

Physics-guided and Energy-based Learning of Interconnected Systems: from Lagrangian to Port-Hamiltonian Systems

Yajie Bao, Vaishnavi Thesma, Atul Kelkar, and Javad Mohammadpour Velni

Abstract—This paper presents a framework for physics-informed energy-based neural network (NN) design to learn models of interconnected systems under the port-Hamiltonian (pH) formalism. In particular, this paper focuses on mechanical systems and incorporates the physical knowledge of Lagrangians into the neural networks to facilitate learning of equations of motion from the data. Moreover, the transformation from the Lagrangian mechanics to the Hamiltonian mechanics is incorporated into the NN architecture and learned from the data such that the learned model is compatible with the pH framework. Then, the structure of input-state-output pH models is imposed on the NN, which guarantees the dissipativity of the learned model. Furthermore, modeling interconnected systems is facilitated by the compositionality property of the pH systems. Additionally, the consistency between the Hamiltonian and Lagrangian is employed for the energy estimation to enable energy-based control. The proposed approach is shown to be computationally more efficient than the existing Lagrangian-based NN design approaches. Furthermore, the learned models with energy estimation are employed for energy-based model predictive control (MPC) design purpose. Experimental results using single (and double) inverted pendulum on carts show that the proposed learning-based approach can achieve an improved performance of model identification compared to the Lagrangian neural networks, accurate estimation of energies and strong control performance.

I. INTRODUCTION

Data-driven methods have been increasingly employed to model unknown system dynamics, where the learned models are compatible with the data in the sense of minimizing metrics for difference (e.g., mean squared error (MSE)), and the accuracy of the learned models is subject to the representativeness of the datasets and the expressiveness of the models. However, known important properties (e.g., dissipativity) of the systems can be missing in the learned model, which limits the out-of-distribution generalization [1] and applicability of the learned models. Incorporating the known knowledge (e.g., physical laws) into learning has been studied to preserve properties. One typical example is Hamiltonian neural networks (HNNs) [2] which aim at training models that respect exact conservation laws.

Authors in [2] proposed to learn Hamiltonians from data using artificial neural networks (ANNs). However, canonical momenta required by HNNs are generally unknown

This work was financially supported by the United States National Science Foundation under awards 1762595 and 1912757.

Y. Bao, V. Thesma, and J. Mohammadpour Velni are with School of Electrical & Computer Engineering, University of Georgia, Athens, GA 30602, USA. {yajie.bao, vaishnavi.thesma25, javadm}@uga.edu.

A. Kelkar is with Department of Mechanical Engineering, Clemson University, Clemson, SC 29634, USA. atul@clemson.edu.

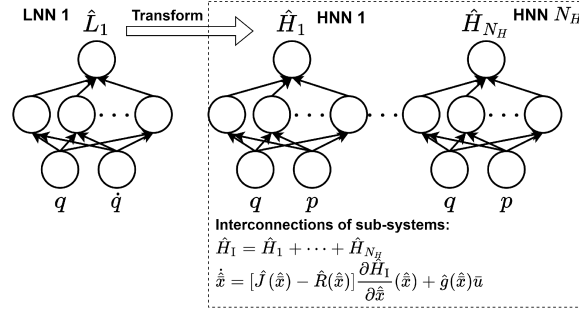


Fig. 1: Physics-guided and energy-based learning of interconnections. The coordinates $(q, \dot{q})$ of data for learning the Euler-Lagrange equations derived from the parameterized Lagrangian (LNN) are transformed to phase space coordinates (p, q) that are later used for the pH modeling of interconnected systems with dissipativity guarantees by the passivity and compositionality properties of the pH systems.

or difficult to compute, and this limits the applicability of HNNs. Furthermore, authors in [3] used an autoencoder to extract phase space coordinates from a collection of observations and used NN/GPs (Gaussian processes) to model Hamiltonian. The autoencoder and the Hamiltonian model were trained jointly, which significantly increases the model complexity for training. Instead, Lagrangian NNs (LNNs) [4] do not require canonical coordinates, and do not restrict the functional form of learned energies. However, LNNs require computing the inverse Hessian of an NN, which decreases the computational efficiency. Furthermore, authors in [5] presented Lagrangian-informed neural networks (LINNs) fused with a sliding mode control design approach for closed-loop model identification (and control) of nonlinear systems.

Moreover, interconnection of different physical domains is ubiquitous in applications. pH systems theory provides a unified framework for modeling systems belonging to different physical domains (mechanical, electrical, hydraulic, thermal, etc.) by recognizing *energy* as 'lingua franca' for multiple domains and identifying ideal system components that capture the main physical characteristics (energy-storage, energy-dissipation, energy-routing, etc.) [6]. For model identification of pH systems, authors in [7] first obtained a descriptor system by interpolating an input-output sequence and then transformed the system into the pH form. Additionally, authors in [8] proposed port-Hamiltonian Neural Networks (pHNNs) to learn time-varying dynamical systems driven by time-dependent control forces and experience energy dissipation. However, pHNNs need momenta data like HNNs and use an L1 penalty that is equal to the L1-norm of the

forcing and damping terms to encourage the network for learning simpler models. Moreover, pHNNs do not consider state-dependent damping matrix and interconnection matrix of system models. Instead, in this paper, we will design an integrated network to incorporate the realizations of pH systems from LINNs by embedding the structure of pH models in the architecture of NNs, as shown by Fig. 1.

