An Application of Geometric Numerical Integration: Forced Variational Integrator Networks on S^3

MATH 273A Project

Hu Hanyang

2025-12-09

1 Introduction

The success of deep learning methods–particularly their flexibility, expressiveness, and remarkable ability to generalize from a finite dataset–may appear counterintuitive at first, as such models are often highly overparameterized yet still avoid overfitting. This can be attributed to the appropriate inductive biases embedded within the design of neural network architecture, which, according to the no-free-lunch theorem [11], are essential for achieving good generalization performance in specific problem domains. When applying neural networks to modeling physical systems, particularly Hamiltonian systems, it is also desirable to encode prior knowledge about the underlying physical laws into the model design.

Specifically, consider a Hamiltonian system

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \\ \dot{\mathbf{p}} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}, \end{aligned} \tag{1.1}$$

where $\mathbf{q}$ and $\mathbf{p}$ denote the coordinates and momenta, respectively, and $\mathcal{H}(\mathbf{p}, \mathbf{q})$ is the Hamiltonian representing the total energy of the system. Suppose we are only given trajectories $\{(\mathbf{q}_t, \mathbf{p}_t)\}_{t=1}^N$ as data samples, how can we model the dynamics by a neural network model while preserving the conservation laws of the system? Moreover, is it possible to extend the modeling framework to incorporate energy dissipation and external forcing, which are important for applications such as robotics control?

One straightforward strategy for incorporating physical priors into neural networks is through the loss function, as exemplified by physics-informed neural networks (PINNs) [7]. In this framework, the network is trained to minimize the residuals of the governing PDEs, along with the initial and boundary conditions, thereby ensuring that the learned function approximately satisfies the physical constraints. For Hamiltonian systems, a specialized approach known as the Hamiltonian Neural Network (HNN) [4] has been proposed, which learns a parameterized Hamiltonian $\mathcal{H}_\theta$ to minimize the following objective:

$$\mathcal{L}_{\text{HNN}} = \left\| \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} - \dot{\mathbf{q}} \right\| + \left\| \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} + \dot{\mathbf{p}} \right\| \tag{1.2}$$

where the training targets $\dot{\mathbf{q}}$ and $\dot{\mathbf{p}}$ may be obtained from analytic time derivatives or finite-difference approximations of the observed trajectories. After training, the model predicts the dynamics by integrating equation (1.1) with a numerical integrator (e.g., a fourth-order Runge–Kutta method). A potential limitation of loss-based approaches is that they typically require training data that sufficiently covers the configuration space to achieve good generalization.

An alternative approach is to focus on inference rather than the loss function. By interpreting a deep residual network as an Euler discretization of an underlying ODE system [2], one can replace the standard Euler step with a geometric numerical integrator, thereby preserving desired invariants such as energy or momentum. This idea is exemplified by the Variational Integrator Network (VIN) [8]. In contrast to HNNs [4], VINs explicitly enforce conservation through their inference scheme, which accounts for their reported advantages over HNNs in low- and moderate-data regimes [8]. For instance, consider a parameterized Lagrangian

$$L_\theta(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\top \mathbf{M}_\theta \dot{\mathbf{q}} - U_\theta(\mathbf{q}) \tag{1.3}$$

and the Velocity-Verlet method

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{q}_k + h\dot{\mathbf{q}}_k - \frac{h^2}{2}\mathbf{M}_\theta^{-1}\nabla U_\theta(\mathbf{q}_k) \\ \dot{\mathbf{q}}_{k+1} &= \dot{\mathbf{q}}_k - h\mathbf{M}_\theta^{-1}\left(\frac{\nabla U_\theta(\mathbf{q}_k) + \nabla U_\theta(\mathbf{q}_{k+1})}{2}\right) \end{aligned} \tag{1.4}$$

can then serve as the feed-forward architecture of the VIN. The parameters θ can be trained by matching model roll-outs to observed trajectories, analogous to training a black-box dynamics model.

The VIN framework comes with the following advantages:

1. VINs inherit the desirable properties of variational integrators, including symplecticity, momentum conservation, and approximate energy conservation. Moreover, the order of accuracy of a variational integrator is determined by how well the discrete Lagrangian approximates the exact discrete Lagrangian, as established by variational error analysis [6].

2. VINs enjoy the flexibility of using a black-box network U_θ for potential energy modeling. Meantime, the notions of kinetic and potential energy increase interpretability.

In this project, our focus is on forced variational integrators, which enable data-efficient and physics-constrained dynamics modeling in control applications.

2 Discrete Variational Mechanics with Forces

2.1 Forced Discrete Systems

In this section, we overview the discrete variational principles underlying forced variational integrators following the presentation in [6]. We start with the continuous setting: consider a configuration manifold Q with a Lagrangian $L : TQ \to \mathbb{R}$ and a control manifold $\mathcal{U}$ with an external force $f_L : TQ \times \mathcal{U} \to T^*Q$, the Hamilton's principle is modified to the Lagrange-d'Alembert principle

$$\underbrace{\delta \int_0^T L(\mathbf{q}(t), \dot{\mathbf{q}}(t))\, dt}_{\text{action integral}} + \underbrace{\int_0^T f_L(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{u}(t)) \cdot \delta \mathbf{q}(t)\, dt}_{\text{virtual work}} = 0 \tag{2.1}$$

subject to $\delta\mathbf{q}(0) = \delta\mathbf{q}(T) = 0$, where $\mathbf{q} : [0, T] \to Q$ is the true path and $\mathbf{u} : [0, T] \to \mathcal{U}$ is the control.

In the discrete setting, we consider the discrete Lagrangian $L_d : Q \times Q \to \mathbb{R}$ and the discrete controlled forces $f_d^{\pm} : Q \times Q \times \mathcal{U} \to T^*Q$ which approximates the virtual work by

$$\int_{t_k}^{t_{k+1}} f_L(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{u}(t)) \cdot \delta\mathbf{q}(t) dt \approx f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_k + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_{k+1} \tag{2.2}$$

then the true discrete trajectory $\{\mathbf{q}_k\}_{k=0}^N$ w.r.t. the control sequence $\{\mathbf{u}_k\}_{k=0}^{N-1}$ satisfies the discrete Lagrange-d'Alembert principle

$$\delta \sum_{k=0}^{N-1} L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + \sum_{k=0}^{N-1} [f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_k + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_{k+1}] = 0 \tag{2.3}$$

subject to $\delta\mathbf{q}_0 = \delta\mathbf{q}_N = 0$. Using this constraint and discrete integration by parts, we have

$$\begin{aligned}
&\delta \sum_{k=0}^{N-1} L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + \sum_{k=0}^{N-1} [f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_k + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_{k+1}] \\
&\quad = \sum_{k=0}^{N-1} [D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) \cdot \delta\mathbf{q}_k + D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) \cdot \delta\mathbf{q}_{k+1}] \\
&\qquad + \sum_{k=0}^{N-1} [f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_k + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \cdot \delta\mathbf{q}_{k+1}] \\
&\quad = \sum_{k=0}^{N-1} [D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k) + f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) + f_d^+(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{u}_{k-1})] \cdot \delta\mathbf{q}_k
\end{aligned} \tag{2.4}$$

which implies the forced discrete Euler-Lagrange equations

$$D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k) + f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) + f_d^+(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{u}_{k-1}) = 0. \tag{2.5}$$

Fix the control sequence $\{\mathbf{u}_k\}_{k=0}^{N-1}$, we can define the forced discrete Legendre transform $\mathbb{F}^{f\pm}L_d : Q \times Q \to T^*Q$ to be

$$\begin{aligned} \mathbb{F}^{f+}L_d &: (\mathbf{q}_k, \mathbf{q}_{k+1}) \to (\mathbf{q}_{k+1}, \mathbf{p}_{k+1}) = (\mathbf{q}_{k+1}, D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k)) \\ \mathbb{F}^{f-}L_d &: (\mathbf{q}_k, \mathbf{q}_{k+1}) \to (\mathbf{q}_k, \mathbf{p}_k) = (\mathbf{q}_k, -D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) - f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k)) \end{aligned} \tag{2.6}$$

so that the momenta $\{\mathbf{p}_k\}_{k=0}^{N}$ are well-defined by the forced discrete Euler-Lagrange equations (2.5) and the connection between the discrete and continuous systems in (2.10) makes sense. Moreover, we can define the forced discrete Euler-Lagrange map and Hamiltonian map by

$$\begin{aligned} F_{L_d} &= (\mathbb{F}^{f-}L_d)^{-1} \circ (\mathbb{F}^{f+}L_d) : (\mathbf{q}_{k-1}, \mathbf{q}_k) \mapsto (\mathbf{q}_k, \mathbf{q}_{k+1}) \\ \tilde{F}_{L_d} &= \mathbb{F}^{f\pm}L_d \circ F_{L_d} \circ (\mathbb{F}^{f\pm}L_d)^{-1} : (\mathbf{q}_k, \mathbf{p}_k) \mapsto (\mathbf{q}_{k+1}, \mathbf{p}_{k+1}) \end{aligned} \tag{2.7}$$

respectively.

