Mathematisches Forschungsinstitut Oberwolfach

Report No. 23/2021

DOI: 10.4171/OWR/2021/23

Mini-Workshop: Analysis of Data-driven Optimal Control (hybrid meeting)

Organized by
Lars Grüne, Bayreuth
Kirsten Morris, Waterloo

9 May – 15 May 2021

Abstract. This hybrid mini-workshop discussed recent mathematical methods for analyzing the opportunities and limitations of data-driven and machine-learning approaches to optimal feedback control. The analysis concerned all aspects of such approaches, ranging from approximation theory particularly for high-dimensional problems via complexity analysis of algorithms to robustness issues.

Mathematics Subject Classification (2010): 47D06, 49L12, 93B52, 93D09.

Introduction by the Organizers

For many complex systems, such as climate models, neural science, epidemiology, there is no mathematical model of the dynamics and/or the model includes very high-dimensional dynamics and nonlinearities. Classical model-based methods for analysis and controller design cannot be used. In the past, a simplified model of the system was used in controller design. This often led to drastic compromises in controller performance.

Feedback control is needed in order to implement a controller. Calculation of the optimal control in feedback form for a nonlinear system requires solution of a Hamilton-Jacobi-Bellman (HJB) equation. This equation has complexity increasing exponentially with system order; this is called the "curse of dimensionality". Computation of feedback controllers that meet stability and performance objectives for nonlinear systems is an open problem, except for very low-order models. Solution of Lyapunov equations is required in many controller designs and some model reduction methods. There are also challenges in solving large-order Lyapunov equations.

This mini-workshop, "Analysis of Data-driven Optimal Control", focused on using data-driven approaches to obtain optimal feedback control for complex systems. Machine learning has had great success in image classification and is now being applied to other areas. It has recently been used to approximately solve HJB and Lyapunov equations of higher order than previously possible. Fundamental questions associated with machine learning include: When does it work well? And why? There are close links between optimization and machine learning. Progress in optimal controller design may also provide insight into machine learning, and more specifically, deep learning, algorithms.

The hybrid mini-workshop was well attended with 16 participants from Europe and North America. Most talks were attended by 14-15 people, despite the issues with varying time zones. Some talks were also attended by participants of the mini-workshop "Mathematics of Dissipation — Dynamics, Data and Control", which took place in Oberwolfach in the same week. The researchers had varied backgrounds that included analysis, computational mathematics, optimization and control theory.

Many of the talks had a connection to machine learning (ML) techniques, particularly analysis of algorithms and the approximation capabilities of ML approaches, with a focus on high-dimensional problems. The remaining talks were related to the analysis of data-driven approaches using algorithms not typically regarded as machine learning.

Regarding ML, Dante Kalise explained in his talk how to combine Pontryagin's maximum principle and HJB equations with modern approximation architectures like tensor train formats and neural networks and showed some first estimates for the computational complexity of such approaches. Nathan Kutz explained how neural networks can be set up in order to represent high-dimensional dynamical systems and how Koopman operator theory can be used for the analysis of such representations. Roberto Guglielmi discussed sensitivity results and regularity for solutions of HJB equations, which are important for the subsequent approximation analysis. Sophie Tarbouriech and Enrique Zuazua both showed how control theoretic methods can be used for analyzing ML techniques, addressing optimization algorithms and universal approximation properties of neural networks, respectively. The talks by Jiequn Han, Lars Grüne and Wei Kang all centered around different aspects of high-dimensional approximations using neural networks. Jiequn Han used the particular structure of stochastic optimal control problems, while Lars Grüne and Wei Kang looked at compositional features of the functions under consideration, which were Lyapunov functions in the first case and the dynamics and cost functions in optimal control in the second case.

Data-driven methods were featured in several of the ML-related talks just discussed (quite prominently in Nathan Kutz' talk, for instance) but also in the remaining talks of the workshop. Péter Koltai discussed in his talk how particular features of complex dynamical systems can be derived from measured data. Florian Dörfler and Sebastian Peitz both looked at how the dynamics governing an optimal control problem can be derived from measured data, though with rather

different mathematical approaches from behavioural systems theory on the one hand and Koopman operator theory on the other hand.

The organizers encouraged participants to take the traditional Wednesday afternoon walk, and take photos that could then be shared. In the late Wednesday afternoon session, all participants contributed photos from their walks, and talked about it. It was nice to have a glimpse of the local life of the participants. This way some of the social aspects of a normal workshop were realized in the virtual format.

Although each participant was in a different location, and none knew all the other participants, discussion sessions were very lively. It was a group of people who, although diverse in background and approaches, had in common that they were open to listening to other ideas. One recurring topic was the advantages/disadvantages of physics-based and data-based approaches. Is a physics-based model always advantageous when there are underlying physical laws? A very important question is what problems, in particular what modelling and controller design problems, can be solved better with ML? There was consensus that ML is advantageous when only data available, but how does it differ from traditional data-based approaches? There was some optimism about the promise of ML for handling complex systems. The notion of a benchmark library of control problems was discussed and it was generally agreed that this would be a good idea, but we did not decide what problems or form it should take. In terms of understanding ML, mathematics has contributed a dynamical systems perspective to ML which has been useful in analysis of algorithms. It is possible that a control systems perspective may further contribute to a better understanding of the fundamentals of ML.

In addition to the discussion sessions with the whole group, several participants met separately to discuss newly discovered relationships between their approaches and possible future collaborations. We were particularly pleased by this because the workshop was held in hybrid form and all but one of the participants attended virtually. Although the interaction was unfortunately not as close as it would have been with an on-site event, some of the important aspects of an Oberwolfach workshop were at least partially realized. All participants were enthusiastic about how much they enjoyed the workshop; several said that it was the "best online workshop" and one of the best workshops ever.

Mini-Workshop (hybrid meeting): Analysis of Data-driven Optimal Control

Table of Contents

Abstracts

Data-enabled predictive control

Florian Dörfler
(joint work with J. Coulson, L. Huang, J. Lygeros, and I. Markovsky)

1. Motivation and Context

Direct vs. indirect: The vast realm of data-driven control methods can be classified into *indirect data-driven control* approaches consisting of sequential system identification and model-based control as well as *direct data-driven control* approaches seeking an optimal decision compatible with recorded data. Both approaches have a rich history, and they have received renewed interest cross-fertilized by novel methods and widespread interest in machine learning. The pros and cons of both paradigms have often been elaborated on: e.g., modeling and identification is cumbersome, its results are often not useful for control (due to, e.g., incompatible uncertainty quantifications), and practitioners generally prefer end-to-end approaches. While direct data-driven control promises to resolve these problems by learning control policies directly from data, the available methods often do not (yet) lend themselves to real-time and safety-critical control systems due to lack of certificates and overburdening computational and sample complexity.

Behavioral approach: In recent years a novel direct data-driven control approach has surfaced in the wake of Willems' *Fundamental Lemma* [1]. The key idea is to take the behavioral perspective and abstract a dynamical system as a set of trajectories (the behavior), e.g., an LTI system is a shift-invariant subspace in the space of all time series. While systems can be represented using parametric models (e.g., state-space equations), Willems et al. showed that the behavior may equally well be represented by a matrix time series containing raw data. This data-driven representation provides a fruitful ground for a blooming research approach to data-driven control that promises to overcome the shortcomings of other direct data-driven control approaches. Whereas the Fundamental Lemma can be taken at face value for deterministic LTI systems, extra care must be taken in the noisy and nonlinear cases. Here we discuss data-enabled predictive control (DeePC), based on the Fundamental Lemma and robustified by means of regularization.

2. Data-Driven and Non-Parametric System Theory

LTI behaviors: Consider the discrete time axis $\mathbb{Z}$, the signal space $\mathbb{R}^q$, and the space of time series $\mathbb{R}^{q\mathbb{Z}}$ consisting of all sequences $(\dots, w(-1), w(0), w(1), \dots)$ with $w(i) \in \mathbb{R}^q$. Consider a permutation matrix Π partitioning each $w(i) = \Pi \begin{bmatrix} u(i) \\ y(i) \end{bmatrix}$, where $u(i) \in \mathbb{R}^m$ and $y(i) \in \mathbb{R}^{q-m}$ are free and dependent variables that will later serve as inputs and outputs. The *behavior* $\mathscr{B}$ is defined as a subset of the space of trajectories, $\mathscr{B} \subset \mathbb{R}^{q\mathbb{Z}}$, and a system as the triple $(\mathbb{Z}, \mathbb{R}^q, \mathscr{B})$. A system

is *linear* if $\mathscr{B}$ is a subspace of $\mathbb{R}^{q\mathbb{Z}}$. Let σ denote the shift operator with action $\sigma w(t) = w(t+1)$. A system is *time-invariant* if $\mathscr{B}$ is shift-invariant: $\sigma\mathscr{B} = \mathscr{B}$.

Representations and LTI complexity: Rather than set-theoretic descriptions, one typically works with *parametric representations* (colloquially termed *models*). For instance, a *kernel representation* with *lag* ℓ specifies an LTI behavior as

$$\mathscr{B} = \ker(R(\sigma)) = \left\{ w \in \mathbb{R}^{q\mathbb{Z}} \,:\, R(\sigma)w = 0 \right\},$$

where $R(\sigma) = R_0 + R_1\sigma + \cdots + R_\ell\sigma^\ell$ is a polynomial matrix of degree ℓ, and the matrices $R_0, R_1, \ldots, R_\ell$ take values in $\mathbb{R}^{(q-m)\times q}$. Alternatively, one can unfold the kernel representation by revealing a latent variable: the state $x(t) \in \mathbb{R}^n$. The *input/state/output* (or *state-space*) *representation* is

$$\begin{aligned}\mathscr{B} = \big\{ w = \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \in \mathbb{R}^{q\mathbb{Z}} \,:\, \exists x \in \mathbb{R}^{n\mathbb{Z}} \text{ such that} \\ \sigma x = Ax + Bu\,,\; y = Cx + Du \big\},\end{aligned}$$

where A, B, C, D are matrices of compatible dimensions. We assume that the lag ℓ (resp., the state dimension n) are minimal. The state dimension n manifests itself in a kernel representation as $n = \sum_{i=1}^{q-m} \ell_i$, where ℓ_i is the lag of the ith row of $R(\sigma)$. Conversely, the lag ℓ manifests itself in a state-space representation as the observability index, i.e., the number of measurements to recover the initial value.

Hence, an LTI system is characterized by the complexity parameters (q, m, n, ℓ). We denote the corresponding class of LTI systems by $\mathscr{L}^{q,n}_{m,\ell}$.

Fundamental Lemma: Let $\mathscr{B}_L$ be the restriction of the behavior to $\mathbb{R}^{qL}$, i.e., the set of trajectories of length L. The restricted behavior $\mathscr{B}_L$ can be described by the above parametric models. Alternatively, we consider a data-driven *image representation* of $\mathscr{B}_L$ via a matrix time series. Consider a set of trajectories $w^d_j(i) \in \mathscr{B}_L$ experimentally obtained from the system. Here the index $i \in \{1, \ldots, L\}$ denotes time, the subscript j denotes the jth experiment, and the superscript d is a shorthand for "data". We define the associated trajectory matrix (or library) as

$$\mathscr{H}_L(w^d) = \begin{bmatrix} w^d_1(1) & w^d_2(1) & \ldots \\ w^d_1(2) & w^d_2(2) & \ldots \\ \vdots & \vdots & \ldots \\ w^d_1(L) & w^d_2(L) & \ldots \end{bmatrix}.$$

We present a necessary and sufficient version of the Fundamental Lemma due to [3]: Consider an LTI system $\mathscr{B} \in \mathscr{L}^{q,n}_{m,\ell}$. The following are equivalent for $L > \ell$:

$$\operatorname{colspan}\left(\mathscr{H}_L(w^d)\right) = \mathscr{B}_L \quad \Longleftrightarrow \quad \operatorname{rank}\left(\mathscr{H}_L(w^d)\right) = mL + n\,.$$

In words, the trajectory matrix $\mathscr{H}_L(w^d)$ parametrizes all L-length trajectories if and only if $\operatorname{rank}\big(\mathscr{H}_L(w^d)\big) = mL+n$. This result extends the original *Fundamental Lemma* [1] which requires input/output partitioning, controllability, persistency of excitation of order $L + n$ (i.e., $\mathscr{H}_{L+n}(u)$ must have full row rank), and a Hankel structure of $\mathscr{H}_L(w^d)$ as sufficient conditions.

3. DeePC and Robustifications

Direct data-driven control: The indirect control approach is based on sequential data collection, pre-processing, identification of a parametric model, state estimation, and finally model-based control. In comparison, we can formulate a compact direct data-driven control problem based on the fundamental lemma:

$$\begin{aligned} &\text{minimize} \quad \text{over } w,\, g \quad f(w - w_r) \\ &\text{subject to} \quad \mathscr{H}_L(w^d)g = \begin{bmatrix} w_{\text{ini}} \\ w \end{bmatrix}. \end{aligned} \tag{1}$$

Here $w \in \mathbb{R}^{qL}$ is the future trajectory to be designed, $w_r \in \mathbb{R}^{qL}$ is a reference to be tracked, $w_{\text{ini}} \in \mathbb{R}^{qT_{\text{ini}}}$ is a prefix trajectory of length $T_{\text{ini}} \geq \ell$ to estimate the initial condition, f is the control cost, and the vector g selects columns of the trajectory matrix according to the fundamental lemma. The data-driven formulation (1) is due to [2] (though with Hankel structure) and assumes perfect data: both offline data w^d and online measurements w_{ini} come from a deterministic LTI system.

DeePC formulation: For real-world applications the problem formulation (1) has to be robustified against noise as well as nonlinearities. Below we summarize the robustifications proposed in [4, 5, 6]. The need for robustification is threefold.

First, note that when implementing (1) in receding-horizon, the online measurement data w_{ini} is noisy leading to infeasible constraint equations. As a remedy, DeePC opts for a moving-horizon least-error estimation and softens these constraints with a slack variable σ penalized in the cost. Second, the data-driven problem (1) is also subject to multiplicative noise via the data matrix $\mathscr{H}_L(w^d)$. This noise can be mitigated offline by pre-processing the trajectory library (e.g., by seeking a low-rank approximation of $\mathscr{H}_L(w^d)$), but in the spirit of direct data-driven control – seeking an online decision based on raw data – DeePC opts for a regularizing the problem (1) with a nonnegative term $h(g)$, which will be justified below. A third minor – yet practicably important – modification is to augment the data-driven LQ problem (1) with constraints $w \in \mathcal{W}$ on inputs and outputs, respectively. These can account for, e.g., saturation, operational limits, or terminal constraints needed for closed-loop stability of the predictive control.

We arrive at the regularized, robustified, and constrained DeePC formulation

$$\begin{aligned} &\text{minimize} \quad \text{over } w \in \mathcal{W},\, g,\, \sigma \quad f(w - w_r) + \gamma \cdot \|\sigma\|_p + \lambda \cdot h(g) \\ &\text{subject to} \quad \mathscr{H}_L(w^d)g = \begin{bmatrix} w_{\text{ini}} + \sigma \\ w \end{bmatrix}, \end{aligned} \tag{2}$$

where γ and λ are nonnegative hyperparameters, and $\|\cdot\|_p$ is a p-norm.

