A Differentiable Newton Euler Algorithm for Multi-body Model Learning

Michael Lutter [*1] **Johannes Silberbauer** [*1] **Joe Watson** [1] **Jan Peters** [1]

1. Introduction

The identification of dynamical systems from data is a powerful tool in Robotics (Åström & Eykhoff, 1971). Learnt analytic models may be used for control synthesis and can be utilized for gravitational and inertial compensation (Spong et al., 2020). Moreover, when used as simulators, they can be used to reduce the sample complexity of data-driven control methods such as Reinforcement Learning (Deisenroth & Rasmussen, 2011; Chua et al., 2018). For these control applications, where out-of-distribution prediction is typically required, the ability to generalize beyond the acquired data is critical. Any modelling error may be exploited by a controller, and such exploitation may result in catastrophic system failure. To ensure sufficient out-of-sample generalization, the model's hypothesis space is an important consideration. Ideally, this space should be defined such that only plausible trajectories, that are physically consistent and have bounded energy, are generated.

Standard black-box models such as deep networks or Gaussian processes, which are a common choice for model learning (Nguyen-Tuong & Peters, 2011; Deisenroth & Rasmussen, 2011; Chua et al., 2018), have a broad hypothesis space and so can simulate dynamics with unbounded energy. To overcome this shortcoming, 'grey-box' models that combine deep networks with physical insights have been recently proposed, e.g., incorporating Lagrangian (Lutter et al., 2019; Lutter & Peters, 2019; Gupta et al., 2019) and Hamiltonian Mechanics (Greydanus et al., 2019) for energy-conserving models. These empirically show better generalization compared to black-box models, but the local representations of deep networks cannot guarantee out-of-sample generalization. Only white-box models (Atkeson et al., 1986; Ting et al., 2006; Traversaro et al., 2016; Wensing et al., 2017; Sutanto et al., 2020), which infer the physical parameters of the system given the analytic equations of motion, can guarantee out-of sample generalization as these models are valid globally. While these combinations of physics with data-driven learning can obtain more robust representations, the usage of physics priors commonly reduces model accuracy compared to black-box methods as most physics priors cannot capture reality to a sufficient fidelity. The priors of Newtonian-, Lagrangian- and Hamiltonian mechanics typically cannot describe the complex nonlinear phenomena of friction, hysteresis and contact. Therefore, the need of a compromise between physics-inspired models and black-box functions approximation is clear.

In this work, we examine a spectrum of hybrid models for the domain of multibody robot dynamics. We motivate a computation graph architecture that embodies the Newton Euler equations, emphasising the utility of the Lie Algebra form in translating the dynamical geometry into an efficient computational structure for learning (Handa et al., 2016). We describe the used actuator models (Section 3) and the virtual parameters (Appendix). In the experiments, we evaluate 26 Newton-Euler based system identification approaches and benchmark these models on the simulated and physical Furuta Pendulum and Cartpole. The comparison shows that the kinematic parameters, required by previous Newton-Euler methods (Atkeson et al., 1986; Sutanto et al., 2020; Ledezma & Haddadin, 2017), can be accurately inferred from data. Furthermore, we highlight that models with guaranteed bounded energy of the uncontrolled system generate non-divergent trajectories, while more general models have no such guarantee. Therefore, their performance strongly depends on the data distribution. The main contributions of this work are the introduction of a white-box model that jointly learns dynamic and kinematics parameters and can be combined with black-box components. We then provide an extensive empirical evaluation on challenging systems and different datasets that elucidates the comparative performance of our grey-box architecture with comparable white- and black-box models.

2. Newton Euler Equations

Consider a set of coupled rigid bodies, such as a robotic manipulator. While the global dynamics of such a system is complex, the behaviour of each individual body and its effect on its neighbours are easily understood. This is the key insight of algorithms such as Recursive Newton Euler Algorithm (RNEA) and Articulated Body (ABA) (Featherstone, 2007), which efficiently solve for the global dynamics

[*] Equal contribution [1] Computer Science Department, Technical University of Darmstadt, Darmstadt, Germany. Correspondence to: Michael Lutter <michael@robot-learning.de>.

Accepted at Workshop on Inductive Biases, Invariances and Generalization, International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

by successively propagating the local solutions forward and backwards. While alternative methods for modelling, such as Lagrangian and Hamiltonian Mechanics, may be appreciated for their mathematical elegance and convenience; the Newton Euler approach is based on global and interpretable physical quantities enabling robust out-of-sample generalization. Moreover, the efficient algorithmic form is applicable for both forward and inverse dynamics We describe the Newton Euler method in its Lie Algebra form (Kim, 2012), which is not only compact, but easy to represent as a differentiable computation graph (Handa et al., 2016).

In this paper we focus on kinematic trees composed on n components. Given a base reference frame, the pose of each body can be described via successive application of an affine transform $\boldsymbol{T}$ containing the 3x3 rotation matrix $\boldsymbol{R}$ and translation $\boldsymbol{p}{\in}\mathbb{R}^3$. To describe the dynamics of each body, we use a 'generalised' velocity $\bar{\boldsymbol{v}}{=}[\boldsymbol{v}, \boldsymbol{\omega}]^\intercal$, which is composed of linear ($\boldsymbol{v}$) and rotational ($\boldsymbol{\omega}$) components. The generalised velocity $\bar{\boldsymbol{v}}_i$ expresses the link velocity in the inertial base frame but expressed in the ith link coordinate frame. The generalised notation is extended to acceleration $\bar{\boldsymbol{a}}$, force $\bar{\boldsymbol{f}}{=}[\boldsymbol{f}, \boldsymbol{\tau}]^\intercal$ (with moment $\boldsymbol{\tau}$), momentum $\bar{\boldsymbol{l}}$, and inertia