Finally, we may establish a correspondence between the continuous and discrete mechanics with forces, so that the variational error analysis for the unforced case may still apply. We define the exact discrete Lagrangian and exact discrete forces to be

$$\begin{aligned} L_d^E(\mathbf{q}_0, \mathbf{q}_1, h) &= \int_0^h L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t))dt \\ f_d^{E+}(\mathbf{q}_0, \mathbf{q}_1, h) &= \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_1} dt \\ f_d^{E-}(\mathbf{q}_0, \mathbf{q}_1, h) &= \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_0} dt \end{aligned} \tag{2.8}$$

where $\mathbf{q}_{0,1} : [0, h] \to Q$ is the solution of the Lagrange-d'Alembert principle (2.1) with the control input $\mathbf{u} : [0, h] \to \mathcal{U}$ and boundary conditions $\mathbf{q}_{0,1}(0) = \mathbf{q}_0$ and $\mathbf{q}_{0,1}(1) = \mathbf{q}_1$. Notice that with the exact discrete forces $f_d^{E\pm}$, the approximation in (2.2) becomes exact

$$\begin{aligned} \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \delta\mathbf{q}_{0,1}(t)dt &= \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \left(\frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_0} \delta\mathbf{q}_0 + \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_1} \delta\mathbf{q}_1 \right) dt \\ &= \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_0} dt \cdot \delta\mathbf{q}_0 \\ &\quad + \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_1} dt \cdot \mathbf{q}_1 \\ &= f_d^{E-}(\mathbf{q}_0, \mathbf{q}_1, h) \cdot \delta\mathbf{q}_0 + f_d^{E+}(\mathbf{q}_0, \mathbf{q}_1, h) \cdot \delta\mathbf{q}_1 \end{aligned} \tag{2.9}$$

We can compute the forced discrete Legendre transform using integration by parts

$$\begin{aligned} \mathbb{F}^{f-}L_d^E(\mathbf{q}_0, \mathbf{q}_1, h) &= -D_1 L_d^E(\mathbf{q}_0, \mathbf{q}_1, h) - f_d^{E-}(\mathbf{q}_0, \mathbf{q}_1, h) \\ &= -\int_0^h \left[\frac{\partial L}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}_{0,1}}{\partial \mathbf{q}_0} + \frac{\partial L}{\partial \dot{\mathbf{q}}} \cdot \frac{\partial \dot{\mathbf{q}}_{0,1}}{\partial \mathbf{q}_0} \right] dt - \int_0^h f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \cdot \frac{\partial \mathbf{q}_{0,1}(t)}{\partial \mathbf{q}_0} dt \\ &= -\int_0^h \underbrace{\left[\frac{\partial L}{\partial \mathbf{q}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} - f_L(\mathbf{q}_{0,1}(t), \dot{\mathbf{q}}_{0,1}(t), \mathbf{u}(t)) \right]}_{= 0 \text{ by the forced Euler-Lagrange equation}} \cdot \frac{\partial \mathbf{q}_{0,1}}{\partial \mathbf{q}_0} dt - \left[\frac{\partial L}{\partial \dot{\mathbf{q}}} \cdot \frac{\partial \mathbf{q}_{0,1}}{\partial \mathbf{q}_0} \right]_0^h \\ &= \frac{\partial L}{\partial \dot{\mathbf{q}}}(\mathbf{q}_{0,1}(0), \dot{\mathbf{q}}_{0,1}(0)) \\ &= \mathbb{F}L(\mathbf{q}_{0,1}(0), \dot{\mathbf{q}}_{0,1}(0)) \end{aligned} \tag{2.10}$$

where $\mathbb{F}L : TQ \to T^*Q$ is the standard Legendre transform, and $\mathbb{F}^{f+}L_d^E(\mathbf{q}_0, \mathbf{q}_1, h) = \mathbb{F}L(\mathbf{q}_{0,1}(1), \dot{\mathbf{q}}_{0,1}(1))$ following similar calculations. Notice that we have used the forced Euler-Lagrange equation, which is the continuous version of (2.5) and can be derived from the Lagrange-d'Alembert principle (2.1). Since the forced Hamiltonian and Lagrangian vector fields are also related by the standard Legendre transform, this shows that the forced exact discrete system is still equivalent to the forced continuous system.

2.2 Preservation of Symplecticity and Momentum

In the forced setting, using the forced discrete Legendre transform (2.6), we can write

$$\begin{aligned}
0 &= \mathbf{d}^2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) \\
&= \mathbf{d}\left(\frac{\partial L_d}{\partial \mathbf{q}_k^i}\mathbf{d}\mathbf{q}_k^i + \frac{\partial L_d}{\partial \mathbf{q}_{k+1}^j}\mathbf{d}\mathbf{q}_{k+1}^j\right) \\
&= \mathbf{d}\left((-\mathbf{p}_k^i - f_d^{-i}(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k))\mathbf{d}\mathbf{q}_k^i + (\mathbf{p}_{k+1}^j - f_d^{+j}(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k))\mathbf{d}\mathbf{q}_{k+1}^j\right) \\
&= \mathbf{d}\mathbf{q}_k^i \wedge \mathbf{d}\mathbf{p}_k^i - \mathbf{d}\mathbf{q}_{k+1}^j \wedge \mathbf{d}\mathbf{p}_{k+1}^j - \underbrace{[\mathbf{d}f_d^{-i}(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) \wedge \mathbf{d}\mathbf{q}_k^i + \mathbf{d}f_d^{+j}(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k)) \wedge \mathbf{d}\mathbf{q}_{k+1}^j]}_{\text{extra terms from forcing}}
\end{aligned} \tag{2.11}$$

which implies that $\mathbf{d}\mathbf{q}_k^i \wedge \mathbf{d}\mathbf{p}_k^i \neq \mathbf{d}\mathbf{q}_{k+1}^j \wedge \mathbf{d}\mathbf{p}_{k+1}^j$ in general due to the presence of forcing, i.e. the forced discrete Hamiltonian map $\tilde{F}_{L_d} : (\mathbf{q}_k, \mathbf{p}_k) \mapsto (\mathbf{q}_{k+1}, \mathbf{p}_{k+1})$ is not necessarily symplectic.

Furthermore, suppose we have a diagonal action $\Phi^{Q\times Q} : G \times Q \times Q \to Q \times Q$ of a group G on $Q \times Q$ and the discrete Lagrangian is G-invariant such that

$$L_d(g \cdot \mathbf{q}_k, g \cdot \mathbf{q}_{k+1}) = L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \qquad \forall g \in G \tag{2.12}$$

then the discrete momentum map $J_d : Q \times Q \to \mathfrak{g}^*$ defined by

$$\langle J_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi\rangle := \langle J(\mathbb{F}^{f\pm} L_d(\mathbf{q}_k, \mathbf{q}_{k+1})), \xi\rangle \tag{2.13}$$

can be preserved by the forced discrete Lagrangian map F_{L_d} under certain conditions.[1]

Specifically, let $\xi \in \mathfrak{g}^*$, we have $L_d(\exp(h\xi)\cdot\mathbf{q}_k, \exp(h\xi)\cdot\mathbf{q}_{k+1}) = L_d(\mathbf{q}_k, \mathbf{q}_{k+1})$. Taking the derivative over h and evaluate at $h = 0$, we have the following identity

$$\langle D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi_Q(\mathbf{q}_k)\rangle + \langle D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi_Q(\mathbf{q}_{k+1})\rangle = 0. \tag{2.14}$$