Furthermore, energy estimation is built into the proposed approach, which enables energy-based control (EBC). EBC has been used for controller synthesis of a large class of under-actuated systems [9], [10] and the pH systems [11]. Moreover, using NNs for EBC of the pH systems has recently been examined [12], [13]. However, the aforementioned EBC methods assumed that the system models were given and did not consider the system constraints. The authors in [14] learned NN models and embedded the model within an energy control law which cannot cope with the system constraints. Instead, in this work, we utilize energy-based MPC with the learned models to intrinsically handle the system constraints. Contribution of this paper lies in the *development of a physics-informed and energy-based approach to learn models for interconnected systems in the pH framework and an energy-based MPC design approach using the learned models*. The rest of this paper is organized as follows: Section II gives the problem statement and introduces Lagrangian mechanics and pH framework. Proposed physics-guided and energy-based modeling and control are introduced in Section III. Section IV presents pH modeling of interconnected systems. Results of our experiments on single inverted pendulum (SIP) and double inverted pendulum (DIP) to evaluate the performance of the proposed methods are presented in Section V. Finally, Section VI provides the concluding remarks.

II. PROBLEM STATEMENT AND RELATED PRELIMINARIES

To avoid the requirements for momentum data, we start with Lagrangians and then learn the transformation from Lagrangian mechanics to Hamiltonian mechanics, which facilitates modeling interconnected systems in the pH framework. Lagrangian mechanics is a reformulation of classical mechanics that is based on the principle of stationary action with energies used to describe motion. The central quantity of Lagrangian mechanics is the Lagrangian L , which is a function of time-dependent q and $\dot{q}$. A function can be taken as L if it generates correct dynamics of the entire system and follows physical laws. There is no unified expression of L for all physical systems. The expression $L = T - V$ in [15] where T and V denote the kinetic and potential energy, respectively, can be used for mechanical systems.

Using the calculus of variations, the Euler-Lagrange equations in vectorized form are as follows

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}}(q, \dot{q}) = \frac{\partial L}{\partial q}(q, \dot{q}), \quad (1)$$

where $q \in \mathbb{R}^{n_q}$ are generalized coordinates. Then, the equations of motion of the system can be derived by substituting L into (1) and adding the generalized forces $\tau \in \mathbb{R}^{n_\tau}$ to the

right-hand side, which results in

$$\frac{\partial^2 L}{\partial \dot{q}^2}(q, \dot{q})\ddot{q} + \frac{\partial^2 L}{\partial q \partial \dot{q}}(q, \dot{q})\dot{q} - \frac{\partial L}{\partial q}(q, \dot{q}) = \tau \quad (2)$$

These equations bypass constraint forces.

Suppose the dynamics of a system can be described by Lagrangian mechanics, a model of the form (2), and a Lagrangian function (that consists of a kinetic energy function and a potential energy function when applicable) will be learned from data $\mathcal{D} = \{(q^{(i)}, \dot{q}^{(i)}, \ddot{q}^{(i)}), \tau^{(i)}\}_{i=1}^N$. It is noted that the inverse model (2) and the loss function $\ell = \frac{1}{N} \sum_{i=1}^N \|\tau^{(i)} - \hat{\tau}^{(i)}\|_2^2$, where $\hat{\cdot}$ refers to an approximation, are employed instead of the forward model $\ell = \frac{1}{N} \sum_{i=1}^N \|\ddot{q}^{(i)} - \hat{\ddot{q}}^{(i)}\|_2^2$, which avoids computing the inverse Hessian of an NN and thus enhances the training efficiency of the NN. While the families of functions that L belongs to are unknown, coarse-grained knowledge exists, including but not limited to the basis functions of L .

Moreover, Hamiltonian mechanics can be derived directly from Lagrange mechanics via the Legendre transformation between the conjugate variables $(q, \dot{q})$ and (q, p) where p is the generalized momentum. Specifically,

$$p = \frac{\partial L}{\partial \dot{q}}(q, \dot{q}) \quad (3)$$

and (1) becomes

$$\dot{q} = \frac{\partial H}{\partial p}, \quad \dot{p} = -\frac{\partial H}{\partial q}, \quad (4)$$

where $H = p^\top \dot{q} - L$ denotes the *Hamiltonian* (i.e., the total energy) of the system.

