Towards Modularity in Floating-Base Deep Lagrangian Networks

Lucas Schulze[1,6], Juliano Decico Negri[2], Victor Barasuol[3], Vivian Suzano Medeiros[2], Marcelo Becker[2], Jan Peters[1,4,5,6], Oleg Arenz[1,6]

***Abstract*— Grey-box methods that embed physical structure into deep learning models offer an attractive alternative to purely black-box learning by improving interpretability, generalization, and sample efficiency. For floating-base systems, Floating-Base Deep Lagrangian Networks (FeLaN) provide a physically consistent parameterization of the inertia matrix that preserves key structural properties such as branch-induced sparsity. However, the original FeLaN formulation still includes a network that depends on all joints, limiting modularity and reuse across related robot morphologies. In this work, we propose a modular extension of FeLaN that decomposes the composite spatial inertia into chain-wise contributions, requiring only one network per kinematic chain. We also introduce an auxiliary loss to improve the identification of individual chain masses. Preliminary results on Spot and Spot with Arm datasets show that the proposed formulation retains inverse-dynamics accuracy close to robot-specific FeLaN models, while reducing the number of trainable parameters and enabling training across related morphologies.**

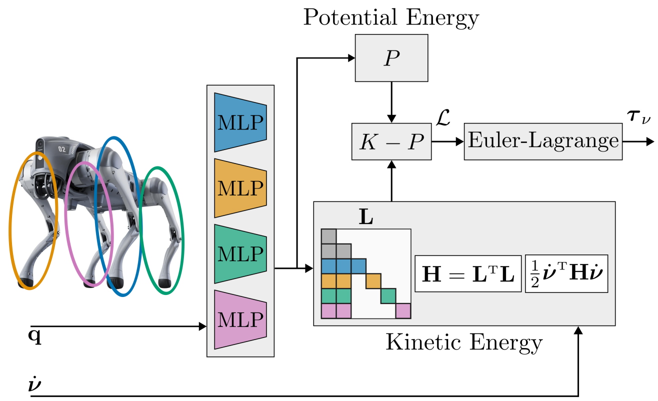

Fig. 1: Modular Floating-Base Deep Lagrangian Networks (FeLaN) for Unitree Go2 quadruped.

I. Introduction

Approaches that incorporate physical structure into deep learning models, commonly referred to as grey-box methods, have emerged as a promising alternative to purely black-box learning, offering improved interpretability, generalization, and sample efficiency. Notable examples include Deep Lagrangian Networks (DeLaN) [1], [2] and Hamiltonian Neural Networks (HNN) [3], which embed Lagrangian and Hamiltonian mechanics, respectively, directly into the model architecture.

Embedding such structure into learned models, therefore, yields a useful inductive bias, especially in low-data regimes [4]. Beyond improving data efficiency, grey-box methods provides a principled way to encode prior physical knowledge. In mechanical systems, properties such as conservation laws arise from invariances and symmetries of the underlying formulation, as stated by Noether's theorem [4].

To address the specific challenges of floating-base systems, Floating-Base Deep Lagrangian Networks (FeLaN) [5] introduced a novel parameterization of the inertia matrix that holds properties such as branch-induced sparsity and full physical consistency of the composite spatial inertia. In its original form, FeLaN uses one network per kinematic chain and an additional network for terms associated with the composite spatial inertia, which takes all joints as input. This coupling limits modularity and requires training separate models for closely related robot morphologies, such as a quadruped with and without an attached arm.

In this work, we present a modular extension of FeLaN that requires only one network per kinematic chain, with each network depending only on the joints of its corresponding chain. We also present an auxiliary loss to improve the identification of the individual chain's mass. We report preliminary results by evaluating the proposed design on combined datasets from related robots, such as Spot with and without an arm.

