



Research Article

Hamiltonian Neural Network 6-DoF Rigid-Body Dynamic Modeling Based on Energy Variation Estimation

Fei Simiao ¹, Huo Lin ², Sun Zhixiao,¹ Wang He,¹ Lu Yuanjie,¹ He Jile,¹ Luo Qing,¹ and Su Qihang¹

¹SADRI Institute, Aviation Industry Corporation of China (AVIC), No. 40 Tawan Street, Shenyang 110031, Liaoning, China

²School of Safety, Shenyang Aerospace University, 37 Daoyi South Street, Shenyang 110136, Liaoning, China

Correspondence should be addressed to Huo Lin; helen0404@icloud.com

Received 23 June 2023; Revised 9 August 2023; Accepted 18 August 2023; Published 5 September 2023

Academic Editor: Lucia Valentina Gambuzza

Copyright © 2023 Fei Simiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study introduces a novel deep modeling approach that utilizes Hamiltonian neural networks to address the challenges of modeling the six degrees of freedom rigid-body dynamics induced by control inputs in various domains such as aerospace, robotics, and automotive engineering. The proposed method is based on the principles of Hamiltonian dynamics and employs an inductive bias in the form of a constructed bias for both conservative and varying energies, effectively tackling the modeling issues arising from time-varying energy in controlled rigid-body dynamics. This constructed bias captures the information regarding the changes in the rigid body's energy. The presented method not only achieves highly accurate modeling but also preserves the inherent bidirectional time-sliding inference in Hamiltonian-based modeling approaches. Experimental results demonstrate that our method outperforms existing techniques in the time-varying six degrees of freedom dynamic modeling of aircraft and missile guidance, enabling high-precision modeling and feedback correction. The findings of our research hold significant potential for the kinematic modeling of time-varying energy systems, parallel system state prediction and control, inverse motion inference, and autonomous decision-making in military applications.

1. Introduction

Six-degree-of-freedom (6-DoF) rigid-body dynamic modeling is crucial in numerous applications, such as aerospace, robotics, and automotive sectors. This modeling assists designers and controllers in understanding the motion attributes and behaviors of rigid bodies, leading to better control strategies. Particularly in high-velocity motion and complex environments, modeling and predicting rigid-body dynamics are vital. The model requires considering factors such as mass, inertia, rotational capability, and equations of motion and accurately forecasting the forthcoming position, velocity, and acceleration parameters. Traditional numerical simulation techniques demand manual model construction and initial condition configurations, often requiring significant human resources and time. However, deep-learning models enable rapid training of an efficient model capable of concurrently predicting motion trajectories and states of

multiple rigid bodies, performing backward inference, and making swift decisions. Consequently, deep-learning models exhibit extensive potential for application in 6-DoF rigid-body dynamic modeling.

In recent years, deep-learning methodologies have emerged as an innovative approach to 6-DoF rigid-body dynamic modeling. Deep neural networks are multilayered models based on neuronal structures that can autonomously extract features from input data and execute intricate nonlinear mappings. Moreover, deep neural networks inherently possess the capacity to perform large-scale batch inferences. Employing deep-learning techniques for learning 6-DoF rigid-body dynamic models entails training neural networks to automatically discern motion patterns and characteristics of rigid bodies from motion data, as opposed to traditional approaches that necessitate manual design of rigid-body dynamic models and feature extraction algorithms.

Deep learning-based dynamic modeling offers several advantages over traditional approaches:

- (1) Deep learning can automatically extract information from data, expediting the dynamic modeling process
- (2) By learning models directly from data containing interference, deep dynamic models exhibit enhanced robustness and resilience against interference
- (3) Deep dynamic models can seamlessly integrate with deep reinforcement learning to generate a unified control strategy
- (4) Inference can be naturally conducted in parallel on a large scale using batch processing
- (5) The model can be easily fine-tuned based on new rigid-body motion data to adapt to novel problems

There are two primary approaches to addressing the problem of 6-DoF rigid-body dynamic modeling based on deep learning. The first approach is knowledge-driven, exemplified by the physics-informed neural network (PINN) method, a physics-constrained reasoning technique grounded in deep neural networks. This method is employed to solve partial differential equations with physical constraints or to incorporate partial differential equations directly into the neural network learning process. For example, PINNs have been utilized to replicate 6-DoF equations of aircraft dynamics [1]. However, due to potential interference or errors in rigid-body flight data, the rigid-body dynamic model generated by the PINN method may be affected by interference, compromising its accuracy.

The alternative approach is data-driven, aiming to learn the underlying physical laws concealed within the collected data. Representative studies include [2, 3] and [4], which strive to learn the precise output corresponding to the input that aligns with the data under the data-driven paradigm. Another approach involves meticulously designing the intrinsic components of deep learning, forming specific induction biases that naturally align with the physical essence embedded within the data. For instance, the authors in [5] devised a robust induction bias for energy conservation, yielding an intriguing byproduct: time reversibility. Building upon, the authors in [5, 6] proposed a deep generative model termed the Hamiltonian generative network (HGN), which can learn the Hamiltonian dynamics of continuous-time evolution systems, exhibiting features such as time reversibility and smooth temporal interpolation.

In the context of 6-DoF equations for rigid bodies, energy is typically not conserved. External control quantities or passive drag forces exert torques on the rigid body, enabling its dynamic and potential energy to fluctuate. These fluctuations largely depend on external conditions in which the rigid body is situated. Therefore, constructing a 6-DoF deep model for rigid bodies requires considering factors that disrupt energy symmetry to ensure the model's accuracy and reliability.

Motivated by prior research [5, 6], this study aims to develop a method that captures the inductive bias of energy changes in rigid bodies as external conditions vary while preserving the high-precision modeling of 6-DoF equations

for rigid bodies and the high-precision forward and backward sliding along the temporal dimension. The proposed method, an energy variational Hamiltonian neural network (VHNN), has been experimentally validated on a relevant dataset [7]. Results demonstrate that VHNN outperforms other methods, such as Hamiltonian neural networks (HNNs) and Hamiltonian graph networks (HGNs), in modeling the 6-DoF equations of rigid bodies and performing temporal sliding.

To ensure a clear presentation of the research content, this paper will be structured into the following sections: Section 2 reviews relevant prior research related to this study, with a particular focus on similar works that employ Hamiltonian methods for deep modeling, which lay the foundation for energy-inductive biases. Section 3 provides a brief introduction to Hamiltonian equations and the six degrees of freedom equations for rigid bodies and extensively discusses the principles of the proposed method, including the neural network module and architecture design, optimization objectives, and specific algorithms. Section 4 assesses the effectiveness and advantages of the proposed method through an analysis of complex dynamic modeling experiments involving aircraft and missiles. Sections 5 and 6 analyze the potential future directions and application prospects of this method and provide a comprehensive evaluation of the method.

2. Related Work

2.1. Dynamic Modeling Using Neural Networks. In recent years, considerable advancements have been made in the domain of dynamic modeling using neural networks. This progress spans various fields, such as fluid dynamic property modeling [8], chemical reaction process modeling [9], microscopic particle dynamic modeling [10, 11], and rigid-body dynamic modeling.

2.2. Rigid-Body Dynamic Modeling. In the realm of rigid-body dynamics, the authors in [12] introduced a graph-based model for simulating the dynamics of joint rigid bodies, facilitating perceptual modeling. They described the architecture and operational principles of Lagrangian graph neural networks (LGNNs) and assessed their efficacy in rendering joint rigid-body processes. The performance of LGNN was corroborated across multiple simulation tasks, yielding superior accuracy in learning physical models compared to alternative models.

Wang et al. [13] proposed a deep learning-based approach for robot dynamic parameter identification and compensation, in addition to the UCM model, effectively addressing the limitations of robot physical dynamic models and enhancing the environmental adaptability of conventional physical models.

Millard et al. [14] presented a differentiable rigid-body dynamic simulator. They employed a variety of techniques for integrating differential equations and computing gradients and compared different parameter estimation methods. In trajectory optimization algorithms, simulation

parameters were procured empirically, and closed-loop model predictive control algorithms were implemented to attain cost and performance optimization.

Sun et al. [15] suggested a deep neural network with dynamic keypoint selection, extracting 6-DoF object pose states from image pixels. Zhang et al. [16] proposed a deep-learning methodology for predicting 6-DoF ship motion, constructing a transformer neural network accounting for the effects of operating conditions on ship dynamics for 6-DoF state transition equations.

2.3. Hamiltonian-Based Models. Research on Hamiltonian neural networks (HNNs) and Hamiltonian generative networks (HGNs) directly models dynamics using Hamiltonian differential equations, resulting in slower divergence rates for extended trajectories. Although these methods bear similarities to neural ODE work [17, 18], Hamiltonian dynamics exhibit time reversibility, rendering HNN and HGN methods more advantageous in terms of computational efficiency and applicability to physical systems and other processes possessing these characteristics.

2.3.1. Hamiltonian Neural Networks (HNNs). HNNs incorporate energy conservation as an inductive bias for neural networks, allowing them to learn conservation laws from data. HNNs learn a parameterized function $H_\theta(q, p)$. To train HNNs, the error between the time derivatives of known coordinates p and q and the symplectic gradient of H concerning input coordinates is minimized. This structure permits HNNs to learn conservation laws from arbitrary coordinates. HNNs have established neural network models for ideal springs, the three-body problem, and other dynamics using Hamilton's equations.

2.3.2. Hamiltonian Generative Networks (HGNs). HGNs constitute a class of generative models that learn time-reversible Hamiltonian dynamics in abstract phase space representations, commencing from image inputs. HGNs learn in three stages. Initially, they encode a sequence of images into initial states S_i in the abstract state space and map these states to a scalar that can be interpreted as Hamiltonian. Subsequently, they estimate dynamics in the abstract state space using Hamiltonian and project the results back into pixel representations employing a deconvolution network. Ultimately, they optimize the network using a loss akin to that of a time-extended variational autoencoder.

2.3.3. Other Hamiltonian-Based Models. Additional Hamiltonian-based models have been developed building upon the foundation of HNNs. The authors in [19] proposed symplectic recurrent neural networks, which utilize symplectic integrators, multistep training, and initial state optimization to learn superior Hamiltonians compared to HNNs. The authors in [20] combined graph networks with differentiable ODE integrators and Hamiltonian inductive bias to predict the dynamics of particle systems. The authors

in [21] targeted dissipative systems' dynamic modeling, proposing an inductive bias related to Hamiltonian and Helmholtz decomposition, achieving favorable results in elementary rigid-body dynamic modeling.

2.4. Limitations and the Proposed Approach. Despite these methods forging new pathways for deep learning-based modeling, energy conservation inductive bias may not be applicable to numerous real-world modeling problems. Specifically, for intricate rigid-body 6-DoF equation modeling, encompassing the aerodynamic characteristics and 6-DoF equations of complex behavior rigid-body systems such as missiles or airplanes, energy is not conserved from the perspective of mechanical energy. Consequently, for such dynamic modeling problems, it is imperative to consider breaking the energy conservation assumption.

This study proposes a rigorous and coherent approach to modeling the dynamics of rigid bodies with variable energy under external perturbations, based on Hamiltonian mechanics. The method distinguishes between two components of energy during object motion: a constant energy component unaffected by external influences, which adheres to the principle of energy conservation, and a variable energy component that changes with variations in external conditions. Building upon this framework, the method incorporates targeted design of deep models to capture biases and achieve high-precision modeling of complex motions. The effectiveness of this approach is validated through experiments involving complex rigid-body motions, such as the guidance processes of aircraft and missiles.

