Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-30 (2025) 317–322

A PINN-Based Approach to Solving the Matching Condition for Energy Shaping Control in Bipedal Locomotion ⋆

Angelos Guan * and **Ge Lv** *

* *Department of Mechanical Engineering, Clemson University, Clemson, SC 29634 USA*
(e-mail: tingrug@clemson.edu, glv@clemson.edu).

Abstract:
We propose a Physics-Informed Neural Network (PINN) framework to solve the nonlinear matching condition - a high-dimensional, first-order partial differential equation (PDE) that forms the core bottleneck to real-time energy shaping control in underactuated locomotion. This PDE is analytically intractable and computationally expensive to approximate numerically previously. We introduce a tailored PINN model design and a new alpha-masking technique that enforces boundary conditions without requiring loss balancing. To our knowledge, this is the first PINN-based controller validated in closed-loop on a biped, achieving stable, energy-efficient gait generation and scalable control design.

Keywords: Robotics, Nonlinear Control Systems, Machine Learning in Modeling, Estimation, and Control, Control of Biomechanical Systems, Control Design

1. INTRODUCTION

Energy shaping control reshapes the energy landscape of a dynamical system to achieve stable closed-loop behaviors with benefits. Common frameworks include the Controlled Lagrangian method (Bloch et al. (2000)) and Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) Ortega et al. (2002). Energy shaping has been widely adopted in robotics, particularly in bipedal locomotion and wearable exoskeletons (Holm and Spong (2008); Lin et al. (2019); Lv et al. (2021)), enabling model-based controllers with improved robustness to underactuation and application-based variability in bipedal devices.

A key step in implementing energy shaping control is solving the matching condition Blankenstein et al. (2002), a set of nonlinear partial differential equations (PDEs) that dictates the achievable energy landscapes for the closed-loop dynamics. While the matching condition is trivially solved for fully-actuated systems, it becomes significantly more challenging to solve for underactuated systems, sometimes even infeasible for high-dimensional systems with varying degrees of underactuation. As a result, energy shaping control has mainly been applied to simplified low degree of freedom (DoF) systems where closed-form solutions are proved to be feasible as in (Holm and Spong (2008)).

The most common strategy to solving the matching condition is to derive global analytical solution, local linearization, or algebraic closed-form solution (Nunna et al. (2015)). One popular approach is to simplify the nonlinear matching condition to a set of quadratic and linear PDEs (Auckly and Kapitanski (2002)), which was built upon by much subsequent research. Suffering from low scalability, such methods are often limited to low-dimensional, simplified systems. Alternative analytical approaches have been explored in (Ryalat and Laila (2016)), where the authors derived closed-form solutions by structuring the mass matrices to cancel out complex terms in the matching condition. Similarly, (Harandi and Taghirad (2021)) also derived analytical solutions for kinetic energy shaping for underactuated systems based on defining special forms for mass matrix, with only one state-related variable, making this approach inflexible compared with more expressive methods. A recent variant (Lin et al. (2024)) uses analytical solutions as basis functions and learns a weighted combination using human torque data via convex optimization. While this adds flexibility and a data-driven aspect, it still depends on the existence of analytical solutions and access to human joint kinematics data.

Numerical solvers have demonstrated potentials in solving the PDEs via approximating the matching condition into ordinary differential equations (Lv et al. (2020)). However, this approach requires an explicit, parameterized formulation and only provides solution at selected sample points. Real-time implementation of this approach will rely on interpolation, which reduces accuracy and requires training models like Gaussian Process Regression to approximate the interpolations. Reinforcement learning (RL) has been applied to solve matching condition numerically (Gheibi et al. (2020)). However, RL-based solvers are usually data-hungry, require long training, and need intensive hyper-parameter tuning to function. In addition, model-free RL

⋆ This work was supported by the National Science Foundation under Award 2340261. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

10.1016/j.ifacol.2025.12.256

methods cannot restrict or guarantee the solution structure explicitly. Lastly, RL models may struggle to generalize to unseen states if the training does not sufficiently explore the state space. More recently, the potential idea of neural-network-based solution has emerged, but they were only explored in theory within the IDA-PBC dynamics (Sanchez-Escalonilla et al. (2024); Massaroli et al. (2022)).