II. Floating-Base Deep Lagrangian Networks

Lagrangian mechanics describes the dynamics of mechanical systems through the Lagrangian $\mathcal{L}$ defined as the difference between kinetic $K = \frac{1}{2}\dot{\boldsymbol{\nu}}^{\mathrm{T}}\mathbf{H}(\boldsymbol{\nu})\dot{\boldsymbol{\nu}}$ and potential P energies. The Euler-Lagrange equations yield the equation of motion

$$\frac{\partial^2 \mathcal{L}(\boldsymbol{\nu}, \dot{\boldsymbol{\nu}})}{\partial^2 \dot{\boldsymbol{\nu}}}\ddot{\boldsymbol{\nu}} + \frac{\partial \mathcal{L}(\boldsymbol{\nu}, \dot{\boldsymbol{\nu}})}{\partial \boldsymbol{\nu}\partial \dot{\boldsymbol{\nu}}}\dot{\boldsymbol{\nu}} - \frac{\partial \mathcal{L}(\boldsymbol{\nu}, \dot{\boldsymbol{\nu}})}{\partial \boldsymbol{\nu}} = \boldsymbol{\tau}_\nu, \qquad (1)$$

where $\boldsymbol{\nu}$ and $\dot{\boldsymbol{\nu}}$ are the generalized positions and velocities; $\mathbf{H}(\boldsymbol{\nu})$ is the inertia matrix; and $\boldsymbol{\tau}_\nu$ includes all external forces acting on the system, e.g., actuated joint torques, contact forces.

To ensure a non-negative kinetic energy via a positive definite $\mathbf{H}$, DeLaN estimates the inertia matrix through its

This work was funded by the German Research Foundation (DFG) - Project number PE 2315/18-1, and by the German Federal Ministry of Research, Technology and Space (BMFTR) - Project number 01IS23057B, and as part of the Robotics Institute Germany (RIG). This project has been supported by a hardware donation by NVIDIA through the Academic Grant Program.

[1]Department of Computer Science, Technical University of Darmstadt, Germany. [2]Mobile Robotics Group, São Carlos School of Engineering, University of São Paulo (EESC-USP), Brazil. [3]Dynamic Legged Systems Lab, Istituto Italiano di Tecnologia (IIT). [4]Hessian.AI. [5]German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning. [6]Robotics Institute Germany (RIG).
Corresponding author: lucas.schulze@robot-learning.de

Cholesky factor $\mathbf{C}$

$$\mathbf{H}(\boldsymbol{\nu}) = \mathbf{C}(\boldsymbol{\nu})\mathbf{C}(\boldsymbol{\nu})^{\mathrm{T}}, \tag{2}$$

where a softplus activation function and an offset are applied to the diagonal elements of $\mathbf{L}$. The potential energy $P(\boldsymbol{\nu})$ is estimated directly as the output of a second network.

For floating-base systems, the generalized coordinates account for both base pose $\mathbf{q}_{\mathbf{B}} \in \mathrm{SE}(3)$, and joint position $\mathbf{q} \in \mathbb{R}^{\mathrm{n_q}}$, leading to $\boldsymbol{\nu} = [\mathbf{q}_{\mathbf{B}}; \mathbf{q}]$. Thus, the inertia matrix includes the composite spatial inertia, which is defined by a stationary mass, and satisfies a triangle inequality for the rotational inertia [5]. These systems, typically composed of multiple kinematic chains, exhibit partial decoupling between the chains. This decoupling occurs because the chains are only coupled through the base, and results in a branch-induced sparsity of the inertia matrix [6].

However, the standard Cholesky factorization typically results in a dense factor $\mathbf{C}$ independent of the sparsity of $\mathbf{H}$ [7], not preserving the branch-induced sparsity or the properties of the composite spatial inertia in DeLaN.

To preserve the branch-induced sparsity, FeLaN estimates the factor $\mathbf{L}$ of the reordered Cholesky factorization

$$\mathbf{H}(\mathbf{q}) = \mathbf{L}(\mathbf{q})^{\mathrm{T}}\mathbf{L}(\mathbf{q}) \tag{3}$$

where $\mathbf{L}(\mathbf{q})$ is a lower triangular matrix as $\mathbf{C}(\mathbf{q})$. This factorization preserves the same sparsity pattern from $\mathbf{H}$ in $\mathbf{L}$, while having the same numerical properties, e.g., positive definiteness [7]. Each component specific to a kinematic chain is estimated by a neural network and depends only on the joint positions $\mathbf{q}_k$ of the respective chain. For a system with $\mathrm{n_k}$ kinematic branches, $\mathbf{L}$ takes the following structure:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{\mathrm{L}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{L}_{\mathrm{RL}} & \mathbf{L}_{\mathrm{R}} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{L}_{\mathrm{1L}} & \mathbf{L}_{\mathrm{1R}} & \mathbf{L}_{1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{L}_{\mathrm{2L}} & \mathbf{L}_{\mathrm{2R}} & \mathbf{0} & \mathbf{L}_{2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\ \mathbf{L}_{\mathrm{n_k L}} & \mathbf{L}_{\mathrm{n_k R}} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{L}_{\mathrm{n_k}} \end{bmatrix}. \tag{4}$$