3. Method

3.1. Hamiltonian. The Hamiltonian equation is one of the important mathematical tools for describing the motion of physical systems, developed based on Hamilton's principle. The Hamiltonian equation describes the laws governing the position and momentum of a system in a generalized coordinate space as a function of time, and its form is as follows:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \quad (1)$$

where q_i represents the generalized coordinates of the system, p_i represents the generalized momentum of the system, $H(q_i, p_i)$ represents the Hamiltonian of the system, and t represents time. These two equations can be referred to as Hamiltonian equations.

Hamiltonian is the energy function of the system, which can be calculated using the Lagrangian $L(q_i, \dot{q}_i)$ of the system and generalized momentum p_i :

$$H(q_i, p_i) = \sum_i p_i \dot{q}_i - L(q_i, \dot{q}_i). \quad (2)$$

The physical significance of the Hamiltonian equation describes the laws governing the motion of physical systems in the generalized coordinate and momentum spaces. The first equation indicates that the rate of change of generalized

coordinates with respect to time is equal to the partial derivative of the Hamiltonian with respect to the generalized momentum, that is, the direction of the system's motion in the generalized momentum space. The second equation indicates that the rate of change of the generalized momentum with respect to time is equal to the negative partial derivative of the Hamiltonian with respect to the generalized coordinates, that is, the direction of the motion of the system in the generalized coordinate space.

The Hamiltonian equation has a wide range of applications in classical mechanics. It can be used to solve many practical problems, such as describing the motion of celestial bodies, gas dynamics, and the motion of electromagnetic fields. The simple Hamiltonian equation of an undamped spring is represented by following formulas (3)–(5). Furthermore, the Hamiltonian equation is a classical correspondence of the Hamiltonian operator in quantum mechanics, which has important theoretical significance:

$$H = \frac{p^2}{2m} + \frac{1}{2}kx^2. \quad (3)$$

Here, H is the Hamiltonian, p is the particle's momentum, m is the mass, k is the spring constant, and x is the displacement of the particle. According to Hamiltonian mechanics, the evolution of momentum and position can be determined using the following equations:

$$\frac{dx}{dt} = \frac{\partial H}{\partial p} = \frac{p}{m}, \quad (4)$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial x} = -kx. \quad (5)$$

3.2. Six Degrees of Freedom Equations for Rigid Bodies. Classical mechanics mainly includes Newtonian mechanics, Lagrangian mechanics, and Hamiltonian mechanics, which primarily describe the motion laws of macroscopic objects under the influence of forces. The 6-DoF equations for a rigid body use Newtonian mechanics to describe the motion state of a rigid object in a three-dimensional space, including changes in its position and attitude. In these equations, the rigid body has three translational degrees of freedom and three rotational degrees of freedom, allowing the object to move and rotate arbitrarily in space. Generally, the six degrees of freedom equations for a rigid body consist of linear motion equations and angular momentum equations, expressed as follows.

Linear motion equations are as follows:

- (1) Force synthesis: $\sum \vec{F} = m\vec{a}$
- (2) Center of mass acceleration: $\vec{a} = (d^2\vec{r}/dt^2)$

Angular momentum equations are as follows:

- (1) Torque synthesis: $\sum \vec{M} = (d\vec{L}/dt)$
- (2) Angular momentum: $\vec{L} = I\vec{\omega}$
- (3) Angular velocity: $\vec{\omega} = (d\vec{\theta}/dt)$

Here, \vec{F} is the external force acting on the rigid body, m is the mass of the rigid body, \vec{a} is the acceleration of the center of mass of the rigid body, \vec{r} is the position vector of

the center of mass of the rigid body, \vec{M} is the torque acting on the rigid body, \vec{L} is the angular momentum of the rigid body, I is the moment of inertia of the rigid body, $\vec{\omega}$ is the angular velocity of the rigid body, and $\vec{\theta}$ is the rotation angle vector of the rigid body.

The relationship between the six degrees of freedom equations and Hamilton's equations can be established through Lagrangian equations. Lagrangian equations are a type of equation used to describe the motion of objects, derived from the principle of least action. By introducing generalized coordinates and generalized momenta, Lagrangian equations can be transformed into Hamilton's equations. For a 6-DoF rigid body, Lagrangian equations can be used to describe its motion laws, and then, the generalized coordinates and generalized momenta can be used to transform them into Hamilton's equations. Therefore, the six degrees of freedom equations for the dynamic model of a rigid body can naturally apply the Hamiltonian representation learning to capture inductive biases.

3.3. Hamiltonian Neural Network Missile Dynamic Modeling Based on Energy Variation Estimation. The energy variation of a 6-DoF rigid body is influenced by various factors related to both the body itself and external conditions or external control variables. On the body side, factors such as the energy output of the propulsion system, attitude control, and guidance system affect the energy variation. In addition, the mass and movement speed of the body are important factors that influence its energy variation. Moreover, external conditions or external control variables, such as the dynamic and spatial characteristics of a target or external control force, also affect the body's energy variation. Therefore, when using deep learning for 6-DoF rigid-body dynamic modeling, it is necessary to incorporate the dynamic characteristics of the external conditions or external control variables. In other words, the establishment of a neural network mapping relationship takes the body's state and the state of external conditions or external control variables as inputs and outputs the body's state at the next moment; i.e., $f(S_t, C_t) \rightarrow S_{t+1}$ requires the introduction of dynamic information of the external conditions or external control variables into the model. In order to achieve bidirectional time sliding for this mapping function, i.e., $f^{-1}(f(S_t, C_t), C_{t+1}) = S_t$, the dynamic information of the external conditions or external control variables must be incorporated into the model.

Six-degree-of-freedom equations encompass the system state and system control variables, which can be influenced by external active control or external conditions. Therefore, when representing dynamic characteristics using a neural network, it effectively establishes a statistical mapping relationship between the joint distribution of the system state and system control variables and the distribution of future system states. In certain scenarios, the system's own control variables may not be directly observable but are influenced by external conditions. Hence, system control variables can be treated as latent variables, enabling the direct construction of a statistical mapping relationship between the

joint distribution of the system state and external conditions and the distribution of future system states.

Similar to HNN, to use the Hamiltonian mechanism to achieve time sliding in the 6-DoF rigid-body dynamic model, without relying on the manual design of the Hamiltonian equation, we need to build a deep-learning model to convert the body state into implicit generalized position and momentum vectors and then convert them into Hamiltonian. On this basis, we introduce the external conditions or influencing factors' dynamic information into the model to describe the variation of the body Hamiltonian with the external conditions or influencing factors' dynamic information. Finally, we train a decoder to restore the generalized position vector to the state of the body. The model can slide forward and backward along the time dimension with high-precision modeling. In other words, we divide f into four parts.

Similar to HNN, the achievement of time sliding in the 6-DoF rigid-body dynamic model using the Hamiltonian mechanism involves the construction of a deep-learning model. This model is responsible for converting the body state into implicit generalized position and momentum vectors, which are further transformed into a Hamiltonian. The manual design of the Hamiltonian equation is not relied upon in this process. The dynamic information of the external conditions or influencing factors is then introduced into the model to depict the variations of the body's Hamiltonian in response to these dynamic factors. Subsequently, a decoder is trained to restore the generalized position vector to the body's state. With this model, high-precision modeling enables forward and backward time sliding along the time dimension. In other words, f is divided into four parts.

3.3.1. Encoder Network

$$q_t, p_t = \text{Encoder}(S_t, C_t; \theta_{\text{enc}}), \quad (6)$$

where θ_{enc} represents the parameters of the encoder network, Encoder represents the encoder neural network, q and p represent the generalized coordinate vector and generalized momentum vector, respectively, and S_t and C_t represent the state vectors of the rigid body and external conditions at time t , respectively.

3.3.2. HAMILTON Network

$$h_t = \text{HAMILTON}(q_t, p_t; \theta_{\text{hnn}}), \quad (7)$$

where θ_{hnn} represents the parameters of the HAMILTON network, HAMILTON represents the Hamiltonian neural network, h represents the Hamiltonian, and q_t and p_t represent the generalized coordinate vector and generalized momentum vector, respectively.

3.3.3. Energy Network

$$\Delta e = \text{Energy}(S_t, q_t, p_t; \theta_{\text{ene}}), \quad (8)$$

where θ_{ene} represents the parameters of the energy network, Energy represents the energy neural network, S_{T_t} represents the state vector of the target at the current time, h represents the Hamiltonian, and Δe represents the energy change.

3.3.4. Decoder Network

$$S_{t+1} = \text{Decoder}(q_{t+1}; \theta_{\text{dec}}), \quad (9)$$

where θ_{dec} represents the parameters of the decoder network, Decoder represents the decoder neural network, q_{t+1} represents the generalized coordinate vector of the missile at the next time step, and S_{t+1} represents the state vector of the missile at the next time step.

The system structure consisting of these four parts is shown in Figure 1.

3.3.5. Positive-Negative Sliding Core. The generalized coordinate vector and the generalized momentum vector at the next time step are calculated as follows:

$$\frac{dq}{dt} = \frac{\partial h}{\partial p} + \frac{\partial \Delta e}{\partial p}, \quad (10)$$

$$\frac{dp}{dt} = \frac{\partial h}{\partial q} - \frac{\partial \Delta e}{\partial q}. \quad (11)$$

In the above equations, q and p represent the generalized coordinate and momentum vectors, respectively, h denotes the Hamiltonian, and Δe represents the energy change. $(\partial h / \partial p)$ and $(\partial h / \partial q)$ are partial derivatives of h with respect to p and q , respectively. Similarly, $(\partial \Delta e / \partial p)$ and $(\partial \Delta e / \partial q)$ are partial derivatives of Δe with respect to p and q , respectively.

We estimate the future state and historical state of dynamic systems from inferred values of the system position and momentum by numerically integrating the Hamiltonian. We explore Euler integration to estimate the value of a function at time $t + dt$ by incrementing the function's value with the value accumulated by the function's derivative, assuming it stays constant in the interval $[t, t + dt]$. For the estimation of future states, that is, the forward sliding of time, Euler integration takes the form:

$$q' = q_{t+dt} = q_t + dt \left(\frac{\partial h}{\partial p} + \frac{\partial \Delta e}{\partial p} \right) \Big|_{p=p_t}, \quad (12)$$

$$p' = p_{t+dt} = p_t - dt \left(\frac{\partial h}{\partial q} - \frac{\partial \Delta e}{\partial q} \right) \Big|_{q=q_t}. \quad (13)$$

For the estimation of historical states, that is, the backward sliding of time, Euler integration takes the form:

$$q' = q_{t-dt} = q_t - dt \left(\frac{\partial h}{\partial p} + \frac{\partial \Delta e}{\partial p} \right) \Big|_{p=p_t}, \quad (14)$$

$$p' = p_{t-dt} = p_t + dt \left(\frac{\partial h}{\partial q} - \frac{\partial \Delta e}{\partial q} \right) \Big|_{q=q_t}. \quad (15)$$