Regularization replaces system identification: Different regularizers $h(g)$ have been proposed. One set of regularizers can be derived from implicit data preprocessing and system identification which are by-passed in the DeePC formulation (2) [6]. These by-passed intermediate steps can be formally modeled as multi-level optimization problems. For example, the outer problem is optimal control based on a model, where the model itself is identified in an inner least-square identification

problem. This particular bi-level problem can be formally reduced and convexified to a single-level problem of the form (2) with regularizer $h(g) = \|Tg\|_p$, where T is a projector accounting for the least-square orthogonality condition. If the inner problem corresponds to denoising by low-rank approximating the data matrix $\mathscr{H}_L(w^d)$, then a convexification leads to $h(g) = \|g\|_1$. This 1-norm regularizer promotes a sparse solution g selecting a sparse basis from the trajectory library.

Regularization to achieve robustness: Another approach leading to regularization is due to distributional robustness [5]. Problem (2) can be abstracted as

$$\text{minimize} \quad \text{over } x \in \mathcal{X} \quad f'(\xi, x) = \mathbb{E}_{\xi\sim\hat{P}}\left[f'(\hat{\xi}, x)\right], \tag{3}$$

where x collects (w, g, σ), ξ are the data samples w^d, and $\hat{P}$ is the associated empirical (sample) distribution. If the solution of the *sample-average problem* (3) is implemented on the real system, one suffers an out-of-sample loss since the true distribution P might have arisen from a complicated nonlinear stochastic process that is only poorly represented by the samples $\hat{P}$. To be robust against such processes, we propose the distributionally robust problem formulation

$$\inf_{x\in\mathcal{X}} \sup_{Q\in\mathbb{B}^p_\epsilon(\hat{P})} \quad \mathbb{E}_{\xi\sim\hat{Q}}\left[f'(\hat{\xi}, x)\right], \tag{4}$$

where $\mathbb{B}^p_\epsilon(\hat{P})$ is a Wasserstein ball centered at $\hat{P}$, of radius $\epsilon > 0$, and with metric induced by the p-norm. One can formally show that, under integrability conditions, the formulation (4) is equivalent to regularized DeePC (2) with λ being the product of ϵ and the Lipschitz constant of the cost, and $h(g) = \|g\|^\star_p$, where $\|\cdot\|^\star_p$ is the dual norm of that one used to construct the Wasserstein ball.

Finally, a similar robustification can be applied to stochastic constraints [5], and similar robustifying regularizations can also be derived deterministically [7].

4. Applications and Validations of DeePC

We demonstrate the utility of DeePC with nonlinear and stochastic case studies (numerical as well as experimental) from the energy and robotics domains [8, 9].

References

[1] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[2] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[3] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," 2020, Available at http://homepages.vub.ac.be/ imarkovs/publications/identifiability.pdf.

[4] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *European Control Conference*, 2019, pp. 307–312.

[5] ——, "Distributionally robust chance constrained data-enabled predictive control," 2020, Available at https://arxiv.org/abs/2006.01702.

[6] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," 2021, Available at arXiv:2101.01273.

[7] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust Data-Enabled Predictive Control: Tractable Formulations and Performance Guarantees," 2021, Available at arXiv:2105.07199.

[8] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," 2019, *Available at arXiv:1911.12151.*
[9] E. Elokda, J. Coulson, P. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," 2019, Available at https://www.research-collection.ethz.ch/.

Curse-of-dimensionality-free deep learning of Lyapunov function using compositionality

Lars Grüne

Introduction. Lyapunov functions are one of the key tools for the stability analysis of nonlinear systems of ordinary differential equations

$$\dot{x} = f(x)$$

with $x \in \mathbb{R}^d$. They do not only serve as a certificate for asymptotic stability of an equilibrium but also allow to give estimates about its domain of attraction or to quantify its robustness with respect to perturbations, for instance, in the sense of input-to-state stability. In the context of this talk, Lyapunov functions serve as a simplified model problem for more general optimal control problems, because just like optimal value functions they can be computed using Hamilton-Jacobi-Bellman (HJB) PDEs, which in case of Lyapunov functions simplify to linear equations [11].

This similarity allows to use neural network-based approximation techniques developed recently for certain classes of high-dimensional PDEs (e.g., [4, 5, 10]) also for Lyapunov functions. While it is well known that any continuous function can in principle be approximated by a neural network, it is also known that this does in general not remove the curse of dimensionality [1, 7], i.e., the fact that the computational effort and the storage effort grows exponentially with the space dimension d. However, under additional structural assumptions on the function to be approximated (here: the Lyapunov function), improved statements are possible. One of the structural assumptions under which this is possible is that the function is compositional [8].

Small-gain theory and compositional Lyapunov functions. A particular form of a compositional function is a separable function

$$V(x) = \sum_{k=1}^{s} V_k(z_k),$$

where the $z_k \in \mathbb{R}^{d_k}$, $k = 1, \ldots, s$, are subvectors of $x \in \mathbb{R}^d$ of moderate dimensions d_k, which do not grow (or at least do not grow too fast) with d. It is relatively easy to check that if the dynamics $\dot{x} = f(x)$ can be decomposed into m decoupled subsystems

$$z_k = f_k(z_k)$$

and for each subsystem there exists a Lyapunov functions V_k, then the function $V(x) = \sum_{k=1}^{m} V_k(z_k)$ is a Lyapunov function for the overall system, which is obviously separable.

Obviously, the existence of s decoupled subsystems is a very strong condition. However, nonlinear small-gain theory provides weaker conditions for the existence of separable Lyapunov functions. It provides conditions on the strength of the interconnection between the subsystems f_k under which the existence of separable Lyapunov functions is still guaranteed. These results come in different forms; a result which yields a Lyapunov function of the separable form from above can be found in [2], related results can be found in, e.g., [9, 3] and many other references.

Neural network structure and approximation result. If the subsystem structure defined by the vectors z_k needed for writing the separable Lyapunov function is known, then the neural network from Figure 1, in which each grey block approximates one of the low dimensional functions V_k, is easily seen to be appropriate for computing an approximation $W(\cdot;\theta) \approx V(\cdot)$.

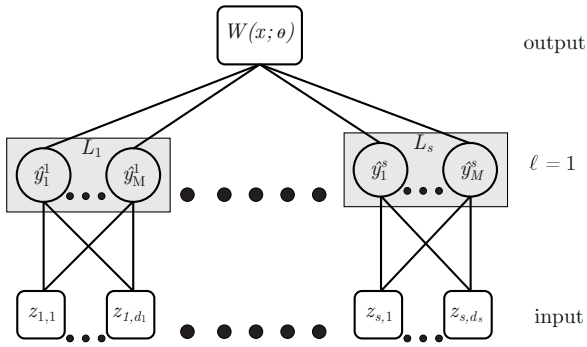

FIGURE 1. Neural network for approximating separable Lyapunov function with known structure

However, it is in general not easy to determine the subsystem structure that would allow for a separable Lyapunov function analytically. As a remedy, we can add another layer to the network, in which a linear change of coordinates is performed that "identifies" the necessary subsystems. Figure 2 shows such a network.

With these networks, it is possible to prove that the number of neurons that is needed for approximating V with a desired accuracy (in the L^∞-sense on a compact set) grows only polynomially in d. For precise statements and assumptions as well as for numerical tests illustrating the practicability of the approach we refer to [6].

Discussion. In the context of this mini-workshop, the results from this talk are related to several other talks. The idea to use a coordinate transformation in order to transform the system to a form in which the considered problem can be easily solved by a neural network is prominently featured in Nathan Kutz' talk. There, instead of being compositional, after transformation the problem can be efficiently represented, e.g., by a Koopman operator of small dimension. Compositionality also played a key role in Wei Kang's talk, with the main difference that in his

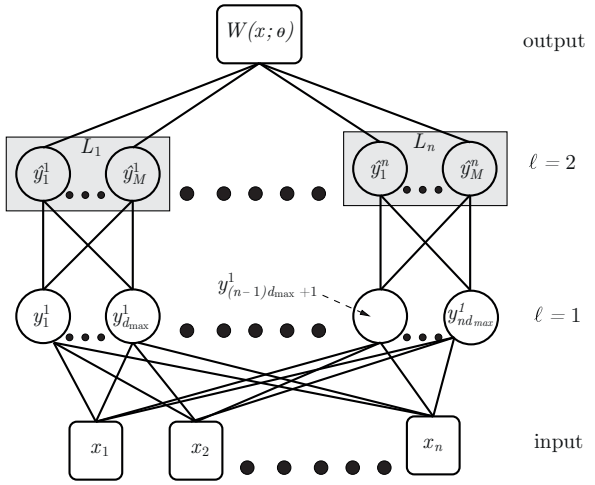

FIGURE 2. Neural network for approximating separable Lyapunov function with unknown structure

talk the ingredients of the problem (dynamics, cost function, ...) are assumed to be compositional while in this talk the Lyapunov function (i.e., the solution of the problem) is assumed to be compositional. The precise relation between these approaches and the respective advantages and disadvantages still need to be explored.

Conceptually, the approach has many similarities with the one presented in Jiequn Han's talk. However, it seems that the compositional structure of the solution is a more general and less demanding assumption than the stucture exploited in his results. Conversely, in Han's talk the problem class is more general and challenging. Finally, the learning schemes presented in this talk (described in [6]) bear certain similarities with methods from Dante Kalise's talk. Here in particular the differences between supervised and non-supervised learning highlighted in D. Kalise's talk deserve attention.

REFERENCES

[1] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals Systems*, **2** (1989), 303–314.

[2] S. Dashkovskiy, H. Ito and F. Wirth, On a small gain theorem for ISS networks in dissipative Lyapunov form, *Eur. J. Control*, **17** (2011), 357–365.

[3] S. N. Dashkovskiy, B. S. Rüffer and F. R. Wirth, Small gain theorems for large scale systems and construction of ISS Lyapunov functions, *SIAM J. Control Optim.*, **48** (2010), 4089–4118.

[4] J. Darbon, G. P. Langlois and T. Meng, Overcoming the curse of dimensionality for some Hamilton-Jacobi partial differential equations via neural network architectures, *Res. Math. Sci.*, **7** (2020), Paper No. 20, 50.

[5] W. E, J. Han and A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, *Commun. Math. Stat.*, **5** (2017), 349–380.

[6] L. Grüne, Computing Lyapunov functions using deep neural networks, *J. Comput. Dyn.*, **8** (2021), 131–152.
[7] H. N. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions, *Neural Computations*, **8** (1996), 164–177.
[8] T. Poggio, H. Mhaskar, L. Rosaco, M. Brando and Q. Liao, Why and when can deep – but not shallow – networks avoid the curse of dimensionality: a review, *Int. J. Automat. Computing*, **14** (2017), 503–519.
[9] B. S. Rüffer, Monotone Systems, Graphs, and Stability of Large-Scale Interconnected Systems, Dissertation, Fachbereich 3, Mathematik und Informatik, Universität Bremen, Germany, 2007.
[10] J. Sirignano and K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364.
[11] V. I. Zubov, *Methods of A.M. Lyapunov and their Application*, P. Noordhoff, Groningen, 1964.

Sensitivity analysis of the value function for infinite dimensional optimal control problems and its relation to Riccati equations

Roberto Guglielmi

The main objective of the presentation is to provide a sensitivity analysis with respect to the initial condition of the value function associated with optimal control problems subject to semilinear parabolic equations. The higher order approximation of the value function associated with the control system is developed up to the second order. Indeed, we derive sufficient conditions which guarantee that the first and the second derivatives of the value function can be expressed in terms of the adjoint equation and a suitably defined Riccati operator. More precisely, we aim at obtaining a local representation of the value function $\mathcal{V}_r$ along optimal trajectories of the form

$$(1)\quad \mathcal{V}_r(z) = \mathcal{V}_r(\bar{z}) + (\bar{p}(0), z-\bar{z})_{2,\Omega} + \frac{1}{2}(P(0)(z-\bar{z}), z-\bar{z})_{2,\Omega} + o\left(\|z-\bar{z}\|^2_{\infty,\Omega}\right),$$

in a neighborhood of the reference initial condition $\bar{z}$, where $\bar{p}(0)$ and $P(0)$ are respectively the values of the adjoint state and the solution of an appropriately defined differential Riccati equation evaluated at the time $t = 0$.

Besides the inherent interest in this topic, the importance of such results resides in the fact that this information can be used for local approximations of the solution to the Hamilton-Jacobi-Bellman equation for optimal control problems. The numerical solution of this equation is impeded by the high dimension of this equation (the so-called *curse of dimensionality*). Therefore one possible numerical approach consists in computing the value function and its derivatives at reference points followed by high-dimensional interpolation. In [12, 13], this idea is exploited on the base of a zero and first order information. This approach is extremely powerful when embedding model-based knowledge of the system into data-driven schemes, where partial information on the system in a finite number of points may enhance deeper insight on the controlled dynamics.

More precisely, given a positive constant $\alpha > 0$ and a target state $y_d : Q \to \mathbb{R}$, we consider the optimization problem

$$\min_{u \in L^2(0,T)} \left[\tilde{J}(z, y, u) = \frac{1}{2} \|y - y_d\|^2_{2,Q} + \frac{\alpha}{2} \|u\|^2_{L^2(0,T)} \right] ,$$

where the state y is subject to the control system

$$\begin{aligned} y_t(x,t) - Ay(x,t) + f(y(x,t)) = u(t)\chi_\omega(x) \,, \quad & (x,t) \in Q \\ y(x,t) = 0 \,, & (x,t) \in \Sigma \\ y(x,0) = z(x) & x \in \Omega \end{aligned}$$

for some initial data $z \in L^\infty(\Omega)$, a control $u \in L^2(0,T)$, a uniformly elliptic operator A, a monotone and smooth nonlinearity f, and given functions $\chi_\omega \in L^\infty(\Omega)$ with support localized in an open subset ω of Ω and $y_d \in L^p(0,T; L^q(\Omega))$, where $p, q \in [2, \infty]$ such that $\frac{1}{p} + \frac{d}{2q} < 1$, $d \geq 2$. The adjoint system associated with the control problem has the form

$$\begin{cases} -p_t + A^* p + f'(y)p = y - y_d \,, & \text{in } Q \,, \\ p = 0 \,, & \text{on } \Sigma \,, \\ p(x,T) = 0 \,, & \text{in } \Omega \,, \end{cases}$$

where A^* stands for the adjoint operator of A.

$$-\frac{\mathrm{d}P(t)}{\mathrm{d}t}\eta + P(t)\tilde{A}(t)\eta + \tilde{A}^*(t)P(t)\eta + P(t)D_1P(t)\eta = D_2(t)\eta \tag{2}$$

for all $\eta \in D(A) = H^2(\Omega) \cap H^1_0(\Omega)$ and with $P(T) = 0$, where $\tilde{A}(t) := A + f'(\bar{y}(t))I$ with domain

$$D(\tilde{A}(t)) = D(A) = \{\varphi \in V : A\varphi \in H\} \quad \text{ for all } \ t \in (0,T) \,.$$

Since we deal with a nonlinear state equation, we admit the existence of locally optimal solutions which are not globally optimal. For this reason, the expansion (1) will be possible in a neighbourhood of locally optimal solutions at which appropriate second order sufficient optimality conditions are satisfied. Indeed, a sufficient condition to obtain the representation (1) is to assume that there exists $\varepsilon > 0$ such that

$$\bar{p}(x,t) f''(\bar{y}(x,t)) < \varepsilon \leq 1 \quad \text{for a.e. } \ (x,t) \in Q \,, \tag{3}$$

which implies a second order sufficient optimality condition. The smallness condition (3) links the optimal adjoint state with the degree of the nonlinearity. A similar condition was used ealier in [11] and also appears in [20] on the study of the turnpike property for semilinear optimal control problems.