A. Port-Hamiltonian Framework

An important subclass of pH systems is of the input-state-output form

$$\begin{cases} \dot{x} = [J(x) - R(x)] \frac{\partial H}{\partial x}(x) + g(x)u \\ y = g^\top(x) \frac{\partial H}{\partial x}(x) \end{cases} \quad (5)$$

with the input u , the state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, and the output y . Also, $J : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ is the interconnection matrix with $J(x) = -J^\top(x)$, $R : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ is the damping matrix specifying the resistive structure with $R(x) = R^\top(x) \succeq 0$. Moreover, $g(x)$ is the input matrix that describes the distribution of the external power into the system. Additionally, (5) satisfies the power-balance as

$$\begin{aligned} \frac{dH}{dt}(x(t)) &= \frac{\partial^\top H}{\partial x}(x) \dot{x} \\ &= - \left(\frac{\partial H}{\partial x}(x) \right)^\top R(x) \frac{\partial H}{\partial x}(x) + y^\top(t)u(t). \end{aligned} \quad (6)$$

Since $R(x) \succeq 0$, we have passivity inequality $\frac{dH}{dt}(x(t)) \leq y^\top(t)u(t)$. Consequently, power-conservation of Dirac structure yields passivity of any pH system if H is bounded from below. In particular, Hamilton's equations (4) can be represented in the port-Hamiltonian framework with $J(x) = \begin{bmatrix} \mathbf{0} & I \\ -I & \mathbf{0} \end{bmatrix}$ and $g(x)u = \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix}$ where I is the identity matrix, $\mathbf{0}$ is the zero matrix with suitable shapes, and $x = [q; p]$.

The problem addressed in this paper is that of learning offline dissipative dynamic models in the pH form with energy estimations for energy-based control, using the dataset $\mathcal{D} = \{(q^{(i)}, \dot{q}^{(i)}, \ddot{q}^{(i)}), \tau^{(i)}\}_{i=1}^N$.

III. PHYSICS-GUIDED ENERGY-BASED MODELING

This section details methods to incorporate the knowledge into NNs to improve the learning efficiency of modeling dynamical systems with energy estimation. Moreover, we present an NN-based approach to transform Euler-Lagrange equations derived from LINNs into the equations formulated by Hamiltonian mechanics that are compatible with the pH framework.

A. Physics-guided Lagrangian Neural Networks (PLNNs)

Since (2) is derived from L , the representation of equations of motion depends on the parameterization of L . We use NNs to represent L for general Lagrangians. However, it is noted that using fully-connected NNs may not help in learning an accurate model due to an inductive bias, which motivated us to further use physical knowledge to improve the learning efficacy of LNNs.

1) *Feature engineering for knowledge embedding*: In machine learning, a feature is an individual measurable property or characteristic of a phenomenon being observed [16], and hence choosing features is crucial for the effectiveness of the learning algorithms. Since the typical knowledge for mechanical systems is that sine and cosine functions of the generalized coordinates exist in the Lagrangians, we can use $[q; \dot{q}; \cos(q); \sin(q)]$ as features for ANNs [5] to improve learning efficiency. Moreover, the set of features (aka basis functions) is supposed to contain the basis functions of the true functions to approximate. Then, Lasso regression can be employed for basis function selection. To summarize, the NNs aim to solve the following optimization problem

$$\min \ell_L := \frac{1}{N} \sum_{i=1}^N \|\tau^{(i)} - \hat{\tau}^{(i)}\|_2^2 + \lambda_T |\alpha| + \lambda_V |\beta| \quad (7a)$$

$$\text{s.t. } \hat{\tau}^{(i)} = \frac{\partial^2 \hat{L}}{\partial \dot{q}^2}(q^{(i)}, \dot{q}^{(i)}) \ddot{q}^{(i)} + \frac{\partial^2 \hat{L}}{\partial q \partial \dot{q}}(q^{(i)}, \dot{q}^{(i)}) \dot{q}^{(i)} - \frac{\partial \hat{L}}{\partial q}(q^{(i)}, \dot{q}^{(i)}) \quad (7b)$$

where $\hat{L}(q^{(i)}, \dot{q}^{(i)}) = \hat{T}(q^{(i)}, \dot{q}^{(i)}) - \hat{V}(q^{(i)}, \dot{q}^{(i)}) = \sum_{j=1}^{N_T} \alpha_j T_j(q^{(i)}, \dot{q}^{(i)}) - \sum_{j=1}^{N_V} \beta_j V_j(q^{(i)}, \dot{q}^{(i)})$ with α_j and β_j denoting the j -th elements of vectors α and β ; λ_T and λ_V are the regularization parameters; $T_j \in \mathcal{T} := \{T_j : X \rightarrow \mathcal{T}\}_{j=1}^{N_T}$ and $V_j \in \mathcal{V} := \{V_j : X \rightarrow \mathcal{V}\}_{j=1}^{N_V}$ with $\mathcal{T}$ and $\mathcal{V}$ denoting the set of values of T and V are the basis functions of the kinetic and potential energy, respectively. However, the inverse model with small prediction errors does not necessarily provide accurate energy estimation that is key to energy-based control. Since the equation of motion by Hamiltonian mechanics determine H only up to an additive constant, the authors in [3] assumed the knowledge of $H_0 = H(q_0, p_0)$ at a single point (q_0, p_0) in the phase space. Then, H_0 introduces another constraint $\hat{H}(q_0, p_0) = H_0$ where $\hat{H}$ is the Hamiltonian estimated by HNNs, to the loss function.