Pair the LHS of the forced discrete Euler–Lagrange equation (2.5) with $\xi_Q(\mathbf{q}_k)$ and use (2.14), we have

$$\begin{aligned}
0 &= \langle D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k) + f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k) + f_d^+(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{u}_{k-1}), \xi_Q(\mathbf{q}_k)\rangle \\
&= -\langle D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi_Q(\mathbf{q}_{k+1})\rangle + \langle D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k), \xi_Q(\mathbf{q}_k)\rangle \\
&\quad + \langle f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_k)\rangle + \langle f_d^+(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{u}_{k-1}), \xi_Q(\mathbf{q}_k)\rangle \\
&= -\langle D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_{k+1})\rangle \\
&\quad + \langle D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k) + f_d^+(\mathbf{q}_{k-1}, \mathbf{q}_k, \mathbf{u}_{k-1}), \xi_Q(\mathbf{q}_k)\rangle \\
&\quad + \langle f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_k)\rangle + \langle f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_{k+1})\rangle \\
&= -\langle \mathbb{F}^{f+} L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi_Q(\mathbf{q}_{k+1})\rangle + \langle \mathbb{F}^{f+} L_d(\mathbf{q}_{k-1}, \mathbf{q}_k), \xi_Q(\mathbf{q}_k)\rangle \\
&\quad + \underbrace{\langle f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_k)\rangle + \langle f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_{k+1})\rangle}_{\text{extra terms from forcing}}
\end{aligned} \tag{2.15}$$

That is to say, when the discrete forces are orthogonal to the group action in the sense that

$$\langle f_d^-(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_k)\rangle + \langle f_d^+(\mathbf{q}_k, \mathbf{q}_{k+1}, \mathbf{u}_k), \xi_Q(\mathbf{q}_{k+1})\rangle = 0, \tag{2.16}$$

then we have

$$\langle J_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \xi\rangle - \langle J_d(\mathbf{q}_{k-1}, \mathbf{q}_k), \xi\rangle = 0 \tag{2.17}$$

i.e., $J_d = (F_{L_d})^* J_d$, the discrete momentum map is preserved by the forced discrete Lagrangian map. This is known as the discrete forced Noether's theorem [6].

Although external forcing generally breaks symplecticity and momentum conservation, the resulting integrator can still be very effective when the system is predominantly governed by its underlying physical laws. This is illustrated, for instance, by a recent work on modeling aerial maneuvers with forced variational integrators [1], which achieves accurate and stable long-term predictions despite the absence of exact conservation properties.

[1] $J : T^*Q \to \mathfrak{g}^*$ is the continuous momentum map given by $\langle J(\alpha_{\mathbf{q}}), \xi\rangle = \langle \alpha_{\mathbf{q}}, \xi_Q(\mathbf{q})\rangle$ where $\xi_Q(\mathbf{q}) := \frac{d}{dh}\exp(h\xi)\cdot\mathbf{q}|_{h=0}$ is the infinitesimal generator.

3 Lie Group Forced Variational Integrator Network

Having established the forced discrete Euler-Lagrange equations (2.5), we could derive forced variational integrators, either from the forced discrete Lagrangian map F_{L_d} or the forced discrete Hamiltonian map $\tilde{F}_{L_d} = \mathbb{F}^{f\pm} L_d \circ F_{L_d} \circ (\mathbb{F}^{f\pm} L_d)^{-1}$, given an appropriate choice of the discrete Lagrangian L_d and discrete controlled forces $f_d^{\pm}$. Applied to the parameterized Lagrangian L_θ and with a neural network used to approximate the discrete forces, this allows us to construct a feed-forward architecture similar to VINs and is referred to as forced variational integrator networks (FVINs) [5].

In this project, we focus on deriving a forced variational integrator network for modeling rigid-body dynamics in control applications, where the underlying configuration space is a Lie group. Since velocity measurements are often available, it is more practical to consider a forced Velocity-Verlet integrator derived from the discrete Hamiltonian map $\tilde{F}_{L_d}$. While a Störmer–Verlet formulation could also be derived from the discrete Lagrangian map F_{L_d} (which is advantageous when velocities are unobserved, such as in settings with raw-image observations), this lies outside the scope of our project.

For simplicity, we start by treating translational and rotational updates separately and assume that the system evolves with a uniform time step $h > 0$.

3.1 Translational Updates

To model the translation in $\mathbb{R}^3$, we consider the separable Newtonian system.

$$L_\theta(\mathbf{x}, \dot{\mathbf{x}}) = \frac{1}{2}\dot{\mathbf{x}}^\top \mathbf{M}_\theta \dot{\mathbf{x}} - U_\theta(\mathbf{x}) \tag{3.1}$$

We choose the following symmetric quadrature approximation

$$L_d(\mathbf{x}_k, \mathbf{x}_{k+1}) = \frac{h}{2}\left(L_\theta\left(\mathbf{x}_k, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) + L_\theta\left(\mathbf{x}_{k+1}, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right)\right) \tag{3.2}$$

By direct computation, we have

$$\begin{aligned}
D_1 L_d(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \frac{h}{2}\frac{\partial L_\theta}{\partial \mathbf{x}}\left(\mathbf{x}_k, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) - \frac{1}{2}\frac{\partial L_\theta}{\partial \dot{\mathbf{x}}}\left(\mathbf{x}_k, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) - \frac{1}{2}\frac{\partial L_\theta}{\partial \dot{\mathbf{x}}}\left(\mathbf{x}_{k+1}, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) \\
&= -\frac{h}{2}\nabla U_\theta(\mathbf{x}_k) - \mathbf{M}_\theta\left(\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) \\
D_2 L_d(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \frac{1}{2}\frac{\partial L_\theta}{\partial \dot{\mathbf{x}}}\left(\mathbf{x}_k, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) + \frac{h}{2}\frac{\partial L_\theta}{\partial \mathbf{x}}\left(\mathbf{x}_{k+1}, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) + \frac{1}{2}\frac{\partial L_\theta}{\partial \dot{\mathbf{x}}}\left(\mathbf{x}_{k+1}, \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) \\
&= \mathbf{M}_\theta\left(\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) - \frac{h}{2}\nabla U_\theta(\mathbf{x}_{k+1})
\end{aligned} \tag{3.3}$$

Using the calculations in (3.3), we can compute the momenta

$$\begin{aligned}
\mathbf{p}_k &= -D_1 L_d(\mathbf{x}_k, \mathbf{x}_{k+1}) - f_d^-(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{u}_k) = \frac{h}{2}\nabla U_\theta(\mathbf{x}_k) + \mathbf{M}_\theta\left(\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) - f_d^- \\
\mathbf{p}_{k+1} &= D_2 L_d(\mathbf{x}_k, \mathbf{x}_{k+1}) + f_d^+(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{u}_k) = \mathbf{M}_\theta\left(\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}\right) - \frac{h}{2}\nabla U_\theta(\mathbf{x}_{k+1}) + f_d^+
\end{aligned} \tag{3.4}$$

based on the forced discrete Legendre transform (2.6), which implies the following update rule

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k + h\mathbf{M}_\theta^{-1}\mathbf{p}_k - \frac{h^2}{2}\mathbf{M}_\theta^{-1}\nabla U_\theta(\mathbf{x}_k) + h\mathbf{M}_\theta^{-1} f_d^- \\
\mathbf{p}_{k+1} &= \mathbf{p}_k - h\frac{\nabla U_\theta(\mathbf{x}_k) + \nabla U_\theta(\mathbf{x}_{k+1})}{2} + f_d^+ + f_d^-
\end{aligned} \tag{3.5}$$

Recall that $\mathbf{p} = \mathbf{M}_\theta \dot{\mathbf{x}}$ for (3.1), if we choose a symmetric representation of the discrete forces

$$f_d^+ = f_d^- = \frac{h}{2} F_\theta(\mathbf{x}_k, \mathbf{u}_k) \tag{3.6}$$

then we will obtain the forced Velocity-Verlet integrator

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + h\dot{\mathbf{x}}_k + \frac{h^2}{2}\mathbf{M}_\theta^{-1}(F_\theta(\mathbf{x}_k, \mathbf{u}_k) - \nabla U_\theta(\mathbf{x}_k)) \\ \dot{\mathbf{x}}_{k+1} &= \dot{\mathbf{x}}_k + h\mathbf{M}_\theta^{-1}\left(F_\theta(\mathbf{x}_k, \mathbf{u}_k) - \frac{\nabla U_\theta(\mathbf{x}_k) + \nabla U_\theta(\mathbf{x}_{k+1})}{2}\right).\end{aligned} \tag{3.7}$$

similar to [5].