Let us point to some related literature. The two paper [1] and [8] are closely related to this presentation. In [1] the authors study the sensitivity with respect to the initial condition of the value function associated with a semilinear parabolic optimal control problem, and derive a second order expansion of the value function but do not establish the relationship to Riccati equations. They consider instead problems with constraints, and under a polyhedricity assumption and second order

sufficient optimality conditions the second order term of the expansion of the value function in [1] is related to an auxiliary quadratic optimal control problem with control constraints. The expansion is achieved by matching local estimates from below and from above, following the methods in [2, 3]. The expansion of the value function has also been considered in [4], in the case of state constraints. Our main interest is to link the second order term of the expansion of the value function to the solution of a suitably defined Riccati equation. For this reason, we must consider an optimal control problem which is well posed without constraints. In this respect let us note that while the constraints create extra technical difficulties they also regularize the underlying problem in a convenient way. In our work, under assumption (3), we prove that the value function is twice Fréchet differentiable along optimal trajectories. This is achieved by differentiating the optimality system with respect to the initial condition, for which we apply an implicit function theorem and use L^∞ regularity of the state and adjoint solutions. In a similar fashion, for a nonlinear Mayer optimal control problem with the state equation given by a differential inclusion in finite dimensions, the paper [8] proves the relation of the second order term in the expansion of the value function with a suitable Riccati operator. Indeed, the authors give conditions to ensure that the value function is twice Fréchet differentiable along optimal trajectories, and moreover that the Hessian of the value function satisfies a matrix differential Riccati equation. In this perspective, the results of the present paper can be seen as an analogue of the results in [8] in infinite dimension (but for different optimal control settings).

In a broader context, the stability and sensitivity of optimization problems under perturbations of parameters has received a considerable amount of attention in the literature, and we can only refer very selectively to related literature. An early work on the differentiability of the solution to parametrized problems in finite dimension dates back to Fiacco [10], where the author applies the implicit function theorem to show the regularity of the solution, under a second order sufficient optimality condition with respect to a strong topology. First and second order optimality conditions are also the main assumptions adopted to analyze stability and sensitivity in several different settings for optimization problems subject to an ordinary differential equation (ODE) in the presence of state and control constraints, see for example [15, 16, 18]. In the context of optimal control problems governed by a partial differential equation (PDE), the work [22] gives conditions to ensure the directional differentiability of the solution under a linear perturbation of the operator and of the source term, with an application to shape sensitivity. In addition, regularity properties of the value function have been further analyzed exploiting the characterization of the value function as the viscosity solution of a Hamilton-Jacobi equation. For example, the paper [14] proves the semi-concavity of the value function for a stochastic optimal control problem, whereas in [5, 6, 7] the semi-concavity and Lipschitz continuity of the value function is established for optimal control problems constrained to semilinear parabolic or hyperbolic equations. In the paper [17], the authors show the stability in L^∞ for the solution of a class of parametrized optimal control problems governed by a semilinear parabolic

equation. Finally, a crucial aspect in our analysis is provided by the L^∞ regularity of the solutions to the nonlinear state equation, deduced in the papers [9, 21].

References

[1] M. Bergounioux and N. Merabet, *Sensitivity analysis for optimal control of problems governed by semilinear parabolic equations*, Control Cybernet. **29** (2000), 861–886.

[2] J.F. Bonnans and R. Cominetti, *Perturbed optimization in Banach spaces. I. A general theory based on a weak directional constraint qualification*, SIAM Journal on Control and Optimization **34** (1996), 1151–1171.

[3] J.F. Bonnans and R. Cominetti, *Perturbed optimization in Banach spaces. II. A theory based on a strong directional constraint qualification*, SIAM Journal on Control and Optimization **34** (1996), 1172–1189.

[4] J.F. Bonnans and P. Jaisson, *Optimal Control of a Parabolic Equation with Time-Dependent State Constraints*, SIAM Journal on Control and Optimization **48** (2010), 4450–4571.

[5] P. Cannarsa, *Regularity properties of solutions to Hamilton-Jacobi equations in infinite dimensions and nonlinear optimal control*, Differential Integral Equations **2** (1989), 479–493.

[6] P. Cannarsa and H. Frankowska, *Value function and optimality condition for semilinear control problems. II. Parabolic case*, Appl. Math. Optim. **33** (1996), 1–33.

[7] P. Cannarsa and H. Frankowska, *Value function and optimality conditions for semilinear control problems*, Appl. Math. Optim. **26** (1992), 139–169.

[8] P. Cannarsa, H. Frankowska and T. Scarinci, *Second-order sensitivity relations and regularity of the value function for Mayer's problem in optimal control*, SIAM Journal on Control and Optimization **53** (2015), 3642–3672.

[9] E. Casas, *Pontryagin's principle for state-constrained boundary control problems of semilinear parabolic equations*, SIAM Journal on Control and Optimization **35** (1997), 1297–1327.

[10] A.V. Fiacco, *Sensitivity analysis for nonlinear programming using penalty methods*, Math. Programming **10** (1976), 287–311.

[11] K. Ito and K. Kunisch, *Augmented Lagrangian-SQP methods for nonlinear optimal control problems of tracking type*, SIAM Journal on Control and Optimization **34** (1996), 874–891.

[12] D. Kalise and K. Kunisch, *Polynomial approximation of high-dimensional Hamilton Jacobi Bellman equations and applications to feedback control of semilinear parabolic PDEs*, SIAM Journal on Scientific Computing **2** (2018), 629–652.

[13] W. Kang and L.C. Wilcox, *Mitigating the Curse of Dimensionality: Sparse Grids Characteristics Method for Optimal Feedback Control and HJB equation*, Computational Optimization and Applications **68** (2017), 289–315.

[14] P.L. Lions, *Viscosity solutions of fully nonlinear second-order equations and optimal stochastic control in infinite dimensions. I. The case of bounded stochastic evolutions*, Acta Math. **161** (1988), 243–278.

[15] K. Malanowski, *Stability analysis for nonlinear optimal control problems subject to state constraints*, SIAM Journal on Control and Optimization **18** (2007), 926–945.

[16] K. Malanowski, *Stability and sensitivity analysis for linear-quadratic optimal control subject to state constraints*, Optimization **56** (2007), 463–478.

[17] K. Malanowski and F. Tröltzsch, *Lipschitz stability of solutions to parametric optimal control problems for parabolic equations*, Z. Anal. Anwendungen **18** (1999), 469–489.

[18] H. Maurer and H. Pesch, *Solution differentiability for nonlinear parametric control problems*, SIAM Journal on Control and Optimization **32** (1994), 1542–1554.

[19] A. Porretta and E. Zuazua, *Long time versus steady state optimal control*, SIAM J. Control Optim. **51** (2013), 4242–4273.

[20] A. Porretta and E. Zuazua, *Remarks on Long Time Versus Steady State Optimal Control*, Mathematical Paradigms of Climate Science, Springer International Publishing (2016), 67–89.

[21] J.P. Raymond and H. Zidani, *Hamiltonian Pontryagin's principles for control problems governed by semilinear parabolic equations*, Appl. Math. Optim. **39** (1999), 143–177.
[22] J. Sokołowski, *Sensitivity analysis of control constrained optimal control problems for distributed parameter systems*, SIAM J. Control Optim. **25** (1987), 1542–1556.

Solving high-dimensional control problems with deep learning

Jiequn Han

Solving high-dimensional control problems and partial differential equations (PDEs) has been a notoriously difficult problem in scientific computing and computational science, due to the well-known curse of dimensionality (CoD): The computational complexity grows exponentially as a function of the dimensionality of the problem. The development of deep learning has provided us new powerful tools to solve these high-dimensional problems. This talk starts with a control-viewpoint of deep learning, discussing the connection between optimizing deep neural networks (DNN) and solving optimal control problems. Inspired by such a connection, we present two lines of research that leverage deep learning to solve high-dimensional control problems: (1) solving stochastic control problems [2], with possible delay effect [4] (2) solving parabolic PDEs based on backward stochastic differential equations (BSDE) [1, 5, 3].

One intimate connection between deep learning and optimal control can be understood from a simple linear-quadratic example. Given the dynamics

$$x_{t+1} = x_t + a_t, \quad x_t \in \mathbb{R}^d, a_t \in \mathbb{R}^d,$$

we aim to solve the control problem

$$\min_{\{a_t\}_{t=0}^{T-1}} \sum_{t=0}^{T-1} \frac{\beta}{2}\|a_t\|^2 + \frac{\gamma}{2}\|x_T\|^2.$$

By dynamic programming, we have the Bellman optimality equation characterizing the value function

$$V_t(x) = \min_a \{\frac{\beta}{2}\|a\|^2 + V_{t+1}(x+a)\}, \quad V_T(x) = \frac{\gamma}{2}\|x\|^2.$$

One can then prove by induction that

$$V_t(x) = \frac{\gamma_t}{2}\|x\|^2, \quad a_t^* = -\frac{\gamma_{t+1}}{\gamma_{t+1}+\beta}x_t,$$

where

$$\gamma_t = \frac{\beta\gamma_{t+1}}{\gamma_{t+1}+\beta}, \quad \gamma_T = \gamma.$$

If we know the optimal control a^* must be in an affine form with respect to the state variable, we can look for a closed-loop control in the form of $a_t = W_t x_t + b_t$,

i.e.,

$$\min_{\{W_t,b_t\}_{t=0}^{T-1}} \mathbb{E}_{x_0} \sum_{t=0}^{T-1} \frac{\beta}{2}\|a_t\|^2 + \frac{\gamma}{2}\|x_T\|^2,$$
$$\text{subject to } x_{t+1} = x_t + W_t x_t + b_t,$$

where the expectation is taken with respect to a specified distribution of the initial state x_0. This problem can be readily interpreted as optimizing a linear residual network with a loss function consisting of a terminal term related to the output x_T and intermidate terms related to the residual block a_t. One can build a deep residual neural netowrk corresponding to this problem and use backpropogation and staochastic gradient descent (SGD) to find near-optimal W_t, b_t. This is an approach dramatically different the dynamic programming. It is flexible to use in general high-dimensions: one can add nonlinear activation or use multiple layers to deal with the general cases. This control example exhibits close resemblance to the DNN-SGD paradigm in deep learning. From an abstract viewpoint, DNN can be viewed as a (discrete) dynamical system, of which residual network is a good example. SGD is a natural consequence when applying GD to the problem in which the objective function is an expectation.

The above methodology can be readily applied to the stochastic control problems, as proposed in [2], which is one of the earliest applications of deep learning to problems in scientific computing. We construct a feedforward neural network at each timestamp to approximate the optimal control and stack them together according to the system's dynamics to form a big deep neural network for end-to-end optimization. The work [3] further extends this idea to the stochastic control problems with a delay effect. A recurrent neural network is introduced to approximate the optimal control to capture the problem-dependent feature of the problem. It turns out the recurrent neural network can solve complex problems more efficiently and accurately than feedforward neural networks in this setting.

The Deep BSDE method was the first deep learning-based numerical algorithm for solving general nonlinear (parabolic) PDEs in high dimensions [1, 5]. It also shares a similar spirit with the above work, in the sense of reformulating the PDE as a stochastic control problem. This is done with the help of BSDEs, hence the name "Deep BSDE method". As a by-product, the Deep BSDE method is also an efficient algorithm for solving high-dimensional BSDEs. The BSDEs can be interpreted as a nonlinear version of the famous Feynman-Kac formula. Under its reformulation, we start with the unknown solution at the initial time and aim to find the optimal control, corresponding to the gradient of the true solution, such that the forward stochastic process corresponding to the solution would match the given terminal condition. Similarly, we approximate the unknown control with feedforward neural networks and employ SGD to find near-optimal parameters. Later the method is extended in combination with the fictitious play technique to solve Nash equilibrium in many-player games [3].

The numerical results in all works mentioned above suggest that the proposed algorithms achieve satisfactory accuracy and, at the same time, can handle rather

high-dimensional problems, say, in 100 dimensions. This opens up new possibilities in economics, finance, and operational research, by considering more realistic and informative high-dimensional states. See more relevant discussion in a recent review article [6].

References

[1] E, W., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5*, 4 (2017), 349–380.

[2] Han, J., and E, W. Deep learning approximation for stochastic control problems. *Deep Reinforcement Learning Workshop, NIPS* (2016).

[3] Han, J., and Hu, R. Deep fictitious play for finding Markovian Nash equilibrium in multi-agent games. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference (MSML)* (2020), vol. 107, pp. 221–245.

[4] Han, J., and Hu, R. Recurrent neural networks for stochastic control problems with delay. *arXiv preprint arXiv:2101.01385* (2021).

[5] Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences 115*, 34 (2018), 8505–8510.

[6] Han, J., Jentzen, A., et al. Algorithms for solving high dimensional pdes: From nonlinear monte carlo to machine learning. *arXiv preprint arXiv:2008.13333* (2020).

Supervised learning for optimal feedback laws

Dante Kalise

(joint work with Giacomo Albi, Behzad Azmi, Sara Bicego, Karl Kunisch)

We study the design of feedback laws for the problem

$$\min_{u(\cdot)\in L^2(t_0,T;\mathbb{R}^m)} J(u;t_0,x) := \int_{t_0}^{\top} \ell(y(t)) + \beta\|u(t))\|_2^2\, dt\,, \qquad \beta > 0\,, \tag{1}$$

subject to $y(t)$ in $\mathbb{R}^n$ being the solution to the control-affine nonlinear dynamics

$$\frac{d}{dt}y(t) = f(y(t)) + g(y(t))u(t)\,, \qquad y(t_0) = x\,. \tag{2}$$

We assume that the running cost $\ell : \mathbb{R}^n \to \mathbb{R}$, the dynamics $f : \mathbb{R}^n \to \mathbb{R}^n$, and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are continuously differentiable. It is well-known that in this case the optimal feedback synthesis is given by the Dynamic Programming Principle, synthesizing the optimal control as

$$u^*(t,x) = \operatorname*{argmin}_{u\in\mathbb{R}^m} \left\{\beta\|u\|_2^2 + \nabla V(t,x)^\top (g(x)u)\right\} = -\frac{1}{2\beta} g^\top(x)\nabla V(t,x)\,, \tag{3}$$

where $V(t,x) : [0,T]\times\mathbb{R}^n \longrightarrow \mathbb{R}$ is the value function of the problem

$$V(t,x) := \inf_{u(\cdot)} \{J(u;t,x) \text{ subject to } (2)\}\,. \tag{4}$$

This value function satisfies a first-order, nonlinear Hamilton-Jacobi-Bellman (HJB) partial differential equation of the form

$$\text{(HJB)} \qquad \begin{cases} \partial_t V(t,x) - \frac{1}{4\beta}\|g(x)^\top \nabla V(t,x)\| + \langle V(t,x), f(x)\rangle + \ell(x) = 0\,, \\ V(T,x) = 0\,, \end{cases}$$

to be solved over the state space of the dynamics $\mathcal{X} \subset \mathbb{R}^n$. We assume that V is C^1, see e.g.[1]. The main difficulty of this approach lies in the construction of a numerical scheme for (HJB) in the high-dimensional case $n \gg 1$, as the computational complexity of traditional numerical schemes often scales exponentially with the dimension of the state space, a phenomenon known as the *curse of dimensionality*. We circumvent this limitation by considering a causality-free approximation of the value function, as proposed in [2]. For a given initial condition, Pontryagin's Maximum Principle (PMP) yields first-order optimality conditions for (1)-(2) in the form of a two-point boundary value problem for a forward-backward coupling between optimal state, adjoint $p^* = (p_1^*, \dots, p_n^*)$, and control variables, denoted by $(y^*(t), p^*(t), u^*(t))$ respectively, which in short reads

$$\text{(TPBVP)} \qquad \begin{cases} \dfrac{d}{dt} y^*(t) = f(y^*(t)) + g(y^*(t))u^*(t)\,, \\ y^*(t_0) = x\,, \\ -\dfrac{d}{dt} p^*(t) = \partial_y (f(y^*(t)) + g(y^*(t))u^*(t))^\top p(t) + \partial_y \ell(y^*(t))\,, \\ p^*(T) = 0\,, \end{cases}$$

closed with the optimality condition

$$u^*(t) = -\frac{1}{2\beta} g^\top(y^*(t)) p^*(t)\,, \qquad \forall t \in (t_0, T)\,. \tag{5}$$

This procedure yields an optimal state-adjoint-control triple originating from the initial condition x. We interpret system (TPBVP) as a representation formula for the solution of (HJB) . In the simplest version of this relation, assuming the solution of the HJB PDE is $\mathcal{C}^2$, it can be shown that the forward-backward dynamics originating from the PMP correspond to the characteristic curves of the HJB equation. This result was further improved in [3], where the PMP was derived from the viscosity solution of a first-order HJB PDE. The precise result linking the solution of the PMP with the characteristic curves of the HJB PDE, and the identification of the adjoint variable as the gradient of the value function can be found in [4, Theorems II.9 and II.10]. More concretely, the computation of a given $V(t^i, x^i)$ can be realized by solving (TPBVP) setting $t_0 = t_i$ and the initial condition $y(t_0) = x^i$, and evaluating the optimal cost (1) using the optimal triple $(y^*(t), p^*(t), u^*(t))$. Moreover, the optimal adjoint verifies $p^*(t) = \nabla V(t, y^*(t))$.