To ensure full physical consistency of the composite spatial inertia, FeLaN employs an additional network to estimate the composite first mass moment $\mathbf{h}$ and the Cholesky factor $\mathbf{L}_{\Sigma}$ of the mass distribution covariance matrix, i.e., $\boldsymbol{\Sigma}_{\mathrm{R}} = \mathbf{L}_{\Sigma}^{\mathrm{T}}\mathbf{L}_{\Sigma}$. The total mass m is estimated as a trainable scalar parameter. The potential energy is directly estimated from $\mathbf{h}$, as described in [5], not requiring an additional network.

III. Modular FeLaN

In this section, we show that the composite spatial inertia of the whole body can be written as the sum of the composite spatial inertias of its individual kinematic chains. Based on this decomposition, the full composite spatial inertia can be obtained by summing the chain-wise estimates, removing the need for a dedicated network for the whole robot.

From the Composite-Rigid-Body (CRB) algorithm [6], the composite spatial inertia $\mathbf{M}_i^c$ of the subtree rooted at body i is given by

$$\mathbf{M}_i^c = \mathbf{M}_i + \sum_{j \in \kappa(i)} \mathbf{X}_j^{\mathrm{T}}\mathbf{M}_j^j\mathbf{X}_j \tag{5}$$

where $\mathbf{M}_i$ is the spatial inertia of body i; $\mathbf{X}_j$ is the spatial transformation from child j to its parent; $\mathbf{M}_j^j$ is the ith child body spatial inertia w.r.t the ith child frame; and $\kappa(i)$ denotes the set of children of body i.

Therefore, the whole body composite spatial inertia $\mathbf{H}_{\mathbf{B}}$ expressed in the base frame can be written as

$$\mathbf{H}_{\mathbf{B}} = \mathbf{M}_b + \sum_{k=1}^{\mathrm{n_k}} \mathbf{X}_k^{\mathrm{T}}\mathbf{M}_k^k\mathbf{X}_k. \tag{6}$$

Since each transformation $\mathbf{X}_k$ depends only on $\mathbf{q}_k$, this yields

$$\mathbf{H}_{\mathbf{B}}(\mathbf{q}) = \mathbf{M}_b + \sum_{k=1}^{\mathrm{n_k}} \mathbf{M}_k(\mathbf{q}_k) \tag{7}$$

here $\mathbf{M}_k(\mathbf{q}_k)$ denotes the composite spatial inertia contribution of chain k expressed in the base frame.

Expanding $\mathbf{H}_{\mathbf{B}}$ gives the corresponding additive relations for the total mass, first mass moment, and rotational inertia

$$\mathrm{m} = \mathrm{m}_b + \sum_{k=1}^{\mathrm{n_k}} \mathrm{m}_k \tag{8}$$

$$\mathbf{h}(\mathbf{q}) = \mathbf{h}_b + \sum_{k=1}^{\mathrm{n_k}} \mathbf{h}_k(\mathbf{q}_k) \tag{9}$$

$$\mathbf{I}_{\mathbf{B}}(\mathbf{q}) = \mathbf{I}_b + \sum_{k=1}^{\mathrm{n_k}} \mathbf{I}_k(\mathbf{q}_k) \tag{10}$$

Therefore, instead of estimating $\mathbf{h}(\mathbf{q})$ and $\mathbf{I}_{\mathbf{B}}(\mathbf{q})$ using a network that takes the full joint configuration $\mathbf{q}$ as input, we estimate the chain-wise quantities $\mathbf{h}_k(\mathbf{q}_k)$ and $\mathbf{I}_k(\mathbf{q}_k)$ using the existing per-chain networks. The base terms, $\mathbf{h}_b$ and $\mathbf{I}_b$, are constant and can be represented as trainable parameters.

Besides being more physically consistent, since the additive structure is enforced explicitly rather than learned implicitly, this formulation also enables reuse of trained chain models across robots with similar morphologies, e.g., a quadruped and a quadruped equipped with an arm.

