcone, *Convergence of the Value Function in Optimal Control Problems with Unknown Dynamics*, 2021
- M. Raissi, P. Perdikaris, G.E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, 2019
- R.S. Sutton, and A.G. Barto, *Reinforcement learning: An introduction*, 2018

Thank you for you attention