In [5], we restrict our attention to a class of smooth and unconstrained nonlinear optimal control problems where the aforedescribed link between PMP and the HJB PDE is direct, and we use it to generate a characteristic-based, causality-free method to approximate $V(t,x)$ and as a by-product $u(t,x)$, without solving (HJB). To do this, we sample a set of initial conditions $\{(t^i, x^i)\}_{i=1}^N$, for

which we compute both $V(t^i, x^i)$ and $\nabla V(t^i, x^i)$ by realizing the optimal trajectory through PMP. This is done by following a reduced gradient approach, in which forward-backward iterative solves of (TPBVP) are combined with a gradient descent method to find the minimizer of $J(u; t_0, x)$. Having collected a dataset $\{t^i, x^i, V(t^i, x^i), \nabla V(t^i, x^i)\}_{i=1}^N$ enriched with gradient information, we fit a polynomial model for the value function

$$V_\theta(t, x) = \sum_{i=1}^{q} \theta_i \Phi_i(t, x) = \langle \theta, \Phi \rangle \,, \tag{6}$$

with $\Phi(t, x) = (\Phi_1(t, x), \ldots, \Phi_q(t, x))$ are elements of a suitable polynomial basis, and the parameters $\theta = (\theta_1, \ldots, \theta_q)$ obtained from a LASSO regression

$$\min_{\theta \in \mathbb{R}^q} \|[\Phi; \nabla\Phi]\,\theta - [V; \nabla V]\|_2^2 + \lambda \|\theta\|_{1,\mathbf{w}} \,, \tag{7}$$

where the matrix $[\Phi; \nabla\Phi] \in \mathbb{R}^{(n+1)N \times q}$ and the vector $[V; \nabla V] \in \mathbb{R}^{(n+1)N}$ include value function and gradient data. The optimal feedback map is recovered as

$$u^*(t, x) = \operatorname*{argmin}_u \; \{\beta \|u\|_2^2 + \nabla V_\theta(t, x)^\top (g(x)u)\} \,. \tag{8}$$

In [6] we extend this framework by considering the infinite horizon case $T \to \infty$, where the use of PMP is not a suitable computational alternative to generate a synthetic dataset. However, assuming dynamics can be expressed in semilinear form

$$\frac{d}{dt} x(t) = A(x(t))x(t) + B(x(t))u(t) \,, \tag{9}$$

and $l(x) = x^\top Q x$ with $Q \succ 0$, a stabilizing feedback operator can be computed as

$$u(x) = -K(x)x = -\frac{1}{\beta} B^\top(x) \Pi(x) x \,. \tag{10}$$

where $\Pi \in \mathbb{R}^{n \times n}$ solves the State-dependent Riccati Equation (SDRE) [7]

$$A^\top(x)\Pi(x) + \Pi(x)A(x) - \frac{1}{\beta}\Pi(x)B(x)B(x)^\top \Pi(x) + Q = 0 \,. \tag{11}$$

The implementation of an SDRE-based controller requires the solution of algebraic Riccati equations at a very fast rate along the trajectory $x(t)$. This computational constraint becomes prohibitive for large-scale dynamical systems. We investigate the training of a suitable artificial neural network to replace this task based on the following alternatives:

- Learning $u(x)$. We train a model for the vector-valued feedback law $u(x) : \mathbb{R}^n \to \mathbb{R}^m$ upon a set of N_s training states $\{x^{(i)}\}_{i=1}^{Ns}$, the solution of the corresponding $\Pi(x)$, and the controls $u(x)$ via (10).
- Learning $V(x)$. We train a model for the scalar function $V(x) : \mathbb{R}^n \to \mathbb{R}$ from $V(x) = x^\top \Pi(x) x$ and its gradient $\nabla V(x) = 2\Pi(x)$, where $\Pi(x)$ is a positive definite solution of (11) for each x in the training set. The feedback law is then expressed as $u(x) = -\frac{1}{2\beta} B(x)^\top \nabla V(x)$.

Acknowledgments. Dante Kalise was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grants EP/V04771X/1, EP/T024429/1, and EP/V025899/1.

References

[1] H. Frankowska. Value function in optimal control. In *Mathematical control theory, Part 1, 2 (Trieste, 2001)*, ICTP Lect. Notes, VIII, pages 516–653. Abdus Salam Int. Cent. Theoret. Phys., Trieste, 2002.

[2] W. Kang and L. C. Wilcox. Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Comput. Optim. Appl.*, 68(2):289–315, 2017.

[3] E. N. Barron and R. Jensen. The Pontryagin maximum principle from dynamic programming and viscosity solutions to first-order partial differential equations. *Trans. Amer. Math. Soc.*, 298(2):635–641, 1986.

[4] N. N. Subbotina. The method of characteristics for the Hamilton-Jacobi equations and its applications in dynamic optimization. *Sovrem. Mat. Prilozh.*, (20, Differ. Uravn.):3–132, 2004. ISSN 1512-1712. English translation appeared in Journal of Mathematical Sciences (New York) [135 (2006), no. 3].

[5] B. Azmi, D. Kalise and K. Kunisch. Optimal Feedback Law Recovery by Gradient-Augmented Sparse Polynomial Regression. *J. Mach. Learn. Res.*, 22: 1-32, 2021.

[6] G. Albi, S. Bicego and D. Kalise. Gradient-augmented Supervised Learning of Optimal Feedback Laws Using State-dependent Riccati Equations. *arXiv:2103.04091* , 2021.

[7] H. T. Banks, B. M. Lewis and H. T. Tran. Nonlinear feedback controllers and compensators: a state-dependent riccati equation approach. *Comput. Optim. Appl.*, 37(2):177–218, 2007.

Compositional features and approximation theory for deep learning

Wei Kang

(joint work with Qi Gong)

It has been widely observed in science and engineering that complicated and high dimensional information input-output relations in real-world applications can be represented as compositions of simple and low dimensional functions. Using power system as an illustrative example, the electric air-gap torque, $\mathbf{P}_e$ in a model of power systems, is a vector valued function that can be approximated by [1, 10]

(1)
$$(\mathbf{P}_e)_i = E_i^2 G_{ii} + \sum_{j=1, j\neq i}^{m} E_i E_j (G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)), \quad 1 \leq i \leq m,$$

where m is the number of generators, δ_i, $i = 1, 2, \cdots, m$, are rotor angles of the generators. Other variables in (1) represent constant parameters. As a function, its input dimension is determined by the number of generators, m, ranging from tens to thousands for networked power systems of various sizes. Nevertheless, this function can be represented as compositions of functions that have low input dimensions, $d \leq 2$, (Figure 1). In this example, we use a layered directed acyclic graph (layered DAG) to represent the compositional function. In the DAG, each node is a function. Several reasons motivate us to study compositional structures and their layered DAGs as a fundamental concept for deep learning.

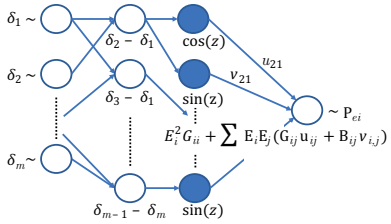

FIGURE 1. Layered DAG of $(\mathbf{P}_e)_i$ defined in (1)

- A widely used idea of representing functions in mathematics is to project a function to a sequence of finite dimensional subspaces. For instance, functions can be represented using Fourier series consisting of $\sin(2n\pi x)$ and $\cos(2n\pi x)$ as basis functions. This representation is closed under linear combination because a linear combination of Fourier series is still in the form of a Fourier series. However, the Fourier series representation is not closed under function composition because the composition of two Fourier series is not in the form of a Fourier series. On the other hand, the family of compositional functions represented by layered DAGs is closed under function composition. In fact, it is proved in [5] that the family of compositional functions is closed under several algebraic operations including linear combination, composition and substitution.
- Obviously, deep neural networks are compositional functions in which each node is an activation function. Proved in [5], some compositional features determine an upper bound of deep neural network complexity for applications such as regression and dynamical system approximation; and the neural networks may share a similar compositional structure with the input-output relation to be approximated.
- Iterative algorithms are computational processes in which a function is repeatedly applied to the result from the previous step. The iterative process is equivalent to a finite sequence of function compositions. Therefore, any function that can be approximated using iterative algorithms, such as the trajectories of differential equations or discrete dynamical systems, can be treated as compositional functions. An approximation theory developed for compositional functions is applicable to a wide spectrum of problems.
- Compositional structure is an inherent property of continuous functions with any input dimension. Kolmogorov's Representation Theorem [4] reveals that any continuous function defined on a d-dimensional cube can be exactly represented by a composition of a set of continuous 1-dimensional functions as

$$f(x_1, \cdots, x_d) = \sum_{q=1}^{2d+1} \phi_q \left(\sum_{p=1}^{d} \psi_{pq}(x_p) \right),$$

where ϕ_q and ψ_{pq} are continuous univariate functions. This implies that every continuous function is a compositional function in which each node has a single input (excluding linear nodes).

Existing approximation theory for neural networks cannot explain why deep learning can solve high dimensional problems. Existing theory typically guarantees a L^2 error upper bound $O(n^{-1/2})$, where $n > 0$ is the number of neurons, i.e., the complexity of the neural network [2, 3] . Although the rate has a constant exponent that is independent of the problem's dimension, a constant C in $O(n^{-1/2})$ can depend on d exponentially. This challenge is called the curse-of-dimensionality, i.e. the error upper bound increases exponentially with the dimension of the problem. For some control system applications, L^∞-norm is preferred over L^2-norm to achieve guaranteed performance in the worst case.

In [5, 6, 7, 9], some fundamental connections between the compositional features of functions and their neural network approximations are revealed. A unified theoretical framework of approximation theory for deep neural networks is developed that is applicable not only to functions as input-output relations, but also to dynamical systems and optimal control. For any continuous function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^q$ associated with a layered DAG, the graph is denoted by $\mathcal{G}^{\mathbf{f}}$. Each node in the DAG represents a function that has relatively low input dimension. Denote the jth node in the ith layer by $f_{i,j}$. The mapping that maps a node to the layer number of the node is denoted by $\mathcal{L}^{\mathbf{f}}$. It is proved in [5] that the following four composition features of $\mathbf{f}$ are critical to the approximation error of deep neural networks.

(2)
$$\begin{cases} \text{Dimension feature } r^{\mathbf{f}}_{max}: & \text{upper bound of } d_{i,j}/m_{i,j}, \\ \text{Volume feature } \Lambda^{\mathbf{f}}: & \text{upper bound of } \{(R_{i,j})^{m_{i,j}}, 1\} \left\|f_{i,j}\right\|_{W^\infty_{m_{i,j},d_{i,j}}}, \\ \text{Lipschitz feature } L^{\mathbf{f}}_{max}: & \text{upper bound of } L_{i,j}, \\ \text{Complexity feature } \mathcal{V}^{\mathbf{f}}_G: & \text{the number of nodes in} \end{cases}$$

where $d_{i,j}$ is the input dimension of $f_{i,j}$ (the jth node in the ith layer of $\mathcal{G}^{\mathbf{f}}$), $m_{i,j}$ is the smoothness of the node, $R_{i,j}$ is the side length of the domain of $f_{i,j}$ (for the simplicity of discussion, we assume the domain of $f_{i,j}$ is a d-cube), $L_{i,j}$ is a Lipschitz constant associated with the node (for its definition, interested readers are referred to [5]).

Theorem 1. ([5]) *Consider a compositional function triplet* $(\mathbf{f}, \mathcal{G}^{\mathbf{f}}, \mathcal{L}^{\mathbf{f}})$ *with* $\mathbf{f} : [-R, R]^d \to \mathbb{R}^q$. *Assume* $f_{i,j} \in W^\infty_{m_{i,j},d_{i,j}}$ *where* $m_{i,j} > 1$. *For any integer* $n_{width} > 0$, *there always exists a deep neural network,* $\mathbf{f}^{NN}$, *that has the following error upper bound,*

(3)
$$\left\|\mathbf{f}(\mathbf{x}) - \mathbf{f}^{NN}(\mathbf{x})\right\|_p \le C_1 L^{\mathbf{f}}_{max} \Lambda^{\mathbf{f}} \left|\mathcal{V}^{\mathbf{f}}_G\right| (n_{width})^{-1/r^{\mathbf{f}}_{max}}, \quad \textit{for all } \mathbf{x} \in [-R, R]^d,$$

where C_1 *is a constant determined by* $\{d_{i,j}, m_{i,j}; \quad f_{i,j} \in \mathcal{V}^{\mathbf{f}}_G\}$. *The complexity (total number of neurons) of* f^{NN} *is* $\left|\mathcal{V}^{\mathbf{f}}_G\right| n_{width}$.

In (3), the only term that depends on the input dimension is the exponential term $-1/r^{\mathbf{f}}_{max}$. The definition of the feature, $r^{\mathbf{f}}_{max}$, in (2) depends on the input dimension of the individual nodes of $\mathbf{f}$, not the input dimension of the overall function. For instance, in the example (1), the input dimension of $\mathbf{P}_e$ depends on the number of generators in the power system, which ranges from tens to thousands. However, $r^{\mathbf{f}}_{max}$ is determined by the largest input dimension of the nodes (assuming a constant smoothness $m_{i,j}$), which is a constant $d_{i,j} = 2$ no matter what the size of the power system is. In general, if the input dimension of individual nodes in a compositional function is bounded, the error upper bound of deep neural network approximation is a polynomial function of the compositional features, $\Lambda^{\mathbf{f}}$, $L^{\mathbf{f}}_{max}$, and $\mathcal{V}^{\mathbf{f}}_G$. For families of functions in which the compositional features do not depend on d exponentially (A function family would be ill conditioned if a feature, for instance the Lipschitz constant, increases exponentially with d), then neural network approximations do not suffer from the curse-of-dimensionality.

