Neural Energy Casimir Control for Port-Hamiltonian Systems

EPFL

Liang Xu, Muhammad Zakwan and Giancarlo Ferrari-Trecate
Institute of Mechanical Engineering, EPFL

National Centre of Competence in Research Automation

Motivations

Application of energy Casimir method relies on availability of suitable **Casimir** and **Lyapunov** functions, whose computation are generally **intractable**. The universal approximation capability of neural networks and available machine learning software motivate us to propose an effective **neural network-based framework** to **learn** these Casimir and Lyapunov functions.

Energy Casimir Control

Set-point control of port-Hamiltonian systems (PHSs):

$$\begin{aligned} \dot{\boldsymbol{x}} &= (\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x}))\frac{\partial H(\boldsymbol{x})}{\partial \boldsymbol{x}} + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}, \\ \boldsymbol{y} &= \boldsymbol{G}^\top(\boldsymbol{x})\frac{\partial H(\boldsymbol{x})}{\partial \boldsymbol{x}}, \end{aligned} \qquad (1)$$

- If desired equilibrium $\boldsymbol{x}^*$ is a local minimum of H, due to the passivity $\dot{H}(\boldsymbol{x}) \leq \boldsymbol{y}^\top \boldsymbol{u}$, negative output feedback, i.e., $\boldsymbol{u} = -\boldsymbol{y}$, would stabilize the system.
- However, if $\boldsymbol{x}^*$ is not a local minimum of H, energy Casimir control is needed.

Casimir function: $C(\boldsymbol{x})$ is a Casimir, if

$$\frac{\partial^\top C}{\partial \boldsymbol{x}}(\boldsymbol{x})(\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x})) = 0. \qquad (2)$$

Casimirs can be used to modify Hamiltonian of system (1), therefore reshaping the local minimum of Hamiltonian.

Energy Casimir Control: Consider the PHS controller

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= [\boldsymbol{J}_c(\boldsymbol{\xi}) - \boldsymbol{R}_c(\boldsymbol{\xi})]\frac{\partial H_c}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) + \boldsymbol{G}_c(\boldsymbol{\xi})\boldsymbol{u}_c, \\ \boldsymbol{y}_c &= \boldsymbol{G}_c^\top(\boldsymbol{\xi})\frac{\partial H_c}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}), \end{aligned} \qquad (3)$$

and the negative feedback interconnection

$$\boldsymbol{u} = -\boldsymbol{y}_c + \boldsymbol{v}, \boldsymbol{u}_c = \boldsymbol{y} + \boldsymbol{v}_c, \qquad (4)$$

Lemma 1 (A. van der Schaft, 2017) *If one can find $\boldsymbol{J}_c(\boldsymbol{\xi})$, $\boldsymbol{R}_c(\boldsymbol{\xi})$, $\boldsymbol{G}_c(\boldsymbol{\xi})$, $H_c(\boldsymbol{\xi})$ for the controller* (3)*, a Casimir function $C(\cdot)$ for the closed-loop system, a function $\Phi : \mathbb{R}^2 \to \mathbb{R}$, and a $\boldsymbol{\xi}^*$, such that the Lyapunov function defined by $V = \Phi(H + H_c, C)$ has a local minimum at $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{\xi}^*)$, i.e.,*

$$\frac{\partial V}{\partial \boldsymbol{z}}|_{\boldsymbol{z}^*} = 0, \quad \frac{\partial^2 V}{\partial \boldsymbol{z}^2}|_{\boldsymbol{z}^*} > 0. \qquad (5)$$

Then the auxiliary inputs

$$\begin{aligned} \boldsymbol{v} &= -\boldsymbol{D}\boldsymbol{G}^\top(\boldsymbol{x})\frac{\partial V}{\partial \boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\xi}), \\ \boldsymbol{v}_c &= -\boldsymbol{D}_c\boldsymbol{G}_c^\top(\boldsymbol{\xi})\frac{\partial V}{\partial \boldsymbol{\xi}}(\boldsymbol{x}, \boldsymbol{\xi}) \end{aligned}$$

with $\boldsymbol{D} = \boldsymbol{D}^\top > 0, \boldsymbol{D}_c = \boldsymbol{D}_c^\top > 0$ asymptotically stabilize the closed-loop system to $(\boldsymbol{x}^*, \boldsymbol{\xi}^*)$.

Obstacles in application: No systematic approaches to design parameters and functions appear in Lemma 1.

Main Results

General Idea

If controller structure matrices $\boldsymbol{J}_c(\boldsymbol{\xi}), \boldsymbol{R}_c(\boldsymbol{\xi}), \boldsymbol{G}_c(\boldsymbol{\xi})$ are fixed a prior, use neural networks (NNs) to learn desired functions H_c, Φ, C and controller state $\boldsymbol{\xi}^*$ such that (5) holds.