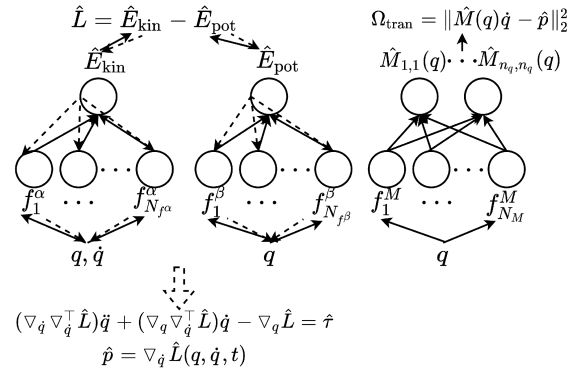


Fig. 2: Architecture of PLNN and regularizers, in which the dashed arrows show the calculus of variations by auto-differentiation.

Moreover, the assumption is not impractical, as H_0 and (q_0, p_0) can be chosen arbitrarily. However, the constraint cannot ensure accurate energy estimates, as shown by the experiments. Instead, by the *conservation of energy* law, we can collect data with Hamiltonians $\mathcal{D}_H$ by simulating the system without external forces from the initial state (q_0, p_0) . Then, we train the model on $\mathcal{D}_H$ with the loss function $\ell_H = \ell_L + \frac{1}{N_{\mathcal{D}_H}} \sum_{i=1}^{N_{\mathcal{D}_H}} \|H_0 - \hat{H}^{(i)}\|_2^2$ where $\hat{H}^{(i)} = \hat{T}(q^{(i)}, \dot{q}^{(i)}) + \hat{V}(q^{(i)}, \dot{q}^{(i)})$. It is noted that there is distribution discrepancy between $\mathcal{D}_H$ without external forces and $\mathcal{D}$ with external forces as discussed in [17]. To learn an accurate model with external forces, we fine-tune the trained model on $\mathcal{D}$ using stochastic gradient descent.

2) *Architecture design*: Based on (3), the estimate $\hat{p}^{(i)} = \frac{\partial \hat{L}}{\partial \dot{q}}(q^{(i)}, \dot{q}^{(i)})$ of p can be obtained from the PLNN by auto-differentiation. Additionally, we can directly use an NN to represent the transformation from $\dot{q}$ to p . In particular, the transformation for mechanical systems follows $p = M(q)\dot{q}$. Therefore, we use an NN denoted by $\hat{M}(q)$ to represent $M(q)$ and have another estimate $\hat{p}_{\text{tran}} = \hat{M}(q)\dot{q}$, as shown by Fig. 2. Then, a regularization term

$$\Omega_{\text{tran}} = \frac{1}{N} \sum_{i=1}^N \|\hat{p}_{\text{tran}}^{(i)} - \hat{p}^{(i)}\|_2^2 \quad (8)$$

is added to the loss functions to have accurate estimation of p . It is noted that Ω_{tran} can be made arbitrarily small using a deep NN. Basis functions can be selected for $\hat{M}(q)$ to incorporate knowledge.

B. Energy-based Control

The energy functions learned by PLNNs enable energy-based control which can be advantageous over classical set-point tracking. We use the energy functions for the formulation of a model predictive controller. At each time step k , the following problem is solved

$$\min_{\mathbf{x}_{0:N_p+1}, \mathbf{u}_{0:N_p}} m(x_{N_p+1}) + \sum_{i=0}^{N_p} l(\hat{T}(x_i), \hat{V}(x_i), u_i) \quad (9a)$$

$$\text{s.t. } u_{\min} \leq u_i \leq u_{\max} \quad (9b)$$

$$(x_i, u_i) \rightarrow x_{i+1} : (7b) \quad (9c)$$

$$g(x_i) \leq 0, \quad \forall i = 1, \dots, N_p + 1, \quad (9d)$$

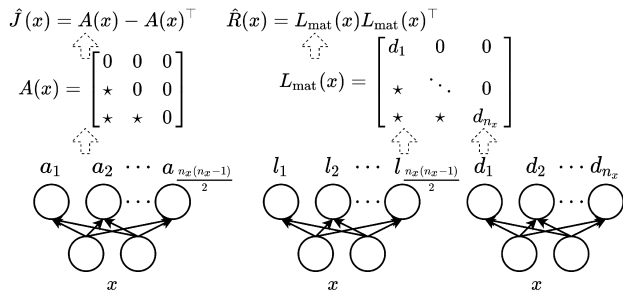


Fig. 3: Architectures of NNs to represent $J(x)$ and $R(x)$. Three NNs are used to learn functions of the off-diagonal elements of $A(x)$ and L_{mat} , and the diagonal elements of L_{mat} , respectively. d_i are ensured to be positive by using non-negative activation functions, e.g., ReLu or Softplus.

where N_p is the prediction horizon, and (9d) represents the state constraints. Then, the control input is $u_k = u_0^*$ where u^* is the optimal solution to (9). Other types of energy-based control are also compatible with PLNNs.