3.2 Rotational Updates

For the rotational update, we consider the parameterized Lagrangian $L_\theta : T(\mathbf{SO}(3)) \to \mathbb{R}$ on the rotation group $\mathbf{SO}(3)$ similar to [3]

$$L_\theta(\mathbf{R}, \dot{\mathbf{R}}) = \frac{1}{2}\omega^\top \mathbf{J}_\theta \omega - U_\theta(\mathbf{R}) \tag{3.8}$$

where $\omega \in \mathbb{R}^3 \cong \mathfrak{so}(3)$ is the angular velocity such that $\dot{\mathbf{R}} = \mathbf{R}[\omega]_\times \in T_{\mathbf{R}}(\mathbf{SO}(3))$. To obtain a discretization similar to (3.2), we need to approximate ω from $(\mathbf{R}_k, \mathbf{R}_{k+1})$. We have $\mathbf{R}_{k+1} = \mathbf{R}_k \exp(h[\omega_k]_\times)$ and notice that $\exp(h[\omega_k]_\times) \approx \mathbb{I} + h[\omega_k]_\times$, hence $[\omega_k]_\times \approx \frac{1}{h}(\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I})$. However, $\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}$ is not skew-symmetric, and therefore direct use of $\omega_k \approx \frac{1}{h}(\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I})^\vee$ does not make sense. Alternatively, we can use the identity $2\mathbf{w}^\top \mathbf{J}\mathbf{w} = -\mathrm{tr}([\mathbf{w}]_\times(2\mathbf{J} - \mathrm{tr}(\mathbf{J})\mathbb{I})[\mathbf{w}]_\times^\top)$, which gives the following discrete Lagrangian

$$\begin{aligned}L_d(\mathbf{R}_k, \mathbf{R}_{k+1}) &= -\frac{1}{2h}\mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]\tilde{\mathbf{J}}_\theta[\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]^\top) - \frac{h}{2}(U_\theta(\mathbf{R}_k) + U_\theta(\mathbf{R}_{k+1})) \\ &= \frac{1}{h}\mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]\tilde{\mathbf{J}}_\theta) - \frac{h}{2}(U_\theta(\mathbf{R}_k) + U_\theta(\mathbf{R}_{k+1}))\end{aligned} \tag{3.9}$$

used by Duruisseaux et al. [3], where $\tilde{\mathbf{J}}_\theta = \mathbf{J}_\theta - \frac{1}{2}\mathrm{tr}(\mathbf{J}_\theta)\mathbb{I}$. For more details, see Appendix A.

Motivated by [9], we can lift the Lagrangian (3.8) on $T\mathbf{SO}(3)$ to TS^3 by the surjective and locally diffeomorphic Lie group homomorphism $\Phi : S^3 \to \mathbf{SO}(3)$

$$\Phi(\mathbf{q}) = (2q_s^2 - 1)\mathbb{I} + 2\mathbf{q}_v\mathbf{q}_v^\top + 2q_s[\mathbf{q}_v]_\times \tag{3.10}$$

where $\mathbf{q} = (q_s, \mathbf{q}_v) \in S^3$ and $S^3 = \{\mathbf{q} \in \mathbb{H} \,|\, \|\mathbf{q}\| = 1\}$ is the set of unit quaternions.[2] Taking the time derivative of the unit norm condition $\|\mathbf{q}\| = \mathbf{q}^*\mathbf{q} = 1$ yields $\mathbf{q}^*\dot{\mathbf{q}} + (\mathbf{q}^*\dot{\mathbf{q}})^* = 0$, which implies that $\mathbf{q}^*\dot{\mathbf{q}}$ is a pure quaternion (i.e., $\mathbf{q}^*\dot{\mathbf{q}} \in \mathbb{H}_p = \{\mathbf{q} \in \mathbb{H} \,|\, q_s = 0\}$). Therefore, $\dot{\mathbf{q}} = (\mathbf{q}^*)^{-1}\xi = \mathbf{q}\xi$ for some $\xi \in \mathbb{H}_p$.

In Appendix B, we will verify that the tangent lift of the map $\Phi : S^3 \to \mathbf{SO}(3)$ acts as

$$(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{q}, \mathbf{q}\xi) \xrightarrow{T_{\mathbf{q}}\Phi} (\Phi(\mathbf{q}), 2\Phi(\mathbf{q})[\xi]_\times) \tag{3.11}$$

and hence we may consider the lifted Lagrangian $\hat{L}_\theta = \Phi^* L_\theta : T(S^3) \to \mathbb{R}$

$$\hat{L}_\theta(\mathbf{q}, \dot{\mathbf{q}}) = 2\xi^\top \mathbf{J}_\theta \xi - U_\theta(\xi). \tag{3.12}$$

To approximate $\xi = \mathbf{q}^*\dot{\mathbf{q}}$ from $(\mathbf{q}_k, \mathbf{q}_{k+1})$, we can use the mid-point rule

$$\mathbf{q}^*\dot{\mathbf{q}} \approx \left(\frac{\mathbf{q}_k + \mathbf{q}_{k+1}}{2}\right)^* \left(\frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{h}\right) = \frac{1}{2h}(\mathbf{q}_k^*\mathbf{q}_{k+1} - \mathbf{q}_{k+1}^*\mathbf{q}_k) = \frac{1}{h}\mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1}) \tag{3.13}$$

similar to [9], which results in the following discrete Lagrangian

$$\begin{aligned}L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) &= \frac{h}{2}\left(L_\theta\left(\mathbf{q}_k, \frac{1}{h}\mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1})\right) + L_\theta\left(\mathbf{q}_{k+1}, \frac{1}{h}\mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1})\right)\right) \\ &= \frac{2}{h}\mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1})^\top \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1}) - \frac{h}{2}(U_\theta(\mathbf{q}_k) + U_\theta(\mathbf{q}_{k+1})).\end{aligned} \tag{3.14}$$

[2]The computations that follow are adapted primarily from [9], with minor modifications to fit our notation and problem setting. For more details, see [9] and [10].

Consider the variation $\mathbf{q}_k(\epsilon) = \mathbf{q}_k \exp(\epsilon(0, \eta_k))$ for $(0, \eta_k) \in \mathbb{H}_p$ such that $\delta \mathbf{q}_k = \frac{d}{d\epsilon}\big|_{\epsilon=0} \mathbf{q}_k(\epsilon) = \mathbf{q}_k(0, \eta_k)$. We can write

$$
\begin{aligned}
D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) \cdot \delta \mathbf{q}_k &= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} L_d(\mathbf{q}_k(\epsilon), \mathbf{q}_{k+1}) \\
&= \frac{2}{h} \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathrm{Im}(\mathbf{q}_k(\epsilon)^* \mathbf{q}_{k+1})^\top \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_k(\epsilon)^* \mathbf{q}_{k+1}) - \frac{h}{2}[U_\theta(\mathbf{q}_k)]^\top H(\mathbf{q}_k)^\top \eta_k \\
&= \frac{4}{h} \mathrm{Im}(\mathbf{q}_k^* \mathbf{q}_{k+1})^\top \mathbf{J}_\theta \left(\frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathrm{Im}(\mathbf{q}_k(\epsilon)^* \mathbf{q}_{k+1}) \right) - \frac{h}{2}[U_\theta(\mathbf{q}_k)]^\top H(\mathbf{q}_k)^\top \eta_k \\
&= \frac{4}{h} \mathrm{Im}(\mathbf{q}_k^* \mathbf{q}_{k+1})^\top \mathbf{J}_\theta \mathrm{Im}\left((\mathbf{q}_k(0, \eta_k))^* \mathbf{q}_{k+1}\right) - \frac{h}{2}[U_\theta(\mathbf{q}_k)]^\top H(\mathbf{q}_k)^\top \eta_k \\
&= \frac{4}{h} \mathrm{Im}(\mathbf{q}_{k+1}^* \mathbf{q}_k)^\top \mathbf{J}_\theta \mathrm{Im}\left(\mathbf{q}_{k+1}^* \mathbf{q}_k(0, \eta_k)\right) - \frac{h}{2}[U_\theta(\mathbf{q}_k)]^\top H(\mathbf{q}_k)^\top \eta_k \\
&= \frac{4}{h} \mathrm{Im}(\mathbf{q}_{k+1}^* \mathbf{q}_k)^\top \mathbf{J}_\theta G(\mathbf{q}_{k+1}^* \mathbf{q}_k)^\top \eta_k - \frac{h}{2}[\nabla U_\theta(\mathbf{q}_k)]^\top H(\mathbf{q}_k)^\top \eta_k
\end{aligned} \tag{3.15}
$$