To overcome limitations of previous approaches, we propose a novel framework that uses Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) to solve the matching condition for energy shaping control. Our framework leverages a neural network to learn a state-dependent transformation matrix that defines the shaped mass matrix and thus the kinetic energy in the closed loop. Our approach addresses PINN training challenges by using a tailored network structure, novel loss functions, and training optimizations—ensuring physical and boundary constraints, enhancing robustness and convergence, and preserving interpretability over other learning-based methods. Unlike traditional methods requiring strong assumptions and closed-form derivations, our approach directly learns smooth and stable solutions with fewer constraints. Compared to prior numerical solutions such as (Zheng and Lv (2023)), our approach greatly reduced the computation time from hours to centiseconds, allowing real-time computation of high accuracy solutions. In comparison to RL-based solvers such as (Gheibi et al. (2020)), our approach converges much faster with significantly fewer training samples required, while allowing structural constraint, enhancing stability and generalization. Most prior studies (Holm and Spong (2008); Lin et al. (2024)) rely on known solutions or supervised data. Compared to hybrid methods, our framework removes reliance on analytic solutions and supervised data, enabling generalization to diverse underactuated systems, while its modular design supports scalability to high-DoF systems, user-defined control objectives, and integration with ground-truth data when available.

2. ENERGY SHAPING CONTROL

We propose a general-purpose PINN-based energy shaping controller for Euler-Lagrange systems, formulated without a fixed control objective. Designed for inner-loop control, it scales to systems with arbitrary DoFs and underactuation, and integrates seamlessly into task-specific optimizations.

2.1 Energy Shaping Control

The Lagrangian of a general dynamical system is given as

$$L(q,\dot{q}) = T(q,\dot{q}) - V(q) = \frac{1}{2}\dot{q}^T M\dot{q} - V(q), \tag{1}$$

where $q \in \mathbb{R}^n$ is the generalized coordinates vector and $\dot{q} \in \mathbb{R}^n$ is its time derivative, $T(q,\dot{q})$ is the kinetic energy, $M \in \mathbb{R}^{n\times n}$ is the positive-definite mass matrix, and $V(q)$ is the potential energy. Based on (1), the equation of motion can be expressed as

$$M\ddot{q} + C\dot{q} + N = Bu, \tag{2}$$

where $C \in \mathbb{R}^{n\times n}$ is the Coriolis/centrifugal matrix, $N \in \mathbb{R}^{n\times 1}$ is the gravitational force vector, and $B \in \mathbb{R}^{n\times m}$ maps the control law $u \in \mathbb{R}^m$ into the overall dynamics. Now consider a new Lagrangian $\tilde{L}(q,\dot{q})$ defined as

$$\tilde{L}(q,\dot{q}) = \frac{1}{2}\dot{q}^T \tilde{M}\dot{q} - \tilde{V}(q), \tag{3}$$

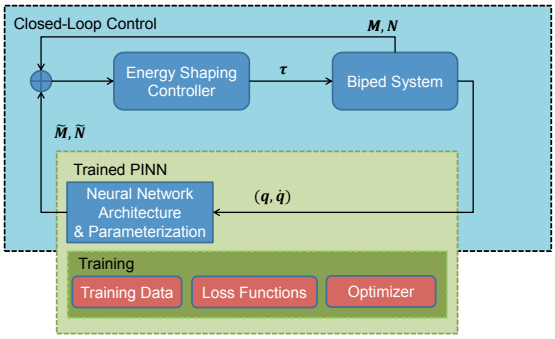


Fig. 1. PINN-based energy shaping control pipeline

which leads to a new equation of motion as

$$\tilde{M}\ddot{q} + \tilde{C}\dot{q} + \tilde{N} = 0, \tag{4}$$

where $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ are defined similar to the matrices in (2). The two systems (2) and (4) are said to match if there exists a control law u that brings (2) into (4). This can be achieved if and only if there exists a full-rank left annihilator $B^\perp \in \mathbb{R}^{(n-m)\times n}$ of B such that matching condition holds, i.e.,

$$B^\perp\left[C\dot{q} + N - M\tilde{M}^{-1}(\tilde{C}\dot{q} + \tilde{N})\right] = 0. \tag{5}$$

When the system is fully-actuated, the matching condition (5) is trivially satisfied for all $\tilde{M}$ and $\tilde{N}$. However, when $m < n$, (5) only admits a limited set of solutions for $\tilde{M}$ and $\tilde{N}$. Once the matching condition is satisfied, the control law u that shapes (2) into (4) is given as:

$$u = (B^TB)^{-1}B^T\left[C\dot{q} + N - M\tilde{M}^{-1}(\tilde{C}\dot{q} + \tilde{N})\right]. \tag{6}$$

2.2 PINN Integration in Closed-Loop Control

We approximate the shaped matrices $\tilde{M}$ and $\tilde{N}$ using a PINN denoted by $\mathcal{N}^{(M)}_{\theta,\dot{q}}(q)$ and $\mathcal{N}^{(N)}_{\theta,\dot{q}}(q)$ for their corresponding weights respectively, i.e.,

$$\tilde{M} = \mathcal{N}^{(M)}_{\theta,\dot{q}}(q), \quad \tilde{N} = \mathcal{N}^{(N)}_{\theta,\dot{q}}(q), \tag{7}$$

where θ denotes the learned weights and parameters of the neural network, and $\dot{q}$ represents that the derivatives of general coordinates were part of the input to the PINN. The framework supports shaping kinetic energy-only ($\tilde{N} = N$), potential energy-only ($\tilde{M} = M$), or both, with different versions of output layer in the unified pipeline. Unlike traditional formulations where M and N depend solely on joint positions, our shaped matrices learned inductive bias on joint velocities as well, which enables adaptive shaping based on the full state $(q,\dot{q})$.