IV. PH MODELING OF INTERCONNECTED SYSTEMS

To further impose the structure of pH model (5) onto the NNs, we enforce the skew-symmetry of $J(x)$ and the positive semidefiniteness of $R(x)$ in the architecture of NNs, as shown in Fig. 3. Since the set of all skew-symmetric matrices of a given size $n \times n$ forms a vector space with dimension $\frac{n(n-1)}{2}$, we use an NN with $\frac{n(n-1)}{2}$ units in the output layer to represent the elements below the main diagonal of A_{n_x} . The elements above the main diagonal and the diagonal elements of A_{n_x} are all 0's. Then,

$$\hat{J}(x) = A(x) - A(x)^\top = -\hat{J}(x)^\top. \quad (10)$$

Moreover, as in [15], we use an NN to represent the entries below the main diagonal and another NN to represent the non-negative diagonal entries, to assemble a lower triangular matrix $L_{\text{matr}}(x)$ with a non-negative diagonal. Then

$$\hat{R}(x) = L_{\text{mat}}(x)L_{\text{mat}}^\top(x) \succeq 0. \quad (11)$$

In particular, for mechanical systems, we have $R(x) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C(x) \end{bmatrix}$, where $C(x) \succeq 0$ can be represented by NNs, similar to (11).

Theorem 1: The NN model with $J(x)$ approximated by (10) and $R(x)$ by (11) is passive throughout the training process.

Proof: Since $\hat{J} = -\hat{J}^\top$, $\hat{R} \succeq 0$ and $H \geq 0$ for any x and weights of the NNs, then, the NN model that represents (5) satisfies the passivity inequality by (6). ■

Theorem 1 guarantees the passivity of the NN model in the pH form, which provides a useful regularization for modeling passive systems by architecture design, while NNs that approximate the ordinary differential equations (ODEs) by minimizing $\ell = \frac{1}{N} \sum_{i=1}^N \|\hat{q}^{(i)} - \tilde{q}^{(i)}\|_2^2$ may not be passive.

Furthermore, by the compositionality of pH systems, the interconnection of pH systems with Dirac structure through

power-conserving interconnection is again pH [6]. Additionally, the Hamiltonian of the interconnected pH system will be the sum of the Hamiltonians of its subsystems while the energy-dissipation relation of the interconnected system will be the union of the energy-dissipating relations of the subsystems. Therefore, we can extend the set of basis functions with those of all the subsystems to model the Hamiltonian, and sum the Hamiltonians of the subsystems to estimate the Hamiltonian of the interconnected pH system.

Additionally, Lagrangian L , as well as $J(x)$ and $R(x)$ can be learned simultaneously by integrating the sub-networks in Section III-A and II-A to solve the following problem

$$\min_{w, \alpha, \beta} \sum_{i=1}^{N_H} (\ell_{L,i} + \Omega_{\text{tran},i}) \quad (12a)$$

$$\text{s.t. } \dot{\hat{x}} = [\hat{J}(\hat{x}) - \hat{R}(\hat{x})] \frac{\partial \hat{H}_1}{\partial \hat{x}}(\hat{x}) + \hat{g}(\hat{x})\bar{u}, \quad (12b)$$

where $\ell_{L,i}$ and $\Omega_{\text{tran},i}$ denote the loss in the form of (7a) and regularizer in the form of (8) for the i -th subsystem, respectively. Furthermore, $\hat{x} = [\hat{x}_1; \dots; \hat{x}_{N_H}]$, $\hat{x}_i = [q_i; \dot{p}_i]$, $\hat{p}_i = \frac{\partial \hat{L}_i}{\partial \dot{q}_i}(q_i, \dot{q}_i)$, and $\hat{H}_1 = \sum_{i=1}^{N_H} \hat{H}_i$.

Proposition 1: Suppose $\mathcal{D}$ is sufficient and the model architecture is determined to contain the family of models that can describe the system denoted by S such that the model f learned from $\mathcal{D}$ achieves a generalization error E that is close to 0. Then, given $\mathcal{D}_1 = \{(\tilde{q}^{(i)}, \dot{\tilde{q}}^{(i)}, \ddot{\tilde{q}}^{(i)}), \bar{\tau}^{(i)}\}_{i=1}^{N_1}$ and the model architecture of the interconnected pH system I where S is a subsystem, the generalization error E_I^1 of the interconnected system model f_I^1 that incorporates f will not exceed the error E_I^2 of the interconnected system model f_I^2 only based on $\mathcal{D}_1$.

It is noted that the accuracy of $\hat{H}_S^1$ from f will not be smaller than that of $\hat{H}_S^2$ from f_I^1 . Using the compositionality property of the pH systems, $H_1 = H_S + H_R$ and $\bar{x} = [x_S; x_R]$, where the subscripts I, S and R denote the interconnected system, the subsystem S , and the remaining subsystems, respectively. Then, the accuracy of $\hat{H}_I^1 = \hat{H}_S^1 + \hat{H}_R^1$ of f_I^1 will not be smaller than that of $\hat{H}_I^2 = \hat{H}_S^2 + \hat{H}_R^2$ of f_I^2 , as f_I^1 and f_I^2 share the same model architecture and are developed based on the same dataset $\mathcal{D}_1$. Therefore, $E_I^1 \leq E_I^2$, which demonstrates the advantage of learning the model of interconnected systems by compositionality.