where $G(\mathbf{q}), H(\mathbf{q})$ are linear operators such that $G(\mathbf{q})^\top \mathbf{x} = \mathrm{Im}(\mathbf{q}(0, \mathbf{x}))$ and $H(\mathbf{q})^\top \mathbf{x} = \mathbf{q}(0, \mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^3$. It can be verified that

$$
\begin{aligned}
G(\mathbf{q}) &= \mathbf{q}_s \mathbb{I} - [\mathbf{q}_v]_\times \\
H(\mathbf{q}) &= (-\mathbf{q}_v, \mathbf{q}_s \mathbb{I} - [\mathbf{q}_v]_\times)
\end{aligned} \tag{3.16}
$$

Notice that $\langle D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), \delta \mathbf{q}_k \rangle = \langle D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), T_\mathbf{e} L_{\mathbf{q}_k}(0, \eta_k) \rangle = \langle T_\mathbf{e}^* L_{\mathbf{q}_k} D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}), (0, \eta_k) \rangle$ by the definition of cotangent lifts. Therefore, we can write

$$
T_\mathbf{e}^* L_{\mathbf{q}_k} D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) = \frac{4}{h} G(\mathbf{q}_{k+1}^* \mathbf{q}_k) \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_{k+1}^* \mathbf{q}_k) - \frac{h}{2} H(\mathbf{q}_k) \nabla U_\theta(\mathbf{q}_k) \tag{3.17}
$$

in the left-trivialized coordinate system. Similarly, we could obtain

$$
T_\mathbf{e}^* L_{\mathbf{q}_{k+1}} D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) = \frac{4}{h} G(\mathbf{q}_k^* \mathbf{q}_{k+1}) \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_k^* \mathbf{q}_{k+1}) - \frac{h}{2} H(\mathbf{q}_{k+1}) \nabla U_\theta(\mathbf{q}_{k+1}). \tag{3.18}
$$

Given left-trivialized discrete forces $f_d^\pm : TS^3 \times \mathcal{U} \to (\mathfrak{s}^3)^*$, we can compute the left-trivialized momenta $\boldsymbol{\pi}_k, \boldsymbol{\pi}_{k+1} \in (\mathfrak{s}^3)^*$ by the forced discrete Legendre transform (2.6) from (3.17) and (3.18)

$$
\begin{aligned}
\boldsymbol{\pi}_k &= -T_\mathbf{e}^* L_{\mathbf{q}_k} D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) - f_d^- \\
&= -\frac{4}{h} G(\mathbf{q}_{k+1}^* \mathbf{q}_k) \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_{k+1}^* \mathbf{q}_k) + \frac{h}{2} H(\mathbf{q}_k) \nabla U_\theta(\mathbf{q}_k) - f_d^- \\
\boldsymbol{\pi}_{k+1} &= T_\mathbf{e}^* L_{\mathbf{q}_{k+1}} D_2 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + f_d^+ \\
&= \frac{4}{h} G(\mathbf{q}_k^* \mathbf{q}_{k+1}) \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}_k^* \mathbf{q}_{k+1}) - \frac{h}{2} H(\mathbf{q}_{k+1}) \nabla U_\theta(\mathbf{q}_{k+1}) + f_d^+
\end{aligned} \tag{3.19}
$$

which implicitly defines the update rule $(\mathbf{q}_k, \boldsymbol{\pi}_k) \mapsto (\mathbf{q}_{k+1}, \boldsymbol{\pi}_{k+1})$.

Furthermore, it is also necessary to establish the relationship between the left-trivialized momentum $\boldsymbol{\pi} \in (\mathfrak{s}^3)^*$ and the trivialized velocity $\xi \in \mathbb{R}^3 \cong \mathfrak{s}^3$. In Appendix C, we will verify that $\boldsymbol{\pi} = 4\mathbf{J}_\theta \xi$.

3.3 Full Algorithm

To avoid ambiguity, we use $\mathbf{x} \in \mathbb{R}^3$ to denote positions and $\mathbf{q} \in S^3$ to denote the unit quaternion representation of rotations, resulting in the following combined Lagrangian $L_\theta : T(\mathbb{R}^3 \times S^3) \to \mathbb{R}$

$$
L_\theta(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{x}}^\top \mathbf{M}_\theta \dot{\mathbf{x}} + 2\xi^\top \mathbf{J}_\theta \xi - U_\theta(\mathbf{x}, \mathbf{q}) \tag{3.20}
$$

where $\xi = \mathrm{Im}(\mathbf{q}^* \dot{\mathbf{q}})$, and choose the discrete Lagrangian L_d by combining the symmetric quadrature approximations in (3.2) and (3.14).

Using $f_d^{\mathbf{x}\pm} \in T^*\mathbb{R}^3$ and $f_d^{\mathbf{q}\pm} \in (\mathfrak{s}^3)^*$ to denote the (left-trivialized) discrete forces, we have the following combined update rules

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{M}_\theta^{-1}\mathbf{p}_k - \frac{h^2}{2}\mathbf{M}_\theta^{-1}\nabla_{\mathbf{x}}U_\theta(\mathbf{x}_k, \mathbf{q}_k) + h\mathbf{M}_\theta^{-1}f_d^{\mathbf{x}-} \tag{3.21}$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k - h\frac{\nabla_{\mathbf{x}}U_\theta(\mathbf{x}_k, \mathbf{q}_k) + \nabla_{\mathbf{x}}U_\theta(\mathbf{x}_{k+1}, \mathbf{q}_{k+1})}{2} + f_d^{\mathbf{x}+} + f_d^{\mathbf{x}-} \tag{3.22}$$

$$\boldsymbol{\pi}_k = -\frac{4}{h}G(\mathbf{q}_{k+1}^*\mathbf{q}_k)\mathbf{J}_\theta\mathrm{Im}(\mathbf{q}_{k+1}^*\mathbf{q}_k) + \frac{h}{2}H(\mathbf{q}_k)\nabla_{\mathbf{q}}U_\theta(\mathbf{x}_k, \mathbf{q}_k) - f_d^{\mathbf{q}-} \tag{3.23}$$

$$\boldsymbol{\pi}_{k+1} = \frac{4}{h}G(\mathbf{q}_k^*\mathbf{q}_{k+1})\mathbf{J}_\theta\mathrm{Im}(\mathbf{q}_k^*\mathbf{q}_{k+1}) - \frac{h}{2}H(\mathbf{q}_{k+1})\nabla_{\mathbf{q}}U_\theta(\mathbf{x}_{k+1}, \mathbf{q}_{k+1}) + f_d^{\mathbf{q}+} \tag{3.24}$$

from (3.5) and (3.19), where $\mathbf{p}_k \in \mathbb{R}^3$ and $\boldsymbol{\pi}_k \in (\mathfrak{s}^3)^* \cong \mathbb{R}^3$ are the corresponding momenta. Notice that when $f_d^{\mathbf{x}\pm} = f_d^{\mathbf{q}\pm} = 0$, this is equivalent to VINs on the unit quaternion group without forces (see [8, 9]).

