Neural Energy Casimir Control

Liang Xu
Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne, Switzerland
LIANG.XU@EPFL.CH

Muhammad Zakwan
Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne, Switzerland
MUHAMMAD.ZAKWAN@EPFL.CH

Giancarlo Ferrari-Trecate
Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne, Switzerland
GIANCARLO.FERRARITRECATE@EPFL.CH

Abstract
The energy Casimir method is an effective controller design approach to stabilize port-Hamiltonian systems at a desired equilibrium. However, its application relies on finding suitable Casimir and Lyapunov functions, which are generally intractable. In this paper, we propose a neural network-based framework to learn these functions. We show how to achieve equilibrium assignment by adding suitable regularization terms in the optimization cost. We also propose a parameterization of Casimir functions for reducing the training complexity. Moreover, the distance between the equilibrium point of the learned Lyapunov function and the desired equilibrium point is analyzed, which indicates that for small suboptimality gaps, the error decreases linearly with respect to the training loss. Our methods are backed up by simulations on a pendulum system.

Keywords: Port-Hamiltonian Systems, Energy Casimir Control, Neural Lyapunov Function, Equilibrium Assignment

1. Introduction
Port-Hamiltonian systems (PHSs) describe real-world applications as the interconnection of dynamical energy-storing, static energy-dissipating, and static lossless energy-routing elements. The interconnections among these elements are pairs of variables called ports. Since energy is the building block among various physical domains, port-Hamiltonian modeling provides a generic framework for describing physical systems, and has been extensively used in various areas, for example, mechanical systems (van der Schaft and Jeltsema, 2014; Schaller et al., 2021; Tsolakis and Keviczky, 2021) and electrical systems (Strehle et al., 2020; Padilla et al., 2020). The increasing significance of the port-Hamiltonian framework for modeling, control, and analysis of multi-physics systems is also evident from several monographs, e.g. Brogliato et al. (2006); van der Schaft (2017).

PHSs are naturally passive, i.e., the increase in the system energy (described by a scalar function, called Hamiltonian), is smaller than the supply rate (the product of the input port and the output port). Passivity implies that the Hamiltonian may serve as a Lyapunov function, and negative output feedback can asymptotically stabilize the PHS to the local minimum of the Hamiltonian. However, if the Hamiltonian does not have a minimum at the desired equilibrium, the Hamiltonian itself alone is not a Lyapunov function and negative output feedback control fails. A well-known method for overcoming this obstacle is the energy Casimir method, which uses additional conserved quantities (called Casimir functions, or Casimirs) in addition to the Hamiltonian for constructing the Lyapunov function. Specifically, this approach interconnects the plant PHS with a controller in the
port-Hamiltonian form and aims to generate Casimir functions, such that the closed-loop system has a well-defined Lyapunov function with a minimum located at the desired equilibrium. Then, an additional damping term is employed to asymptotically stabilize the closed-loop system to the desired equilibrium (Ortega et al., 1999, 2008).

Unfortunately, the computation of Casimirs and Lyapunov functions requires to solve a set of partial differential equations (PDEs), which usually are analytically intractable and computationally challenging (Wu et al., 2020). Due to the universal approximation capability of neural networks (NNs) and various tools available for training NNs, in this paper, we propose a NN based approach to compute Casimirs and Lyapunov functions and shape the equilibrium of the closed-loop system without solving convoluted PDEs. There is a also growing interest in using NNs for designing controllers and certifying the closed-loop stability. Chang et al. (2020) and Dai et al. (2021) propose iterative procedures to construct a neural controller and a neural Lyapunov function with provable stability guarantees. Yin et al. (2021) and Pauli et al. (2021) abstract NNs using integral quadratic constraints and analyze the stability of feedback systems with neural controllers. Graph NNs are used for distributed control in Yang and Matni (2021) and Gama and Sojoudi (2021), and sufficient conditions on input-state stability are further provided in Gama and Sojoudi (2021). Richards et al. (2018) and Chen et al. (2021) use NNs to learn the region of attraction of nonlinear systems for obtaining stability and safety certificates. The above works mainly deal with either linear systems or very general classes of non-linear systems.

