Learning Mechanical Systems by Hamiltonian Neural Networks

Eva Dierkes[1,*] and **Kathrin Flaßkamp**[2]

[1] Center for Industrial Mathematics, University of Bremen

[2] Systems Modeling and Simulation, Saarland University

The great success of machine learning in image processing and related fields also motivated its application to dynamical system identification. In particular, neural networks are trained to learn equations of motion and thus provide an alternative to first-principle modeling. While these black-box algorithms are quite flexible regarding the system structure, they often have difficulties in learning basic physics laws which are intrinsic system properties. Recently, Hamiltonian neural networks (HNN) were introduced to explicitly learn the total energy of a system in order to overcome the lack of physical rules. However, Hamiltonian systems often also contain other structures such as symmetries in terms of further invariances. In this contribution, we extend HNN such they respect system invariances in addition to the Hamiltonian. The proposed extension leads to a trade off between energy conservation and conservation of invariance properties, which we investigate exemplarily.

1 Introduction

The developments in industry and technology have led to the construction of increasingly complex systems such that their modeling is often non-trivial. However, these systems typically come with large amounts of data. Thus, together with the rapid development of neural networks, this fostered the popularity of data-based models. Since neural networks are per se black-box models, they are not aware of physical laws. This is why physics-informed learning emerged as a field of research. Within this area, we focus on a special network architecture that has been developed for Hamiltonian systems. This work extends the existing structure so that it is possible to also take system symmetries into account.

2 Hamiltonian systems

Hamiltonian Systems form an important subgroup in the field of mechanical systems (for detailed information we refer to [2]). This field is closely linked to Lagrangian mechanics by the bijective Legendre transformation. In Hamiltonian mechanics, the system is fully described by the scalar valued Hamiltonian $H(q, p)$, where q are generalized coordinates and p the corresponding momenta. The Hamiltonian represents the total energy of the system and the dynamics are described by the *Hamilton's equations*: $\dot{q}_i = \partial H(q, p)/\partial p_i$, $\dot{p}_i = -\partial H(q, p)/\partial q_i$.

In nature, many systems have symmetries such as periodicity, translational or rotational invariances. Here, we consider continuous symmetries which can be represented by actions Φ_g of a Lie group G. Roughly speaking, Φ_g is called a *symmetry* of the mechanical system, if it maps trajectories to trajectories, i.e. it commutes with the system's flow. In this contribution, we focus on position invariance with respect to one, w.l.o.g. the i-th, coordinate. This kind of symmetry is described by $G = \mathbb{R}$ with $\Phi_g(q, \dot{q}) = (q + v_i g, \dot{q})$ (v_i denotes the i-th unit vector). The invariance in q_i leads to a constant momentum $\dot{p}_i = -\partial H(q, p)/\partial q_i = 0$.

3 Neural networks architectures

Neural networks build an input–output relation by nodes that are connected by multiple nested and/or parallel affine transformations, where to each a non-linear activation function is attached. The weights and biases of the transformations (together denoted by θ) are fitted to minimize a given loss function.

Hamiltonian neural network [1] Hamiltonian neural networks learn the Hamiltonian instead of the ode. Thus, the neural network represents $H_\theta = f_{\text{HNN}}(q, p)$ and the ode of the system is derived by *Hamilton's equations* as described above. The training of the network is done using the loss function $\mathcal{L}_{\text{HNN}} = \|\dot{q} - \partial H_\theta/\partial p\|_2^2 + \|\dot{p} + \partial H_\theta/\partial q\|_2^2$. The derivatives of the network output H_θ with respect to the inputs q and p can be obtained using algorithmic differentiation.

Hamiltonian neural network including system invariances In order to make use of known invariances in the system, we propose to add a penalty term to the loss function, $\mathcal{L}_{\text{HNN\_I}} = (1 - \alpha)\mathcal{L}_{\text{HNN}} + \alpha\gamma \|\partial H_\theta/\partial q_i\|_2^2$. Where γ is used as a scaling parameter to bring both parts of the objective function to the same order of magnitude. The parameter α can be used to vary between the two goals of being energy preserving and position invariant. Theoretically, there is a solution (the exact Hamiltonian) which minimizes both equally well. The network is an approximation of the system and thus this additional term helps to find a better local minimum during fitting.

---

* Corresponding author: e-mail eva.dierkes@uni-bremen.de

PAMM · Proc. Appl. Math. Mech. 2021;**21**:1 e202100116.   www.gamm-proceedings.com