Training Objective

Denote with $H_{c,\theta_1}, \Phi_{\theta_2}, C_{\theta_3}$ the NN approximations of H_c, Φ and C, respectively. The training targets are

1. C_{θ_3} must be a Casimir function for the closed-loop system, i.e., the following must be satisfied
$$\frac{\partial^\top C_{\theta_3}}{\partial \boldsymbol{z}}(\boldsymbol{z})\left(\boldsymbol{J}_{cl}(\boldsymbol{z}) - \boldsymbol{R}_{cl}(\boldsymbol{z})\right) = 0. \qquad (6)$$
2. V_θ defined as $V_\theta = \Phi_{\theta_2}(H + H_{c,\theta_1}, C_{\theta_3})$ must have a local minimum at $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{\xi}^*)$ for some $\boldsymbol{\xi}^*$, i.e., the following must be satisfied
$$\frac{\partial V_\theta}{\partial \boldsymbol{z}}|_{\boldsymbol{z}^*} = 0, \quad \frac{\partial^2 V_\theta}{\partial \boldsymbol{z}^2}|_{\boldsymbol{z}^*} > 0. \qquad (7)$$

To satisfy (6), one could grid the region of interest and require (6) holds on all grid points. However, the computational complexity is high, only approximate Casimir will be learned and stability cannot be rigorously guaranteed.

Reduce Training Complexity by Casimir Parameterization

Suppose $\boldsymbol{J}_{cl}$ and $\boldsymbol{R}_{cl}$ are constant. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ be the basis of $\ker(\boldsymbol{J}_{cl} - \boldsymbol{R}_{cl})$. Then

$$\frac{\partial C}{\partial \boldsymbol{z}}(\boldsymbol{z}) = \sum_{i=1}^{r} \alpha_i(\boldsymbol{z})\boldsymbol{v}_i \qquad (8)$$

for some scalar functions $\alpha_i(\boldsymbol{z})$. A candidate function C therefore is

$$C(\boldsymbol{z}) = K\left(\sum_{i=1}^{r} \beta_i(\boldsymbol{z}^\top \boldsymbol{v}_i)\right) \qquad (9)$$

for some scalar functions $K(\cdot), \beta_i(\cdot)$. As a result, the constraint (6) is inherently satisfied and one only needs to minimize a loss function measuring the violation of (7).

Loss Function Minimization

Use neural networks $K_{\theta_3}, \beta_{i,\theta_{i+3}}$ to approximate K and β_i. Construct the neural Lyapunov function as

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

and solve the training problem

$$\min_{\theta_1,\ldots,\theta_{r+3},\boldsymbol{\xi}^*} \left\|\frac{\partial V_\theta(\boldsymbol{z})}{\partial \boldsymbol{z}}|_{\boldsymbol{z}^*}\right\| + \mathrm{ReLU}\left(-\lambda_{\min}\left(\frac{\partial^2 V_\theta(\boldsymbol{z})}{\partial \boldsymbol{z}^2}|_{\boldsymbol{z}^*} - a\boldsymbol{I}\right)\right), \qquad (10)$$

where $a\boldsymbol{I}$ with $a > 0$ is to promote local convexity of V_θ.

Performance Guarantees

Suppose the loss in (10) after training is ϵ. When ϵ is sufficiently small, the distance between the local minimum $\bar{z}$ of V_θ and the desired equilibrium $\boldsymbol{z}^*$ is also small, satisfying

$$\|\bar{z} - \boldsymbol{z}^*\| \leq \frac{\epsilon}{a - \epsilon}, \qquad (11)$$

that is, **the control bias decreases linearly w.r.t. training loss**.