In [5], the algebraic frame of compositional functions is applicable to iterative computational algorithms because iterative algorithms form a special family of compositional functions. The trajectories of ordinary differential equations can be approximated by iterative algorithms, such as the Euler method. Therefore, Theorem 1 is directly applicable to differential equations.

Theorem 2. *Consider the following ODE*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{f}(\mathbf{x}) \in \mathbb{R}^d, t \in [0, T]. \tag{4}$$

in which $\mathbf{f}$ *is a compositional function with a layered DAG* $\mathcal{G}^{\mathbf{f}}$ *in which all nodes satisfy the same assumption as in Theorem 1. Let* $\boldsymbol{\phi}(t; \mathbf{x})$ *represent the solution at* t *with initial state* $\mathbf{x}$*. Then for any integer* $n_{width} > 0$*, there exists a deep neural network,* $\boldsymbol{\phi}^{NN} : \mathbb{R}^d \to \mathbb{R}^d$*, satisfying*

$$\left\| \boldsymbol{\phi}(T; \mathbf{x}) - \boldsymbol{\phi}^{NN}(\mathbf{x}) \right\|_p \leq C(n_{width})^{-1/r^{\mathbf{f}}_{max}}, \quad \mathbf{x} \in D^{\mathbf{f}}. \tag{5}$$

where C *is a polynomial function of the compositional features,* $\Lambda^{\mathbf{f}}$*,* $L^{\mathbf{f}}_{max}$*, and* $\mathcal{V}^{\mathbf{f}}_G$*.*

In [5], the same approximation theory is also applied to optimal control to prove that the feedback law of optimal control can be approximated by deep neural networks with an error boud that depends on the compositional features as a polynomial function. Inspired by the results in [5], we believe that the study of compositional features is essential to the fundamental question of why deep learning is able to solve high dimensional problems. For future research, we will study the role of compositional features in some fundamental and long term open problems in deep learning, such as the error bounds in the regression of high dimensional nonsmooth functions, solving PDEs without the curse-of-dimensionality, and finding efficient and effective data size and distribution for both training and validation.

References

[1] P. M. Anderson and A. A. Fouad. Power Systems Control and Stability, IEEE Press, 1994.
[2] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. on Information Theory, **39** (1993), 930–945.
[3] W. E, C. Ma, S. Wojtowytsch and L. Wu, *Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't*, arXiv:2009.10713v3, 2020.
[4] A. N. Kolmogorov, *On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition*, In Doklady Akademii Nauk, Russian Academy of Sciences **114:4** (1957), 953-956.
[5] W. Kang and Q. Gong, *Neural network approximation of compositional functions with applications to dynamical systems*, arXiv:2012.01698v1, 3 Dec 2020.
[6] H. N. Mhaskar, *Neural networks for optimal approximation of smooth and analytic functions*, Neural Computation, **8:1** (1996) 164-177.
[7] H. N. Mhaskar and T. Poggio, *Deep vs. shallow networks: an approximation theory perspective*, arXiv: 1608.03287v1, 2016.
[8] T. Nakamura-Zimmerer, Q. Gong, W. Kang, *Adaptive deep learning for high-dimensional Hamilton-Jacobi-Bellman equations*, SIAM J. Scientific Computing **43:2** (2021), 1221-1247.
[9] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, *Why and when can deep - but not shallow - networks avoid the curse of dimensionality: a review*, arXiv:1611.00740v5, 2017.
[10] J. Qi, K. Sun, and W. Kang, *Optimal PMU placement for power system dynamic state estimation by using empirical observability Gramian*, IEEE Trans. Power Systems **30:4** (2015), 2041-2054.

Persistent dynamical structures and their manipulation

Péter Koltai
(joint work with various coauthors)

The identification of persistent forecastable structures in complicated dynamics is vital for a robust prediction (or manipulation) of such systems in a potentially sparse-data setting. We characterise and find such structures in non-stationary fluid-dynamical applications. Then, we show how these objects relate to so-called collective variables known from statistical physics. We have recently developed a data-driven technique to find collective variables in molecular systems. These generalize to other applications as well.

1. Coherent sets

Let a time-dependent velocity field v on $\mathbb{R}^d$ be given. We call a time-parametrized family of sets, $(A_t)_{t\geq 0}$, *coherent*, if it contains the flow associated with v in a manner that is robust to noise. More precisely, for some small $\varepsilon > 0$ consider the process x_t governed by the SDE $dx_t = v(t, x_t)dt + \varepsilon\, dw_t$, and the (upper) escape rate from $(A_t)_t$,

$$E((A_t)_t) = -\limsup_{t\to\infty} \frac{1}{t} \log \mathbb{P}\big(x_r \in A_r,\, 0 \leq r \leq t\big),$$

where $\mathbb{P}$ refers to the law of the standard Wiener process w_t, and x_0 is (say) uniformly distributed over the bounded set A_0. The family is said to be coherent if the associated escape rate is small.

Our question is whether we have a characterization of coherent families $(A_t)_t$ that allows for their numerical computation.

Let us consider the periodically-forced case, i.e., where $t \mapsto v(t, \cdot)$ is τ-periodic with $\tau > 0$. We consider the equivalent augmented form of the above SDE,

$$\begin{aligned} \dot{\theta}_t &= 1 \\ dx_t &= v(\theta_t, x_t)dt + \varepsilon\, dw_t \end{aligned}$$

which is a time-homogeneous process on $\tau S^1 \times \mathbb{R}^d$. As such, it has a generator $\mathcal{G}$ acting on twice differentiable L^1 functions over $\tau S^1 \times \mathbb{R}^d$. Eigenfunctions of $\mathcal{G}$ can be used to find coherent families:

Theorem 1. [4] Let $\mathcal{G}f = \kappa f$, $\kappa \neq 0$. Then, $\int f(\theta, \cdot) = 0\ \forall\theta$ and for the family of sets

$$A_\theta := \{f(\theta, \cdot)\ \geq 0\}$$

it holds true that

$$E((A_\theta)_\theta) \leq -\mathfrak{Re}(\kappa).$$

Thus, if we find eigenfunctions f of $\mathcal{G}$ for eigenvalues close to zero, the family of sets extracted from the eigenfunction has a small escape rate. Various grid-based discretization techniques are readily available to solve the eigenproblem, we refer to [4, 5]. An unsolved problem is whether there exist methods that allow an efficient solution in higher dimensions.

Theorem 1 can be extended to the case where the system is not periodically-forced and only considered on a finite time interval, and in this case escape rate is replaced by an "escape ratio". A similar statement holds true, where the generator (in particular, the augmented process) needs to be adapted [5].

This framework is especially advantageous if we consider the problem of mixing manipulation. There, the velocity field v is given, together with a set of possible additive velocity perturbations δv. We would like to enhance (or suppress) the advection-dominated (but slightly diffusive) mixing in the system, as measured by the eigenvalues κ of the generator $\mathcal{G}$. Different eigenmodes allow us to target different structures. To target the overall (worst-case) mixing, one should target the second eigenvalue. It turns out, that the optimization problem

$$\min / \max\ \kappa(\mathcal{G}(v + \delta v)) \quad \text{s.t. } \|\delta v\|_{L^2} \leq C$$

is very well approachable, mostly due to the linear dependence of the generator $\mathcal{G}$ on the velocity field v. Details are given in [5], see also [7]. Advantages of the linear dependence on the control were discussed in the talk of Sebastian Peitz.

Coherent sets are one particular example of robust dynamical structures, making them well-forecastable objects, central to complex fluid flows. In higher-dimensional complex systems a structural reduction of a similar sort becomes highly important if one is interested in a robust forecast—or simple manipulation—of the system at hand. This brings us to our next topic.

2. Collective variables

Depending on the context, collective variables are known by the names "reaction coordinates" or "order parameters", and their definitions might slightly deviate. Our interpretation is that collective variables are sufficient to give a closed-form description of the considered system's dominant dynamical behavior.

To make this more precise, let us consider a time-discrete homogeneous process $(x_t)_t$ on $\mathbb{R}^N$, where usually $N \gg 1$. A (smooth) function $\xi : \mathbb{R}^N \to \mathbb{R}^r$, where usually $r \ll N$, is called a *collective variable*, if $\mathrm{Law}(\xi(x_{t+1}))$ can be to a large accuracy expressed by $\mathrm{Law}(\xi(x_t))$ in a functional relationship, without any explicit additional knowledge about the full state x_t. This idea generalizes in a straightforward way to time-continuous processes. A more rigorous characterization of this fact is given in [2], where it is required that the dominant spectrum of $\mathcal{G}$, the generator of the process $(x_t)_t$, is pointwise close to the dominant spectrum of $P_\xi \mathcal{G} P_\xi$, where P_ξ denotes the L^2-orthogonal projection from $L^2(\mathbb{R}^N)$ to $\{f \in L^2 \mid f = g \circ \xi \text{ for some } g : \mathbb{R}^r \to \mathbb{R}\}$.

How to find collective variables if only the high-dimensional description of the system is known? For this we consider the *transition density function* $p^\tau(\cdot,\cdot)$ of the system (assuming it exists), defined by

$$\mathrm{Law}(x_{t+\tau}) = \int p^\tau(x_t, \cdot) d\mathrm{Law}(x_t).$$

We showed:

Theorem 2. [2] Let the system be reversible. It the set $\{p^\tau(x,\cdot) \mid x \in \mathbb{R}^N\} \subset L^1$ is ε-close to a r-dimensional (smooth) manifold, then there is a (smooth) r-dimensional collective variable that reproduces dominant timescales (i.e., the dominant spectrum of $\mathcal{G}$) up to $\mathcal{O}(\varepsilon)$.

Theorem 2 gives a constructive statement in the sense that one can build a computational procedure based on this. One needs to approximate transition density functions at selected anchor points x, yielding a point cloud in function space. Then one needs to parametrize this point cloud with a small number of coordinates, e.g., by manifold learning techniques. For details we refer to [2].

Collective variables are strongly motivated by statistical physics, and especially molecular dynamics, where such collective variables can help immensely in understanding for instance protein folding processes. We have found, however, that the concept is informative in other applications as well.

This brings us back to fluid dynamics and – first – to coherent sets. It is natural to state the above coherence problem in a "Lagrangian" setting: What are the most coherent *material* families of sets $(A_t)_t$? Here, "material" refers to the fact that A_t should be the set A_0 evolved by the flow governed by $\dot{x}_t = v(t, x_t)$. This suggests to consider the stochastic process $dx_t = v(t, x_t)dt + \varepsilon\, dw_t$ in the co-evolving coordinate system evolved by the flow of $\dot{x}_t = v(t, x_t)$. That is in the limit of $\varepsilon \to 0$ a reversible process, and we can consider collective variables for it. This has been done in [1], and some results are shown in Fig. 1. This example shows

that trajectories that have similar collective variables will belong most likely to the same coherent set – they are having the same behavior from a mixing perspective.

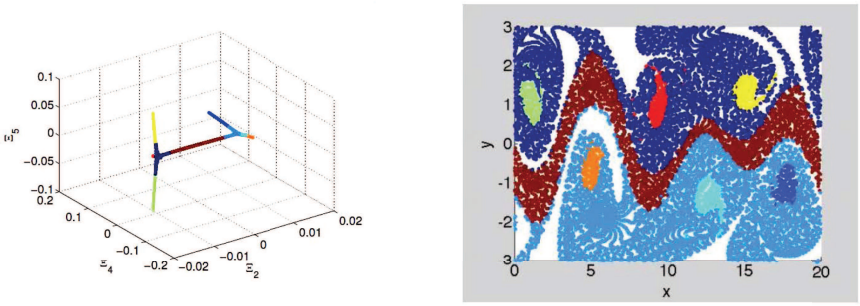

FIGURE 1. On the left we see three-dimensional collective variables for a Rossby-perturbed Bickley Jet flow (trajectories of which are shown on the right). The coloring is obtained by clustering points on the left. Each point on the left corresponds to an evolving trajectory on the right, where we impose the same coloring. Evolving these trajectories shows that trajectories of identical colors form coherent sets (the right-hand side shows the trajectories midway through their evolution).

Another application in fluid dynamics is given in [8], where we find two-dimensional collective variables in a Rayleigh–Bénard convection in a cylindrical container, observed through temperature measurements. This is, in its physical description, an infinite dimensional system ($N = \infty$) governed by the Navier–Stokes equations. Although it is not reversible, collective variables seem to exist in this case as well.

An example of collective variables in social dynamics can be found in [6]. Therein, the high-dimensional system describes the evolution of each individual's opinion in a fixed population, while collective variables are expected to be fractions of opinions in the whole populations (or some distinguished clusters of it).

3. Discussion

While the derivation of reduced models for complex high-dimensional systems has been considered in the respective fields for decades, a unified view – overarching the fields – on variables / observables that allow such a reduction is only emerging (cf. also Nathan Kutz's talk). Such low-dimensional representations are expected not just to help with the prediction of systems, but, naturally, allow us to leverage this structure for manipulating / controlling the process. The talks of Dante Kalise and Jiequn Han considered numerical techniques (such as tensor decomposition and deep learning) for control of high dimensional systems; and these could be speculated to work well because the problems at hand featured a low-dimensional

structure that the method could implicitly target. A more explicit manifestation of a low-dimensional structure in the Lyapunov function of high-dimensional stable equilibria appeared in Lars Grüne's talk.

References

[1] R. Banisch and P. Koltai. *Understanding the geometry of transport: diffusion maps for Lagrangian trajectory data unravel coherent sets.* **Chaos** 27(3): 035804, 2017.

[2] A. Bittracher, P. Koltai, S. Klus, R. Banisch, M. Dellnitz, and Ch. Schütte. *Transition manifolds of complex metastable systems: Theory and data-driven computation of effective dynamics.* **J. Nonlinear Sci.**, 28(2): 471–512, 2018.

[3] G. Froyland. *Dynamic isoperimetry and the geometry of Lagrangian coherent structures.* **Nonlinearity**, 28(10), 3587, 2015.

[4] G. Froyland and P. Koltai. *Estimating long-term behavior of periodically driven flows without trajectory integration.* **Nonlinearity** 30, pp. 1948–1986, 2017.

[5] G. Froyland, P. Koltai, and M. Stahn. *Computation and optimal perturbation of finite-time coherent sets for aperiodic flows without trajectory integration.* **SIAM J. Appl. Dyn. Sys.** 19(3), 1659–1700, 2020.

[6] L. Helfmann, J. Heitzig, P. Koltai, J. Kurths, and C. Schütte. *Statistical Analysis of Tipping Pathways in Agent-Based Models.* arXiv preprint arXiv:2103.02883, 2021.

[7] P. Koltai, H. C. Lie, and M. Plonka. *Fréchet differentiable drift dependence of Perron-Frobenius and Koopman operators for non-deterministic dynamics* **Nonlinearity**, 32, pp. 4232–4257, 2019.

[8] P. Koltai and S. Weiss. *Diffusion maps embedding and transition matrix analysis of the large scale flow structure in turbulent Rayleigh–Bénard convection.* **Nonlinearity** 33(4): 1723, 2020.