Fig. 1 illustrates the overall structure of our framework. PINN is trained following the procedure in Sec. 3. Once trained, the PINN outputs the shaped energy matrices $\tilde{M}$ and $\tilde{N}$ based on the current state in the control loop, which are used to compute the control torque u. This structure applies the learned neural network from offline learning to online control, preserving interpretability and enforcing physical constraints during deployment.

2.3 Coriolis Matrix Computation

Calculating the Coriolis matrix $C \in \mathbb{R}^{n\times n}$ is often a computational bottleneck, with direct derivation often infeasible. In this paper, we opted to use an analytical

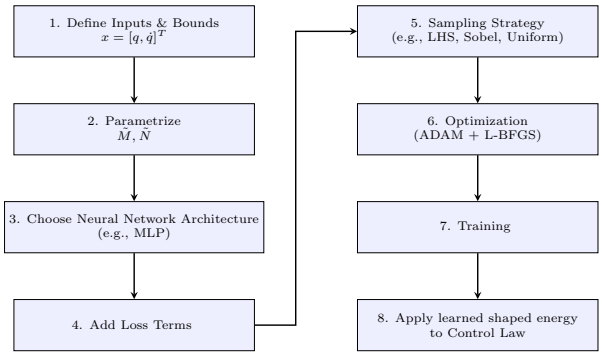


Fig. 2. The proposed PINN structure and workflow.

solution for the Coriolis matrix using structure-aware learning parameterization similar to Ortega et al. (2002); Zheng and Lv (2023). Generally, the shaped mass matrix $\tilde{M}$ can be framed as a function of the original mass matrix M and the neural network output $\mathcal{N}_\theta(q, \dot{q})$, e.g.,

$$\tilde{M} = f(M, \mathcal{N}_\theta(q, \dot{q})). \tag{8}$$

The matrix $\tilde{C}$ can be derived analytically based on the function f and the known form of M, bypassing explicit partial gradient computation in the control loop.

3. PHYSICS-INFORMED NEURAL NETWORK

3.1 General Framework

The proposed PINN utilizes neural networks as universal function approximators, approximating solutions to PDEs by embedding the governing physics into the loss function during training (Raissi et al. (2019)). PINNs have recently been theoretically introduced to energy shaping under the IDA-PBC structure in (Plaza et al. (2022)). In our approach, a neural network takes input joint states $(q, \dot{q})$, and is trained to predict the shaped mass and potential terms $\tilde{M}$ and $\tilde{N}$, enabling real-time control without requiring analytical form of the shaped matrices.

Fig. 2 summarizes our training and deployment pipeline. The process begins by defining the input domain and selecting a parametrization for the shaped matrices. Random training samples are drawn (Raissi et al. (2019)) using Latin Hypercube Sampling (LHS) or Sobol sampling across the input domain based on the achievable ranges of joint configurations. The model is trained with selected losses and optimizers such as Adaptive Moment Estimation (ADAM) (Kingma and Ba (2017)) and Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) (Liu and Nocedal (1989)) optimization algorithms.

3.2 Loss Functions

Only residual and boundary losses are theoretically essential. Others aid convergence and are task-specific. The boundary loss can be omitted when using the proposed alpha-masking technique. Bounds are selected to balance model expressiveness and training stability; users may adjust the proposed loss functions or add custom losses for desired control behavior. The overall total loss is given as:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{k} w_i \mathcal{L}_i, \tag{9}$$

$$w_i = \text{clip}\left(\frac{1}{1 + \log(1 + \mathcal{L}_i + \epsilon)},\ 0.5,\ 5.0\right), \tag{10}$$

where k is the total number of loss terms, ϵ is a small number added for numerical stability, and each loss $\mathcal{L}_i$ is balanced with a logarithmic weighting scheme clamped between 0.5 and 5.0.