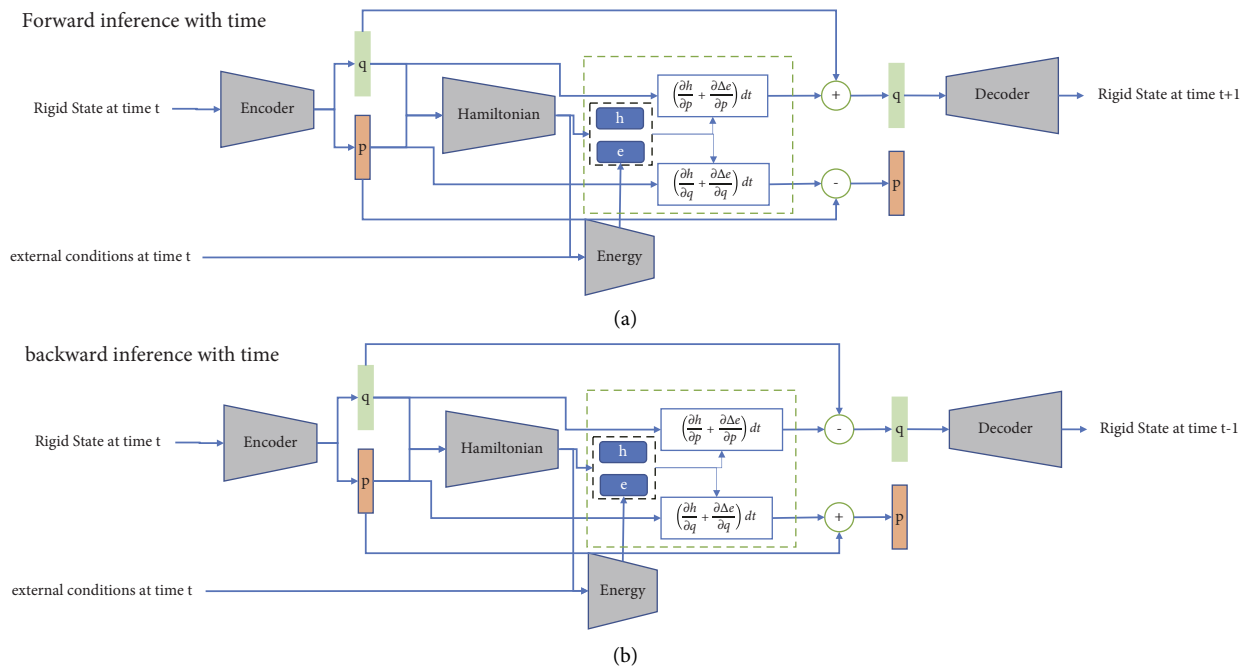


FIGURE 1: The transmission path of the rigid states and external conditions or control variables in VHNN. (a) Forward inference with respect to time and (b) backward inference with respect to time.

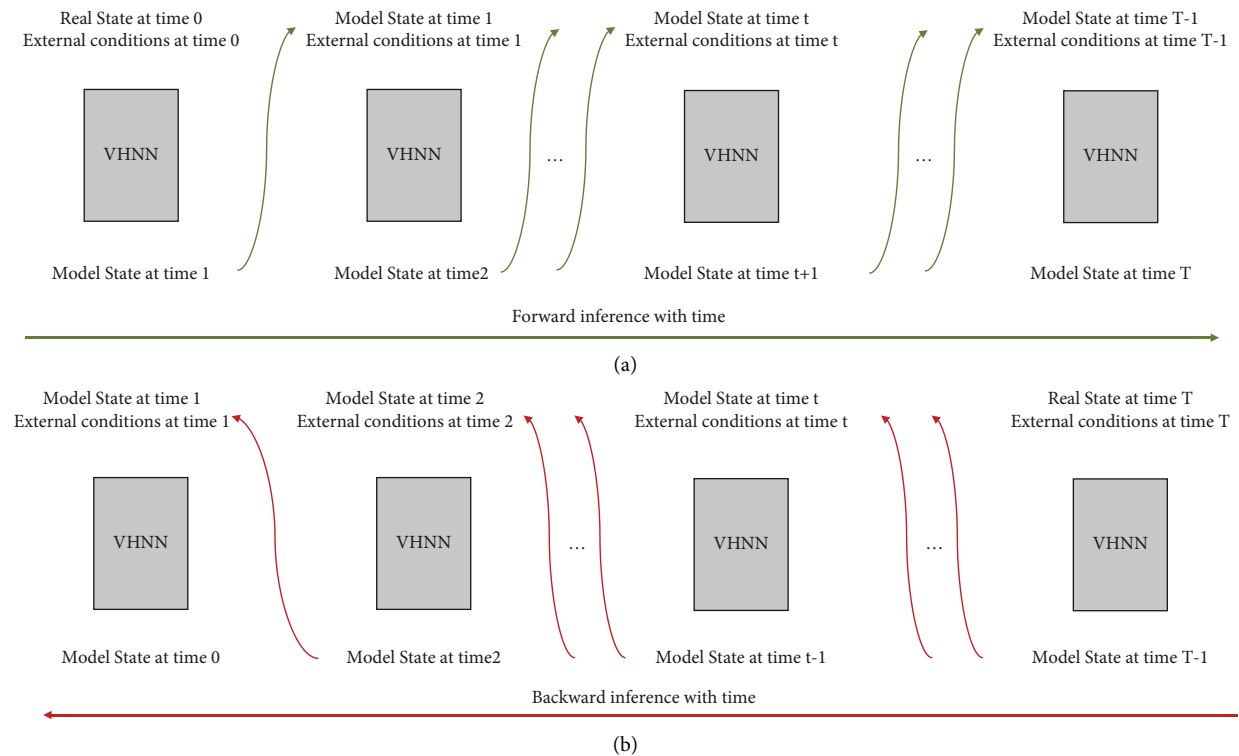


FIGURE 2: The VHNN network model performs inference over time and interdependence between the input and output. (a) Forward inference with respect to time and (b) backward inference with respect to time.

Require: $D = \{(S_t, C_t), S_{t+1}\}_{t=1: N}$
Ensure: $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}$
(1) Initialize $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}$.
(2) Loop until the loss converges:
(3) **while** loss not converged **do**
(4) $S_t, C_t \rightarrow$ Encoder, and output q and p .
(5) Input $q, p \rightarrow$ HAMILTON, and output h .
(6) Input $p, C_t \rightarrow$ Energy, and output Δe .
(7) Calculate the next state of the missile q', p' using formula (12) and (13).
(8) Input $q' \rightarrow$ Decoder, and output S'_{t+1} .
(9) Calculate the mechanical energy E from S'_{t+1} .
(10) Calculate the loss using (16) and gradient.
(11) Update all $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}$.
(12) **end while**
return $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}$

ALGORITHM 1: VHNN forward inference and training algorithm.

Require: $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}, \{C_t\}_{t=1: N}$
Ensure: D_{back}
(1) init $D_{back} = \{S_t\}$
(2) **for** t in $T: 1$ **do**
(3) $S_t, C_t \rightarrow$ Encoder, and output q and p .
(4) Input $q, p \rightarrow$ HAMILTON, and output h .
(5) Input $p, C_t \rightarrow$ Energy, and output Δe .
(6) Calculate the next state of the missile q', p' using formula (14) and (15)
(7) Input $q' \rightarrow$ Decoder, and output S_{t-1} .
(8) add S_{t-1} to D_{back}
(9) **end for**
return $\theta_{enc}, \theta_{hnn}, \theta_{ene}, \theta_{dec}$

ALGORITHM 2: VHNN backward inference.

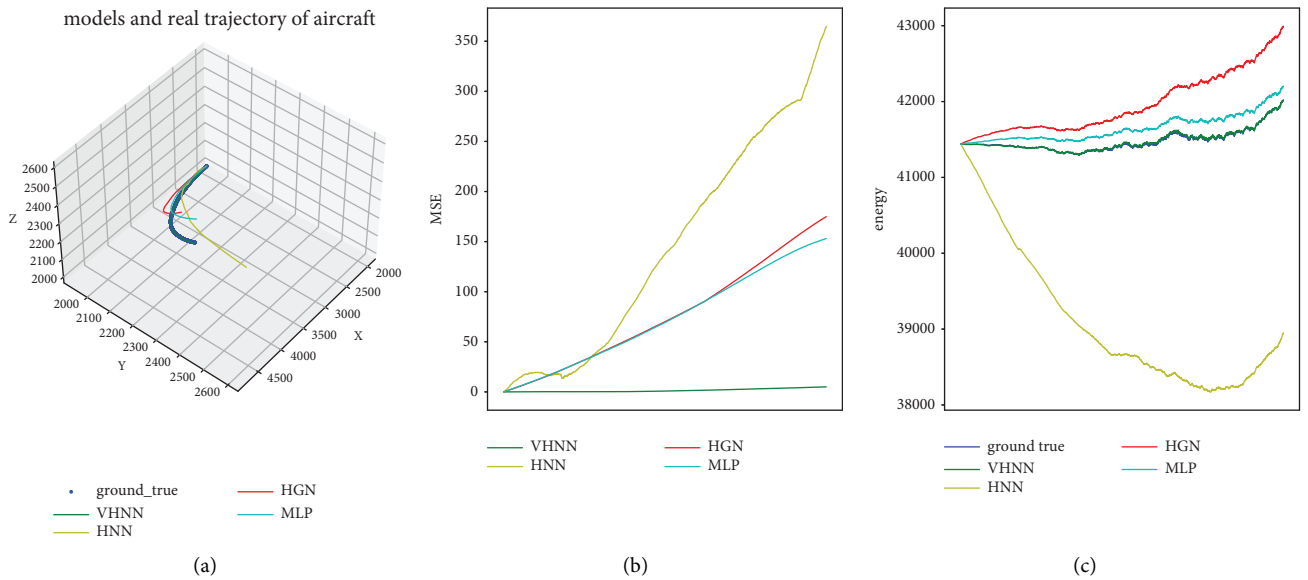


FIGURE 3: The trajectories of all comparative models and the actual model under identical initial conditions, with aircraft performing the same random action sequences (a). In addition, the mean absolute error (MAE) of all models' coordinates in relation to the actual model (b) and the energy-time variation curves of all models and the actual aircraft model (c).

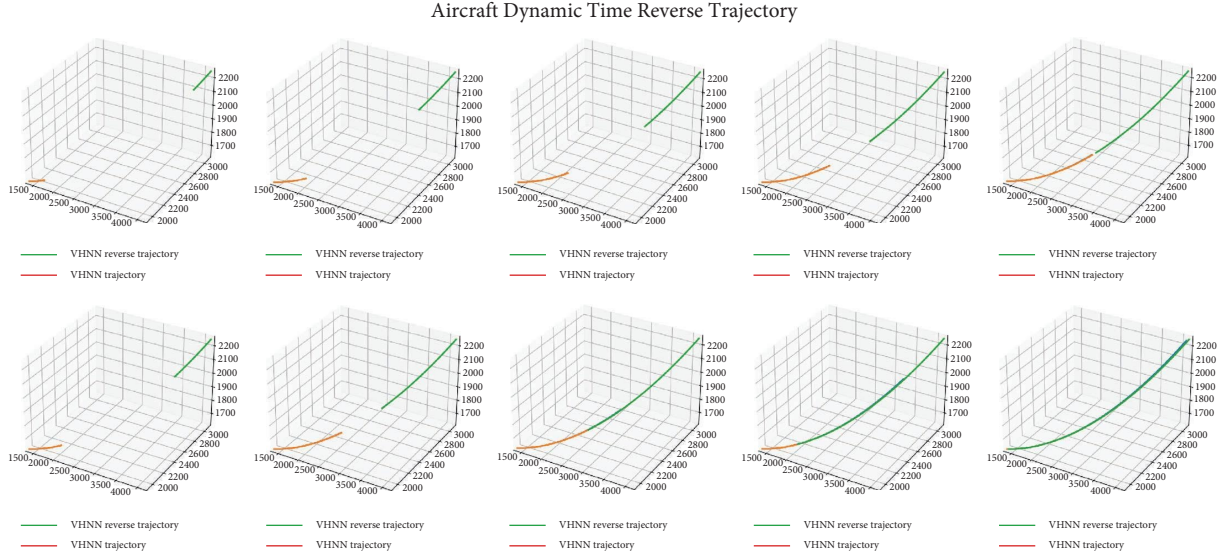


FIGURE 4: A comparison of forward and backward extrapolation over time in the aircraft modeling experiment using the VHNN model.

