Introducing Monge-GPs: A new class of physics-informed Gaussian Processes (Extended abstract)

J. Moser[1], **C. Albert**[1], **S. Ranftl**[2,3]
(1) ITPCP, Graz University of Technology, Austria
(2) Courant Institute, New York University, New York, USA
(3) Division of Applied Mathematics, Brown University, USA

1 Introduction

Hybrid approaches combining differential equations and machine learning, commonly referred to as physics-informed machine learning, have gained significant attention in recent years. Prominent examples include Physics-Informed Neural Networks (PINNs) [1] and Physics-Informed Gaussian Processes (PIGPs), the latter naturally providing uncertainty quantification. PIGPs encode differential constraints directly in the covariance kernel, and existing approaches can be roughly grouped into two schools of thought. Operator-based constructions apply differential operators to a base kernel, yielding systematic and algorithmic methods, but are often restricted to controllable systems [2] or specific classes of differential equations [3] or may require many auxiliary outputs [4] and a relatively large amount of data. In contrast, Mercer-type constructions build kernels from problem-specific solution components such as Green's functions [5] or fundamental solutions [6]; while typically data efficient, they rely on analytical insight and substantial manual derivation. We propose Monge-GPs, a hybrid construction based on Monge parametrization that unifies operator-based kernels as in [2] and Mercer kernels as in [6]. By parametrizing the controllable dynamics algorithmically and restricting problem-specific design to a low-dimensional autonomous component, the approach substantially reduces the need for manual kernel design, and stays data efficient while lifting the restriction to controllable systems.

2 We present: Monge GPs - The theory

Let's suppose we have a system of linear ODEs or PDEs, represented by a full row-rank operator matrix R and unknown solutions η of the form

$$R\eta = 0. \tag{1}$$

This system is called *controllable*, if all possible solutions in a function space (for example C^∞) can be written in the parametrized form $\eta = B\zeta$ for any $\zeta \in C^\infty$, where B is an operator matrix that forms the null-space of R, i.e. $RB = 0$. Gaussian processes are closed under linear operators, so we can construct a parametrized GP by transforming the kernel with this parametrization matrix such as in [2]. We need a base kernel k_0, whose realizations are dense in C^∞, in order to be able to realize the ζ in our solution - the standard RBF kernel is the natural choice here. The parametrized kernel is then simply $K(\mathbf{x}, \mathbf{x}') = B(\mathbf{x})B^T(\mathbf{x}')k_0(\mathbf{x}, \mathbf{x}')$. We see that this parametrization approach depends crucially on$B$ being non-trivial and expressive enough, and therefore per definition on controllability.

If the system is not fully controllable however, we propose to instead decompose the solution using the so-called Monge parametrization as used in [7], leading to our Monge-GP. The basic idea can be formalized as follows: We can split the system matrix R into its controllable part R' and its autonomous part represented by R'', such that $R = R''R'$. Since R' is controllable, it has

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

a parametrization matrix B (i.e. $R'B = 0$), as well as a right-inverse matrix $(R')^{-1}$. Defining $R'\eta =: \xi$, we define a new, but equivalent, system of equations

$$R\eta = 0 \Leftrightarrow \big(R''\xi = 0 \wedge R'\eta = \xi\big). \tag{2}$$

This means we first need to find the solutions of the autonomous problem $R''\xi = 0$. This corresponds to a Mercer-type construction applied to the reduced autonomous subsystem, which is typically much simpler than the original system since the parametrizable component has already been separated. Given our found solution ξ (for $n \geq 1$ independent vector-valued solutions, ξ will be a matrix with n columns), we can go ahead and solve the inhomogeneous equation $R'\eta = \xi$, whose general solution can be split into a homogeneous and a particular part $\eta = \eta_h + \eta_p$. With the parametrization matrix B we can write the general homogeneous solution as $\eta_h = B\zeta$ for any $\zeta \in C^\infty$. Furthermore, by inverting R', we can find a particular solution to the inhomogeneous part of our problem, thus writing the general solution as

$$\eta = B\zeta + ((R')^{-1}\xi)\vec{C} \quad \forall \zeta \in C^\infty, \vec{C} \in \mathbb{R}^n. \tag{3}$$