In this paper, we focus on the port-Hamiltonian models and use NNs to tackle the computational obstacles in the application of the energy Casimir method. We show how to transform the Casimir design problem into a parametric optimization problem. We use NNs to approximate the controller Hamiltonian, the Casimirs and the Lyapunov function for the closed-loop system. To assign the minimum of the Lyapunov function to the desired equilibrium point, we use a novel training cost that penalizes the norm of the Jacobian and the eigenvalues of the Hessian of the neural Lyapunov function evaluated at the desired equilibrium point. Moreover, we incorporate PDE constraints on Casimirs in the cost as a regularization term. However, embedding PDE constraints in the cost might significantly increase the computational complexity of the learning process. Therefore, we further present a NN-based parameterization of Casimirs to satisfy the PDE constraints by design, which achieves a trade-off between the representation generality and the training speed. Moreover, we provide an upper bound on the difference between the desired and achieved equilibrium point in terms of the training loss, which shows that for small suboptimality gaps, the equilibrium assignment error scales linearly with the training loss. We experiment on the set-point control of pendulum systems and show that this method asymptotically stabilizes the pendulum to the desired equilibrium and learns a neural Lyapunov function with a large region of attraction.

**Organization.** The paper is organized as follows. Section 2 recalls preliminaries on PHSs and the classic energy Casimir method. Section 3 presents our main results on the parameterization of Casimirs and a novel optimization cost for achieving the equilibrium assignment. Section 4 discusses simulation experiments and concluding remarks are given in Section 5.

**Notations.** For a function $H : \mathbb{R}^n \rightarrow \mathbb{R}$, $\frac{\partial H}{\partial x}(x)$ denotes the column vector of partial derivatives of $H$. $\frac{\partial^T H}{\partial x}(x)$ denotes the transpose of $\frac{\partial H}{\partial x}(x)$.

### 2. Preliminaries

In this section, we briefly introduce PHSs and the energy Casimir control method.
2.1. Port-Hamiltonian System

An input-state-output PHS can be expressed as

\[
\dot{x} = (J(x) - R(x)) \frac{\partial H(x)}{\partial x} + G(x)u, \quad y = G^\top(x) \frac{\partial H(x)}{\partial x},
\]

(1)

where \(x \in \mathbb{R}^n\) is the state, \(u \in \mathbb{R}^m\) is the input, and \(y \in \mathbb{R}^m\) is the measured output; \(J(x), R(x)\) are the interconnection and damping matrices, respectively, satisfying \(J(x) = -J^\top(x)\) (skew-symmetric) and \(R(x) = R^\top(x) \geq 0\); \(G(x)\) is the input matrix and is assumed full rank; \(H : \mathbb{R}^n \to \mathbb{R}\) is a continuously differentiable function and is called the Hamiltonian of the system. By construction, one has the passivity property

\[
\dot{H}(x) \leq y^\top u,
\]

(2)

which is extensively used in the stability analysis of PHSs (van der Schaft, 2017).

2.2. Casimir Function

Casimirs are conserved quantities along the state trajectories of a PHS.

**Definition 1 (Casimir function)** A continuously differentiable function \(C(x) : \mathbb{R}^n \to \mathbb{R}\) is a Casimir function for (1), if it satisfies

\[
\frac{\partial^\top C}{\partial x}(x)(J(x) - R(x)) = 0, \quad x \in \mathbb{R}^n.
\]

It follows that for \(u = 0\),

\[
\frac{d}{dt} C = \frac{\partial^\top C}{\partial x}(x)[J(x) - R(x)] \frac{\partial H}{\partial x}(x) = 0,
\]

and, therefore, Casimir functions are invariant along system trajectories. Moreover, one can prove that \(C(x)\) is a Casimir function for (1) if and only if the following condition holds (van der Schaft, 2017, Proposition 6.4.2)

\[
\frac{\partial^\top C}{\partial x}(x)J(x) = 0 \quad \text{and} \quad \frac{\partial^\top C}{\partial x}(x)R(x) = 0, \quad x \in \mathbb{R}^n.
\]

(3)

2.3. Energy Casimir Method

We are interested in set-point control of (1), that is in designing a controller to stabilize (1) around a given set-point \(x^*\). If \(x^*\) is a local minimum of the Hamiltonian function \(H\), due to the passivity property (2), a negative output feedback controller, i.e., \(u = -y\), would achieve the goal (van der Schaft, 2017). However, if \(x^*\) is not a local minimum of \(H\), this approach can not be applied and alternative methods are needed.