1. Residual Loss:

$$\mathcal{L}_{\text{res}} = \frac{1}{N}\sum_{i=1}^{N} ||B^{\perp}[(C_i\dot{q}_i + N_i) - M_i\tilde{M}_i^{-1}(\tilde{C}_i\dot{q}_i + \tilde{N}_i)]||_2,$$

$$C\dot{q} = D_q(M\dot{q})\dot{q} - \frac{1}{2}\nabla_q^T\left(\dot{q}^T M\dot{q}\right), \tag{11}$$

where N is the number of training samples, and the subscript i indicates the dynamic terms at the i-th sample state $(q_i, \dot{q}_i)$. The Coriolis term $C\dot{q}$ is computed from the mass matrix with $D_q(\cdot)$ denoting the Jacobian with respect to q, and ∇_q is the gradient operator.

2. Control Loss:

$$u_i = ||B^{\dagger}[(C_i\dot{q}_i + N_i) - M_i\tilde{M}_i^{-1}(\tilde{C}_i\dot{q}_i + \tilde{N}_i)]||_2, \tag{12}$$

$$\mathcal{L}_{\text{control}} = \frac{1}{N}\sum_{i=1}^{N} \text{Softplus}(u_i - u_{\text{bound}})^2, \tag{13}$$

where u_i denotes the predicted control torque at the i-th sample state and $B^{\dagger} := (B^TB)^{-1}B^T$ denotes the left pseudo-inverse of B. The control loss $\mathcal{L}_{\text{control}}$ penalizes torque magnitudes exceeding a chosen bound through a smooth, differentiable quadratic penalty. The Softplus function is defined as $\text{Softplus}(x) = \ln(1 + e^x)$, ensuring differentiability near the threshold.

3. Boundary Loss:

$$\mathcal{L}_{\text{bnd}} = \|\tilde{M}(x_0) - M(x_0)\|_F, \tag{14}$$

where $x_0 = (q_0, \dot{q}_0)^T$ denotes the initial state. The loss penalizes the difference between predicted mass matrix $\tilde{M}(x_0)$ and $M(x_0)$ using the Frobenius norm (subscript F) to enforce exact matching at this boundary point.

4. Deviation Loss:

$$\mathcal{L}_{\text{dev}} = \frac{1}{N}\sum_{i=1}^{N} \text{Softplus}\left(\max(D_i - \text{bound}, 0)\right)^2, \tag{15}$$

$$D_i = \frac{\left\|\tilde{M}(x_i) - M(x_i)\right\|_F}{\|M(x_i)\|_F + \epsilon}, \tag{16}$$

where $x_i = (q_i, \dot{q}_i)^T$ is the i-th state, and D_i is the relative deviation between shaped and original mass matrices with ϵ for numerical stability. This loss penalizes relative deviations above bound using a Softplus-smoothed quadratic term.

5. Eigenvalue Positivity Loss:

$$\mathcal{L}_{\text{eig\_pos}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} \text{Softplus}\left(-\lambda_j(\tilde{M}(x_i))\right)^2, \tag{17}$$

where λ_j is the eigenvalue of $\tilde{M}(x_i)$, k being the total number of eigenvalues. This loss penalizes negative eigenvalues to ensure positive definiteness of the mass matrix.

6. Eigenvalue Range Loss:

$$\mathcal{L}_{\text{eig\_range}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\big(\text{Softplus}(a_i - \lambda_j(\tilde{M}(x_i))) + \text{Softplus}(\lambda_j(\tilde{M}(x_i)) - b_i)\big). \quad (18)$$

This loss (18) bounds the eigenvalues of $\tilde{M}(x_i)$ within a scaled range $[a_i, b_i]$ based on those of $M(x_i)$, where $[a_i, b_i] = [(1-\alpha)\lambda_{\min},\ (1+\alpha)\lambda_{\max}]$ with $\lambda_{\min}$ and $\lambda_{\max}$ being the minimum and maximum eigenvalues of $M(x_i)$.

7. Sparse Sample Loss: Sparse Sample Loss $\mathcal{L}_{\text{sparse}}$ uses the same equation for residual loss (11) to compute mean residual loss over a small batch of randomly sampled data resampled at each epoch. This loss helps mitigate overfitting towards main training data.

3.3 Alpha-Masking for Embedding Boundary Condition

Conventional PINNs impose boundary conditions via an explicit boundary loss term where the boundary domain is overly sampled to match the general sample for computing the residuals. Training becomes unstable when the boundary consists of few or only a single point, as is often the case in robotic systems. To address this imbalance, we propose an *α-masking* mechanism that embeds the boundary condition into the model architecture, eliminating the need for boundary loss balancing, by constructing a function that satisfies or approximates

$$\alpha(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{otherwise} \end{cases} \quad (19)$$

To construct the shaped matrix mass, we choose:

$$\alpha(x) = (\|x\|_2 > \varepsilon), \quad \varepsilon = 10^{-6} \quad (20)$$

$$\tilde{M}(x) = M(x_0) + \alpha(x)\cdot f(M(x), \mathcal{N}_\theta(x)). \quad (21)$$