How can we translate this into our kernel function? The homogeneous part of the solution is controllable and can therefore be translated into the kernel through parametrization just as discussed earlier. The autonomous, i.e. particular part of the solution in contrast has more constraints and can therefore in general be expressed through its fundamental solutions $\boldsymbol{\phi} = (R')^{-1}\xi$ as a Mercer kernel

$$k_p(\mathbf{x}, \mathbf{x}') = \sum_{ij} \phi_i(\mathbf{x})\Sigma_{ij}\phi_j(\mathbf{x}')^T \qquad \phi_i, \phi_j \in \boldsymbol{\phi}. \tag{4}$$

with a positive semi-definite, and in general diagonal, covariance Σ. Since GPs are closed under addition, we can then form a full GP that is able to realize all controllable and autonomous solutions of our system by adding the parametrized and autonomous kernel. The full solution kernel is then simply

$$K(\mathbf{x}, \mathbf{x}') = B(\mathbf{x})B^T(\mathbf{x}')k_0(\mathbf{x}, \mathbf{x}') + k_p(\mathbf{x}, \mathbf{x}'). \tag{5}$$

3 Example: Bipendulum

Let's consider the Bipendulum ODE system also discussed in [3, 8]. It consists of two pendula, of lengths $\ell_1 = \ell_2 = 1$ that are connected on top by a rod that moves according to its acceleration $u(t)$. The linearized system is represented by

$$R\eta = \begin{pmatrix} \partial_t^2 + 1 & 0 & -1 \\ 0 & \partial_t^2 + 1 & -1 \end{pmatrix} \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ u(t) \end{pmatrix} = 0. \tag{6}$$

setting the gravitational acceleration to $g = 1$ for simplicity. The parametrizable part of our system R', the corresponding parametrization matrix B and right-inverse matrix $(R')^{-1}$ are

$$R' = \begin{pmatrix} 1 & -1 & 0 \\ 0 & \partial_t^2 + 1 & -1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 \\ 1 \\ \partial_t^2 + 1 \end{pmatrix} \qquad (R')^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1. \end{pmatrix} \tag{7}$$

Since $R' \neq R$, we need to define the autonomous subsystem represented by R'' by factorizing R, yielding

$$R''\xi = \begin{pmatrix} \partial^2 + 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \xi_0 \\ \xi_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$\Rightarrow \xi_1 = 0, \; \xi_0 \in \text{span}\{\cos(t), \sin(t)\}. \tag{8}$$

This leads to the autonomous solution

$$\eta_p = (R')^{-1}\xi = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \cos(t) & \sin(t) \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \cos(t) & \sin(t) \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{9}$$

and therefore the full solution therefore reads

$$\eta = \begin{pmatrix} 1 \\ 1 \\ \partial^2 + 1 \end{pmatrix} \zeta + \begin{pmatrix} \cos(t) & \sin(t) \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \vec{C} \qquad \zeta \in C^\infty, \vec{C} \in \mathbb{R}^2. \tag{10}$$

All that is left to do is to translate this into a kernel function. Since the autonomous solution can be represented by the tuple $\phi(t) = \text{span}\{\begin{pmatrix} \cos(t) \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sin(t) \\ 0 \\ 0 \end{pmatrix}\}$, the corresponding kernel is a Mercer kernel as in eq. 4. Choosing $\Sigma = \mathbb{I}$ and exploiting trigonometric identities, we get the autonomous kernel

$$k_p(t, t') = \begin{pmatrix} \cos(t - t') & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{11}$$

and therefore the total Monge-GP kernel reads

$$\begin{aligned} k(t, t') &= B(t) k_0(t, t') B^T(t') + k_p(t, t') \\ &= \begin{pmatrix} k_0(t, t') + \cos(t - t') & k_0(t, t') & (\partial_{t'}^2 + 1) k_0(t, t') \\ k_0(t, t') & k_0(t, t') & (\partial_{t'}^2 + 1) k_0(t, t') \\ (\partial_t^2 + 1) k_0(t, t') & (\partial_t^2 + 1) k_0(t, t') & (\partial_t^2 + 1)(\partial_{t'}^2 + 1) k_0(t, t') \end{pmatrix}. \end{aligned} \tag{12}$$

We can now implement this kernel in python using the python package `PCGP`[1] [9] and use it to either solve the differential equation for any initial conditions, or to solve the inverse problem of estimating ℓ from available data, as discussed for example in [9]. In Fig. 1 we show a solution to the bipendulum problem for a sinusoidal input $u(t) = \sin(2t)$ with randomly chosen boundary conditions and additive Gaussian noise of $\sigma = 0.03$ on the input. The analytical solution for the particular boundary conditions is marked with a red dotted line. Even with few conditioning points, the posterior closely matches the analytical solution.