Targeted use of deep learning for physics-informed model discovery

J. Nathan Kutz

(joint work with Steven L. Brunton, Kathleen Champion, Bethany Lusch, Henning Lange)

Deriving governing parsimonious physical laws and constraints has until very recently been accomplished by using expert knowledge, asymptotic reductions and imposed constraints. These derivations have been uninformed by data except in a qualitative way. Modern data-driven discovery has been recently pioneered (Brunton, Proctor & Kutz, PNAS 2016) by simple sparse regression tools such as the *sparse identification of nonlinear dynamics* (SINDy). Although attempts have been made to use neural network (NN) architectures to learn physics, there remains critical issues concerning generalizability, interpretability, overfitting and significant data requirements, limiting their usefulness and computational tractability for on-line learning. Regardless of the method used, they are compromised in practice by limited data, corruption due to noise, unmeasured latent variables, parametric dependencies, and unaccounted for multi-scale physics. We will overcome these challenges by developing new priors (regularizers) based on physical knowledge, scalable filtering algorithms, and representation formats (coordinates) based on sparse, low-rank, and targeted neural network models. In addition to the SINDy

architecture, Kutz and co-workers have recently pioneered leading strategies for sensor placement, coordinate discovery and parametric reduced order models.

My goal is to specifically learn physically interpretable models of dynamical systems from off-line and/or on-line streaming data. Furthermore, we consider systems for which no baseline physical model or parametric form is known, although partial physics knowledge can easily be included. Thus we seek to perform system identification for dynamical systems from data $\mathbf{y} \in \mathbb{R}^m$ in high dimensional state spaces $\mathbf{x} \in \mathcal{X}_1 \subset \mathbb{R}^n$, where $n \gg 1$ and $m \ll n$. Specifically

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, t, \boldsymbol{\Theta}, \boldsymbol{\Omega}), \\ \mathbf{y}_k &= h(t_k, \mathbf{x}(t_k)) + \boldsymbol{\Xi} \end{aligned} \tag{1}$$

where the dynamics are prescribed by $f : \mathcal{X}_1 \to \mathbb{R}^n$ the observation operator is $h : \mathcal{X}_1 \to \mathbb{R}^m$, the frequency of observations are given by $\mathbf{y}_k$ which are measured at the times t_k. Observations are compromised by measurement noise $\boldsymbol{\Xi}$, which is typically described by some probability distribution (e.g. a normal distribution $\boldsymbol{\Xi} \sim N(\mu, \sigma)$). The dynamics is prescribed by a set of parameters $\boldsymbol{\Theta}$. Moreover, the dynamics may be subject to stochastic effects characterized by $\boldsymbol{\Omega}$.

My goal: Given p measurements $\mathbf{y}_k$ arranged in the matrix $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2\ \cdots\ \mathbf{y}_p] \in \mathbb{R}^{m \times p}$, infer the dynamics $f(\cdot)$ with parametrization Θ, the measurement operator $h(\cdot)$, or a proxy model of the true system, so that tasks such as control and forecasting can be accomplished. Adding to the difficulty of the task are multiscale and multiphysics problems. In this case, the governing dynamics (1) for a two time scale (fast and slow, although generally there are more scales) should be modified to

$$\frac{d\mathbf{x}_1}{dt} = f_1(\mathbf{x}_1, \mathbf{x}_2, t, \tau, \boldsymbol{\Theta}_1, \boldsymbol{\Omega}_1), \qquad \frac{d\mathbf{x}_2}{d\tau} = f_2(\mathbf{x}_1, \mathbf{x}_2, t, \tau, \boldsymbol{\Theta}_2, \boldsymbol{\Omega}_2) \tag{2}$$

where $\tau = \epsilon t$ ($\epsilon \ll 1$) is a slow scale. If $h(\cdot)$ is not the identity and/or ξ_k is not zero, then we are in the case of *imperfect data.* This problem can be one of *online* learning where the update must occur in real-time and with no possibility of repeating the experiment. This is an ill-posed problem whose solution must be accomplished through judiciously chosen regularization and, in the case of (2), through first decomposing the data into its constitutive fast (t) and slow (τ) timescales.

Solving the ill-posed problem (1) is a fundamental scientific and mathematical challenge. To date, it has only been accomplished in highly specialized settings with typically full state measurements and clean (low-noise) data. Significant mathematical innovations will have to be developed in order to make this a general and robust architecture. The multiscale (2) is uncharted territory requiring the integration of a tremendously broad set of mathematical tools, something exhibited by the PI Kutz in tackling this grand scale and basic science challenge. The methodology can be broadly applied and the PI Kutz has extensive research work in DoD relevant application areas of directed energy, scientific computing, fluid dynamics and neuroscience where it can be test bedded.

Sensors. Everything starts with data acquisition. This is often completely overlooked in machine learning methods that assume that access to the correct variables are available. Thus learning the mapping $h(\cdot)$ from the measurements to the state space through (1b) is important to learn. For many complex systems, the latent variable space is an important aspect of the discovery process. Time-delay embeddings, and recourse to Taken's theorem, help establish a critical connection to dynamical systems theory and a potential reconstruction of the full state space. There are four critical aspects to developing a robust sensing framework: (i) sensor placement, (ii) sensor cost, (iii) discovery of the measurement map $h(\cdot)$, and (iv) multi-model data integration from diverse sensor types (e.g. video, audio, thermal, etc). These tasks must all be integrated at the front end of the learning process.

Kutz and co-workers have developed some of the earliest rigorous mathematical results on formulating optimal sensor placement and minimal cost strategies for complex spatio-temporal systems. These techniques are based upon low-rank linear subspaces that can be used to great effect with greedy sampling techniques. New fundamental mathematical innovations are required for generating near-optimal greedy selection procedures for nonlinear manifold embeddings typical of real data. Neural networks can be used for decoder networks capable of producing a highly improved mapping between the data to the underlying state space. It would be also necessary to use the time-delay embedding structure to try and reconstruct, as best as possible, the latent variables and reframe the greedy algorithms based upon the time-delay data. To date, it is unknown what the limits and mathematical possibilities are for using such a method to extract the full state variable $\mathbf{x}$ from measurements $\mathbf{y}_k$. In addition to extracting critical information on the state space, our recent neural network architectures are capable of denoising data sets in a manner that is comparable, and in many cases better, than Kalman filtering methods. This would be a pre-processing step for data streams where the underlying model is known, unknown or partially known. Potentially helping improve these results are multi-modal data fusion techniques which can be potentially used to help improve decision making or predictions. Sensors are critical for determining $h(\cdot)$.

Coordinate Discovery. Data processed from the multi-modal sensors are then used to discover a transformation $\mathbf{z} = g(\mathbf{x})$ where a parsimonious, low-dimensional dynamics can be constructed

$$\dot{\mathbf{z}} = F(\mathbf{z}, t, \boldsymbol{\Theta}, \boldsymbol{\Omega}) \tag{3}$$

where $\mathbf{z} \in \mathcal{X}_1 \subset \mathbb{R}^r$ is an r-dimensional ($r \ll n$) model of the physics specified by $F(\cdot)$. Ultimately, the discovery of the nonlinear transform $g(\cdot)$, through training neural network autoencoders, gives the coordinates for parsimonious dynamics $F(\cdot)$ required for the reduced models for the intelligent agent. These are computationally tractable models of its own dynamics and environment.

If only limited data is available, then it may be required to produce a low-fidelity, on-line model using a linear map. This can be done with an r-rank POD mode truncation of the snapshots of $\mathbf{x}$. Dynamic mode decomposition, or a Koopman

approximation using augmented state-space measurements, can then be used on this low-rank subspace to produce the best-fit linear model through the data. This provides a baseline architecture for diagnostics and forecasting. As more data is required, a full nonlinear mapping and nonlinear model can be used to refine the results, both in terms of building a lower-rank nonlinear subspace and for producing a parsimonious nonlinear dynamics. As sufficient data is acquired from the sensors, the data-discovery pipeline then produces the flow:

$$\mathbf{y} \in \mathbb{R}^m(\text{measurements}) \rightarrow \mathbf{x} \in \mathbb{R}^n(\text{state space}) \rightarrow \mathbf{z} \in \mathbb{R}^r(\text{reduced order model})$$

with two mappings to discover: h and g. With limited data, SVD provides a linear approximation.

Learning Physics, Discrepancy Models, and Physics Constraints. The backbone of the proposal is the highly effective method for model discovery: the SINDy algorithm. It leverages simple sparse regression to discover nonlinear dynamical systems from data. The basic algorithmic structure of SINDy has been modified by Kutz and co-workers to discover parametrically-dependent systems, identify nonlinear control laws, infer biological networks, discover spatio-temporal systems, and identify nonlinear systems in the low-data limit. The SINDy algorithm posits a large set of potential candidate functions that comprise $F(\cdot)$ (or $f(\cdot)$ for the full state vector), then uses sparsity to determine the dominant terms. A library of candidate nonlinear functions $\boldsymbol{\Theta}(\mathbf{Z})$ is constructed from $\mathbf{Z}$. It is possible to relate the time derivatives in $\dot{\mathbf{Z}}$ to the candidate nonlinearities by $\dot{\mathbf{Z}} = \boldsymbol{\Theta}(\mathbf{Z})\boldsymbol{\Xi}$, where each column $\boldsymbol{\xi}_k$ in $\boldsymbol{\Xi}$ is a vector of coefficients that determines which terms are active (nonzero). Sparsity promoting algorithms are used to ensure that most of the entries of the column $\boldsymbol{\xi}_k$ are zero. By identifying the sparse coefficient vectors $\boldsymbol{\xi}_k$, a model of the nonlinear dynamics (3) is found. *The SINDy framework is the key regularization for solving the ill-posed problem (1).*

One of the most attractive aspects of SINDy: the underlying mathematics is simply a linear, overdetermined system of equations. The solution is achieved by regularizing with sparsity. Because of this simple architecture, additional constraints can be easily placed on the linear regression, including regularizations corresponding to conservation of mass, energy, etc. Moreover, discrepancy models can be easily accommodated since known physics terms can be easily incorporated into the regression structure. The flexibility of this backbone algorithm is remarkable and will serve as the core architecture for the intelligent agent in discovering nonlinear reduced order models for its environment. Alternatively, one can learn these constraints from the data itself. Examples of known physics that may be explicitly encoded include: non-negativity of physical variables (e.g., pressure, temperature, chemical concentration, etc.), known dynamics (e.g., growth rates, frequencies, etc.), conservation laws (e.g., mass, momentum, energy, chemical elements), and symmetries. Our perspective is driven by the need for parsimonious representations that are efficient, avoid overfitting, and provide minimal descriptions of the dynamics on interpretable intrinsic coordinates. Such models additionally offer the best hope for generalizability.

References

[1] S. L. Brunton, J. L. Proctor, and J. N. Kutz. *Discovering governing equations from data by sparse identification of nonlinear dynamical systems.* Proceedings of the national academy of sciences **113**.15 (2016) 3932-3937.

[2] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, 2019. *Data-driven discovery of coordinates and governing equations.* Proceedings of the National Academy of Sciences, **116**(45) (2019) 22445-22451.

[3] B. Lusch, J. N. Kutz, and S. L. Brunton. *Deep learning for universal linear embeddings of nonlinear dynamics.* Nature communications **9**.1 (2018): 1-10.

[4] H. Lange, S. L. Brunton, and J. N. Kutz. *From Fourier to Koopman: Spectral methods for long-term time series prediction.* Journal of Machine Learning Research **22**, no. 41 (2021): 1-38.

The QuaSiModO approach for universal transformations of data-driven models to control systems

Sebastian Peitz

(joint work with Katharina Bieker)

This work pursues the central task to efficiently solve *optimal control problems* for complex and thus expensive-to-evaluate dynamical systems. Mathematically speaking, we consider the following problem over the time horizon $p \cdot \Delta t$:

$$\text{(I)} \qquad \begin{aligned} &\min_{u \in U^p} J(y) = \min_{u \in U^p} \sum_{i=0}^{p-1} P(y_{i+1}) \\ &\text{s.t.} \quad y_{i+1} = \Phi(y_i, u_i), \qquad i = 0, 1, 2, \dots, \end{aligned}$$

where y_i and $u_i \in U$ are the system state and control at time instant $t_i = i\Delta t$, with U being the set of admissible controls, e.g., $U = [u^{\mathsf{min}}, u^{\mathsf{max}}]$. The objective function (for instance, the distance to some desired trajectory y^{ref}) is denoted by P, and Φ describes the flow of the underlying dynamical system (e.g., an ordinary or a partial differential equation) over the time increment Δt. The solution of (I) yields the optimal control u^* and corresponding state y^*. Feedback control can then be achieved via *Model Predictive Control (MPC)* [4], i.e., by solving (I) repeatedly over a short horizon and applying the first entry u_0^* to the real system which is running simultaneously.

A substantial challenge that we often face is the fact that the efficient prediction (and, by extension, control) of complex dynamical systems is hindered by the fact that the system dynamics are either very expensive to simulate or even unknown. Researchers have been investigating ways to accelerate the solution by using data for decades, the *Proper Orthogonal Decomposition (POD)* being an early and very prominent example [12]. More recently, the major advances in data science and machine learning have lead to a plethora of new possibilities, for instance artificial neural networks such as *Long Short-Term Memory (LSTM)* Networks [5] or *Reservoir Computers / Echo State Networks* [6], regression-based frameworks for the identification of nonlinear dynamics [3], or numerical approximations of the *Koopman operator* [11, 9, 7], which describes the linear dynamics of observable

functions. These methods facilitate the efficient simulation and prediction of high-dimensional spatio-temporal dynamics using measurement data, without requiring prior system knowledge. As a consequence of the success of data-driven prediction, many approaches for control have been presented over the past decades. However, a drawback is that the construction of surrogate models with inputs is often much more tedious and also problem-specific and data hungry [1, 2].

The approach we present here to solve (I) via surrogate models while avoiding the aforementioned issues is based on modifying the control problem instead of adjusting the surrogate modeling to the control setting. The resulting framework, which we call *QuaSiModO*, consists of the following steps (cf. also Figure 1):

(1) **Qua**ntization of the the admissible control U (for instance by replacing the interval $U = [u^{\min}, u^{\max}]$ by the bounds $V = \{u^{\min}, u^{\max}\}$);
(2) **Si**mulation of the autonomous systems (e.g., $\Phi_{u^{\min/\max}}(y) = \Phi(y, u^{\min/\max})$;
(3) **Mod**eling of the individual systems – using either the full state y or some observable $z = f(y)$ – via an arbitrary "off-the-shelf" surrogate modeling technique (POD, neural network, Koopman operator, etc.);
(4) **O**ptimization using the resulting set of autonomous surrogate models and relaxation techniques.

This interplay between continuous and integer control modeling as well as between the full system state and observed quantities (e.g., measurements) allows us to utilize the best of both worlds, namely

- integer controls for efficient data-driven modeling using arbitrary predictive models,
- continuous control inputs for real-time control, and
- existing error bounds for predictive models.

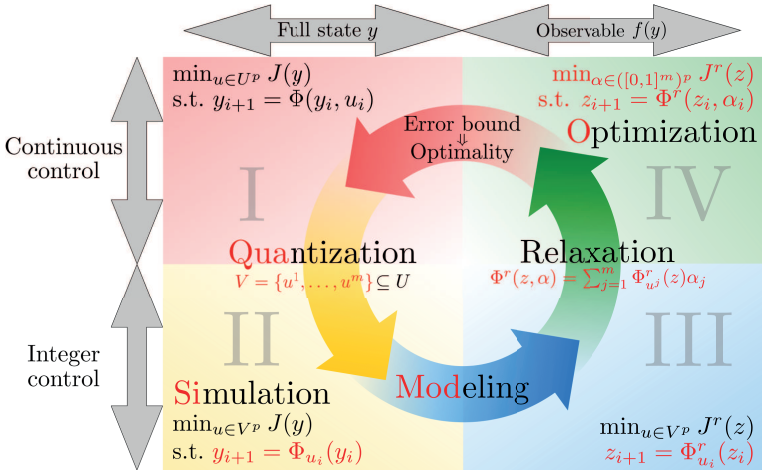

FIGURE 1. The QuaSiModO framework consisting of the four steps Quantization, Simulation, Modeling and Optimization [8].