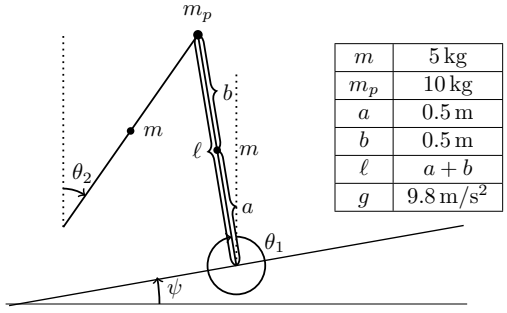
In (20), $f(M(x), \mathcal{N}_\theta(x))$ denotes the original shaped mass matrix output from neural network parametrization. By construction, this enforces $\tilde{M}(x_0) = M(x_0)$, satisfying the boundary constraint exactly without additional loss terms. While an alternative strategy is to scale the weight w_i for boundary loss (14) by the number of training samples N, we found that the alpha-masking mechanism converges more smoothly and guarantees zero boundary error by design. Both methods produce nearly identical torque profiles and shaped dynamics under identical training parameters, but alpha-masking avoids turbulent early training and loss weight tuning. It can be naturally extended to multiple discrete boundary points by constructing smooth selector functions. This approach is best suited for sparse, structured boundary manifolds.

4. SIMULATIONS AND DISCUSSION

In this section, we detail the specific implementation of kinetic energy shaping on a two-DoF compass gait biped model as a proof of concept study for validation purposes.

4.1 Simulation Setup

We chose the two-DoF biped model from Holm and Spong (2008) (Fig. 3) for simulation purposes, with its controller serving as our benchmark for comparison. We chose this model for validation given its prevalence in verifying control efficacy for bipedal walking and exoskeletons. The biped consists of two rigid links with mass m and length $\ell = a+b$, with upper body and hip mass lumped together as m_p. The configuration vector $q = [\theta_1, \theta_2]^T \in \mathbb{R}^2$ includes absolute angles of the stance and swing legs from the vertical, with corresponding velocities $\dot{q} \in \mathbb{R}^2$. For this biped model, we apply kinetic energy shaping as a case study (i.e., $\tilde{N} = N$) using a learned matrix (to be defined in Sec. 4.2.) The shaped mass matrix $\tilde{M}$ was inferred from the network, while the Coriolis term is computed analytically from $\tilde{M}$ and (11). For implementation details, see footnote [1].



m	5 kg
m_p	10 kg
a	0.5 m
b	0.5 m
ℓ	$a+b$
g	9.8 m/s^2

Fig. 3. The two-DoF compass-gait biped model.

4.2 PINN Models Result Comparison

We adopted a standard multilayer perceptron architecture [2] with two hidden layers (width 1024), `tanh` activations, and layer normalization, which empirically outperformed alternatives we tested such as residual networks, convolutional networks and transformers, likely due to the lack of spatial or sequential structure in the input. Train loss exhibited fast exponential decay when boundary conditions were hard-coded using the proposed alpha-masking method, stabilizing near zero by epoch 1000. In contrast, models trained with boundary loss showed sharp fluctuations before epoch 100 but followed a similar decay trend thereafter. We compared five PINN-based controllers trained on the same input states. **KMK**, our baseline, uses $\tilde{M} = K^T M K$ with diagonal $K(q, \dot{q}) = \text{diag}(k_1(q, \dot{q}), k_2(q, \dot{q})) \in [0.7, 1.3]$, ensuring positive definiteness of $\tilde{M}$ because M is positive-definite. **KMKhb** uses the same form but replaces the boundary loss with alpha-masking (Sec. 3.3) to enforce $\tilde{M}(0) = M(0)$. **KMKrw** reduces hidden width from 1024 to 512 and **KMKik** increases bounds to $[0.5, 1.5]$, both exhibit overfitting, leading to suboptimal control, highlighting the importance of parameter tuning in neural networks. **KM** uses $K(q, \dot{q}) = \begin{bmatrix} k_1(q,\dot{q}) & k_2(q,\dot{q}) \\ k_2(q,\dot{q}) & k_3(q,\dot{q}) \end{bmatrix}$ in $\tilde{M} = K \odot M$, which does not inherently preserve positive definiteness, thus requiring positive eigenvalue loss(17). **KMK** and **KMKhb** yield the best residual accuracy and generalization; alpha-masking improves convergence and guarantee boundary enforcement. Stability is assessed via eigenvalues of the linearized Poincaré map (Westervelt et al. (2018)).

[1] Simulations were conducted in `MATLAB`, with controlled trajectory computed using `ode45`. The trained PINNs were developed in PyTorch, exported via ONNX, and integrated into the `MATLAB` control loop using the Deep Learning Toolbox.