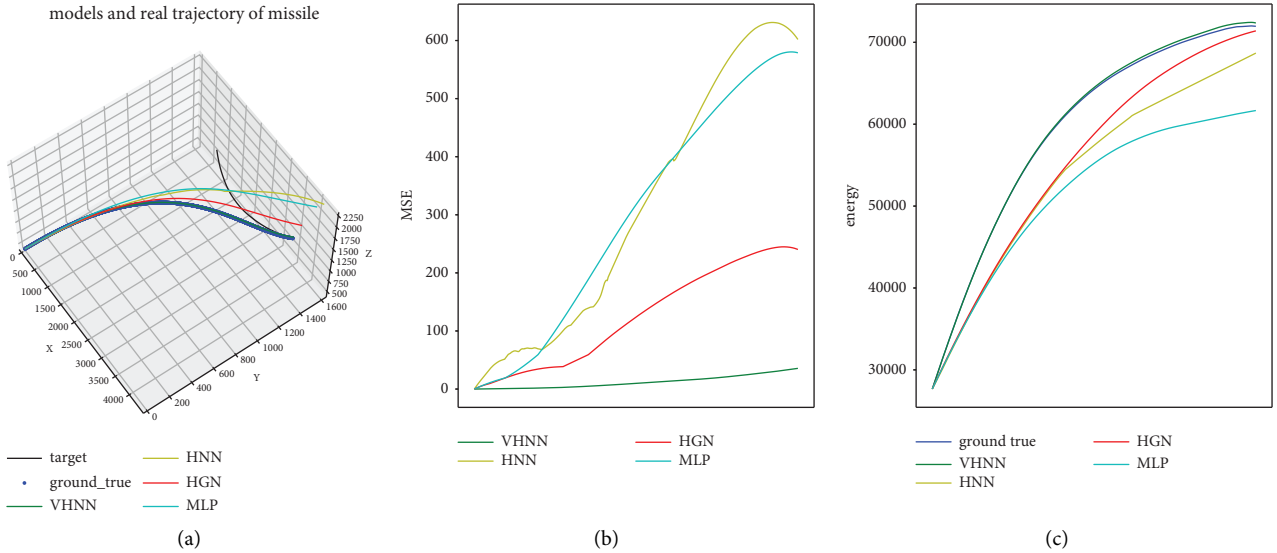


FIGURE 5: The trajectories of various comparative models and the actual model when the missile and targeted object share identical initial states, and the targeted aircraft performs the same random action sequences (a). Moreover, the mean absolute error (MAE) of all models' coordinates compared to the actual model (b) and the energy-time variation curves of all models along with the actual missile model (c).

The objective function of design philosophy of the loss function is to approach the true value in terms of both state and energy, which can be expressed as “encouraging the inferred posterior to match a prior:”

$$\begin{aligned}
 \mathcal{L}(\theta_{\text{enc}}, \theta_{\text{hnn}}, \theta_{\text{ene}}, \theta_{\text{dec}}; S_i, C_i, S_{i+1}) \\
 = \frac{1}{N} \sum_{i=1}^N \left[E_{P_{\theta_{\text{enc}}}(q_i^S)} \left[\log P_{\theta_{\text{hnn}}, \theta_{\text{ene}}, \theta_{\text{dec}}}(S'_i | q_i) \right] \right. \\
 \left. + KL(P_{\theta_{\text{enc}}, \theta_{\text{hnn}}, \theta_{\text{ene}}}(h + \Delta e | S_i) \| E_m(S'_i)) \right] \quad (16)
 \end{aligned}$$

The first term of the equation represents the negative log-likelihood expectation of the future system state S'_i given the current system state S_i and generalized positional variables. It quantifies the reconstruction error between the predicted system state and the true system state. By minimizing this term, the model aims to improve the accuracy of sequence generation. The second term represents KL divergence between the conditional distribution of the sum of future system-conserved energy and the change in energy relative to the current system state and the true energy distribution of the future system. It encourages the latent representation to approximate the

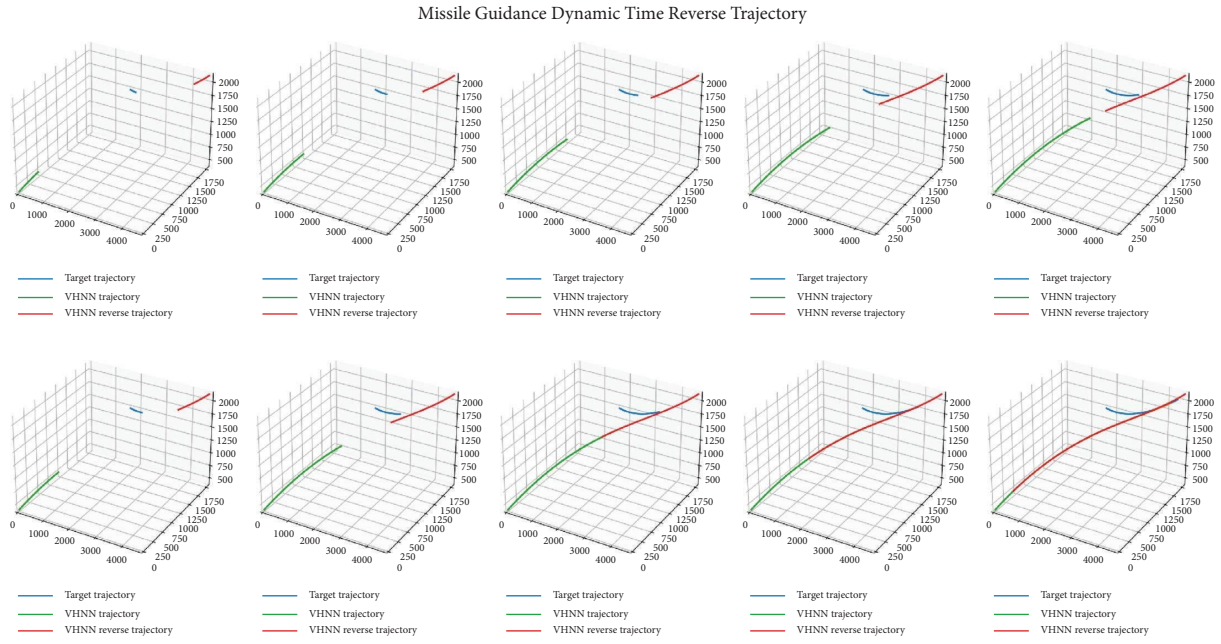


FIGURE 6: Comparison between forward and backward extrapolation over time in the VHNN model during the missile guidance modeling experiment.

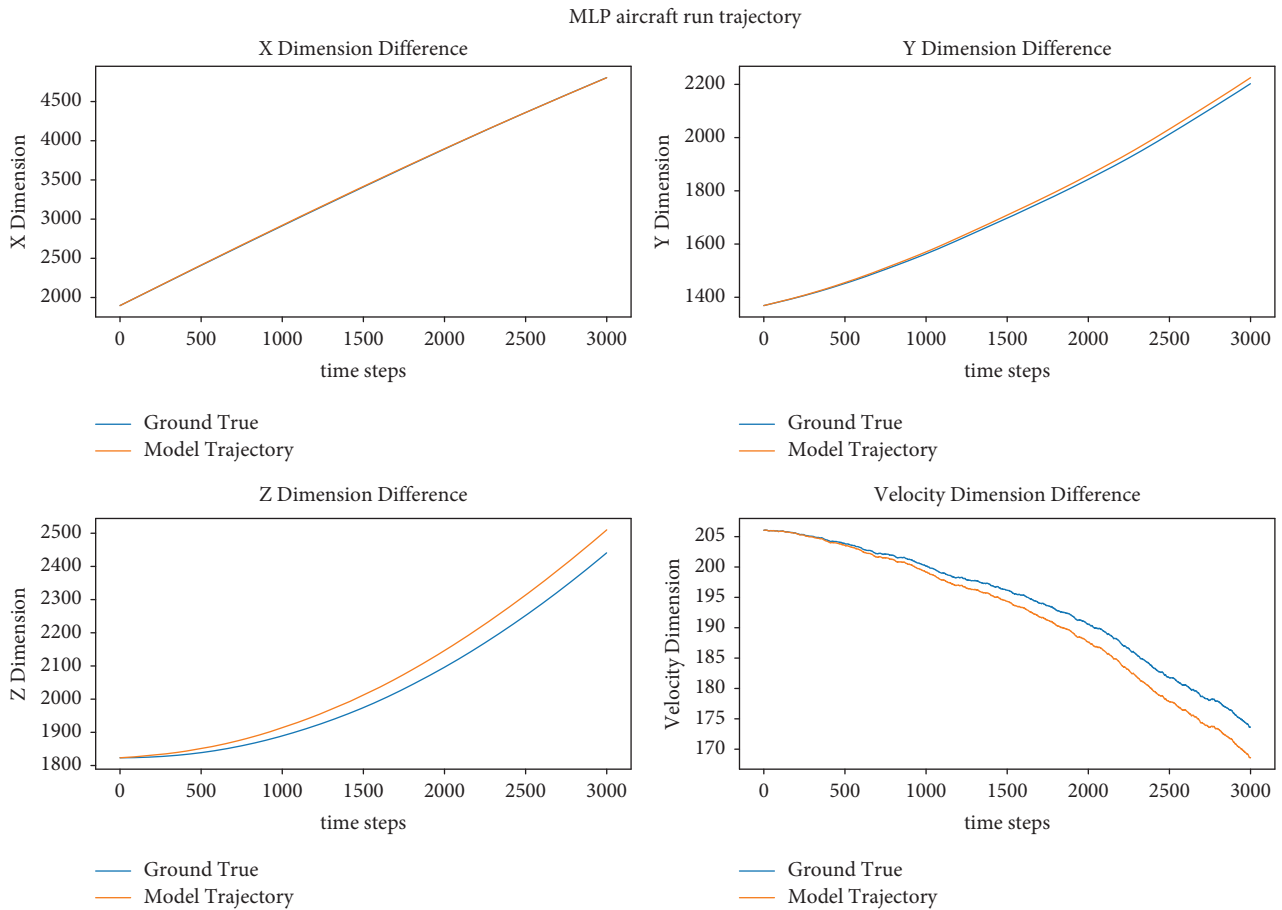


FIGURE 7: Aircraft dynamic modeling experiment simulation trajectory of the MLP model, incorporating differences in the Cartesian coordinate system's three-axis coordinates and disparities in velocity magnitudes.