Let $\xi_k \in \mathfrak{s}^3 \cong \mathbb{R}^3$ such that $\mathbf{q}_{k+1} = \mathbf{q}_k \exp(\xi_k)$. We could apply a root-finding algorithm (e.g., Newton's method) to solve ξ_k for (3.23) given $(\mathbf{q}_k, \boldsymbol{\pi}_k)$ and obtain $(\mathbf{q}_{k+1}, \boldsymbol{\pi}_{k+1})$ from the definition of ξ_k and (3.24), similar to the approach in [3]. More specifically, we are solving equations of the form

$$G(\exp(-\xi_k))\mathbf{J}_\theta\mathrm{Im}(\exp(-\xi_k)) = C \tag{3.25}$$

where the constant C depends on the input $(\mathbf{x}_k, \mathbf{p}_k, \mathbf{q}_k, \boldsymbol{\pi}_k, \mathbf{u}_k)$ and the model parameters θ. For simplicity, we compute the Jacobian for the LHS of (3.25) using automatic differentiation, although it could potentially be further optimized by deriving the corresponding analytical expressions or applying the implicit function theorem (i.e., implicit differentiation).

Because the covering map $\Phi : S^3 \to \mathbf{SO}(3)$ is two-to-one ($\Phi(\mathbf{q}) = \Phi(-\mathbf{q})$), the physical rotation must be independent of the quaternion sign. To prevent the learned LieFVIN model from producing inconsistent predictions for $\mathbf{q}$ and $-\mathbf{q}$, we enforce sign invariance on all black-box components as follows:

$$\begin{aligned}\hat{\mathbf{J}}_\theta(\mathbf{q}) &= \tfrac{1}{2}\big(\mathbf{J}_\theta(\mathbf{q}) + \mathbf{J}_\theta(-\mathbf{q})\big),\\ \hat{U}_\theta(\mathbf{x}, \mathbf{q}) &= \tfrac{1}{2}\big(U_\theta(\mathbf{x}, \mathbf{q}) + U_\theta(\mathbf{x}, -\mathbf{q})\big),\\ \hat{F}_\theta^\pm(\mathbf{x}, \mathbf{q}, \mathbf{u}) &= \tfrac{1}{2}\big(F_\theta^\pm(\mathbf{x}, \mathbf{q}, \mathbf{u}) + F_\theta^\pm(\mathbf{x}, -\mathbf{q}, \mathbf{u})\big).\end{aligned}$$

Here, $\mathbf{J}_\theta(\mathbf{q})$ is the learned (potentially configuration-dependent) inertia matrix and $F_\theta^\pm$ are the neural networks approximating the discrete forces $f_d^{\mathbf{q}\pm}$.

Lastly, given input data $(\mathbf{x}, \mathbf{R}, \dot{\mathbf{x}}, \dot{\mathbf{R}}) \in T\mathbf{SE}(3)$, we can convert to

$$(\mathbf{x}, \mathbf{q}, \mathbf{p}, \boldsymbol{\pi}) = (\mathbf{x}, \mathbf{q}, \mathbf{M}_\theta\dot{\mathbf{x}}, 2\mathbf{J}_\theta(\mathbf{R}^\top\dot{\mathbf{R}})^\vee) \in T^*(\mathbb{R}^3 \times S^3) \tag{3.26}$$

where $\mathbf{q} = \Phi^{-1}(\mathbf{R}) \in S^3$, since $\mathbf{p} = \mathbf{M}_\theta\dot{\mathbf{x}}$, $\boldsymbol{\pi} = 4\mathbf{J}_\theta\xi$ and $\xi = \frac{1}{2}(\mathbf{R}^\top\dot{\mathbf{R}})^\vee$. Furthermore, the discrete forces are related by $f_d^{\mathbf{R}\pm} = \frac{1}{2}f_d^{\mathbf{q}\pm}$.[3] This conversion is invertible, hence we can always replace the original LieFVIN on $T\mathbf{SE}(3)$ [3] with the algorithm on the quaternion representation $T^*(\mathbb{R}^3 \times S^3)$.

4 Experiments

We evaluate the algorithm described in Section 3.3 (which we refer to as **S3FVIN**) on a planar pendulum, where we only need to model the rotation. Our implementation and experiment settings follow the original LieFVIN code [3], but the LieFVIN on $\mathbf{SO}(3)$ is replaced with our S^3-based approach.[4]

We have implemented the following three variants of **S3FVIN**:

(1) Enforcing sign-invariance $\mathbf{J}_\theta(\mathbf{q}) = \mathbf{J}_\theta(-\mathbf{q})$; $U_\theta(\mathbf{q}) = U_\theta(-\mathbf{q})$, and $F_\theta^\pm(\mathbf{q}, \mathbf{u}) = F_\theta^\pm(-\mathbf{q}, \mathbf{u})$;

(2) Using a fixed inertia matrix $\mathbf{J}_\theta$ (instead of a configuration-dependent one in the original setting);

(3) Plain (imposing no restrictions).

[3] The factor $\frac{1}{2}$ accounts for the double effect of the quaternion in the rotation action [10], see Appendix B.
[4] The PyTorch implementation is available at `https://github.com/hanyang-hu/S3FVIN`.

We observe that both (1) and (2) successfully recover the ground-truth inertia matrix $\mathbf{J}_\theta$ and the control gain $g(\mathbf{q})$ (s.t. $f_d^{\mathbf{q}-} = g(\mathbf{q})\mathbf{u}$ and $f_d^{\mathbf{q}-} = 0$). As expected from the cotangent lift of $\Phi : S^3 \to \mathbf{SO}(3)$, the learned $g(\mathbf{q})$ is twice the value reported in [3]. In contrast, (3) fails to learn these quantities accurately, which leads to noticeable energy drift.

Upon closer inspection, (2) shows superior energy conservation, as expected from its use of a fixed inertia matrix $\mathbf{J}_\theta$. On the other hand, allowing a configuration-dependent $\mathbf{J}_\theta(\mathbf{q})$ increases model expressiveness but may slightly compromise long-term energy stability. Therefore, the choice should be made based on the specific problem requirements.

We also observe in Fig. 2 that the sign-invariant **S3FVIN** converges markedly faster and more stably than other variants of **S3FVIN** and the original **SO3FVIN**, which operates on 3×3 rotation matrices. This improvement stems from the fact that unit quaternions provide a compact 4-dimensional representation of 3D rotations, avoiding the redundancies in the 9-dimensional matrix form. To verify that the advantage is purely representational, we implemented an **SO3FVIN** variant in which all black-box components ($\mathbf{J}_\theta$, U_θ, and $F_\theta^\pm$) operate on a unit quaternion $\mathbf{q} \in \Phi^{-1}(\mathbf{R})$ chosen as a preimage of $\mathbf{R}$ under the Lie group homomorphism $\Phi : S^3 \to \mathbf{SO}(3)$.[5] This single modification removes representational redundancy and achieves convergence nearly identical to **S3FVIN**, while retaining the faster per-iteration speed of **SO3FVIN** (due to the analytical Jacobian used in Newton's iteration). Therefore, for $\mathbf{SO}(3)$-based implementations seeking optimal training behavior, internally using the conversion $\mathbf{R} \mapsto \mathbf{q}$ is a recommended practice.

5 Conclusion

In this project, we explored **S3FVIN**, a Lie group forced variational integrator network based on unit quaternions for learning controlled rigid-body dynamics. Our experiments suggest two potentially useful takeaways: (1) explicitly enforcing sign invariance is a simple yet essential inductive bias for obtaining physically plausible results for **S3FVIN**; and (2) working directly with unit quaternions generally leads to faster and more stable training than using 3×3 rotation matrices, largely due to the more compact representation. In practice, however, we still recommend using the original LieFVIN on $\mathbf{SE}(3)$, coupled with an internal conversion from rotation matrices to quaternions.

[5] We backpropagate through the differentiable rotation-to-quaternion transformation provided by the *kornia* package.