[2] The model was trained on 5000 LHS-sampled input states using the ADAM optimizer for 2000 epochs with cosine learning rate warmup (20 epochs), final learning rate 10^{-3}, and L_2 regularization 10^{-6}. $u_{\text{bound}} = 40$ Nm, deviation bound = 0.5.

Table 1 reports the final training losses. "N/A" under $\mathcal{L}_{\text{eig\_pos}}$ indicates models that are inherently positive semi-definite and do not require positive eigenvalue regulation. Table 2 reports the average and maximum residual losses for each model over the unseen test sets. **1Step** and **10Step** are simulated trajectories of one and ten gait cycles under an existing energy shaping controller from (Holm and Spong (2008)). **LHS**, **Uni**, and **Sobel** are samples over $(q, \dot{q}) \in \mathbb{R}^4$ under their possible ranges, generated via LHS, uniform sampling, and Sobol sequences, respectively. These sets assess performance comprehensively over input domain. **KMK** and **KMKhb** consistently achieve the lowest errors, while **KMKik** generalizes poorly. Nonzero residual loss in the tables reflects approximation error from model uncertainty, similar to using nominal models in model-based control to approximate true system dynamics.

4.3 Result Comparison

We compare the controller performance based on closed-dynamics learned from the following strategies: **KMK**, **KMKhb**, and **KMKrw** (as defined in Sec. 4.2), against **Passive** (no control) and two analytic baseline controllers **H-6** and **H6** adopted from (Holm and Spong (2008)), with gain $k = -6$ and $k = 6$, respectively. Controller performance is assessed using six key metrics, chosen to reflect energy efficiency, control effort, and gait characteristics. Fig. 4 (left, center) shows average energy expenditure over a gait cycle and total control energy as functions of step velocity. Fig. 4 (right) and Fig. 5 (left) show control torques and their ranges across the gait cycle, measuring smoothness and control effort. Gait characteristics are evaluated by step length versus velocity in Fig. 5 (center) and phase portraits of each joint in Fig. 5 (right), which provide insight into gait shape and stability. Figs. 4 and 5 show that PINN-based controllers achieve comparable or better performance without manual tuning of parameters like k, highlighting their flexibility and effectiveness. The normalized energy expenditure metric used in Fig. 4 (left) is defined as (Martin and Schmiedeler (2014)):

$$E_{\text{metric}} \approx \frac{\sum_{i=1}^{N_T} u^2(i)\,\Delta t(i)}{T[(2m+m_p)gl]^2}, \tag{22}$$

Table 1. Final Trained Losses

Result	KMK	KMKhb	KMKrw	KM	KMKik
Stability	Yes	Yes	Yes	Yes	No
$\mathcal{L}_{\text{res}}$	7.66e-4	5.07e-4	1.07e-3	2.06e-3	3.22e-1
$\mathcal{L}_{\text{control}}$	0.0	0.0	0.0	0.0	0.0
$\mathcal{L}_{\text{bnd}}$	3.0e-7	0.0	1.5e-6	9.0e-7	3.79e1
$\mathcal{L}_{\text{dev}}$	2.55e-1	2.55e-1	2.55e-1	2.50e-1	6.73e-1
$\mathcal{L}_{\text{eig\_range}}$	3.07e-1	3.07e-1	3.07e-1	3.14e-1	2.93e-1
$\mathcal{L}_{\text{sparse}}$	5.0e-4	9.0e-4	2.0e-3	1.9e-3	3.46e-1
$\mathcal{L}_{\text{eig\_pos}}$	N/A	N/A	N/A	2.13e-1	N/A

Table 2. Residual Losses on Test Set

Result	KMK	KMKhb	KMKrw	KM	KMKik
LHS avg	1.02e-3	6.25e-4	1.25e-3	2.21e-3	0.334
LHS max	7.11e-2	8.31e-2	6.87e-2	0.112	2.54
Sobel avg	1.01e-3	6.59e-4	1.39e-3	2.21e-3	0.329
Sobel max	7.25e-2	8.64e-2	7.59e-2	0.118	2.47
Uni avg	1.10e-3	6.74e-4	1.36e-3	2.18e-3	0.333
Uni max	7.98e-2	6.58e-2	7.57e-2	9.77e-2	2.58
1Step avg	1.17e-3	3.26e-4	2.97e-3	1.63e-3	0.623
1Step max	5.32e-3	3.94e-3	2.45e-2	8.70e-3	1.53
10Step avg	1.23e-3	3.55e-4	2.61e-2	1.67e-3	0.638
10Step max	6.32e-3	4.61e-3	3.17e-3	1.16e-2	1.59

where g is the gravity constant, T is the step duration, and N_T the number of sample steps within a step.