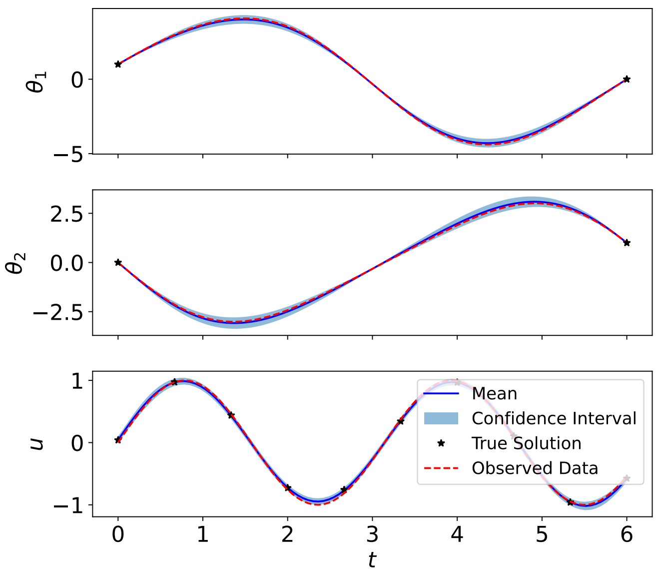

Figure 1: Example of the solution of a forward problem using Monge-GPs. The mean is depicted in dark blue, the 2σ confidence interval is light blue, the analytical solution is marked in red, and the training data are shown as black stars.

[1]freely available on github: https://github.com/moserjo/PCGP

4 Ongoing work

We are currently comparing this approach to others [4, 3, 6] in performance and expect it to work well, especially in the low data regime. It is a general approach that is specifically not restricted to controllable systems, and can be used both in ODE and PDE settings with constant coefficients. We are currently looking into well-definedness for systems with non-constant coefficients using for example Weyl algebras (allowing for polynomial and rational coefficients) as a suitable operator ring. We are also currently looking into finding the autonomous solution algorithmically, where grade filtration [10] may be a promising candidate. If successful, we will implement an automatic construction of the Monge-kernel as an extension in the python package `PCGP` [9].

5 Acknowledgements

The authors thank the reviewers for their helpful feedback. J.M. would also like to thank A. Quadrat for the introduction to homological algebra and grade filtration and fruitful discussion thereof. J.M. and C.A. were financially supported by the Austrian Research Promotion Agency (FFG) project VENTUS within the AI for Green funding program, Grant no. 910263. S.R. was financially supported by the Austrian Science Fund (FWF), Grant DOI: 10.55776/J4774.

References

[1] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[2] Markus Lange-Hegermann. Algorithmic Linearly Constrained Gaussian Processes. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/68b1fbe7f16e4ae3024973f12f3cb313-Abstract.html.

[3] Andreas Besginow and Markus Lange-Hegermann. Constraining Gaussian processes to systems of linear ordinary differential equations. *Advances in Neural Information Processing Systems*, 35: 29386–29399, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bcef27c5825d1ed8757290f237b2d851-Abstract-Conference.html.

[4] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, November 2017. ISSN 0021-9991. doi: 10.1016/j.jcp.2017.07.050. URL https://www.sciencedirect.com/science/article/pii/S0021999117305582.

[5] Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Linear Latent Force Models Using Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, November 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2013.86. URL https://ieeexplore.ieee.org/abstract/document/6514873.

[6] Christopher G. Albert. Gaussian Processes for Data Fulfilling Linear Differential Equations. *Proceedings*, 33(1):5, 2019. ISSN 2504-3900. doi: 10.3390/proceedings2019033005. URL https://www.mdpi.com/2504-3900/33/1/5. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[7] A Quadrat and D Robertz. Parametrizing all solutions of uncontrollable multidimensional linear systems. *IFAC Proceedings Volumes*, 38(1):49–54, 2005.

[8] Jean-Francois Pommaret. *Partial Differential Control Theory: Mathematical tools*, volume 530. Springer Science & Business Media, 2001.

[9] Johanna Moser, Christopher Albert, and Sascha Ranftl. Parameter learning with physics-consistent gaussian processes. *In preparation*, 2026.

[10] Alban Quadrat. Grade filtration of linear functional systems. *Acta Applicandae Mathematicae*, 127(1): 27–86, 2013.