The energy Casimir method tackles this issue by introducing a controller in the PHS form and designing an appropriate Casimir and a Lyapunov function to place the closed-loop system minimum at \((x^*, \xi^*)\) for certain controller equilibrium point \(\xi^*\). The design procedure is thoroughly
described in van der Schaft (2017) and summarized below for convenience. Consider the PHS controller
\[
\dot{\xi} = [J_c(\xi) - R_c(\xi)] \frac{\partial H_c}{\partial \xi}(\xi) + G_c(\xi)u_c, \quad y_c = G_c^T(\xi) \frac{\partial H_c}{\partial \xi}(\xi),
\]
(4)
where \(\xi \in \mathbb{R}^{n_c}, u_c \in \mathbb{R}^m, y_c \in \mathbb{R}^m\) are the controller state, input, and output, respectively; \(J_c(\xi) = -J_c^T(\xi), R_c(\xi) = R_c^T(\xi) \geq 0\) and \(G_c(\xi)\) is of full rank. The continuously differentiable function \(H_c : \mathbb{R}^{n_c} \rightarrow \mathbb{R}\) is the controller Hamiltonian. By coupling the plant (1) with the controller (4) via the standard negative feedback interconnection
\[
u = -y_c + v, \quad u_c = y + v_c,
\]
(5)
where \(v, v_c\) are auxiliary input signals that will be specified later, we obtain the closed-loop system
\[
\begin{bmatrix}
\dot{x} \\
\dot{\xi}
\end{bmatrix}
= \begin{bmatrix}
J(x) & -G(x)G_c^T(\xi) \\
G_c(\xi)G_c^T(x) & J_c(\xi)
\end{bmatrix}
- \begin{bmatrix}
R(x) & 0 \\
0 & R_c(\xi)
\end{bmatrix}
\begin{bmatrix}
v \\
v_c
\end{bmatrix}
- \begin{bmatrix}
\frac{\partial H_c}{\partial \xi}(x) \\
\frac{\partial H_c}{\partial \xi}(\xi)
\end{bmatrix},
\]
(6)
which is also a PHS, with state space \(\mathbb{R}^{n+n_c}\), Hamiltonian \(H(x) + H_c(\xi)\), interconnection structure matrix \(J_{cl}(x, \xi)\), dissipation matrix \(R_{cl}(x, \xi)\), inputs \((v, v_c)\) and outputs \((y, y_c)\). For simplicity, we denote the state for the closed-loop system as \(z\), i.e., \(z = (x, \xi)\).

The control design procedure and the corresponding closed-loop stability properties are described in the following Lemma, which is a simplified version of Proposition 7.1.8 in van der Schaft (2017).

**Lemma 2** If one can find \(J_c(\xi), R_c(\xi), G_c(\xi), H_c(\xi)\) for the controller (4), a Casimir function \(C(z)\) for the closed-loop system (6), a function \(\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}\), and a \(\xi^* \in \mathbb{R}^{n_c}\), such that the Lyapunov function defined by \(V = \Phi(H, H_c, C)\) has a local minimum at \(z^* = (x^*, \xi^*), \) i.e.,
\[
\frac{\partial V}{\partial z}|_{z^*} = 0, \quad \frac{\partial^2 V}{\partial z^2}|_{z^*} > 0.
\]
(7)
Then the auxiliary signal
\[
v = -DG^T(x) \frac{\partial V}{\partial x}(x, \xi), \quad v_c = -D_cG_c^T(\xi) \frac{\partial V}{\partial \xi}(x, \xi)
\]
with \(D = D^T > 0, D_c = D_c^T > 0\) asymptotically stabilizes (6) to \((x^*, \xi^*)\).

The main obstacle to the application of the energy Casimir method is that there is no systematic way to design parameters and functions appearing in Lemma 2.
3. Neural Energy Casimir Control

In this section, we describe our method, which uses NNs to approximate the functions in Lemma 2 and transforms the design problem into a parametric optimization problem.

We assume that $J_c(\xi)$, $R_c(\xi)$, $G_c(\xi)$ are fixed a priori so that one only need to find candidate functions $H_c, \Phi, C$ and the controller state $\xi^*$ such that (7) holds. We denote with $H_{c,\theta_1}, \Phi_{\theta_2}, C_{\theta_3}$ the NN approximations of the functions $H_c, \Phi$ and $C$, respectively, where $\theta_1, \theta_2, \theta_3$ represent NN parameters. We have the following optimization targets.