Simulation Result

Set-point control of pendulum system

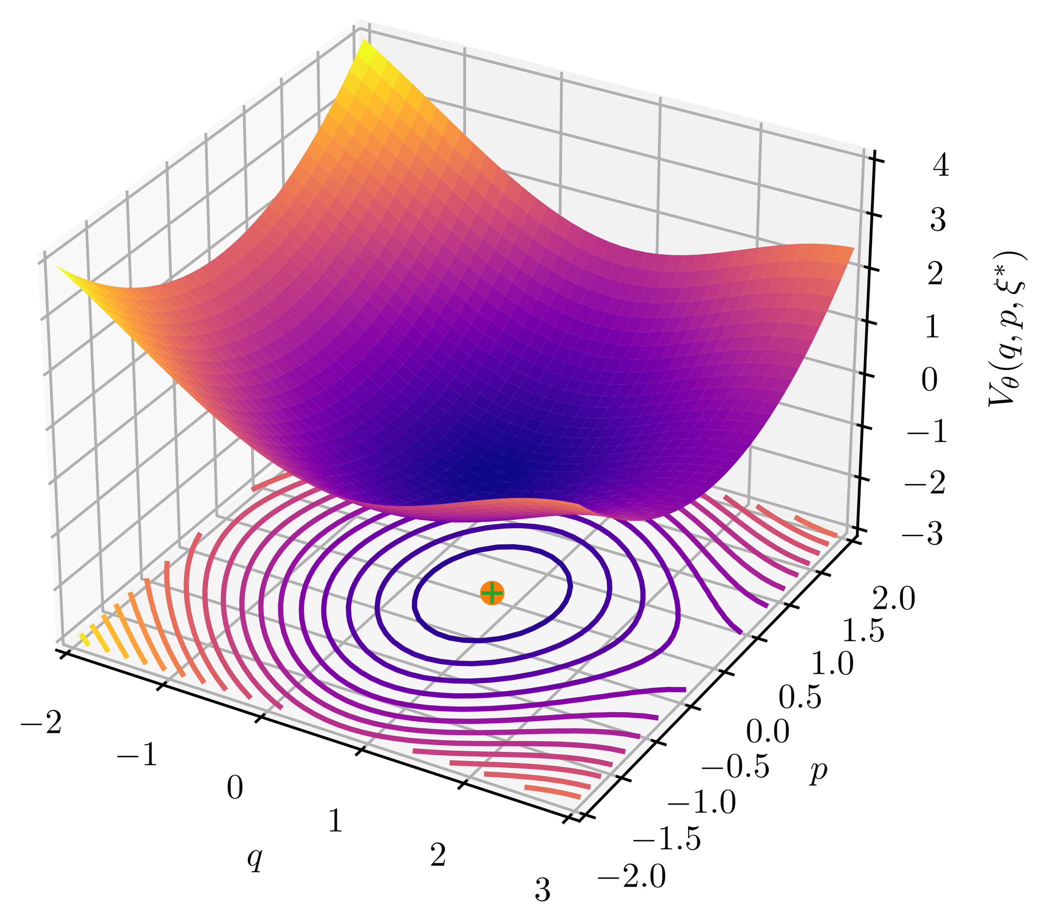

Fig. 1 Learned Lyapunov function $V_\theta(q, p, \xi^*)$

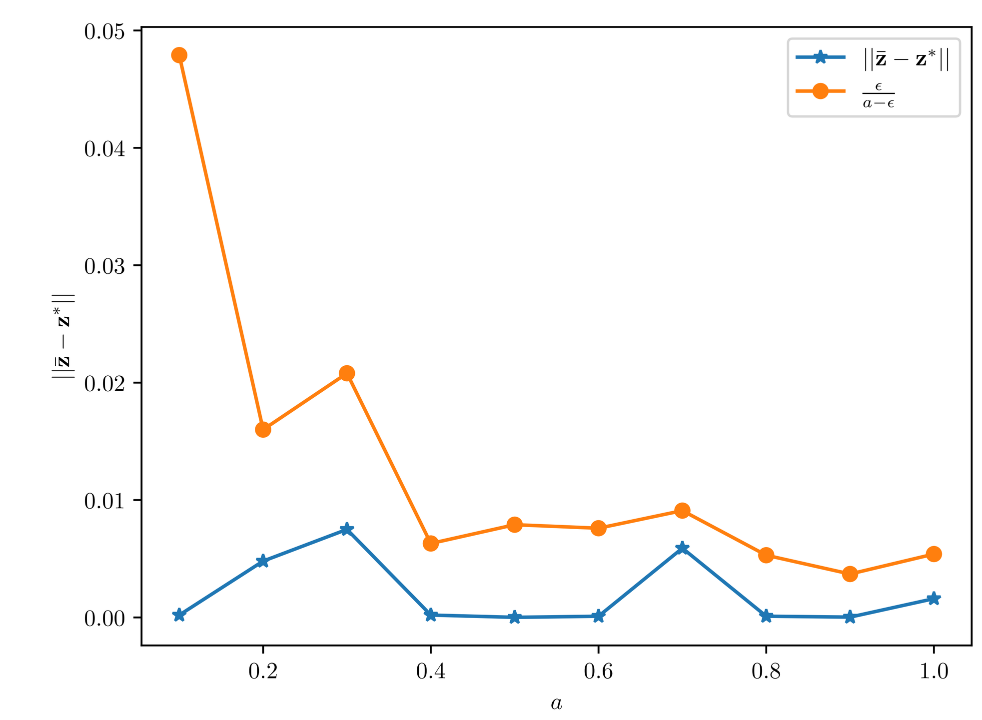

Fig. 2 The error $\|\bar{z} - \boldsymbol{z}^*\|$ and the bound $\frac{\epsilon}{a-\epsilon}$ for different values of a

Conclusion

- Propose an NN-based approach to facilitate the design of energy Casimir control. Do not require solving convoluted PDEs for equilibrium assignment. The difference between the desired and achieved equilibrium point can be bounded in terms of the training loss.
- Further work will be devoted to extending the proposed framework to other controller design procedures for PHSs.