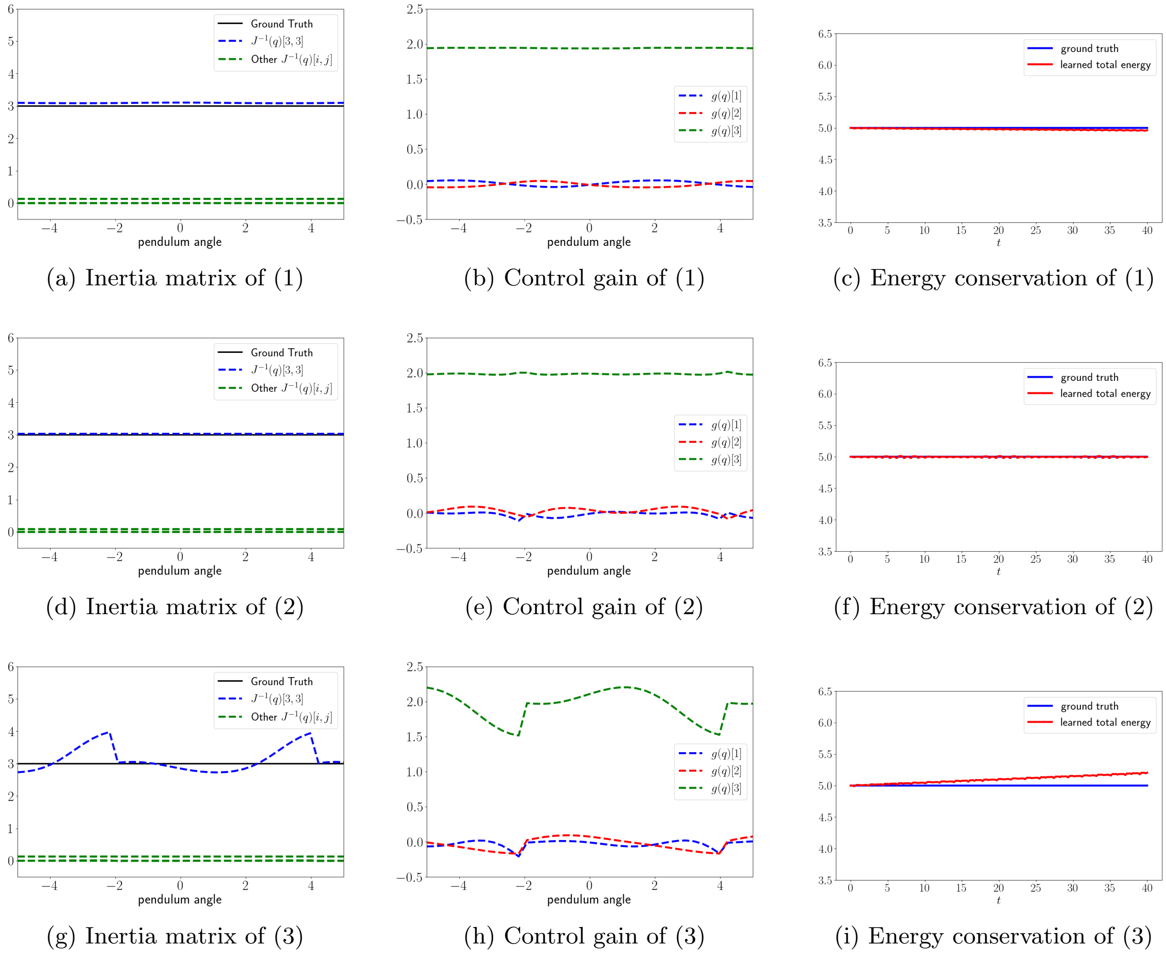

Figure 1: Visualization of the learned inertia matrix $\mathbf{J}_\theta$, control gain $g(\mathbf{q})$, and long-term energy behavior without control input. Enforcing either sign invariance or a configuration-independent inertia matrix is essential for obtaining a well-behaved model of the planar pendulum dynamics.

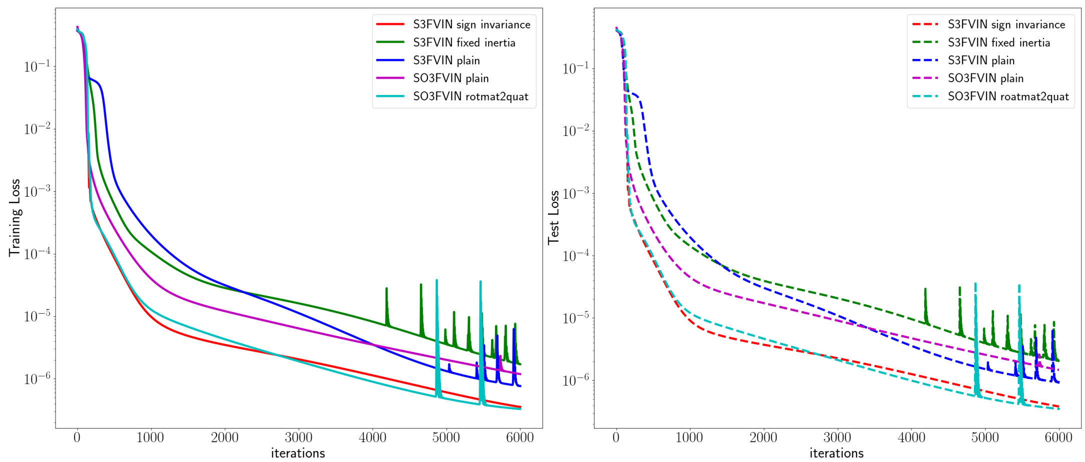

Figure 2: Loss curves for different variants of **S3FVIN** and **SO3FVIN**. The sign-invariant **S3FVIN** converges fastest and most stably. **SO3FVIN** can achieve comparable performance by parameterizing each black-box component using the transformation $\mathbf{R} \mapsto \mathbf{q}$, but exhibits less stable training. The latter benefits from less wall-clock time due to the analytical Jacobian used in Newton's method.

References

[1] Scott Beck, Chuong Nguyen, Thai Duong, Nikolay Atanasov, and Quan Nguyen. High accuracy aerial maneuvers on legged robots using variational integrator discretized trajectory optimization. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10253–10260, 2025.

[2] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[3] Valentin Duruisseaux, Thai Duong, Melvin Leok, and Nikolay Atanasov. Lie group forced variational integrator networks for learning and control of robot systems, 2023.

[4] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks, 2019.

[5] Aaron Havens and Girish Chowdhary. Forced variational integrator networks for prediction and control of mechanical systems, 2021.

[6] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.

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

[8] Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Peter Deisenroth. Variational integrator networks for physically structured embeddings, 2020.

[9] Xuefeng Shen and Melvin Leok. Lie group variational integrators for rigid body problems using quaternions, 2017.

[10] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics, 2021.

[11] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Comput.*, 8(7):1341–1390, October 1996.

A Discrete Lagrangian on SO(3)

We state the well-known Lagrange's formula and the fact that 3×3 skew-symmetric matrices can be used to represent cross products as matrix multiplications. They can be verified by direct computations.

Lemma A.1. (Lagrange's formula) Let $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^3$, we have $(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} = (\mathbf{u} \cdot \mathbf{w})\mathbf{v} - (\mathbf{v} \cdot \mathbf{w})\mathbf{u}$.

Lemma A.2. Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$, we have $[\mathbf{v}]_\times \mathbf{w} = \mathbf{v} \times \mathbf{w}$.

Corollary A.3. Let $\mathbf{x} \in \mathbb{R}^3$, we have $[\mathbf{x}]_\times^2 = \mathbf{x}\mathbf{x}^\top - \|\mathbf{x}\|^2 \mathbb{I}$.

Proof. For any $\mathbf{v} \in \mathbb{R}^3$, we have

$$\begin{aligned} [\mathbf{x}]_\times^2 \mathbf{v} &= \mathbf{x} \times (\mathbf{x} \times \mathbf{v}) \\ &= -(\mathbf{x} \times \mathbf{v}) \times \mathbf{x} \\ &= (\mathbf{v} \cdot \mathbf{x})\mathbf{x} - (\mathbf{x} \cdot \mathbf{x})\mathbf{v} \\ &= (\mathbf{x}\mathbf{x}^\top - \|\mathbf{x}\|^2 \mathbb{I})\mathbf{v} \end{aligned} \tag{A.1}$$

which concludes the proof. □

Claim. Let $\mathbf{w} \in \mathbb{R}^3$ and $\mathbf{J}$ be a 3×3 symmetric matrix, we have

$$2\mathbf{w}^\top \mathbf{J} \mathbf{w} = -\mathrm{tr}([\mathbf{w}]_\times (2\mathbf{J} - \mathrm{tr}(\mathbf{J})\mathbb{I})[\mathbf{w}]_\times^\top). \tag{A.2}$$