V. EXPERIMENTAL RESULTS AND VALIDATION

The proposed learning methods are validated using simulations of single (and double) inverted pendulums.

A. Validation on a Single Inverted Pendulum (SIP)

The energy functions of the SIP are

$$T = \frac{1}{2}(m_0 + m_1)\dot{x}_{\text{pos}}^2 + \frac{1}{2}m_1L_1^2\dot{\theta}_1^2 + m_1L_1\dot{x}_{\text{pos}}\dot{\theta}_1\cos(\theta_1),$$

$$V = -m_1gL_1\cos(\theta_1).$$

Then, from (2), the equations of motion for SIP are

$$(m_0 + m_1)\ddot{x}_{\text{pos}} + m_1L_1\ddot{\theta}_1\cos(\theta_1) - m_1L_1\dot{\theta}_1^2\sin(\theta_1) = f,$$

$$m_1L_1\ddot{x}_{\text{pos}}\cos(\theta_1) + m_1L_1^2\ddot{\theta}_1 + m_1gL_1\sin(\theta_1) = 0. \quad (13)$$

This experiment intends to learn a pH model from the simulation data of the SIP system (13) and then design an energy-based MPC with the learned pH model to erect the pendulum at desired positions and to stabilize the pendulum in the up position while avoiding obstacles. In this experiment, we consider a circle obstacle with center c and radius r . Table I shows the parameters of the simulations. The initial state is considered to be $[x_{\text{pos}}; \theta_1; \dot{x}_{\text{pos}}; \dot{\theta}_1] = [0; \pi; 0; 0]$.

TABLE I: Simulation parameters.

Parameter	Value	Parameter	Value
m_0	0.6 kg	g	9.8 m/s^2
m_1	0.2 kg	c	(0, 0.6)
L_1	0.5 m	r	0.3
f	$[-4, 4]$ N	m_2	0.2 kg

1) *Experimental setup*: We applied a random input sequence drawn from the uniform distribution $\mathcal{U}(-5, 5)$ to the system (13) to generate data. A sampling time of 0.04 s is used and 100,000 data points were collected. We used the first 80,000 points for training and the rest 20,000 for testing. First, we used $\mathcal{T}_1 = \{\dot{x}_{\text{pos}}^2, \dot{\theta}_1^2, \dot{x}_{\text{pos}}\dot{\theta}_1\cos(\theta_1)\}$ and $\mathcal{V}_1 = \{\cos(\theta_1)\}$. Linear combinations of basis functions in $\mathcal{T}_1$ and $\mathcal{V}_1$ were used to represent T and V . Then, we used $\mathcal{T}_2 = \{\dot{x}_{\text{pos}}, \dot{\theta}_1, \cos(\theta_1)\}$ as the basis functions of $\hat{T}$ to test the effect of basis functions on the modeling performance. $\mathcal{T}_2$ was fed into a fully-connected NN with 2 hidden layers to represent T . Each hidden layer consists of 500 units and uses Softplus as activation functions while output layer has 1 unit without activation functions. For model optimization, we use Adam optimizer in Keras [18]. The learning rate of Adam is set to be 0.01 and decay to be $1e-6$. All the other parameters of Adam are set as default. We trained the model with $\mathcal{T}_1$ and $\mathcal{V}_1$ for 1,000 epochs and the model with $\mathcal{T}_2$ and $\mathcal{V}_1$ for 1,000 epochs with batch size of 2048.

For the formulation of the MPC, we chose $N_p = 20$, $l = \hat{V} + (x_{\text{pos}} - x_{\text{pos,des}})^2$, $m = \hat{T} + \hat{V}$, and considered the constraint (9d) to be $g = 1.05r - d_2 \geq 0$ to avoid the obstacle where d_2 is the distance between the pole and the center of the circle, and r is the radius. To solve (9) with ODEs, we employed the orthogonal collocation on finite elements discretization approach with Gauss-Radau collocation points [19] implemented in [20]. The collocation degree was set to 2 and the number of finite elements for the states within a time-step to 2.

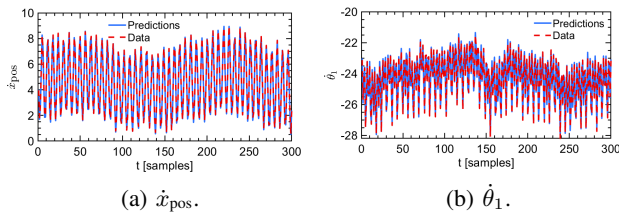


Fig. 4: Validation of the learned models on the testing set. For the sake of clarity, only the first 300 testing data points are shown here. $\text{MAE}_{\dot{x}_{\text{pos}}} = 0.05$ and $\text{MAE}_{\dot{\theta}_1} = 0.12$.