1. $C_{\theta_3}$ should be a Casimir function for the closed-loop system, i.e., the following condition should be satisfied
   \[
   \partial^T C_{\theta_3}(z) J_{cl}(z) = 0 \quad \text{and} \quad \partial^T C_{\theta_3}(z) R_{cl}(z) = 0, \quad z \in \mathbb{R}^{n+n_c}.
   \]  

2. $V_{\theta}$ defined as $V_{\theta} = \Phi_{\theta_2}(H, H_{c,\theta_1}, C_{\theta_3})$ should have a local minimum at $z^* = (x^*, \xi^*)$ for some $\xi^*$, i.e., the following condition should be satisfied
   \[
   \frac{\partial V_{\theta}}{\partial z} \big|_{z^*} = 0, \quad \frac{\partial^2 V_{\theta}}{\partial z^2} \big|_{z^*} > 0.
   \]

To meet the requirement (8), albeit in an approximate way, one could grid the region of interest, so obtaining a set of points $\Omega = \{z_i\}_{i=1}^N$, and require (8) holds on all grid points. To satisfy (9), one could minimize the norm of the gradient $\frac{\partial V_{\theta}}{\partial z} \big|_{z^*}$ and penalize the case that the smallest eigenvalue of the Hessian $\frac{\partial^2 V_{\theta}}{\partial z^2} \big|_{z^*}$ is negative. In order to consider both goals simultaneously, we propose to train NNs to minimize the following cost

\[
\sum_{z_i \in \Omega} \left( \left\| \frac{\partial^T C_{\theta_3}}{\partial z} \big|_{z_i} J_{cl}(z_i) \right\| + \left\| \frac{\partial^T C_{\theta_3}}{\partial z} \big|_{z_i} R_{cl}(z_i) \right\| \right) + \left\| \frac{\partial V_{\theta}}{\partial z} \big|_{z^*} \right\| + \text{ReLU} \left( -\lambda_{\text{min}} \left( \frac{\partial^2 V_{\theta}}{\partial z^2} \big|_{z^*} \right) \right),
\]

where the ReLU function is defined as $\text{ReLU}(x) = \max(0, x)$.

The above cost function involves the Jacobian and Hessian of NN outputs w.r.t. some inputs. Existing machine learning frameworks, such as pytorch, provide functions to calculate these quantities, thus allowing to compute and optimize the cost (10).

However, since we need to grid the region of interest and minimize the terms $\left\| \frac{\partial^T C_{\theta_3}}{\partial z} J_{cl} \right\|$ and $\left\| \frac{\partial^T C_{\theta_3}}{\partial z} R_{cl} \right\|$ on all grid points, the computational requirement for training might be high. Moreover, only approximate Casimirs will be learned and the closed-loop stability cannot be rigorously guaranteed. In the following, we propose to explicitly parameterize Casimirs to avoid these issues.

3.1. Parameterization of Casimir Functions

In this section, we introduce a parameterization of Casimir functions for satisfying the constraints (3) by design. As a result, the first two terms in the cost (10) can be removed, which simplifies the training.

1. Hereafter, we use $\theta$ to denote parameters of all the neural networks.
We first assume that $J_{cl}$ and $R_{cl}$ are constant matrices, and then discuss the more general case where $J_{cl}$ and $R_{cl}$ are functions of the state $z$. Suppose the dimension of $\ker J_{cl} \cap \ker R_{cl}$ is $r$, and let $v_1, \ldots, v_r$ be the basis of $\ker J \cap \ker R_{cl}$. Then from (3), we have

$$\frac{\partial C}{\partial z}(z) = \sum_{i=1}^{r} \alpha_i(z) v_i$$

for some functions $\alpha_i(z) : \mathbb{R}^{n+n_c} \rightarrow \mathbb{R}$, $i = 1, \ldots, r$. A candidate function $C$ satisfying (11) is

$$C(z) = K\left(\sum_{i=1}^{r} \beta_i(z^\top v_i)\right)$$

for some functions $K : \mathbb{R} \rightarrow \mathbb{R}$, $\beta_i : \mathbb{R} \rightarrow \mathbb{R}$, $i = 1, \ldots, r$. Indeed, in this case, $\frac{\partial C}{\partial z}$ verifies

$$\frac{\partial C}{\partial z} = \frac{\partial K}{\partial \text{input}} \sum_{i=1}^{r} \frac{\partial \beta_i}{\partial \text{input}} v_i = \sum_{i=1}^{r} \alpha_i(z) v_i,$$

where $\alpha_i(z) = \frac{\partial K}{\partial \text{input}} \frac{\partial \beta_i}{\partial \text{input}}$, and $\frac{\partial K}{\partial \text{input}}, \frac{\partial \beta_i}{\partial \text{input}}$ represent the gradient of $K(\cdot)$ and $\beta_i(\cdot)$ with respect to their input, respectively.