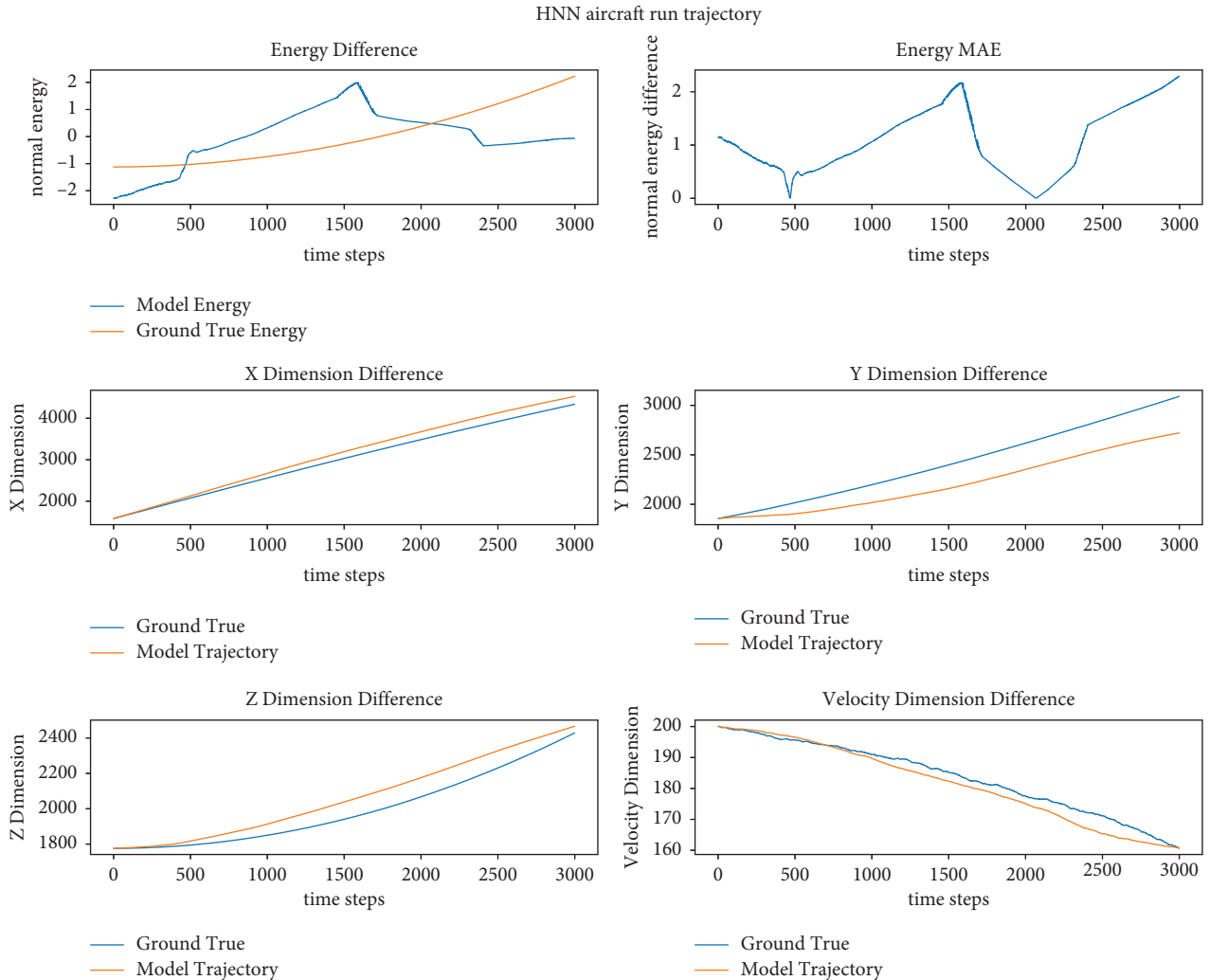


FIGURE 8: Aircraft dynamic modeling experiment simulation trajectory of the HNN model, comprising normalized energy discrepancies, mean absolute error (MAE) of normalized energies, distinctions in the Cartesian coordinate system’s three-axis coordinates, and disparities in velocity magnitudes.

prior distribution. Minimizing this term helps the model learn additional latent representations of energy changes and inductive biases.

The forward inference and training algorithms of VHNN as well as the backward inference algorithm are shown in Algorithm 1 and 2. Figures of forward and reverse deduction are shown in Figures 1 and 2.

4. Experiment

In order to verify the effectiveness of the proposed rigid-body dynamic modeling method, we selected two complex rigid-body 6-DoF dynamic equations as experimental subjects and conducted comparative experiments using the multilayer perceptron (MLP), Hamiltonian neural network (HNN), Hamiltonian graph network (HGN), and proposed variational Hamiltonian neural network (VHNN) method. The experimental subjects are derived from the 6-DoF aircraft equation model in [7] and a missile guidance model that includes proportional navigation. Both tasks belong to high-precision regression of rigid-body dynamic

models, as the error at each moment accumulates over time and has a cumulative effect on future states. This requires deep modeling to have a smaller single-step mean absolute error (MAE) and a feedback correction mechanism for a long-term cumulative error. That is, the deep model is required to learn the inherent laws of rigid-body dynamics, as shown in Algorithm 1 and 2.

4.1. Deep Modeling Experiment of 6-DoF Aircraft Equations.

First, the advantages of our method were demonstrated using the data generated by the 6-DoF aircraft model in [7]. The experimental subject was a 6-DoF aircraft model with state variables including the position, velocity, heading angle, and pitch angle, and control variables were accelerations in three directions. Our task was to ensure that the response generated by the deep model was as consistent as possible with the real aircraft model when any control variable was applied at any given current state. Moreover, it was hoped that the final state was as consistent as possible with the model-generated final state after a given initial

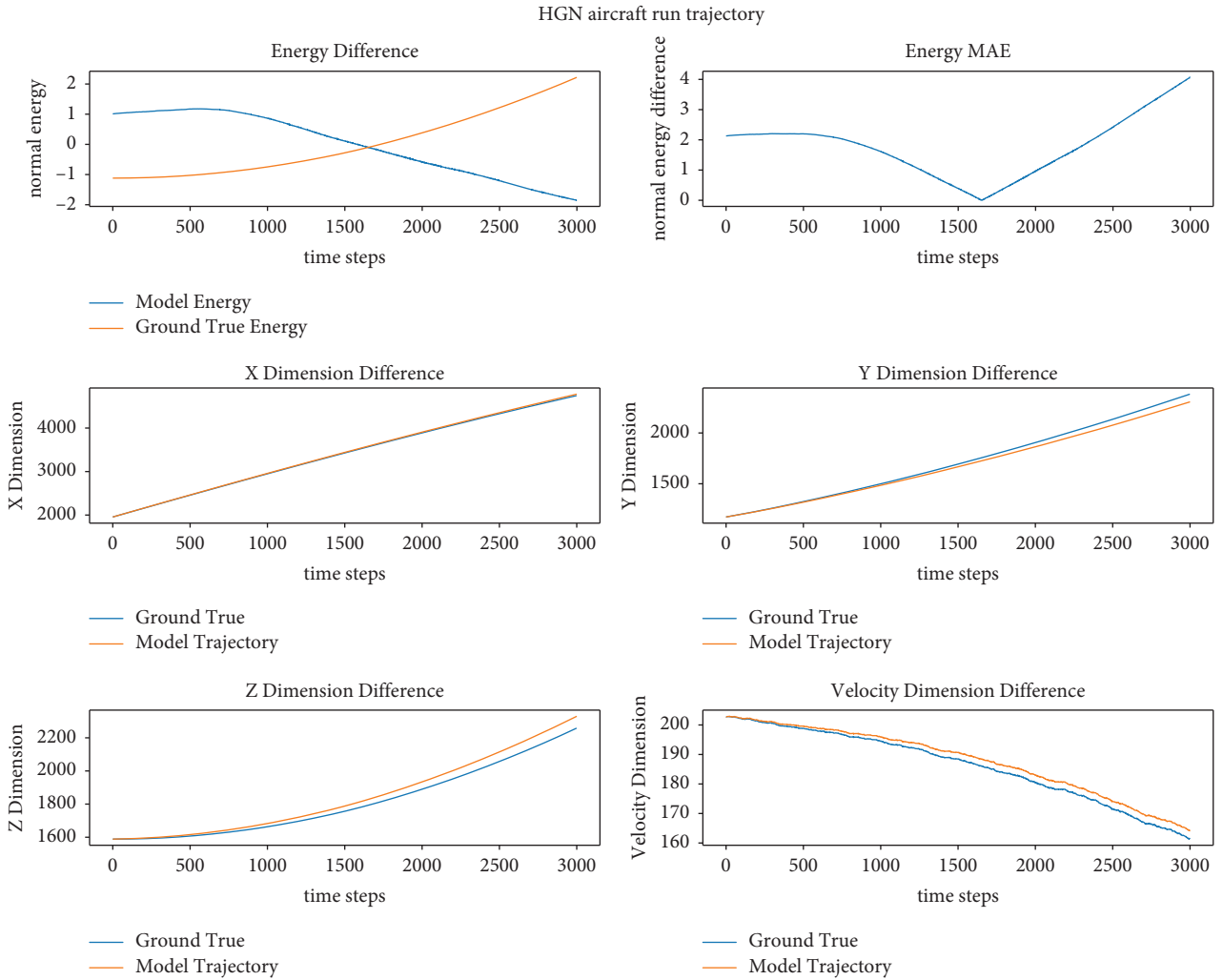


FIGURE 9: Aircraft dynamic modeling experiment simulation trajectory of the HGN model, encompassing normalized energy discrepancies, mean absolute error (MAE) of normalized energies, variations in the Cartesian coordinate system’s three-axis coordinates, and disparities in velocity magnitudes.

state and a sequence of control variables at equal intervals for a period of time. Experiments were conducted using MLP, HNN, HGN, and the proposed VHNN method, with specific experimental settings detailed in Appendix A. The comparison of various methods is shown in Figure 3.

In Figure 3, the 3D trajectories of different models show that the real aircraft trajectory can be almost perfectly cloned by the VHNN method proposed in this paper, while the trajectory begins to diverge from the first half by both MLP and HGN, but the overall trend is close to the real aircraft trajectory. A significant deviation from the beginning of the trajectory is produced by the HNN model. In terms of the mean absolute error (MAE) of the coordinates, a relatively low level is maintained by VHNN and HGN, with an MAE close to 0 throughout for VHNN, while HGN begins to increase linearly slightly after 1500 steps. The process of error accumulation is clearly shown by the MAE curve of MLP, while HNN is in a divergent state. In terms of energy correlation, the highest degree of overlap with the real aircraft trajectory is achieved by VHNN, indicating that

there is a causal relationship between VHNN’s good energy control and MAE regression accuracy. Energy conservation is tended to be maintained in the first half by HNN, leading to excessive deviation accumulation, and energy is in a divergent state in the second half. More charts about this experiment can be found in Appendix A.

Like other Hamiltonian-based deep modeling methods, the characteristic of time-reversed inference is exhibited by VHNN. For related charts of time-reversed inference, please refer to Figure 4 in Appendix A.

The experimental results indicate that (1) in the case of aircraft, where a rigid-body motion equation accepts external control variables, energy plays a crucial role in dynamic changes and (2) VHNN achieves high-precision modeling of rigid-body dynamics controlled by external conditions due to the introduction of inductive bias for energy changes.