2) *Results and discussion*: The trained LNN model with $\mathcal{T}_1$ and $\mathcal{V}_1$ achieved mean absolute errors (MAE) of $\text{MAE}_{\dot{q}} =$

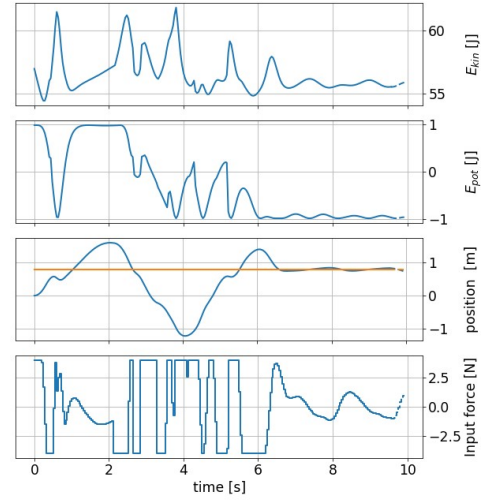


Fig. 5: Control results using $\mathcal{T}_2$ and $\mathcal{V}_1$.

0.42 and $\text{MAE}_L = 4.58$ on the testing set, and can be used for energy-based MPC to achieve the control objectives. However, with $\mathcal{T}_2$ and $\mathcal{V}_1$, the LNN model achieved $\text{MAE}_{\dot{q}} = 3.84$ but is too complex for MPC while using less units for hidden layers results in optimization failure from the inverse Hessian operation of an NN and larger MAEs. Instead, using the proposed approach with $\mathcal{T}_2$ and $\mathcal{V}_1$, a fully-connected NN that is composed of 1 hidden layer with 10 hidden units can be trained efficiently and achieved $\text{MAE}_{\tau} = 0.26$ and $\text{MAE}_L = 39.39$ after training for 5,000 epochs. Fig. 4 shows the comparison between the predictions of the learned model and the data from the system simulation on the testing set. Furthermore, the designed MPC with $N_p = 10$, $l = 5\hat{V} + 10(x_1 - x_{1,\text{des}})^2$, $m = \hat{T} + 50\hat{V}$ using the learned model successfully avoided the obstacles and erected the pendulum at the desired position. The control results are shown in Fig. 5. It is noted that the minimum estimate of the kinetic energy is not 0, as the data determines T only up to an additive constant.

B. Validation on Double Inverted Pendulum (DIP)

Based on the model of an SIP, we learn the DIP model to validate the approach described in Section IV. The energy functions of DIP are

$$\begin{aligned}
 T_{\text{cart}} &= \frac{1}{2} m_0 \dot{x}_{\text{pos}}^2 \\
 T_{p_1} &= \frac{1}{2} m_1 \left((\dot{x}_{\text{pos}} + l_1 \dot{\theta}_1 \cos(\theta_1))^2 + (l_1 \dot{\theta}_1 \sin(\theta_1))^2 \right) \\
 &\quad + \frac{1}{2} J_1 \dot{\theta}_1^2 \\
 T_{p_2} &= \frac{1}{2} m_2 \left((\dot{x}_{\text{pos}} + L_1 \dot{\theta}_1 \cos(\theta_1) + l_2 \dot{\theta}_2 \cos(\theta_2))^2 \right. \\
 &\quad \left. + (l_1 \dot{\theta}_1 \sin(\theta_1) + l_2 \dot{\theta}_2 \sin(\theta_2))^2 \right) + \frac{1}{2} J_2 \dot{\theta}_2^2 \\
 V &= -m_1 g l_1 \cos(\theta_1) - m_2 g (L_1 \cos(\theta_1) + l_2 \cos(\theta_2)).
 \end{aligned}$$

Since the second pendulum is linked to the first one, the set of basis functions for $\hat{L}$ was extended to include the interactions. Specifically, $\mathcal{T} = F \otimes F$ where

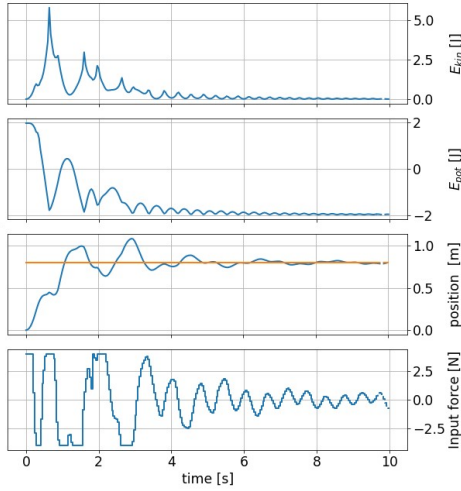


Fig. 6: Control results for the DIP example.

$F = [\dot{x}_{\text{pos}}; \dot{\theta}_1 \sin(\theta_1); \dot{\theta}_1 \cos(\theta_1); \dot{\theta}_2 \sin(\theta_2); \dot{\theta}_2 \cos(\theta_2); \dot{\theta}_1; \dot{\theta}_2]$ and $\otimes$ denotes the Kronecker product, and $\mathcal{V} = [\cos(\theta_1); \cos(\theta_2)]$.