QuaSiModO successively transforms Problem (I) into related control problems that – as long as the predictive surrogate model is sufficiently accurate – yield optimal trajectories y^* that are close to one another. From (I) to (II), we quantize the control, meaning that only a finite set $V \subseteq U$ of inputs is feasible. This allows us to replace the non-autonomous dynamical system $\Phi(y, u)$ by a finite set of autonomous systems $\Phi_{u^j}(y)$, each corresponding to one entry $u^j \in V$. While introducing an artificial drawback from the control perspective (Problem (II) is a mixed-integer optimal control problem), we can now easily introduce an equivalent Problem (III) that is based on surrogate models $\Phi^r_{u^i}(z)$ for a reduced quantity $z = f(y)$. Here, the function f is an *observable* which maps measurements from the state space of the full system to the space of measurements (which may be of significantly smaller dimension). As the transformation from (II) to (III) acts on a set of autonomous systems, we can approximate the individual systems Φ_{u^j} from individual measurement data sets, using whichever method we prefer.

In order to mitigate the disadvantages with respect to the complexity of the control problem, the problem of selecting an optimal input from V is relaxed by determining the optimal convex combination of the autonomous systems:

$$\text{(IV)} \qquad \begin{aligned} \min_{\alpha \in ([0,1]^m)^p} J^r(z) &= \min_{\alpha \in ([0,1]^m)^p} \sum_{i=0}^{p-1} P^r(z_{i+1}) \\ \text{s.t. } z_{i+1} = \Phi^r(z_i, \alpha_i) &= \sum_{j=1}^{m} \alpha_{i,j} \Phi^r_{u^j}(z_i) \quad \text{and} \quad \sum_{j=1}^{m} \alpha_{i,j} = 1. \end{aligned}$$

Problem (IV) is again continuous – with respect to the input α. For control affine systems, we can now determine $u^* = \sum_{j=1}^m \alpha_j^* u^j$ and directly apply it to the real system. For non-affine systems, we use the sum up rounding algorithm from [10], by which a control corresponding to one of the quantized inputs is applied to the real system.

Besides the ability to include arbitrary predictive models into the QuaSiModO framework, an important aspect is that existing error bounds for the chosen surrogate model can easily be included, see [8] for a detailed description The availability of error bounds is of particular importance for engineering systems, where safety is of utmost importance (e.g., for aircraft or autonomous vehicles). The bounds guarantee the performance of a controller and – more importantly – will automatically become stronger with future developments in the field of data-driven modeling.

We have tested the QuaSiModO framework on a variety of dynamical systems, observable functions and surrogate modeling techniques, cf. Figure 2, a detailed description is given in [8]. For instance, we can control the lift force acting on a cylinder (determined by the velocity and pressure fields governed by the 2D Navier–Stokes equations) without any knowledge of the flow field using the standard LSTM framework included in *TensorFlow*, and stabilize the Mackey-Glass equation using a standard echo state network. This highlights the flexibility and

broad applicability of the method and the success of the technique in constructing data-driven feedback controllers.

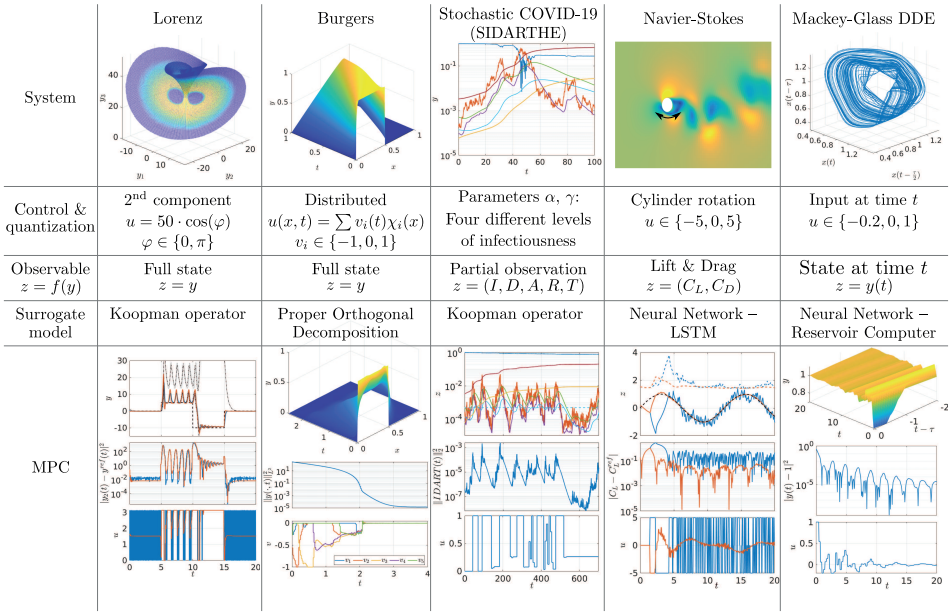

FIGURE 2. MPC using QuaSiModO applied to various combinations of systems and surrogate models [8].

REFERENCES

[1] M. Bergmann, L. Cordier, and J.-P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids*, 17:1–21, 2005.

[2] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, 34:577–591, 2020.

[3] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[4] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer International Publishing, 2 edition, 2017.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004.

[7] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020.

[8] S. Peitz and K. Bieker. On the Universal Transformation of Data-Driven Models to Control Systems. *arXiv:2021.04722*, 2021.

[9] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.
[10] S. Sager, H. G. Bock, and M. Diehl. The integer approximation error in mixed-integer optimal control. *Mathematical Programming*, 133(1-2):1–23, 2012.
[11] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
[12] L. Sirovich. Turbulence and the dynamics of coherent structures part I: coherent structures. *Quarterly of Applied Mathematics*, XLV(3):561–571, 1987.

Algorithms analysis via encapsulated nonlinearity Lyapunov based framework

Sophie Tarbouriech
(joint work with Milan Korda)

1. Introduction

In this talk we address the way to deal with some optimization algorithms via a dynamical system and control theory point of view, according to recent works as [Lessard et al., 2016], [Michalowsky et al., 2020]. Indeed, we want revisit the study of an algorithm behavior by noting that it can be viewed as a dynamical system with feedback, allowing us to use the tools of control theory [Taylor et al., 2018], [Michalowsky et al., 2020]. More especially, we consider fast and simple algorithms and therefore, algorithms based on first-order methods.

Hence, by considering the following unconstrained optimisation problem

$$\min_{z\in\mathbb{R}^n} f(z)$$

the main goal is to check that

$$\| z_k - z^\star \| \leq \delta\rho^k \| z_0 - z^\star \|$$

with z_0 the initial condition, $z^\star$ the optimal value, $\rho \in (0,1)$ the convergence rate, $\delta > 0$. Note that other objectives could be addressed as function error $f(z_k) - f(z^\star)$, $\| \nabla f(z_k) \|^2$, ... The general method of interest to solve such an optimization is defined as

$$z_{k+1} = z_k + \beta(z_k - z_{k-1}) - \alpha\nabla f(z_k + \gamma(z_k - z_{k-1})) \tag{1}$$

with ∇f the gradient of f and some positive tuning parameters α, β and γ. Let us cite the three main methods entering in this class of methods [Lessard, 2018], i.e., Gradient method, Heavy ball method and Nesterov accelerated method. See also, [Polyak, 1987], [Boyd and Vandenberghe, 2004], [Beck, 2014] for other methods and discussions.

2. Problem formulation

We focus on the Heavy ball (HB) method, which corresponds to (1) in which we choose $\gamma = 0$. This method can then be interpreted as a dynamical discrete-time system:

$$(2) \qquad \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k \\ u_k & = & \nabla f(y_k) \\ y_k & = & Cx_k \\ z_k & = & Dx_k \end{array}$$

with $x_k = \begin{bmatrix} z_k \\ z_{k-1} \end{bmatrix}; A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix}; B = \begin{bmatrix} -\alpha \\ 0 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \end{bmatrix} = D.$

Given Assumptions on f and ∇f, two complementary problems can be addressed:

- The analysis problem: given α and β, find the best convergence rate ρ.
- The design problem: given a desired $\rho \in (0,1)$ find α and β.

To address both problems above defined, we study the function $f(z)$:

$$f(z) = \begin{cases} u_0 k_g z & \text{if} \quad k_g z > u_0 \\ \frac{k_g^2 z^2}{2} & \text{if} \quad |k_g z| \leq u_0 \\ -u_0 k_g z & \text{if} \quad k_g z < -u_0 \end{cases}$$

with the minimizer $z^\star = 0$ ($z \in \mathbb{R}^n$, $n = 1$). That corresponds to the following gradient, which is a saturation:

$$\nabla f(z) = \begin{cases} u_0 k_g & \text{if} \quad k_g z > u_0 \\ k_g^2 z & \text{if} \quad |k_g z| \leq u_0 \\ -u_0 k_g & \text{if} \quad k z_g < -u_0 \end{cases} = k_g sat_{u_0}(k_g z)$$

with $u_0 > 0$ the level of saturation. From (2) that leads to study the following *closed-loop system*:

$$(3) \qquad \begin{array}{rcl} x_{k+1} & = & A_0 x_k + B_0 \phi_k \\ \phi_k & = & sat_{u_0}(Kx_k) - Kx_k \end{array}$$

with $A_0 = A + BK = \begin{bmatrix} 1+\beta-\alpha k_g^2 & -\beta \\ 1 & 0 \end{bmatrix}; B_0 = k_g B = \begin{bmatrix} -\alpha k_g \\ 0 \end{bmatrix}; K = \begin{bmatrix} k_g & 0 \end{bmatrix}$. where now the input of the system is a dead-zone-like nonlinearity.

3. Main technique

By leveraging on the framework associated to the class of systems (3), we can propose some theoretical conditions in order to solve both problems above stated (see, for example, [Tarbouriech et al., 2011]).

3.1. **Analysis problem.** Hence, to address the analysis problem, we fix a pair α, β such that matrix A_0 is Schur-Cohn. It is also important to note that that 1) the eigenvalues of A are: $1, \beta$, and 2) matrix A_0 is not stable for any pair of (α, β), as depicted on Fig. 1.

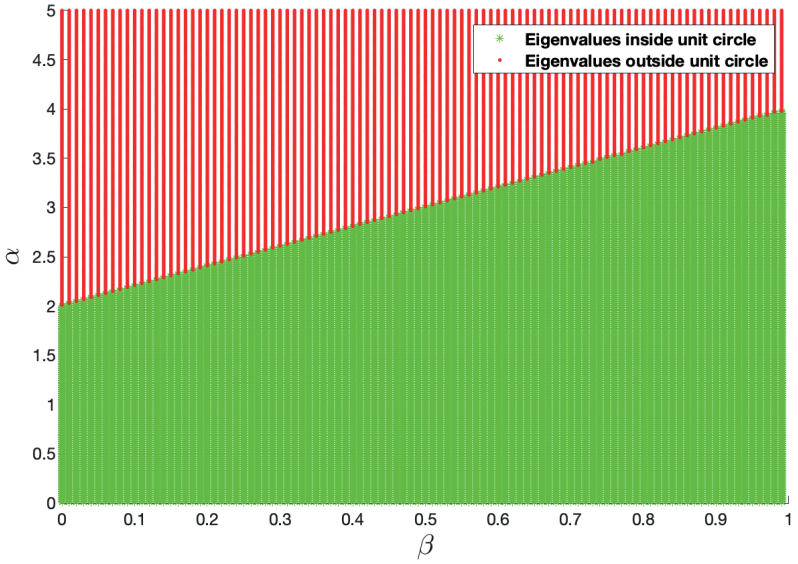

Figure 1. Illustration of area of stability (in green) or instability (in red) for different values of (α, β)

First, we are trying to ensure the global asymptotic (or exponential) stability of the closed loop, that is for any initial condition z_0, and therefore search for

- a Lyapunov function $V(x_k) > 0$, $V(0) = 0$,
- some properties to embed the nonlinearity ϕ_k as $h_g(x_k, \phi_k) \geq 0$ for $x_k \in \mathbb{R}^2$

such that along the trajectories of system (3):

$$V(x_{k+1}) - \rho V(x_k) + \tau_g h_g(x_k, \phi_k) < 0, \forall x_k \in \mathbb{R}^2$$

with $\rho > 0$, $\tau_g > 0$. Thanks to theoretical conditions expressed as linear (or quasi-) matrix inequalities (LMIs), which reveal to be unfeasible, and simulations confirming this, we can show that the system (3) is not globally asymptotically stable with the decay rate $\sqrt{\rho}$, or equivalently the HB algorithm is not globally convergent with the convergence rate $\sqrt{\rho}$.

Nevertheless, we can modify the way to encapsulate the dead-zone nonlinearity to provide theoretical conditions ensuring that the system (3) is locally asymptotically stable with the decay rate $\sqrt{\rho}$, or equivalently the HB algorithm is locally convergent with the convergence rate $\sqrt{\rho}$. At this aim, we search for

- a Lyapunov function $V(x_k) > 0$, $V(0) = 0$,
- some properties to embed the nonlinearity ϕ_k as $h_l(x_k, \phi_k) \geq 0$ for $x_k \in \Omega \subset \mathbb{R}^2$

such that along the trajectories of system (3):

$$V(x_{k+1}) - \rho V(x_k) + \tau_l h_l(x_k, \phi_k) < 0,$$
$$\forall x_k \in \mathcal{E}(V, 1) = \{x_k \in \mathbb{R}^2; V(x_k) \leq 1\} \subseteq \Omega$$

with $\rho > 0$, $\tau_g > 0$. The region of admissible initial conditions, for which the stability or the convergence is guaranteed, is then characterized as a level set of the Lyapunov function $V(x_k)$. Such a region is illustrated on Fig.2, where two trajectories are depicted: $x_0 \in \mathcal{E}(P)$ and the convergence to zero is ensured (in red) and $x_0 \notin \mathcal{E}(P)$ and the trajectories converge to a limit cycle (in blue). The

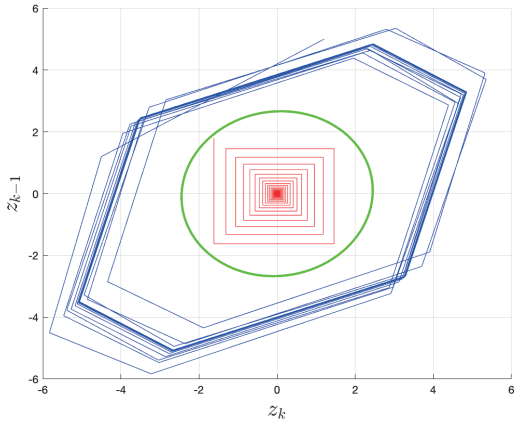

FIGURE 2. Illustration of the behavior for 2 initial conditions with $u_0 = 2$, $k_g = 1$ and $(\alpha, \beta) = (1.9, 0.9)$ in or outside the level set (depicted in green)

classical trade-off between the performance (ρ) and the size of the level set $\mathcal{E}(P)$ of guaranteed initial conditions is also illustrated.