4.2. *Proportional Navigation Missile Guidance Deep Modeling Experiment.* Subsequently, the slightly more complex rigid-body dynamic model of a missile was employed as the

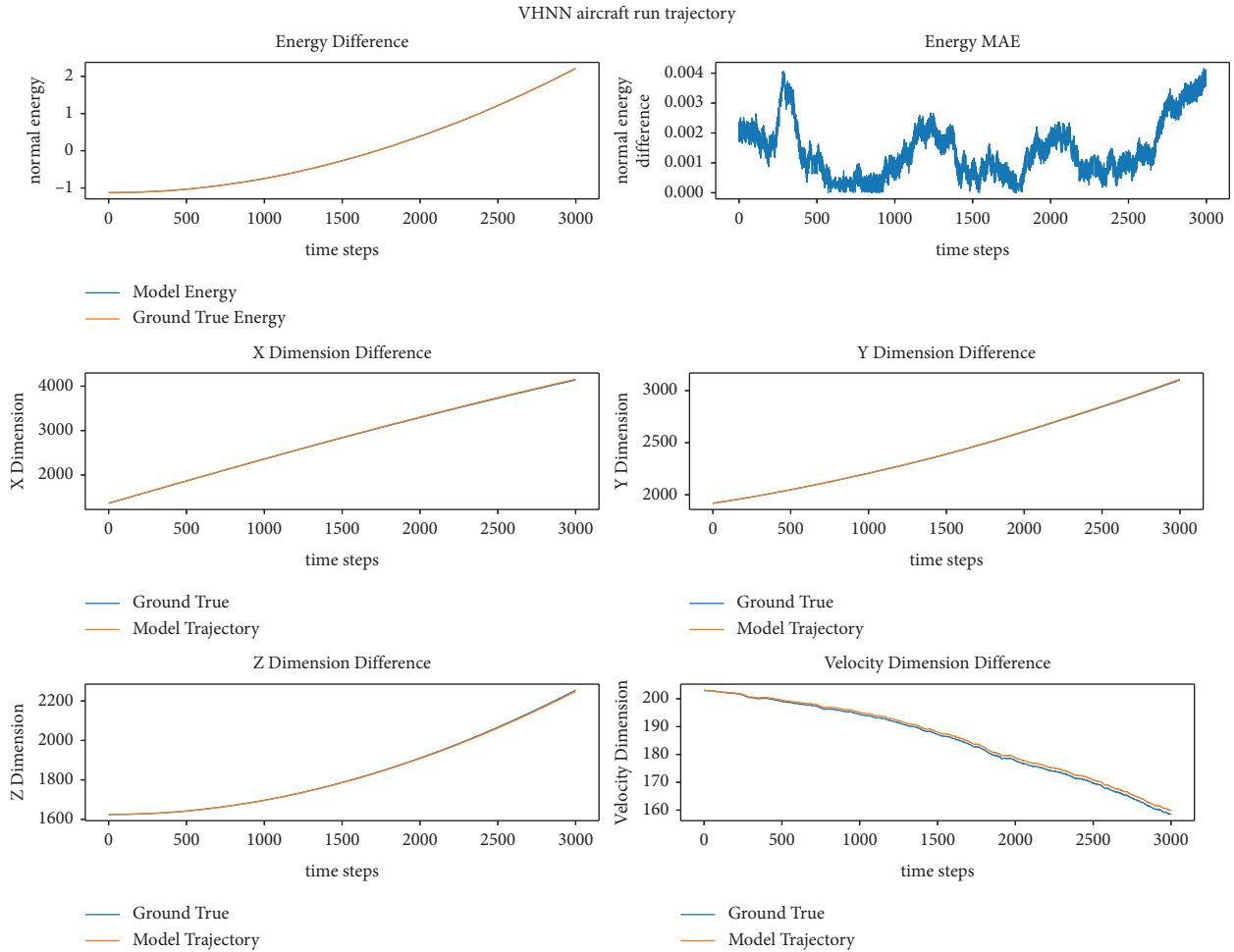


FIGURE 10: Aircraft dynamic modeling experiment simulation trajectory of the VHNN model, including normalized energy discrepancies, mean absolute error (MAE) of normalized energies, differences in the Cartesian coordinate system’s three-axis coordinates, and disparities in velocity magnitudes.

experimental subject, with the control variables implicitly determined by the dynamic target being attacked and generated by a proportional navigation algorithm based on the changes in the relative position of the attacked target and the missile. This results in corresponding control variables that enable the missile to intersect and complete the attack on the target’s motion trajectory. The effectiveness of our method was validated using the 6-DoF missile model combined with the proportional navigation algorithm in [7]. The state variables of this model include the current position, velocity, heading angle, and pitch angle of the missile, with the state of the attacked target considered as control variables, namely, the position, velocity, heading angle, and pitch angle of the attacked target. Our task is to ensure that the next moment’s missile state generated by the deep model is as consistent as possible with the original model when any given missile and attacked target’s current state are provided. Moreover, we aim for the final state to be as consistent as possible with the model-generated final state after a given initial state of the missile and the attacked target, along with a sequence of motion states of the attacked target at equal intervals for a period of time. Specific experimental settings

can be found in Appendix A. The comparison of various methods is shown in Figure 5.

The figure demonstrates that the VHNN model remains the best-performing model, almost perfectly replicating the response of the original model and achieving an almost perfect overlap with the original model trajectory throughout the entire missile guidance cycle. In terms of the MAE curve, although VHNN exhibits a slight deviation in the later stage, it still maintains a very low level. MLP and HNN, on the other hand, display larger deviations. From the energy curve, VHNN continues to achieve the best overlap with the real missile energy curve. More charts about this experiment can be found in Appendix A. For related charts of time-reversed inference, please refer to Figure 6 in Appendix A.

The experimental results demonstrate that (1) whether the external control variables of the rigid-body dynamic model are implicit and unknown but can be influenced by certain other conditions, estimating the overall energy change of dynamics remains highly beneficial for modeling under such condition-based dynamics. For instance, the dynamic control variables of

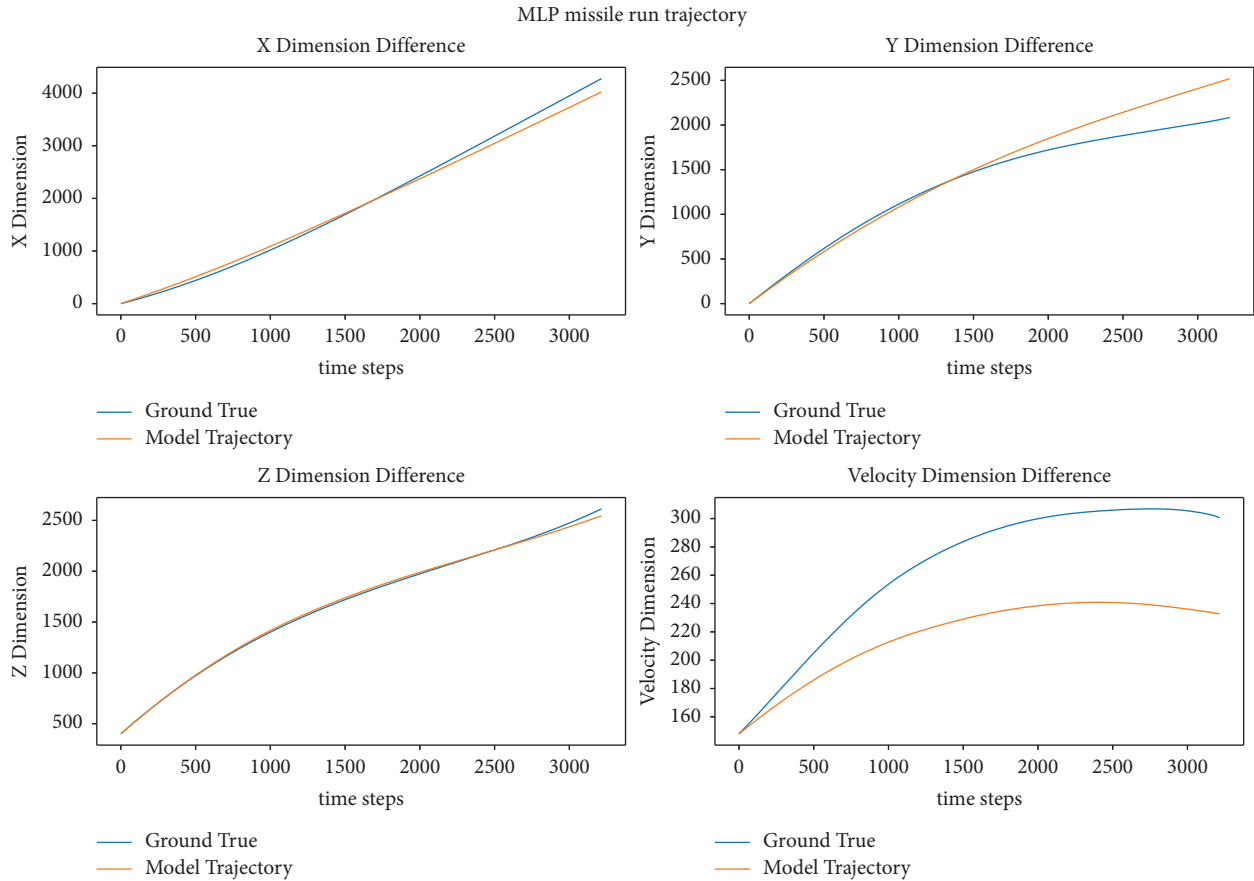


FIGURE 11: Trajectory simulation of the missile guidance dynamic modeling experiment using the MLP model, including the differences in the Cartesian coordinate system's three-axis coordinates and the differences in velocity scalar magnitudes.

missiles are calculated using a proportional navigation algorithm based on the state of the attacked target and their own state, although this is not explicitly considered in deep modeling; (2) the VHNN model still exhibits good performance for deep modeling of such implicit external control variables.

5. Discussion and Future Work

The experimental results of this study demonstrate that significant advantages have been achieved in the modeling of rigid-body 6-DoF equations through our proposed Hamiltonian-based method (VHNN). In both experiments, higher accuracy and lower errors are exhibited by the VHNN model, particularly in terms of trajectory overlap with the original model, surpassing other methods.

First, during the 6-DoF modeling experiment of aircraft, it was discovered that the dynamic changes of rigid-body motion equations, subjected to external control inputs like those found in aircraft, are significantly influenced by energy. The VHNN model, which incorporates inductive bias for energy changes, enables high-precision modeling of rigid-body dynamics controlled by external conditions. In comparison to other

methods, the VHNN model demonstrates superior performance in terms of the mean absolute error (MAE) and energy correlation.

Second, in the deep modeling experiment of missile proportional guidance, it was found that even when the external control input of the rigid-body dynamics model is implicit and unknown, estimating the overall energy changes in dynamics remains highly advantageous for modeling such condition-based dynamics. The VHNN model retains its effectiveness in deep modeling of implicit external control inputs, outperforming other methods in both MAE curves and energy curves.