Proof. We have

$$\begin{aligned} \mathbf{w}^\top \mathbf{J} \mathbf{w} &= \mathrm{tr}(\mathbf{J}\mathbf{w}\mathbf{w}^\top) \\ &= \mathrm{tr}(\mathbf{J}([\mathbf{w}]_\times^2 + \|\mathbf{w}\|^2 \mathbb{I})) \\ &= \mathrm{tr}([\mathbf{w}]_\times \mathbf{J} [\mathbf{w}]_\times) + \|\mathbf{w}\|^2 \mathrm{tr}(\mathbf{J}) \end{aligned} \tag{A.3}$$

and

$$\begin{aligned} \mathrm{tr}([\mathbf{w}]_\times (2\mathbf{J} - \mathrm{tr}(\mathbf{J})\mathbb{I})[\mathbf{w}]_\times) &= 2\mathrm{tr}([\mathbf{w}]_\times \mathbf{J} [\mathbf{w}]_\times) - \mathrm{tr}([\mathbf{w}]_\times^2)\mathrm{tr}(\mathbf{J}) \\ &= 2\mathrm{tr}([\mathbf{w}]_\times \mathbf{J} [\mathbf{w}]_\times) - \mathrm{tr}(\mathbf{w}\mathbf{w}^\top - \|\mathbf{w}\|^2 \mathbb{I})\mathrm{tr}(\mathbf{J}) \\ &= 2\mathrm{tr}([\mathbf{w}]_\times \mathbf{J} [\mathbf{w}]_\times) + 2\|\mathbf{w}\|^2 \mathrm{tr}(\mathbf{J}) \end{aligned} \tag{A.4}$$

since $\mathrm{tr}(\|\mathbf{w}\|^2 \mathbb{I}) = 3\|\mathbf{w}\|^2$. As $[\mathbf{w}]_\times = -[\mathbf{w}]_\times^\top$ by definition, this concludes the proof. □

Assuming that $\mathbf{R}_{k+1}^\top \mathbf{R}_{k+1} = \mathbb{I}$, we have

$$\begin{aligned} \mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]\tilde{\mathbf{J}}_\theta [\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]^\top) &= \mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}][\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]^\top \tilde{\mathbf{J}}_\theta) \\ &= \mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} \mathbf{R}_{k+1}^\top \mathbf{R}_k - \mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbf{R}_{k+1}^\top \mathbf{R}_k + \mathbb{I}]\tilde{\mathbf{J}}_\theta) \\ &= \mathrm{tr}([\mathbb{I} - \mathbf{R}_k^\top \mathbf{R}_{k+1}]\tilde{\mathbf{J}}_\theta) + \mathrm{tr}([\mathbb{I} - \mathbf{R}_{k+1}^\top \mathbf{R}_k]\tilde{\mathbf{J}}_\theta) \\ &= -2\mathrm{tr}([\mathbf{R}_k^\top \mathbf{R}_{k+1} - \mathbb{I}]\tilde{\mathbf{J}}_\theta) \end{aligned} \tag{A.5}$$

since $\tilde{\mathbf{J}}_\theta = \mathbf{J}_\theta - \frac{1}{2}\mathrm{tr}(\mathbf{J}_\theta)\mathbb{I}$ is symmetric. This justifies the discrete Lagrangian in equation (3.9).

B Tangent Lift of $\Phi\colon S^3 \to \mathbf{SO}(3)$

It is well-known that $\Phi(\mathbf{q})\mathbf{x} = \mathbf{q}\mathbf{x}\mathbf{q}^*$ for any $\mathbf{x} \in \mathbb{R}^3$.[6] Therefore, for any fixed $\mathbf{x} \in \mathbb{R}^3$, we have

$$\begin{aligned}\frac{d}{dt}\Phi(\mathbf{q})\mathbf{x} &= \frac{d}{dt}(\mathbf{q}\mathbf{x}\mathbf{q}^*)\\ &= \dot{\mathbf{q}}\mathbf{x}\mathbf{q}^* + \mathbf{q}\mathbf{x}\dot{\mathbf{q}}^*\\ &= \mathbf{q}\xi\mathbf{x}\mathbf{q}^* + \mathbf{q}\mathbf{x}(-\xi\mathbf{q}^*)\\ &= \mathbf{q}(\xi\mathbf{x} - \mathbf{x}\xi)\mathbf{q}^*\\ &= \Phi(\mathbf{q})[2(\xi \times \mathbf{x})]\\ &= [2\Phi(\mathbf{q})[\xi]_\times]\mathbf{x}\end{aligned} \tag{B.1}$$

since $\dot{\mathbf{q}} = \mathbf{q}\xi$ for $\xi \in \mathbb{H}_p$ and $\xi^* = -\xi$. Consequently, the tangent lift of Φ is given by

$$(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{q}, \mathbf{q}\xi) \xrightarrow{T_{\mathbf{q}}\Phi} (\Phi(\mathbf{q}), 2\Phi(\mathbf{q})[\xi]_\times). \tag{B.2}$$

Remark. With a slight abuse of notations, we can compute

$$\begin{aligned}\xi\mathbf{x} &= (\xi_i\mathbf{i} + \xi_j\mathbf{j} + \xi_k\mathbf{k})(x_i\mathbf{i} + x_j\mathbf{j} + x_k\mathbf{k})\\ &= -(\xi_i x_i + \xi_j x_j + \xi_k x_k) + (\xi_j x_k - \xi_k x_j)\mathbf{i} + (\xi_k x_i - \xi_i x_k)\mathbf{j} + (\xi_i x_j - \xi_j x_i)\mathbf{k}\\ &= -\xi \cdot \mathbf{x} + \xi \times \mathbf{x}\end{aligned} \tag{B.3}$$

and similarly $\mathbf{x}\xi = -\mathbf{x} \cdot \xi + \mathbf{x} \times \xi$. Therefore, we have

$$\xi\mathbf{x} - \mathbf{x}\xi = 2(\xi \times \mathbf{x}) \tag{B.4}$$

which is used in (B.1).

C Left-Trivialized Momenta and Velocities

Taking any tangent variation $\delta\dot{\mathbf{q}}(t) \in T_q S^3$, we have

$$\begin{aligned}\left\langle \frac{\partial \hat{L}_\theta}{\partial \dot{\mathbf{q}}}, \delta\dot{\mathbf{q}} \right\rangle &= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \hat{L}_\theta(\mathbf{q}, \dot{\mathbf{q}} + \epsilon\delta\dot{\mathbf{q}})\\ &= 2\frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathrm{Im}(\mathbf{q}^*(\dot{\mathbf{q}} + \epsilon\delta\dot{\mathbf{q}}))^\top \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}^*(\dot{\mathbf{q}} + \epsilon\delta\dot{\mathbf{q}}))\\ &= 4\mathrm{Im}(\mathbf{q}^*\dot{\mathbf{q}})^\top \mathbf{J}_\theta \mathrm{Im}(\mathbf{q}^*\delta\dot{\mathbf{q}})\end{aligned} \tag{C.1}$$

Notice that for any $\eta \in \mathfrak{s}^3 \cong \mathbb{R}^3$, we have

$$\begin{aligned}\langle \boldsymbol{\pi}, \eta \rangle &= \left\langle T_{\mathbf{e}}^* L_{\mathbf{q}} \frac{\partial \hat{L}_\theta}{\partial \dot{\mathbf{q}}}(\mathbf{q}, T_{\mathbf{e}} L_{\mathbf{q}}\xi), \eta \right\rangle\\ &= \left\langle \frac{\partial \hat{L}_\theta}{\partial \dot{\mathbf{q}}}(\mathbf{q}, T_{\mathbf{e}} L_{\mathbf{q}}\xi), T_{\mathbf{e}} L_{\mathbf{q}}\eta \right\rangle\\ &= \left\langle \frac{\partial \hat{L}_\theta}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \mathbf{q}\xi), \mathbf{q}\eta \right\rangle\\ &= 4\xi^\top \mathbf{J}_\theta \eta\end{aligned} \tag{C.2}$$

which implies that $\boldsymbol{\pi} = 4\mathbf{J}_\theta\xi$.

---

[6] Here we slightly abuse notation by identifying $\mathbf{x} \in \mathbb{R}^3$ with the pure quaternion $(0, \mathbf{x}) \in \mathbb{H}_p$; the product $\mathbf{q}\mathbf{x}\mathbf{q}^*$ is then interpreted as quaternion multiplication, which then yields a pure quaternion corresponding to a vector in $\mathbb{R}^3$.