https://doi.org/10.1002/pamm.202100116   © 2021 The Authors. *Proceedings in Applied Mathematics & Mechanics* published by Wiley-VCH GmbH.   **1 of 2**
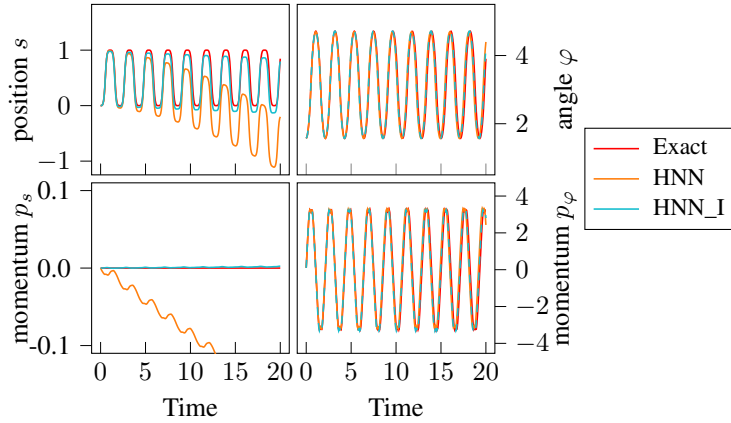
Fig. 1: Evaluated states if the cart pole. Evaluation of different (learned) models with starting point $[0, \pi/2, 0, 0.1]$.
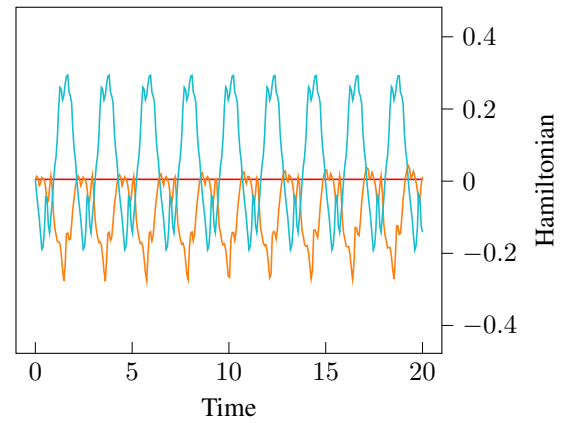

Fig. 2: Hamiltonian of the cart pole evaluated for the states shown in Fig. 1.

4   Results

The results are shown using the cart-pole example, where the position of the cart s and the angle of the pole φ lead to the Hamiltonian given by

$$H(\varphi, p_s, p_\varphi) = \frac{ap_s^2 + 2bp_sp_\varphi\cos\varphi + cp_\varphi^2}{2ac - b^2\cos^2\varphi} - D\cos\varphi,$$

where $a = ml^2$, $b = ml$, $c = M + m$ and $D = -mgl$. l denotes the length and m the weight of the pole, M corresponds to the mass of the cart and $g = 9.81$ to gravitation. For simplicity $l = m = M = 1$.

The Hamiltonian does not depend on s and thus, the penalty term is defined by $\|\partial H_\theta/\partial s\|_2^2$. Training data is collected using 1500 simulations of $3\,\mathrm{s}$ duration with different initial values in a predefined range. Gaussian noise with standard deviation $\sigma^2 = 0.01$ is added to all data points and the data set is split into $70\,\%$ training and $30\,\%$ validation data.

An evaluation of the trained models is shown in Fig. 1 and the corresponding Hamiltonian in Fig. 2. In the lower left plot of Fig. 1 one can see the momentum p_s, which should be a conserved quantity. The proposed extension, denoted by HNN_I, conserves the symmetry much better compared to the original HNN while also keeping the Hamiltonian within an acceptable range.

In the definition of the loss function $\mathcal{L}_{\mathrm{HNN\_I}}$, the weights α and γ have been introduced. To recommend weighting parameters for the cart-pole system, a parameter study is conducted for $\alpha \in [0, 0.99]$, while keeping $\gamma = 10$ fixed. The values for $\|\partial H_\theta/\partial s\|_2^2$ are shown in Fig. 3 for all runs. At the same time, the values for the first term of the loss function, i.e. $\mathcal{L}_{\mathrm{HNN}}$ remain in the same order of magnitude while α is varied. This support the previously mentioned point that both parts in the objective are not truly contradictory. Eventually, we choose $\alpha = 0.5$ by which we obtain $\|\partial H_\theta/\partial s\|_2^2 = 0.048$.
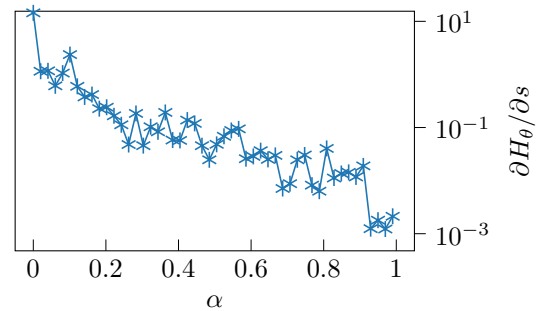

Fig. 3: Final values of the invariance criterion for different weighting in the loss function $\mathcal{L}_{HNN\_I}$.

5   Conclusion

In this paper, an extension to Hamiltonian neural networks was presented which is based on known symmetries in terms of invariances within data from Hamiltonian system simulations. Numerical results for the cart-pole example show that this extension helps the network to achieve improved results in structure-preserving system simulation with neural network.

References

[1] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian Neural Networks, Advances in Neural Information Processing Systems (2019).
[2] R. Abraham and J. Marsden, Foundations of Mechanics, American Mathematical Society, Providence, Rhode Island (2008).