3.2. **Design problem.** To address the complementary problem of designing the parameters α and β, we rewrite matrices A_0 and B_0 in which in this step α and β

are the decision variables:

$$
\begin{array}{rcl}
A_0 & = & \left[\begin{array}{cc} 1+\beta-\alpha k_g^2 & -\beta \\ 1 & 0 \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \end{array}\right] + \left[\begin{array}{c} 1 \\ 0 \end{array}\right] K_0 \left[\begin{array}{cc} -k_g^2 & 0 \\ 1 & -1 \end{array}\right] \\
& = & A_1 + B_1 K_0 C_1 \\
B_0 & = & \left[\begin{array}{c} -\alpha k_g \\ 0 \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \end{array}\right] K_0 \left[\begin{array}{c} -k_g \\ 0 \end{array}\right] = B_1 K_0 C_2 \\
K_0 & = & \left[\begin{array}{cc} \alpha & \beta \end{array}\right]
\end{array} \tag{4}
$$

Hence, we want to design the *gain* K_0. Note that as previously matrix A_0 has to be Schur-Cohn.

The technique to design the parameters α and β follows the same procedure as developed in the local stability analysis case, by using slack variables and adequate change of variables. By considering the objective to enlarge the size of the level set $\mathcal{E}(P)$, one can illustrate that for a given performance level ρ, α is growing and β is decreasing (towards 0).

4. conclusion

This talk allows of illustrating on a simple case how the framework developed for systems with isolated nonlinearities (as saturation) can be used. Indeed, one can reinterpret some optimization algorithms as the stability/stabilization of discrete-time systems with a isolated nonlinearity in the input (corresponding to the gradient of the function to optimize $f(z)$). Not only some performance level (as the convergence rate) but also the characterization of the the basin of attraction of the minimizer $z^\star$ can be studied. Some robustness issues (for example if there is an error in the gradient $\nabla f(z_k)$) could be added as supplementary constraints.

These preliminary ingredients pave the way for several studies, as considering other types of algorithms [Michalowsky et al., 2020], as considering more complex functions to optimize (for example piecewise convex function, non-convex, ...), as adding constraints in the optimization problem:

$$
\begin{array}{c}
\min_z f(z) \\
\text{subject to } g(x) \geq w
\end{array}
$$

Finally, we could take inspiration of the recent results in the framework of nonlinear and hybrid systems to improve the class of algorithms of optimization [Le and Teel, 2021].

References

[Beck, 2014] Beck, A. (2014). *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Le and Teel, 2021] Le, J. and Teel, A. (2021). Hybrid heavy-ball systems: Reset methods for optimization with uncertainty. In *American Control Conference (ACC)*, New Orleans, USA.

[Lessard, 2018] Lessard, L. (2018). Robust control approach to algorithm analysis and design. In *CDC workshop on the Intersections of Machine Learning and Parameter Estimation in Control*, Miami, FL, USA.

[Lessard et al., 2016] Lessard, L., Recht, B., and Packard, A. (2016). Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. OPTIM*, 26(1):57–95.

[Michalowsky et al., 2020] Michalowsky, S., Scherer, C., and Ebenbauer, C. (2020). Robust and structure exploiting optimisation algorithms: an integral quadratic constraintapproach. *Int. J. of Control*, pages 1–24.

[Polyak, 1987] Polyak, B. T. (1987). *Introduction to optimization*. Optimization Software, New York, USA

[Tarbouriech et al., 2011] Tarbouriech, S., Garcia, G., Gomes da Silva Jr, J.-M., and Queinnec, I. (2011). *Stability and Stabilization of Linear Systems with Saturating Actuators*. Springer, London.

[Taylor et al., 2018] Taylor, A., B. Van Scoy, and Lessard, L. (2018). Lyapunov functions for first-order methods: Tight automated convergence guarantees. In *35th Int. Conference on Machine Learning,*, Stockholm, Sweden.

Neural ODE for classification, approximation and transport

Enrique Zuazua

(joint work with Domènec Ruiz-Balet)

In the past decade, deep learning has had a tremendous impact on technology, in a wide range of applications ([1]). Deep learning techniques are employed, for instance, in *supervised learning*, whose main goal is finding suitable approximations of an unknown function $f : \mathcal{X} \to \mathcal{Y}$ given a large sample $\{(x_i, y_i)\}_{i=1}^N$ of it. One can typically distinguish between two types of problems: we refer to *classification* (resp. *regression*) when $\mathcal{Y}$ is discrete (resp. continuous).

The so-called *residual neural networks* (ResNets, [4]) are a class of deep neural networks that can be written in the form of a discrete dynamical system

$$\begin{cases} \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + W^k \sigma(A^k \mathbf{x}_i^k + b^k) \\ \mathbf{x}_i^0 = x_i \end{cases} \tag{1}$$

for the layers $0 \leq k \leq N_{\text{layers}} - 1$, to be solved with input data x_i, $1 \leq i \leq N$, with the same discrete parameters $A^k, W^k \in \mathbb{R}^{d\times d}$, $b^k \in \mathbb{R}^d$ to be found, that play the role of controls. The nonlinearity σ is the so-called activation function, which is applied component-wise. $N_{\text{layers}} \geq 1$ is the number of layers or *the depth* of the ResNet.

ResNets can also be understood as Euler schemes of a continuous-time formulation (Neural ODEs) ([5])

$$\begin{cases} \dot{\mathbf{x}}_i(t) = W(t)\sigma(A(t)\mathbf{x}_i(t) + b(t)) & \text{for } t \in (0, T) \\ \mathbf{x}_i(0) = x_i \in \mathbb{R}^d, \ 1 \leq i \leq N \end{cases} \tag{2}$$

where $W, A \in L^\infty((0, T); \mathbb{R}^{d\times d})$ and $b \in L^\infty((0, T); \mathbb{R}^d)$ are time-dependent control functions aimed to steer the dynamics of the system simultaneously for all initial data $x_i \in \mathbb{R}^d$, $1 \leq i \leq N$.

Typically the free-parameters of a neural network are found by means of optimization algorithms out of the ensemble of available samples. The objective

functional is defined through the samples $\{x_i, y_i\}_{i=1}^N$ of the unknown function f. For a general mathematical study of the optimal control problems see [7, 8, 9].

In these notes we briefly present the main results in [6], focusing in the case of the ReLU activation function (Figure 1).

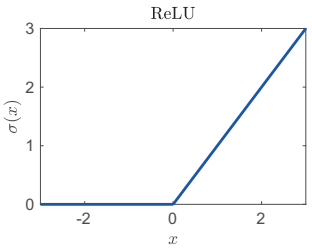

FIGURE 1. ReLU activation function, $\sigma(x) = \max\{0, x\}$.

Let us first discuss the classification and interpolation problems, which can be formulated as simultaneous control problems.

Let us assume that the data are distributed in M classes, $\{1, ..., M\}$. We first build a partition of $\mathbb{R}^d$, $\{S_m\}_{m=1}^M$. The classification problem consists then on finding controls W, A, b so that the solutions of (2) satisfy

$$\phi_T(x_i; W, A, b) \in S_m \iff y_i = m \tag{3}$$

where ϕ_T is the solution of (2) at time T with initial data x_i and controls W, A, b. Note that W, A, b are the same for every input $x_i, i = 1, ...N$.

We are therefore facing a simultaneous control problem, the targets being the sets S_m of the partition of the Euclidean space. Reaching these targets suffices to assure that the classification has been successfully achieved.

The interpolation problem is more challenging, the goal being to drive the trajectories to the N targets $\{z_i\}_{i=1}^N \subset \mathbb{R}^d$ exactly:

$$\phi_T(x_i; W, A, b) = z_i,\ i = 1, ..., N. \tag{4}$$

These simultaneous controllability properties are rare for classical dynamical systems in mechanics. For instance, it is an impossible task for linear systems, when all trajectories are assumed to satisfy the same dynamics. The key property that allows Neural ODEs to achieve this hard task is that the activation function vanishes in half-space. The choice of A and b allows to determine that half-space, in a time-dependent fashion, while the matrix W allows to orient the vector field in the active half-space in which the sigmoid is active (see Figure 2). Employing these flows in an adequate time-dependent fashion, one can achieve both classification and the simultaneous control (see Figure 3).

An even more ambitious goal of supervised learning is to approximate a given function. This can be also obtained in an approximate manner (in the L^2-sense for instance). In fact, the simultaneous control property of (2), can be interpreted

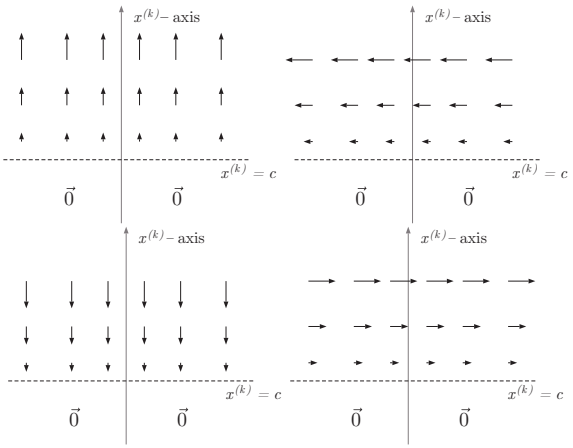

FIGURE 2. Some of the flows generated by a Neural ODE with the ReLU activation function.

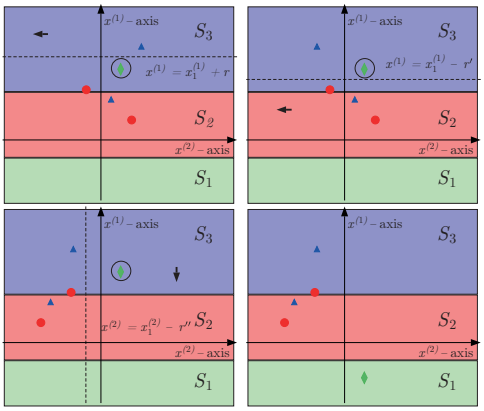

FIGURE 3. Qualitative representation of the procedure applied to classify the data by a partition made out of parallel strips. Each point is sent to the corresponding strip, according to its color. The dotted line represents the hyperplane chosen as interface between the active and frozen half-spaces and the black arrow the direction of the vector field in the active half-space.

as the approximate control property of the following neural transport equation in the Wasserstein distance

$$\begin{cases} \partial_t \rho + \mathrm{div}_x \big[\left(W(t)\boldsymbol{\sigma}(A(t)x + b(t))\right) \rho \big] = 0 \\ \rho(0) = \rho^0 \in C_c(\mathbb{R}^d). \end{cases}$$

One can even achieve a simultaneous control result for a finite number of Neural transport equations:

$$\begin{cases} \partial_t \rho_m + \mathrm{div}_x \left[(W(t)\boldsymbol{\sigma}(A(t)x + b(t))\rho_m) \right] = 0, & m = 1, ..., M \\ \rho_m(0) = \rho_m^0 \in C_c(\mathbb{R}^d), & m = 1, ..., M \\ \mathrm{supp}(\rho_m^0) \cap \mathrm{supp}(\rho_{m'}^0) = \emptyset & \text{if } m \neq m'. \end{cases}$$

The interested reader is refereed to the article [6] for more precise statements and the corresponding proofs.

Acknowledgments. Authors have been funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No.765579-ConFlex, the Alexander von Humboldt-Professorship program, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement NO. 694126-DyCon), the Transregio 154 Project "Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks" of the German DFG, grant MTM2017-92996-C2-1-R COSNET of MINECO (Spain) and by the Air Force Office of Scientific Research (AFOSR) under Award NO. FA9550-18-1-0242.

References

[1] LeCun, Y., Bengio, Y., and Hinton, G. *Deep learning.* Nature 521, 7553 (2015), 436–444.
[2] Cybenko, G. *Approximation by superpositions of a sigmoidal function.* Math. Control Signals Systems (1989), 303–314.
[3] Pinkus, A. *Approximation theory of the mlp model in neural networks.* Acta Numer. 8, 1 (1999), 143–195.
[4] He, K., Zhang, X., Ren, S., and Sun, J. *Deep residual learning for image recognition.* In Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 770–778.
[5] E, W. A proposal on machine learning via dynamical systems. *Commun. Math. Stat. 5*, 1 (2017), 1–11.
[6] Ruiz-Balet, D., and Zuazua, E. *Neural ODE control for classification, approximation and transport.* arXiv preprint arXiv:2104.05278 (2021).
[7] Esteve, C., Geshkovski, B., Pighin, D., and Zuazua, E. *Large-time asymptotics in deep learning.* arXiv preprint arXiv:2008.02491 (2020).
[8] Geshkovski, B. *Control in moving interfaces and deep learning.* PhD Thesis (2021).
[9] Esteve Yagüe, C., and Geshkovski, B. *Sparse approximation in learning via neural ODEs.* arXiv preprint arXiv:2102.13566 (2021).

Reporter: Lars Grüne

Participants

Dr. Michael Heinrich Baumann
Mathematisches Institut
Universität Bayreuth
95447 Bayreuth
GERMANY

Prof. Dr. Florian Dörfler
Automatic Control Laboratory
ETH Zürich
Physikstrasse 3
8092 Zürich
SWITZERLAND

Prof. Dr. Weinan E
Department of Mathematics
Princeton University
Fine Hall
Washington Road
Princeton, NJ 08544-1000
UNITED STATES

Prof. Dr. Lars Grüne
Mathematisches Institut
Lehrstuhl für Angewandte Mathematik
Universität Bayreuth
95440 Bayreuth
GERMANY

Prof. Dr. Roberto Guglielmi
Dept. of Mathematics
University of Waterloo
Waterloo ON N2L 3G1
CANADA

Dr. Jiequn Han
Department of Mathematics
Princeton University
Fine Hall
Washington Road
Princeton, NJ 08544-1000
UNITED STATES

Dr. Dante Kalise
School of Mathematical Sciences
University of Nottingham
University Park Campus
Nottingham NG7 2RD
UNITED KINGDOM

Prof. Dr. Wei Kang
Dept. of Applied Mathematics
Naval Postgraduate School
832 Dyer Rd.
Monterey CA 93943
UNITED STATES

Avneet Kaur
Dept. of Applied Mathematics
University of Waterloo
200 University Avenue West
Waterloo ON N2L 3G1
CANADA

Dr. Péter Koltai
Freie Universität Berlin
Dept. of Mathematics & Computer Science
Arnimallee 6
14195 Berlin
GERMANY

Milan Korda
LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex 4
FRANCE

Lisa Krügel
Mathematisches Institut
Universität Bayreuth
Postfach 101251
95447 Bayreuth
GERMANY

Dr. Nathan Kutz
Department of Applied Mathematics
University of Washington
Lewis Hall 201
Seattle, WA 98195-3925
UNITED STATES

Brian Mao
Dept. of Applied Mathematics
University of Waterloo
200 University Avenue West
Waterloo ON N2L 3G1
CANADA

Prof. Dr. Kirsten Anna Morris
Department of Applied Mathematics
University of Waterloo
Waterloo ON N2L 3G1
CANADA

Dr. Sebastian Peitz
Institut für Mathematik
Universität Paderborn
Warburger Str. 100
33098 Paderborn
GERMANY

Prof. Dr. Sophie Tarbouriech
LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex 4
FRANCE

Prof. Dr. Enrique Zuazua
Department Mathematik
Universität Erlangen-Nürnberg
Cauerstr. 11
91058 Erlangen
GERMANY