The experimental results of this study indicate that the VHNN model exhibits significant advantages in addressing complex rigid-body motion problems encountered in aircraft and missile guidance processes. This offers a novel approach to solving practical engineering applications' rigid-body dynamics modeling problems. In the military domain, deep dynamic models of missiles and fighter jets can be seamlessly integrated with deep reinforcement learning to holistically generate control strategies. Specifically, for missile deep modeling, the VHNN model can accurately extrapolate the positions of numerous incoming missiles in batches, which is of great

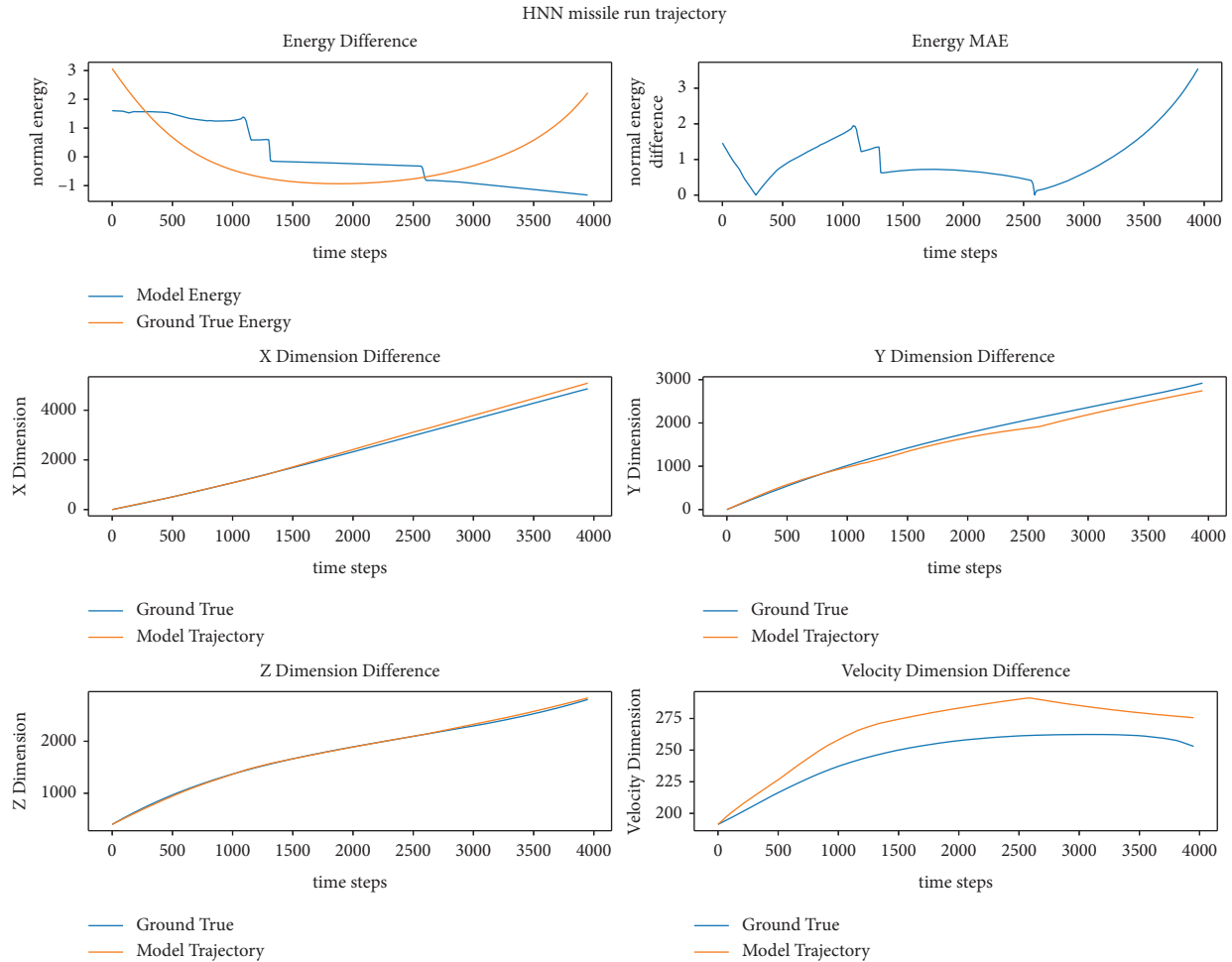


FIGURE 12: Trajectory simulation of the missile guidance dynamic modeling experiment employing the HNN model, encompassing the disparities in normalized energy, normalized energy's mean absolute error (MAE), deviations in the Cartesian coordinate system's three-axis coordinates, and the dissimilarities in velocity scalar magnitudes.

importance for offensive and defensive purposes. The Hamiltonian-based deep modeling method also possesses the additional capability of performing backward inference over time. This implies that given the current state of a missile, its historical trajectory can be deduced conveniently, enabling rapid localization of missile launch coordinates and targeted countermeasures.

However, it is important to note that this study solely focuses on the experimental validation of aircraft and missile motion, which is a typical example of rigid-body motion. In the missile guidance modeling experiment, the modeling

issue of unobservable system control variables was initially considered. These two experimental cases represent complex rigid-body motions and serve as highly representative examples, demonstrating the capability of our approach to perform deep modeling of complex rigid bodies with variable system energy in the presence of unobservable control variables. Nonetheless, further exploration is required in practical applications to establish an inductive bias for the VHNN model when dealing with incomplete or missing system states. Particularly, when system states are incomplete or unobservable, it becomes necessary to address

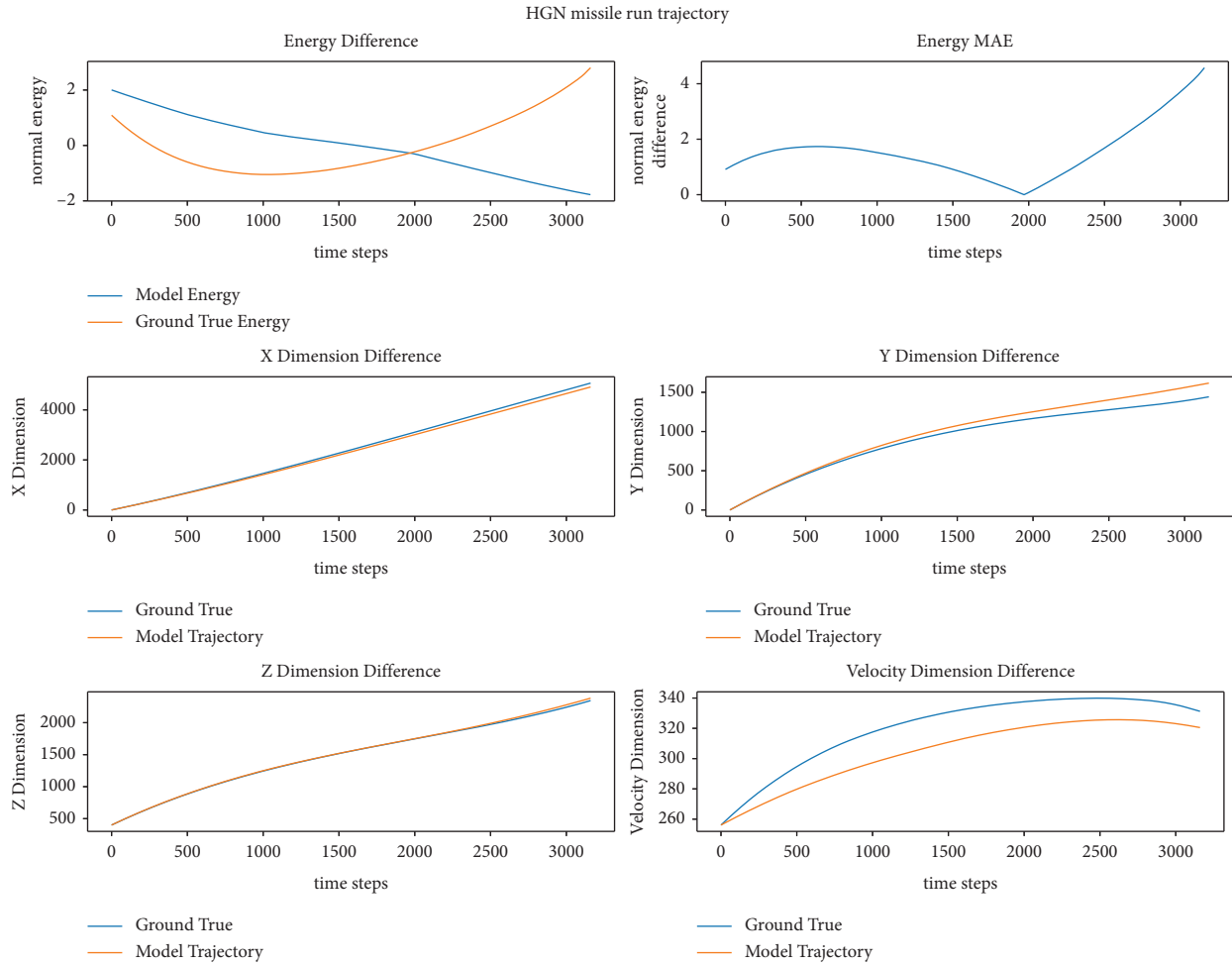


FIGURE 13: Trajectory simulation of the missile guidance dynamic modeling experiment using the HGN model, comprising the discrepancies in normalized energy, normalized energy’s mean absolute error (MAE), variations in the Cartesian coordinate system’s three-axis coordinates, and the distinctions in velocity scalar magnitudes.

how to represent these incomplete system states within the neural network and design appropriate optimization functions that induce the model to learn the correlation between the missing system states and the observed ones. Currently, one possible approach we consider is deepening the design

of deep models using time-series inference methods in a latent variable space to tackle this problem. In addition, further research can be conducted on parameter optimization and structural design of the VHNN model to enhance its performance and applicability.

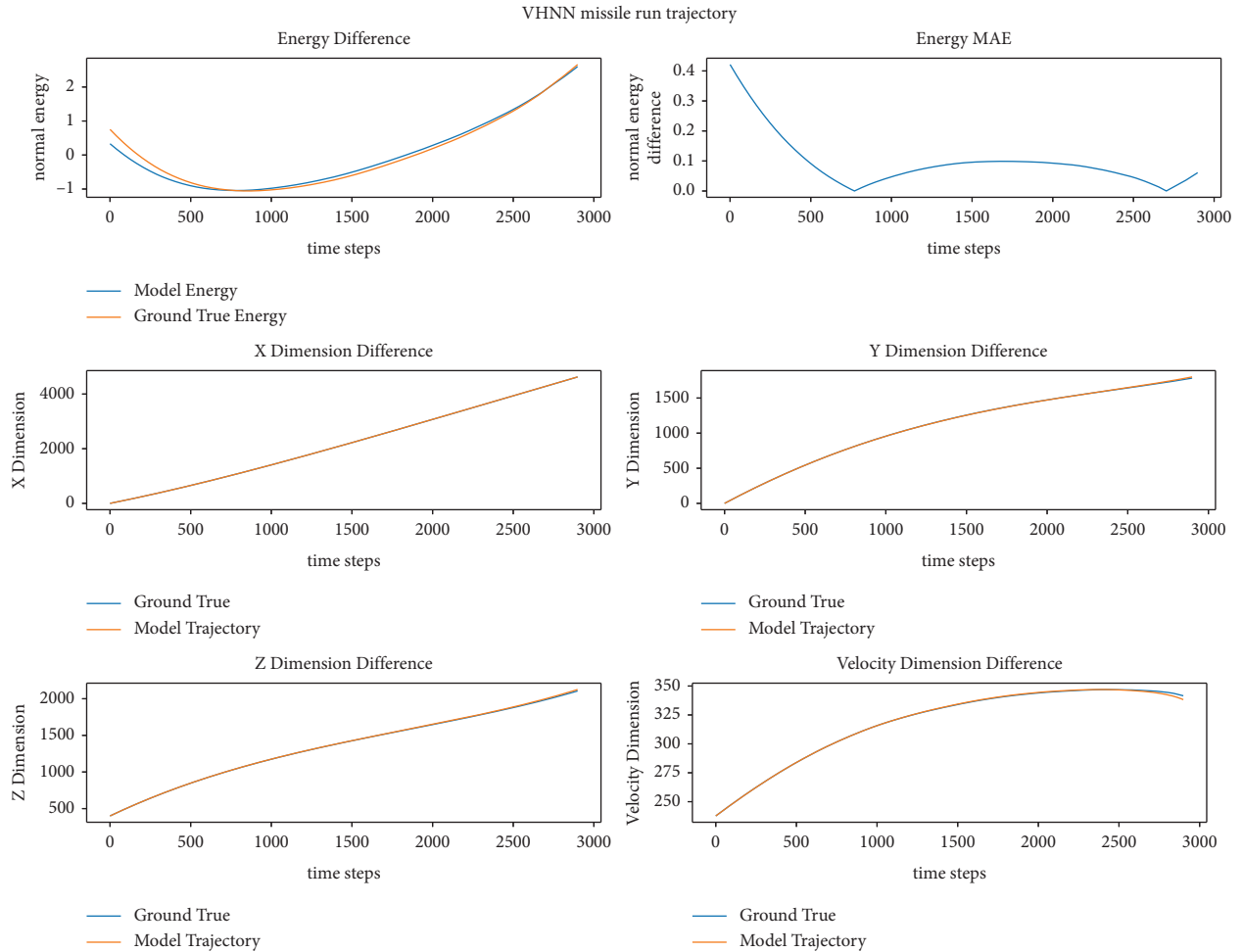


FIGURE 14: Trajectory simulation of the missile guidance dynamic modeling experiment conducted with the VHNN model, incorporating the differences in normalized energy, normalized energy's mean absolute error (MAE), fluctuations in the Cartesian coordinate system's three-axis coordinates, and the discrepancies in velocity scalar magnitudes.