If $J_{cl}$ and $R_{cl}$ depends on $z$, i.e., in the form of $J_{cl}(z)$, $R_{cl}(z)$, then from (3), we have

$$\frac{\partial C}{\partial z}(z) = \sum_{i=1}^{r(z)} \alpha_i(z) v_i(z),$$

(13)

where $v_i(z), i = 1, \ldots, r(z)$ form a basis of $\ker J_{cl}(z) \cap \ker R_{cl}(z)$. Denote the RHS of (13) as $F(z)$. For the existence of $C$ satisfying (13), $F(z)$ must satisfy the integrability condition (Lee, 2012, Equation 11.21)

$$\frac{\partial F_i}{\partial z_j}(z) = \frac{\partial F_j}{\partial z_i}(z), \quad \forall i, j, z$$

(14)

where $F_i$ and $z_i$ are the $i$-th element of $F$ and $z$, respectively. The requirement (14) is the main difficulty in providing a universal parameterization of general Casimirs. Hereafter, we restrict our attention on the case where $F(z) = [F_1(z_1), \ldots, F_{n+n_c}(z_{n+n_c})]^\top$, that is the $i$-th element of $F$ is only a function of the $i$-th element of $z$. A candidate $C$ satisfying (13) is then given by

$$C = K(\int F_1(z_1)dz_1 + \ldots + \int F_{n+n_c}(z_{n+n_c})dz_{n+n_c}),$$

for some continuously differentiable function $K : \mathbb{R} \rightarrow \mathbb{R}$.

### 3.2. Neural Energy Casimir Control with Parameterized Casimir Functions

With the proposed parameterization of Casimir functions, we modify the neural energy Casimir control design as follows, where we assume $J_{cl}, R_{cl}$ are constant matrices\(^2\). We use the NNs

\(^2\) When $J_{cl}, R_{cl}$ are functions of $z$, and if we can find a parameterization of $C$, the modified neural energy Casimir control can be obtained similarly.
$H_{c,\theta_1}(), \Phi_{\theta_2}()$ to approximate $H_c()$ and $\Phi()$ in Lemma 2, respectively, and use the NNs $K_{\theta_3}()$ and $\beta_{i,\theta_{r+3}}()$ to approximate $K()$ and $\beta_i()$, $i = 1, \ldots, r$ in (12), respectively, where $\theta_i, i = 1, \ldots, r+3$ are NN parameters. Then we define the neural Lyapunov function

$$V_\theta(z) = \Phi_{\theta_2}(H(x), H_{c,\theta_1}(\xi), K_{\theta_3}(\sum_{i=1}^r \beta_{i,\theta_{r+3}}(z^Tv_i)))$$

and solve the following optimization problem

$$\min_{\theta_1,\ldots,\theta_{r+3},\xi^*} \| \frac{\partial V_\theta(z)}{\partial z} \|_{z^*} + \text{ReLU} \left( -\lambda_{\min} \left( \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} - a I \right) \right),$$

(15)

where the term $aI$ with $a > 0$ is added to the Hessian regularization to promote local convexity of $V_\theta$ around the closed-loop equilibrium. This can improve the control performance as shown in the next section.

3.3. Performance Analysis

In this section, we analyze the performance of the neural energy Casimir control method. We show that when the loss (15) is sufficiently small after training, the distance between the equilibrium point of the learned Lyapunov function and the desired equilibrium point $z^*$ is also small. Suppose after training, we obtain a small loss $\epsilon$

$$\epsilon = \| \frac{\partial V_\theta(z)}{\partial z} \|_{z^*} + \text{ReLU} \left( -\lambda_{\min} \left( \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} - a I \right) \right).$$

(16)

From (16), we have

$$\| \frac{\partial V_\theta(z)}{\partial z} \|_{z^*} \leq \epsilon,$$

(17)

$$\text{ReLU} \left( -\lambda_{\min} \left( \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} - a I \right) \right) \leq \epsilon.$$  

(18)

Since $\epsilon$ is close to zero and $a$ is larger than zero, we have that $a - \epsilon > 0$. Then from (18) we have $\frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} \geq (a - \epsilon) I$. By left and right multiplying the above inequality with $\left( \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} \right)^{-\frac{1}{2}}$, we have $\left( \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} \right)^{-1} \leq \frac{1}{a - \epsilon} I$, therefore

$$\| \left( \frac{\partial^2 V_\theta(z)}{\partial z^2} \right)|_{z^*}^{-1} \| \leq 1/(a - \epsilon),$$

(19)

where the matrix norm is the spectral norm.