1) *Experimental setting:* To collect data of Hamiltonians, we first simulate DIP without external forces for 10 time steps from different initial states of the form $[0; \theta_1; \theta_2; 0; 0; 0]$ where θ_1, θ_2 are from the 100 uniform grid points in the range $[0.03, 3.11]$ rad. Then, the Hamiltonians are conserved in the 10 steps and equal to the potential energies that are easy to evaluate for each initial state. Additionally, the sampling rate is 0.04s. In this way, we collected a dataset with Hamiltonian measurements. For learning the model of DIP, we first train a model on the dataset $\mathcal{D}_H$ and then fine-tune the model using the dataset collected by applying random inputs drawn from $\mathcal{U}(-4, 4)$ to the system as V-A.1. L1 norm regularization parameter of $\lambda_T = 1e - 5$.

2) *Results and discussion:* The trained model on $\mathcal{D}_H$ achieved testing error of $\text{MAE}_H = 5.30e - 4$ but $\text{MAE}_L = 50.71$ on $\mathcal{D}$, which shows the distributions discrepancy between $\mathcal{D}_H$ and $\mathcal{D}$. After fine-tuning for 8,000 epochs, $\text{MAE}_T = 0.10$ and $\text{MAE}_L = 12.35$. Furthermore, the designed MPC with $N_p = 10$, $l = \hat{V} + 10(x_1 - x_{1,\text{des}})^2$, $m = 10\hat{T} + 10\hat{V}$ using the learned model successfully avoided the obstacle with $c = (0, 0.6)$ and $r = 0.1$, and erected the pendulum at the desired position. The control results are shown in Fig. 6.

VI. CONCLUDING REMARKS

In this paper, a physics-guided and energy-based neural network learning approach was proposed to learn an accurate model of interconnected systems from data in the pH framework. The proposed approach provided an inverse model, the transformation from Lagrangian mechanics to Hamiltonian mechanics, and an accurate energy estimate which was used later for the interconnected system identification and energy-based MPC design. The architecture design of the neural network that represented the interconnected systems was informed by the Hamiltonian of subsystems and the compositionality of pH systems. Moreover, the passivity of

learned models was ensured throughout the training process by enforcing the skew-symmetry/positive semi-definiteness of matrix functions via neural network architecture design. The learned energy functions were employed to formulate the energy-based MPC design. Finally, experiments on single (and double) inverted pendulum showed that the proposed methods can learn an accurate model, achieve strong control performance with bounded plant-model mismatch, and be applied to interconnected systems of high complexities.

REFERENCES

- [1] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," 2021.
- [2] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *arXiv preprint arXiv:1906.01563*, 2019.
- [3] T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis, "On learning Hamiltonian systems from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 12, p. 121107, 2019.
- [4] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.
- [5] Y. Bao, V. Thesma, and J. M. Velni, "Physics-guided and neural network learning-based sliding mode control," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 705–710, 2021.
- [6] A. Van Der Schaft, "Port-Hamiltonian systems: an introductory survey," in *Proceedings of the international congress of mathematicians*, vol. 3. Citeseer, 2006, pp. 1339–1365.
- [7] K. Cherifi, V. Mehrmann, and K. Hariche, "Numerical methods to compute a minimal realization of a port-hamiltonian system," 2019.
- [8] S. A. Desai, M. Mattheakis, D. Sondak, P. Protopapas, and S. J. Roberts, "Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems," *Physical Review E*, vol. 104, no. 3, p. 034312, 2021.
- [9] R. Ortega, M. W. Spong, F. Gómez-Estern, and G. Blankenstein, "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment," *IEEE transactions on automatic control*, vol. 47, no. 8, pp. 1218–1233, 2002.
- [10] W. Zhong and H. Rock, "Energy and passivity based control of the double inverted pendulum on a cart," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01)(Cat. No. 01CH37204)*. IEEE, 2001, pp. 896–901.
- [11] A. Van Der Schaft and D. Jeltsema, "Port-Hamiltonian systems theory: An introductory overview," *Foundations and Trends in Systems and Control*, vol. 1, no. 2-3, pp. 173–378, 2014.
- [12] S. S. E. Plaza, R. Reyes-Báez, and B. Jayawardhana, "Total energy shaping with neural interconnection and damping assignment-passivity based control," in *4th Annual Learning for Dynamics & Control Conference*. arXiv, 2022.
- [13] L. Xu, M. Zakwan, and G. Ferrari-Trecate, "Neural energy casimir control," *arXiv preprint arXiv:2112.03339*, 2021.
- [14] M. Lutter, K. Listmann, and J. Peters, "Deep Lagrangian networks for end-to-end learning of energy-based control for under-actuated systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7718–7725.
- [15] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: Using physics as model prior for deep learning," *arXiv preprint arXiv:1907.04490*, 2019.
- [16] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [17] Y. Bao, J. Mohammadpour Velni, and M. Shahbakhti, "An online transfer learning approach for identification and predictive control design with application to RCCI engines," in *Dynamic Systems and Control Conference*, vol. 84270. American Society of Mechanical Engineers, 2020, p. V001T21A003.
- [18] F. Chollet et al., "Keras," <https://keras.io>, 2015.
- [19] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.
- [20] S. Lucia, A. Tăulea-Codrean, C. Schoppmeyer, and S. Engell, "Rapid development of modular and sustainable nonlinear model predictive control solutions," *Control Engineering Practice*, vol. 60, pp. 51–62, 2017.