6. Conclusion

A deep modeling approach for rigid-body dynamics based on Hamiltonian dynamic neural networks is proposed in this study. The temporal variations in energy, including factors and rules influencing energy changes, are captured by incorporating an inductive bias into our method. The concept of Hamiltonian neural networks serves as the foundation for this approach. The superior accuracy and feedback correction achieved by our method are demonstrated through experimental modeling of the 6 degrees of freedom dynamics of aircraft and missiles. It is argued that this approach is particularly suitable for extracting fundamental physical laws governing rigid-body dynamics under the influence of external control variables. More accurate and efficient model support for precise rigid-body motion control is provided, especially in domains and military applications that necessitate large-scale distributed predictive control of system dynamics, batch retrospective of historical states through dynamics, and the construction of differentiable simulation models.

Moving forward, further optimization of our method is aimed at enhancing the accuracy and robustness of the model, enabling more precise and efficient rigid-body control. In addition, integration of this method into a model-based reinforcement learning framework is planned to explore further application directions.

Appendix

A. Experimental Setup and Plots

A.1. Dataset Generation

A.1.1. Aircraft Dynamic Modeling Experiment. Given the initial position of an aircraft, the aircraft model in [7] calculates the guidance behavior based on the current state and control commands and outputs the next state of the aircraft. The control commands consist of six types of discrete instructions, each of which is composed of accelerations in the x , y , and z directions of the Cartesian coordinate system. These six instructions represent level flight at constant speed, horizontal right turn, horizontal left turn, climbing, diving,

and horizontal acceleration. The data features include two parts: (1) the one-hot representation of the aircraft's control command index at the current moment and (2) the current state of the aircraft, including the x , y , and z positions in the Cartesian coordinate system, scalar velocity, pitch angle, and heading. The data labels not only contain the aircraft state at the next moment (0.005 seconds later) but also include the x , y , and z positions, scalar velocity, pitch angle, and heading in the Cartesian coordinate system.

Each flight trajectory lasts 3,000 time steps, with the initial state of the aircraft randomly distributed in the x , y , and z directions of the Cartesian coordinate system within a range of 1,000 meters. The entire dataset consists of 200 random flight trajectories, approximately 600,000 samples, of which 80% are used for training and 20% for testing.

A.1.2. Missile Dynamic Modeling Experiment. The cloning or modeling object employed in this study is the missile model from [7], and the proportional guidance method is adopted. Given the initial state of the missile and the target, the missile model calculates the corresponding guidance maneuvers based on its current state and the relative state of the target and outputs the next state of the missile using the RK4 numerical integration method with an adjustable time step based on the proportional guidance formula. The missile state comprises six dimensions: coordinates x , y , z in the inertial coordinate system, scalar velocity, heading, and pitch angle. Similarly, the state of the target aircraft includes these six dimensions, represented by x , y , z , v , ψ , and γ . After providing the initial state of the missile and the target aircraft, the target aircraft can freely execute various maneuvers. The missile adjusts its three accelerations according to the different states generated by the target and ultimately intercepts the target aircraft.

The data features include two parts: (1) the current state of the target aircraft, including the x , y , and z positions in the Cartesian coordinate system, scalar velocity, pitch angle, and heading; (2) the current state of the missile, including the x , y , and z positions in the Cartesian coordinate system, scalar velocity, pitch angle, and heading. The data labels not only contain the missile state at the next moment (0.005 seconds later) but also include the x , y , and z positions, scalar velocity, pitch angle, and heading in the Cartesian coordinate system.

Since the time for the missile to attack the target aircraft is not fixed, each missile flight trajectory lasts between 2,000 and 4,000 time steps. To maintain generality, the missile launch coordinates are fixed at the origin, and the initial state of the aircraft is randomly distributed in the x , y , and z directions of the Cartesian coordinate system within a range of 1,000 meters. The entire dataset consists of 200 random flight trajectories, approximately 600,000 samples, of which 80% are used for training and 20% for testing.

A.2. Experimental Details. The VHNN model constructs a residual module with a three-layer MLP neural network with 256 hidden units, connecting the encoder, decoder, and Hamiltonian network or energy network. Each network uses this residual module as the main body to ensure smooth

gradient flow between different network structures. The encoder takes data features as inputs and outputs 32-dimensional generalized coordinates q and 32-dimensional generalized momenta p . The energy and Hamiltonian networks take q , p , and the target state as inputs and output a 1-dimensional abstract energy. The energy derivatives with respect to p and q are calculated using Hamilton's equations (10) and (11), and the next moment's generalized coordinates q' and momenta p' are obtained using Euler integration within the time interval Δt . The next moment's rigid-body state is then decoded by the decoder.

Four models are trained using the same dataset: a three-layer MLP with 512 hidden units, VHNN, and HNN and HGN models with network structures and hyperparameters similar to VHNN. It is worth noting that the original HGN paper primarily focuses on image sequences, but techniques related to image sequences are not necessary for the rigid-body model cloning task. Therefore, only the core idea of HGN is retained for comparison. In addition, to incorporate an input mechanism for external control variables, the conditional input is integrated into the encoder module based on the experimental feature requirements, while the decoder still outputs the next moment's state.

The energy comparisons of each model in Appendices B and C are formed by scaling the Hamiltonian or VHNN energy to the mechanical energy of the true trajectory, resulting in a contrast of energy differences.

A.3. Aircraft Dynamic Modeling Experiment Plots. Deep aircraft dynamic modeling experiment results are shown in Figures 4, 7–10.

A.4. Missile Guidance Dynamic Modeling Experiment Plots. Deep modeling experimental results of missile guidance dynamics are shown in Figures 6, 11–14.

Data Availability

All data have been generated with <https://github.com/ajfrewin/pn-guidance>, which allows anyone to regenerate the data used in this paper.

Disclosure

A preprint version [22] of this article was submitted and cited as a reference in the reference list.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Fei Simiao contributed equally to this work.

Acknowledgments

The authors would like to express their sincere gratitude to Professor Yang Yu from Nanjing University for his invaluable guidance and inspiration in the initial stages of this

research paper. The authors would also like to extend their appreciation to Dr. Tu, also from Nanjing University, for his insightful suggestions on feature design. Their expertise and support have been instrumental in shaping the direction and quality of this work.

References

- [1] T. Stachiw, A. Crain, and J. Ricciardi, "A physics-based neural network for flight dynamics modelling and simulation," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 9, no. 1, pp. 13–20, 2022.
- [2] D. Guo, Z. Xie, X. Sun, and S. Zhang, "Dynamic modeling and model-based control with neural network-based compensation of a five degrees-of-freedom parallel mechanism," *Machines*, vol. 11, no. 2, p. 195, 2023.
- [3] J. Zhang, T. Zhang, and Y. Wang, "Gnn-based structural dynamics simulation for modular buildings," in *Pattern Recognition and Computer Vision: 5th Chinese Conference, PRCV 2022*, pp. 245–258, Springer, Shenzhen, China, 2022.
- [4] D. Jia, Y. Su, and C. Li, "Deep convolutional neural network for 6-dof image localization," 2016, <https://arxiv.org/abs/1611.02776>.
- [5] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [6] P. Toth, D. J. Rezende, A. Jaegle, S. Racani`ere, A. Botev, and I. Higgins, "Hamiltonian Generative Networks," 2019, <https://arxiv.org/abs/1909.13789>.
- [7] A. Frewin, "pn-guidance," 2019, <https://github.com/ajfrewin/pn-guidance>.
- [8] M. Ramezanizadeh, M. H. Ahmadi, M. A. Nazari, M. Sadeghzadeh, and L. Chen, "A review on the utilized machine learning approaches for modeling the dynamic viscosity of nanofluids," *Renewable and Sustainable Energy Reviews*, vol. 114, Article ID 109345, 2019.
- [9] N. Sharma and Y. Liu, "A hybrid science-guided machine learning approach for modeling chemical processes: a review," *AIChE Journal*, vol. 68, no. 5, Article ID 17609, 2022.
- [10] Y. Wang, J. M. Lamim Ribeiro, and P. Tiwary, "Machine learning approaches for analyzing and enhancing molecular dynamics simulations," *Current Opinion in Structural Biology*, vol. 61, pp. 139–145, 2020.
- [11] N. E. Jackson, M. A. Webb, and J. J. de Pablo, "Recent advances in machine learning towards multiscale soft materials design," *Current Opinion in Chemical Engineering*, vol. 23, pp. 106–114, 2019.
- [12] R. Bhattoo, S. Ranu, and N. Krishnan, "Learning articulated rigid body dynamics with Lagrangian graph neural network," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29789–29800, 2022.
- [13] S. Wang, X. Shao, L. Yang, and N. Liu, "Deep learning aided dynamic parameter identification of 6-dof robot manipulators," *IEEE Access*, vol. 8, pp. 138102–138116, 2020.
- [14] D. Millard, E. Heiden, S. Agrawal, and G. S. Sukhatme, "Automatic Differentiation and Continuous Sensitivity Analysis of Rigid Body Dynamics," 2020, <https://arxiv.org/abs/2001.08539>.
- [15] H. Sun, T. Wang, and E. Yu, "A dynamic keypoint selection network for 6dof pose estimation," *Image and Vision Computing*, vol. 118, Article ID 104372, 2022.
- [16] M. Zhang, G. Taimuri, J. Zhang, and S. Hirdaris, "A deep learning method for the prediction of 6-dof ship motions in real conditions," *Proceedings Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 17, 2023.
- [17] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," in *International Conference on Machine Learning*, pp. 2078–2087, PMLR, New York, NY, USA, 2018.
- [18] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "Ffjord: free-form continuous dynamics for scalable reversible generative models," 2018, <https://arxiv.org/abs/1810.01367>.
- [19] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic Recurrent Neural Networks," 2019, <https://arxiv.org/abs/1909.13334>.
- [20] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, "Hamiltonian Graph Networks with Ode Integrators," 2019, <https://arxiv.org/abs/1909.12790>.
- [21] A. Sosanya and S. Greydanus, "Dissipative Hamiltonian Neural Networks: Learning Dissipative and Conservative Dynamics Separately," 2022, <https://arxiv.org/abs/2201.10085>.
- [22] S. Fei, L. Huo, Z. Sun et al., "Hamiltonian Neural Network 6-dof Rigid Body Dynamic Modeling Based on Energy Variation Estimation," *Research Square*, vol. 54, 2023.