The gradient $\frac{\partial V_\theta(z)}{\partial z} |_z$ can be expanded using Taylor’s theorem as

$$\frac{\partial V_\theta(z)}{\partial z} |_z = \frac{\partial V_\theta(z)}{\partial z} |_{z^*} + \frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} (z - z^*) + o(z - z^*)$$

(20)

If $\epsilon$ is sufficiently small, from (17), $\frac{\partial V_\theta(z)}{\partial z} |_{z^*}$ is also close to zero. Since $\frac{\partial^2 V_\theta(z)}{\partial z^2} |_{z^*} > 0$, then in a sufficiently small neighborhood of $z^*$, there exists $\tilde{z}$, such that $\frac{\partial V_\theta(z)}{\partial z} |_{\tilde{z}} = 0$. Moreover, the
continuity of $\frac{\partial^2 V_\theta(z)}{\partial z^2}|_{z^*}$ implies that $\frac{\partial^2 V_\theta(z)}{\partial z^2}|_{z^*} > 0$, which means $\bar{z}$ is a local minimum of $V_\theta$ in a small neighborhood of $z^*$. From (20), we also have

$$\|\bar{z} - z^*\| \leq \|\frac{\partial^2 V_\theta(z)}{\partial z^2}|_{z^*}\| \left( \|\frac{\partial V_\theta(z)}{\partial z}\|_{z^*} + \|\alpha(z - z^*)\| \right),$$

Omitting the high-order terms and in view of (19), we can obtain

$$\|\bar{z} - z^*\| \leq \frac{\epsilon}{a - \epsilon}.$$ 

The above result implies that if the cost (15) to close to zero after minimization, we can effectively reduce the set-point control bias caused by the NNs approximation errors.

**Remark 3** The (neural) energy Casimir method can only guarantee the local stability of the desired equilibrium, since it only requires the desired equilibrium to be a local minimum of the Lyapunov function. However, from the simulation example given in the next section, we can observe that the learned Lyapunov function has a large region of attraction. One could tune the region of attraction achieved with the neural energy Casimir method by including additional regularization terms in the cost (15). For example, similar to Chang et al. (2020), we can sample the region of interest to obtain data points $\{z_1, \ldots, z_N\}$, and add the regularization term $\text{ReLU}(\frac{1}{N} \sum_{i=1}^{N} (\|z_i\|^2 - \gamma V_\theta(z_i)))$ to the cost (15) with $\gamma$ to be a hyperparameter, which regulates how fast the Lyapunov function value increases with respect to its input. Moreover, after training, we can follow similar procedures outlined in Chang et al. (2020) and use satisfiability modulo theory solvers to verify whether the learned $V_\theta(z)$ is a Lyapunov function over all state vectors in the region of interest.

### 4. Simulations

In this section, we illustrate the performance of the proposed control design method via simulations. We consider the set-point control of the pendulum system described in (van der Schaft, 2017, Example 7.1.12)

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \sin(q) \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = p, \quad (22)$$

where $q$ is the angle and $p$ is the momentum, and $u$ is the input torque. The system (22) is in the PHS form (1) with

$$H = \frac{1}{2} p^2 + (1 - \cos(q)), \quad J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$ 

Suppose we want to stabilize the pendulum at the non-zero angle $q^* = \frac{\pi}{4}$ with $p^* = 0$. We consider the controller

$$\dot{\xi} = y + v_c, \quad u = -\frac{\partial H_c(\xi)}{\partial \xi} + v,$$
which corresponds to (4) with $J_c = 0$, $R_c = 0$, $G_c = 1$, interconnected with the plant (22) as in (5). Therefore, the closed-loop system is in the PHS form (6) with

$$J_{cl} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad R_{cl} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. $$

Since $J_{cl}$ has a 1-dimensional kernel space spanned by the vector $[1, 0, -1]^T$, from (12), the Casimir function can be parameterized as $C = K(q - \xi)$ for some function $K : \mathbb{R} \rightarrow \mathbb{R}$. We could use a NN to approximate the function $\Phi()$ and optimize its parameters. Alternatively, we can also manually choose the form of $\Phi()$, which though loss generality but can improve training speed. In this simulation, we select $\Phi = H + H_c + C$.