4.4 Discussion

The proposed PINN-based controllers (**KMK**, **KMKhb**, **KMKrw**) achieve comparable or better performance than the analytic shaping controllers (Holm and Spong (2008)) (**H-6**, **H6**) for the selected k values used in their formulation. **KMK** and **KMKhb**, which only differ in how the boundary is enforced, achieve nearly identical learned mass matrices, phase portraits, and torque profiles, demonstrating robustness and convergence toward optimal solution for the matching condition. Simulation shows the learned controllers recover from varied initial states and remain stable under small perturbations, demonstrating practical viability. **KMKrw**, with reduced layer width, still achieves comparable gait patterns but incurs higher energy costs, highlighting a tradeoff between model simplicity and performance. All three PINN-based models enable stable control and energy-efficient gaits on par with analytical baselines. In contrast, **KMKik** fails to stabilize the system, suffering from high residual loss and training stagnation due to overly wide output bounds. This demonstrates the importance of structural design and parameter tuning for learning physically meaningful dynamics.

Although the framework extends to arbitrary DoF, high-dimensional PINNs are known to introduce numerical challenges. Matrix inverses in matching condition introduce strong nonlinearity, leading to vanishing gradients and instability, especially with near-singular matrices. This limitation remains underexplored in PINN literature and is the focus of our ongoing work on improving training robustness for high dimensional systems.

5. CONCLUSION AND FUTURE WORK

We present a PINN-based framework for learning closed-loop dynamics for energy shaping control, addressing the core limitations of prior methods requiring analytic solutions or data-driven approximations. When applied to a two-DoF biped model, our controllers generated stable, energy-efficient gaits comparable to conventional methods, its lower computational time compared to numerical solvers demonstrates feasibility for real-time use. Current limitations include numerical instability from near-singular mass matrices and inference latency in higher dimension. Future work will focus on accelerating computation via model compression, improving training stability for ill-conditioned matrices, and enhancing robustness to disturbances through adversarial training and human-in-the-loop optimization for real world implementation.

REFERENCES

Auckly, D. and Kapitanski, L. (2002). On the λ-equations for matching control laws. *SIAM J. Control Optim.*, 41(5), 1372–1388.

Blankenstein, G., Ortega, R., and Van Der Schaft, A.J. (2002). The matching conditions of controlled lagrangians and ida-passivity based control. *Int. J. Control*, 75(9), 645–665.

Bloch, A., Leonard, N., and Marsden, J. (2000). Controlled lagrangians and the stabilization of mechanical systems. i. the first matching theorem. *IEEE Trans. Autom. Control*, 45(12), 2253–2270.

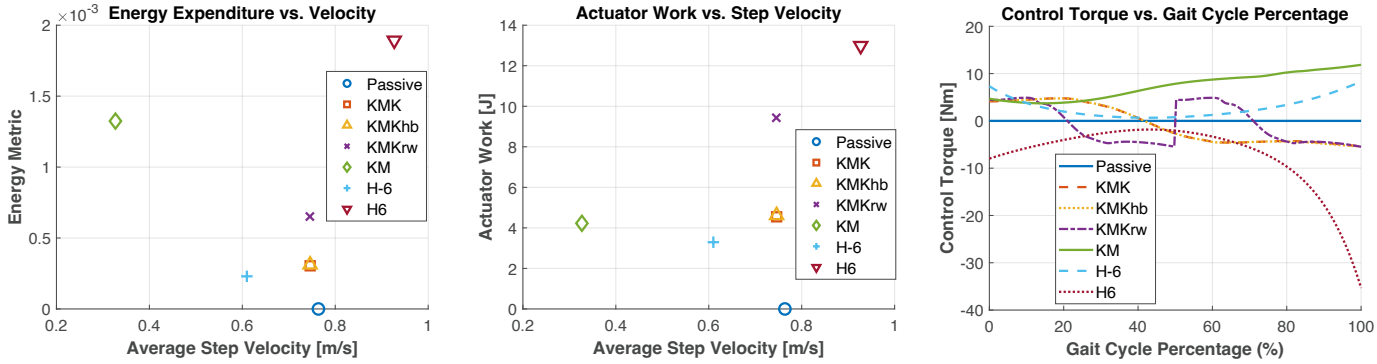


Fig. 4. **Left**: average energy expenditure, **center**: actuator work over velocity, **right**: torque profiles of different controllers.