A. Identifiability of Individual Chain Masses

The decomposition of $\mathbf{I}_{\mathbf{B}}(\mathbf{q})$ and $\mathbf{h}(\mathbf{q})$ into chain components does not introduce additional challenges, since gradients with respect to each chain configuration arise naturally through their explicit dependence on $\mathbf{q}_k$. However, identifying the individual components of m is more challenging, since the inertia matrix depends only on their sum.

To recover an additional learning signal for each chain mass, we first consider the single-rigid-body case. For a rigid body i, define the joint-space inertia block $\mathbf{H}_i$ and the linear coupling term $\mathbf{H}_{i\mathrm{L}}$ as

$$\mathbf{H}_i = \mathbf{L}_i^{\mathrm{T}}\mathbf{L}_i = \mathrm{m}_i\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i + \mathbf{J}_{\mathrm{r}\,i}^{\mathrm{T}}\mathbf{I}_{\mathrm{r}i}\mathbf{J}_{\mathrm{r}i}, \tag{11}$$

$$\mathbf{H}_{i\mathrm{L}} = \mathbf{L}_i^{\mathrm{T}}\mathbf{L}_{i\mathrm{L}} = \mathrm{m}_i\mathbf{J}_i, \tag{12}$$

TABLE I: Inverse dynamics NMSE and mass estimation errors of FeLaN and Modular for Spot and Spot with Arm.

Method	Test Spot	Test Spot with Arm	$\lvert\Delta \mathrm{m}\rvert_{\mathrm{Total}}$	$\lvert\Delta \mathrm{m}\rvert_{\mathrm{Base}}$	$\lvert\Delta \mathrm{m}\rvert_{\mathrm{Arm}}$	$\lvert\Delta \mathrm{m}\rvert_{\mathrm{Legs}}$	# Parameters
FeLaN Spot	**6.32**	–	3.32	–	–	–	3994
FeLaN Spot with Arm	–	**9.88**	**1.49**	–	–	–	5920
Modular FeLaN ($w_{\mathrm{m}} = 0$)	8.28	10.11	2.46	11.17	1.73	2.81	5735
Modular FeLaN ($w_{\mathrm{m}} = $ 1e-3)	8.25	10.11	2.46	**5.81**	**1.73**	**1.10**	5735

where $\mathbf{L}_i$ and $\mathbf{L}_{i\mathrm{L}}$ are blocks of the reordered Cholesky factor in (4); $\mathbf{I}_{\mathrm{r}i}$ is the rotational inertia; $\mathbf{J}_i$ and $\mathbf{J}_{\mathrm{r}i}$ are the linear and rotational jacobians, respectively. Combining (11) and (12) yields

$$\mathbf{L}_{i\mathrm{L}} = \mathbf{H}_i^{-\mathrm{T}}\mathbf{H}_{i\mathrm{L}} \tag{13}$$

Expanding the product of the Cholesky factor in (3), the block associated with the total mass can be written as

$$\mathbf{U}_i = \mathbf{L}_{i\mathrm{L}}^{\mathrm{T}}\mathbf{L}_{i\mathrm{L}} = \mathbf{H}_{i\mathrm{L}}^{\mathrm{T}}(\mathbf{L}_i\mathbf{L}_i^{\mathrm{T}})^{-1}\mathbf{H}_{i\mathrm{L}} = \mathbf{H}_{i\mathrm{L}}^{\mathrm{T}}(\mathbf{H}_i)^{-1}\mathbf{H}_{i\mathrm{L}} \tag{14}$$

From (11), and using $\mathbf{I}_{\mathrm{r},i} \succeq 0$, we have

$$\mathbf{H}_i \succeq \mathrm{m}_i\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i. \tag{15}$$

Assuming that $\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i$ is nonsingular, this implies

$$\mathbf{H}_i^{-1} \preceq \frac{1}{\mathrm{m}_i}(\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i)^{-1}. \tag{16}$$

Substituting into the previous expression gives

$$\mathbf{U}_i \preceq \mathrm{m}_i\,\mathbf{J}_i^{\mathrm{T}}(\mathbf{J}_i\mathbf{J}_i^{\mathrm{T}})^{-1}\mathbf{J}_i, \tag{17}$$

or the equivalent projector form according to the adopted Jacobian convention. Therefore, under the corresponding full-rank assumption on $\mathbf{J}_i$,

$$\mathbf{U}_i \preceq \mathrm{m}_i\mathbf{1}_3 \tag{18}$$

which implies

$$\lambda_{\max}(\mathbf{U}_i) \leq \mathrm{m}_i. \tag{19}$$

For a kinematic tree, analogous expressions follow from composing bodies within each chain. However, in this case $\mathbf{U}_i$ contains cross-terms between bodies, which makes a direct extension of (19) nontrivial. Establishing an equivalent bound for the multi-body case is left for future work.