Under these settings, traditional energy Casimir control method requires to design the functions $K(\cdot)$ and $H_c(\cdot)$ and $\xi^*$, such that the following conditions holds

$$\sin q^* + \frac{\partial K}{\partial \text{input}}|_{q^*-\xi^*} = 0,$$

$$- \frac{\partial K}{\partial \text{input}}|_{q^*-\xi^*} + \frac{\partial H_c}{\partial \xi}|_{\xi^*} = 0,$$

$$\begin{bmatrix} \cos q^* + \frac{\partial^2 K}{\partial \text{input}^2}|_{q^*-\xi^*} & 0 & - \frac{\partial^2 K}{\partial \text{input}^2}|_{q^*-\xi^*} \\ 0 & 1 & 0 \\ - \frac{\partial^2 K}{\partial \text{input}^2}|_{q^*-\xi^*} & 0 & \frac{\partial^2 H_c}{\partial \xi^2}|_{\xi^*} \end{bmatrix} > 0,$$

which are not straightforward to satisfy. Instead, in neural energy Casimir control, we can use NNs to approximate $K$ and $H_c$ and optimize the network parameters to satisfy the above constraints. We use multilayer perception (MLP) NNs to approximate the function $K$ and the controller Hamiltonian $H_c$. Specifically, we use an MLP with 1 input layer, 1 hidden layer, and 1 output layer, where the width of the hidden layer is 64, to approximate $K$. We use an MLP with 1 input layer, 1 hidden layer, and 1 output layer, where the width of the hidden layer is 32, to approximate $H_c$. Moreover, the optimizer is selected to be ADAM (Kingma and Ba, 2014) with a step size $1 \times 10^{-4}$. We then train the NNs by optimizing the cost (15), where $a$ is selected to be $a = 0.5$. The control gain is selected as $D = 5, D_c = 6$. The NN structures and the optimizer step size have been chosen by trial and error. We gradually increase the depth and width of the NNs, and decrease the optimizer step size, until we can obtain a sufficiently small training loss. The choices of $a, D, D_c$ do not matter, as long as they are positive definite 3. After training for 2000 epochs, we let $p = 0, \xi = \xi^*$ and obtain the learned Lyapunov function versus $q$, shown in Fig. 1. We can observe that it achieves the minimum around the desired equilibrium point $q = \frac{\pi}{4}$. Moreover, the learned Lyapunov function is convex when $q \in A = [0, 2]$, which implies the region of attraction is at least $A$ and therefore quite large. Next, we randomly generate 10 initial states, and plot the closed-loop system response in Fig. 2, which shows that the trajectories converge to the desired equilibrium, and demonstrates the effectiveness of the proposed control method.

After training, we obtain the loss as $\epsilon = 0.0050$ and the controller equilibrium as $\xi^* = -0.5693$. Therefore, the upper bound in (21) is $\frac{\epsilon}{\partial \epsilon} = 0.0101$. Moreover, we randomly select

3. The code for the simulation is built on pytorch-lightning and can be accessed at: https://github.com/DecodEPFL/neural_energy_casimir_control

9
Figure 1: Learned Lyapunov function $V_\theta(q, 0, \xi^*)$, where the orange point denotes the value of the Lyapunov function at the desired equilibrium $q = \frac{\pi}{4}$.

1 trajectory from the 10 generated trajectories, and find that the trajectory converges to $([7.9093 \times 10^{-1}, -2.2693 \times 10^{-4}, -5.6329 \times 10^{-1}])$, which is also the minimum $\bar{z}$ of the learned Lyapunov function. Since $z^* = [0.7854, 0.0000, -0.5693]$, we have $\|\bar{z} - z^*\| = 0.0082$. As a result, the bound $\|\bar{z} - z^*\| \leq \frac{\epsilon}{a - \epsilon}$ (see (21)) is verified. Furthermore, we train the NNs under the same setting with different $a$. In Fig. 3, we plot the $\frac{\epsilon}{a - \epsilon}$ and the error $\|\bar{z} - z^*\|$ against different values of $a$. We can observe that in all cases, the bound (21) is verified.

5. Conclusions

We propose a NN-based approach to extend the conventional energy Casimir method for control of PHSs. Our neural energy Casimir method does not require solutions of convoluted PDEs for the equilibrium assignment. We use parameterizations of Casimir functions and provide regularizers penalizing the Jacobian and Hessian of the neural Lyapunov function at the desired equilibrium point to achieve local stability. Moreover, we prove that the difference between the desired and achieved equilibrium point can be bounded in terms of the training loss. Further work will be devoted to analyze the performance of different classes of deep neural networks (DNNs) including Hamiltonian-DNNs (Galimberti et al., 2021) which match the Hamiltonian structure of the system and controller.