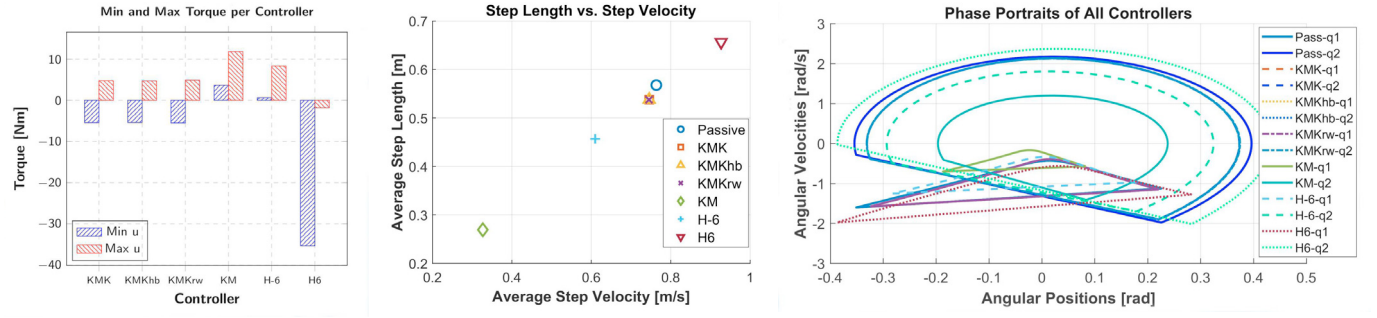


Fig. 5. **Left**: torque range, **center**: step length vs. step velocity, **right**: phase portraits with different controllers.

Gheibi, A., Ghiasi, A.R., Ghaemi, S., and Badamchizadeh, M.A. (2020). Interconnection and damping assignment control based on modified actor–critic algorithm with wavelet function approximation. *ISA Trans.*, 101, 116–129.

Harandi, M.R.J. and Taghirad, H.D. (2021). On the matching equations of kinetic energy shaping in ida-pbc. *J. Franklin Inst.*, 358(16), 8639–8655.

Holm, J.K. and Spong, M.W. (2008). Kinetic energy shaping for gait regulation of underactuated bipeds. In *Proc. of the 2008 IEEE Int. Conf. on Control Appl.*, 1232–1238.

Kingma, D.P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Lin, J., Divekar, N., Lv, G., and Gregg, R.D. (2019). Energy shaping control with virtual spring and damper for powered exoskeletons. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 3039–3045. IEEE.

Lin, J., Thomas, G.C., Divekar, N.V., Peddinti, V., and Gregg, R.D. (2024). A modular framework for task-agnostic, energy shaping control of lower limb exoskeletons. *IEEE Trans. Control Syst. Technol.*, 32(6), 2359 – 2375.

Liu, D.C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1), 503–528.

Lv, G., Lin, J., and Gregg, R.D. (2021). Trajectory-free control of lower-limb exoskeletons through underactuated total energy shaping. *IEEE access*, 9, 95427–95443.

Lv, G., Xing, H., Lin, J., Gregg, R.D., and Atkeson, C.G. (2020). A task-invariant learning framework of lower-limb exoskeletons for assisting human locomotion. In *2020 American Control Conference (ACC)*, 569–576. IEEE.

Martin, A.E. and Schmiedeler, J.P. (2014). Predicting human walking gaits with a simple planar model. *J. Biomech.*, 47(6), 1416–1421.

Massaroli, S., Poli, M., Califano, F., Park, J., Yamashita, A., and Asama, H. (2022). Optimal energy shaping via neural approximators. *SIAM J. Appl. Dyn. Syst.*, 21(3), 2126–2147.

Nunna, K., Sassano, M., and Astolfi, A. (2015). Constructive interconnection and damping assignment for port-controlled hamiltonian systems. *IEEE Trans. Autom. Control*, 60(9), 2350–2361.

Ortega, R., van der Schaft, A., Maschke, B., and Escobar, G. (2002). Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems. *Automatica*, 38(4), 585–596.

Plaza, S.S.E., Reyes-Báez, R., and Jayawardhana, B. (2022). Total energy shaping with neural interconnection and damping assignment-passivity based control. In *Proc. Learn. Dyn. Control Conf.*, 520–531.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378, 686–707.

Ryalat, M. and Laila, D.S. (2016). A simplified ida-pbc design for underactuated mechanical systems with applications. *Eur. J. Control*, 27, 1–16.

Sanchez-Escalonilla, S., Zoboli, S., and Jayawardhana, B. (2024). Robust neural ida-pbc: passivity-based stabilization under approximations.

Westervelt, E.R., Grizzle, J.W., Chevallereau, C., Choi, J.H., and Morris, B. (2018). *Feedback control of dynamic bipedal robot locomotion*. CRC press.

Zheng, S. and Lv, G. (2023). A two-layer human-in-the-loop optimization framework for customizing lower-limb exoskeleton assistance. In *Proc. of Amer. Control Conf.*, 3913–3920.