Motivated by (19), we define the following auxiliary loss for each chain mass:

$$l_{\mathrm{m}} = \sum_{k=0}^{\mathrm{n_k}} \mathrm{mean}\Big(\mathrm{softplus}\big(\boldsymbol{\lambda}_{\mathbf{U}k} - \mathrm{m}_k\big)\Big), \tag{20}$$

where

$$\boldsymbol{\lambda}_{\mathbf{U}k} = \lambda_{\max}(\mathbf{L}_{k\mathrm{L}}^{\mathrm{T}}\mathbf{L}_{k\mathrm{L}})\,. \tag{21}$$

This term provides an additional learning signal for each chain mass and improves identifiability in practice, as shown in Sec. IV. Note that this auxiliary term acts as a soft constraint and does not affect other physical consistency properties. The auxiliary loss l_{m} is combined with the original FeLaN loss l_{FeLaN}, yielding the optimization problem

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} l_{\mathrm{FeLaN}} + w_{\mathrm{m}}l_{\mathrm{m}} \tag{22}$$

IV. Experiments

In this section, we present preliminary results using the datasets for Spot and Spot with Arm published in [5] and available online[1]. To evaluate the proposed modular formulation, we trained four models: FeLaN on Spot, FeLaN on Spot with Arm, Modular FeLaN with the additional loss (20), and Modular FeLaN without this additional term ($w_{\mathrm{m}} = 0$). All networks are implemented with 2 layers of 16 neurons.

The modular models were trained on a combined dataset of both robots, totaling 24k samples, evenly split between Spot and Spot with Arm. In contrast, each original FeLaN model was trained separately on its corresponding robot dataset, also using 24k samples. Therefore, each individual FeLaN had access to twice as much data from its specific morphology as the modular model had for each robot.

Table I reports the normalized mean square error (NMSE) of inverse dynamics on a fixed test set of 6k samples for each robot. The modular formulation yields slightly higher NMSE than the specific FeLaN. This is expected, since the two platforms are not identical apart from the arm: the datasets come from two different Spot robots, one of which permanently carries an arm. Thus, the modular model must not only account for the arm but also average the differences between the two robots. Importantly, it does so while using less robot-specific data than the individual FeLaN models.

Both modular variants achieved similar inverse dynamics performance. However, including the auxiliary mass loss reduced the deviation $\lvert\Delta \mathrm{m}\rvert$ between the estimated chain masses and the nominal mass values, without affecting the inverse dynamics performance or the total mass estimation.

V. Conclusion

In this work, we presented a modular extension of FeLaN that decomposes the composite spatial inertia into chain-wise terms, removing the need for a dedicated network that depends on all joints. This yields a more modular architecture with fewer parameters and enables training across related robot morphologies. We also introduced an auxiliary loss for improved identification of individual chain masses. Preliminary results on similar morphologies show that the proposed formulation achieves comparable predictive performance while improving modularity and parameter efficiency.

References

[1] M. Lutter *et al.*, "Deep lagrangian networks: Using physics as model prior for deep learning," in *International Conference on Learning Representations*, 2019.

[1] `https://schulze18.github.io/felan_website`

[2] M. Lutter and J. Peters, “Combining physics and deep learning to learn continuous-time dynamics models,” *The International Journal of Robotics Research*, 2023.
[3] S. Greydanus *et al.*, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems*, 2019.
[4] J. Watson *et al.*, “Machine learning with physics knowledge for prediction: A survey,” *Trans. on Machine Learning Research*, 2025.
[5] L. Schulze, J. D. Negri, V. Barasuol, V. S. Medeiros, M. Becker, J. Peters, and O. Arenz, “Floating-base deep lagrangian networks,” 2026.
[6] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer US, 2008.
[7] ——, “Efficient factorization of the joint-space inertia matrix for branched kinematic trees,” *The International Journal of Robotics Research*, June 2005.