4. For the sake of reproducibility, we fixed the random seed before carrying out the simulations.
Figure 2: Closed-loop response under neural Casimir controller, where the dashed line represents the desired equilibrium

Figure 3: The error $\| \bar{z} - z^* \|$ and the bound $\frac{\xi}{a-\epsilon}$ with respect to different values of $a$
Acknowledgments

This research is supported by the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40 180545).

References

Bernard Brogliato, Rogelio Lozano, Bernhard Maschke, and Olav Egeland. Dissipative Systems Analysis and Control: Theory and Applications. Communications and control engineering. Springer London, Limited, London, 2006.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, page 3245–3254. Curran Associates Inc., 2020.

Shaoru Chen, Mahyar Fazlyab, Manfred Morari, George J Pappas, and Victor M Preciado. Learning region of attraction for nonlinear systems. arXiv preprint arXiv:2110.00731, 2021.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. arXiv preprint arXiv:2109.14152, 2021.

Clara Lucía Galimberti, Luca Furieri, Liang Xu, and Giancarlo Ferrari-Trecate. Hamiltonian deep neural networks guaranteeing non-vanishing gradients by design. arXiv preprint arXiv:2105.13205, 2021.

Fernando Gama and Somayeh Sojoudi. Graph neural networks for distributed linear-quadratic control. In Learning for Dynamics and Control, pages 111–124. PMLR, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

John Lee. Introduction to Smooth Manifolds, volume 218 of Graduate texts in mathematics. Springer New York, New York, NY, 2012. ISBN 1441999817.

Romeo Ortega, Arjan Van der Schaft, Bernhard Maschke, and Gerard Escobar. Energy-shaping of port-controlled Hamiltonian systems by interconnection. In Proceedings of the 38th IEEE Conference on Decision and Control, pages 1646–1651. IEEE, 1999.

Romeo Ortega, Arjan Van der Schaft, Fernando Castanos, and Alessandro Astolfi. Control by interconnection and standard passivity-based control of port-Hamiltonian systems. IEEE Transactions on Automatic control, 53(11):2527–2542, 2008.

G. P. Padilla, J. C. Flores Paredes, and M. C. F. Donkers. A port-Hamiltonian approach to complete vehicle energy management: A battery electric vehicle case study. In 2020 American Control Conference (ACC), pages 288–294, 2020.

Patricia Pauli, Dennis Gramlich, Julian Berberich, and Frank Allgöwer. Linear systems with neural network nonlinearity: Improved stability analysis via acausal Zames-Falb multipliers. arXiv preprint arXiv:2103.17106, 2021.
Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In Conference on Robot Learning, pages 466–476. PMLR, 2018. ISBN 2640-3498.

Manuel Schaller, Friedrich Philipp, Timm Faulwasser, Karl Worthmann, and Bernhard Maschke. Control of port-Hamiltonian systems with minimal energy supply. *European Journal of Control*, 62:33–40, 2021.

Felix Strehle, Martin Pfeifer, Albertus Johannes Malan, Stefan Krebs, and Sören Hohmann. A scalable port-Hamiltonian approach to plug-and-play voltage stabilization in DC microgrids. In 2020 IEEE Conference on Control Technology and Applications (CCTA), pages 787–794. IEEE, 2020.

Anastasios Tsolakis and Tamas Keviczky. Distributed IDA-PBC for a class of nonholonomic mechanical systems. *IFAC-PapersOnLine*, 54(14):275–280, 2021.

Arjan van der Schaft. *L2-Gain and Passivity Techniques in Nonlinear Control*. Communications and Control Engineering. Springer International Publishing, Cham, 3rd ed. 2017. edition, 2017.

Arjan van der Schaft and Dimitri Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3):173–378, 2014.

Chengshuai Wu, Arjan van der Schaft, and Jian Chen. Stabilization of port-Hamiltonian systems based on shifted passivity via feedback. *IEEE Transactions on Automatic Control*, 66(5):2219–2226, 2020.

Fengjun Yang and Nikolai Matni. Communication topology co-design in graph recurrent neural network based distributed control. *arXiv preprint arXiv:2104.13868*, 2021.

He Yin, Peter Seiler, and Murat Arcak. Stability analysis using quadratic constraints for systems with neural network controllers. *IEEE Transactions on Automatic Control*, 2021.