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{J} & m[\boldsymbol{p}_m] \\ m[\boldsymbol{p}_m]^\intercal & m\boldsymbol{I} \end{bmatrix} \tag{1}$$

with link mass m, link inertia $\boldsymbol{J}$ and link center of mass (CoM) position $\boldsymbol{p}_m$ relative to the body frame, and $[\boldsymbol{x}]$ mapping the vector $\boldsymbol{x}$ to a skew-symmetric matrix such that $\boldsymbol{x}{\times}\boldsymbol{y}{=}[\boldsymbol{x}]\boldsymbol{y}$. The benefit of the Lie Algebra formulation is apparent when considering the kinematics, dynamics and differential equations. Transforming the kinematic terms (e.g. $\bar{\boldsymbol{v}}, \bar{\boldsymbol{a}}$) of the ith body in the jth frame is a compact linear operations through use of the adjoint transform of $\boldsymbol{T}$ ($\text{Ad}_{\boldsymbol{T}}$), i.e.,

$$\bar{\boldsymbol{v}}_j = \text{Ad}_{\boldsymbol{T}_{j,i}} \bar{\boldsymbol{v}}_i, \qquad \bar{\boldsymbol{a}}_j = \text{Ad}_{\boldsymbol{T}_{j,i}} \bar{\boldsymbol{a}}_i. \tag{2}$$

In the Lie algebra formulation, the kinematic terms can be described in the SE(3) manifold due to the properties of the affine transform $\boldsymbol{T}$. Dynamic terms (e.g. $\bar{\boldsymbol{l}}, \bar{\boldsymbol{f}}$) can be shown to act in the *dual space* for SE(3), DSE(3). Hence, these terms can be transformed between frames using the coadjoint operator $\text{Ad}^*_{\boldsymbol{T}_{j,i}}{=}\text{Ad}^\intercal_{\boldsymbol{T}_{j,i}}$, i.e.,

$$\bar{\boldsymbol{l}}_j = \text{Ad}^*_{\boldsymbol{T}_{j,i}} \bar{\boldsymbol{l}}_i, \qquad \bar{\boldsymbol{f}}_j = \text{Ad}^*_{\boldsymbol{T}_{j,i}} \bar{\boldsymbol{f}}_i. \tag{3}$$

Differential equations in the inertial base frame can also be expressed compactly, i.e., the Newton-Euler equation is described by

$$\bar{\boldsymbol{f}} = \frac{d}{dt}\bar{\boldsymbol{l}} = \bar{\boldsymbol{M}}\bar{\boldsymbol{a}} - \text{ad}^*_{\bar{\boldsymbol{v}}}\bar{\boldsymbol{M}}\bar{\boldsymbol{v}}, \;\; \text{where} \;\; \text{ad}^*_{\bar{\boldsymbol{v}}} = \begin{bmatrix} [\boldsymbol{\omega}] & \boldsymbol{0} \\ [\boldsymbol{v}] & [\boldsymbol{\omega}] \end{bmatrix}.$$

These transformations and differential equations can be embedded in articulated body algorithm (ABA) and the Recursive Newton Euler Algorithm (RNEA). These different formulation yield more compact and intuitive description compared to the original formulation (Featherstone, 2007), e.g., the ABA algorithm in Lie algebra is described in Algorithm 1.

3. Actuator Models

The models of the previous sections mostly focused on simulating rigid body dynamics. This representation does commonly not capture reality with sufficient fidelity for mechanical systems with actuation. Actuators exerting the control signal are affected by non-linear transformation of the set-points, hysteresis and friction. To be able to learn such models with non-ideal actuators, we augment the rigid-body dynamics model with an actuator model. This actuation model can either be white-box model relying on existing friction models or black-box models. We define six different joint independent actuator models, building from $\tau = \tau_d$,

$$\begin{aligned}
\text{Viscous:} \quad & \tau = \tau_d {-} \mu_v \dot{q}, \\
\text{Stribeck:} \quad & \tau = \tau_d {-} \text{sign}(\dot{q}) \left(f_s {+} f_d \exp\left(-\nu_s \dot{q}^2\right)\right) - \mu_v \dot{q}, \\
\text{NN Friction:} \quad & \tau = \tau_d {-} \text{sign}(\dot{q}) \, \| f_{\text{NN}}(q, \dot{q}; \psi_F) \|_1, \\
\text{NN Residual:} \quad & \tau = \tau_d {-} f_{\text{NN}}(q, \dot{q}; \psi_R), \\
\text{FF-NN:} \quad & \tau = f_{\text{NN}}(\tau_d, q, \dot{q}; \psi_M),
\end{aligned}$$

where τ_d is the desired torque. The Viscous, Stribeck and MLP friction actuator models are guaranteed to learn a stable uncontrolled system (i.e., τ_d:=0) as all possible actuator parameters can only dissipate energy, i.e., $\dot{E} = \boldsymbol{\tau}^T \dot{\boldsymbol{q}} \leq 0 \; \forall \; \theta_F$. The Residual and MLP actuator models are capable of generating energy and so the performance will depend on the training data.

4. Experimental Setup

In the following the experimental setup containing the used systems, datasets and models is described.

4.1. Systems

For the comparison we choose the Quanser Cartpole and the Quanser Furuta Pendulum as these systems are under-actuated and the physical systems have different peculiarities, which make system identification challenging. System identification for under-actuated systems is harder compared to fully-actuated systems as all data points lie on a low dimensional manifold and one cannot capture the whole space.

4.2. Datasets

To evaluate the impact of data quality, we are evaluating the performance on three different datasets with different levels of complexity. The simulated data from uniform

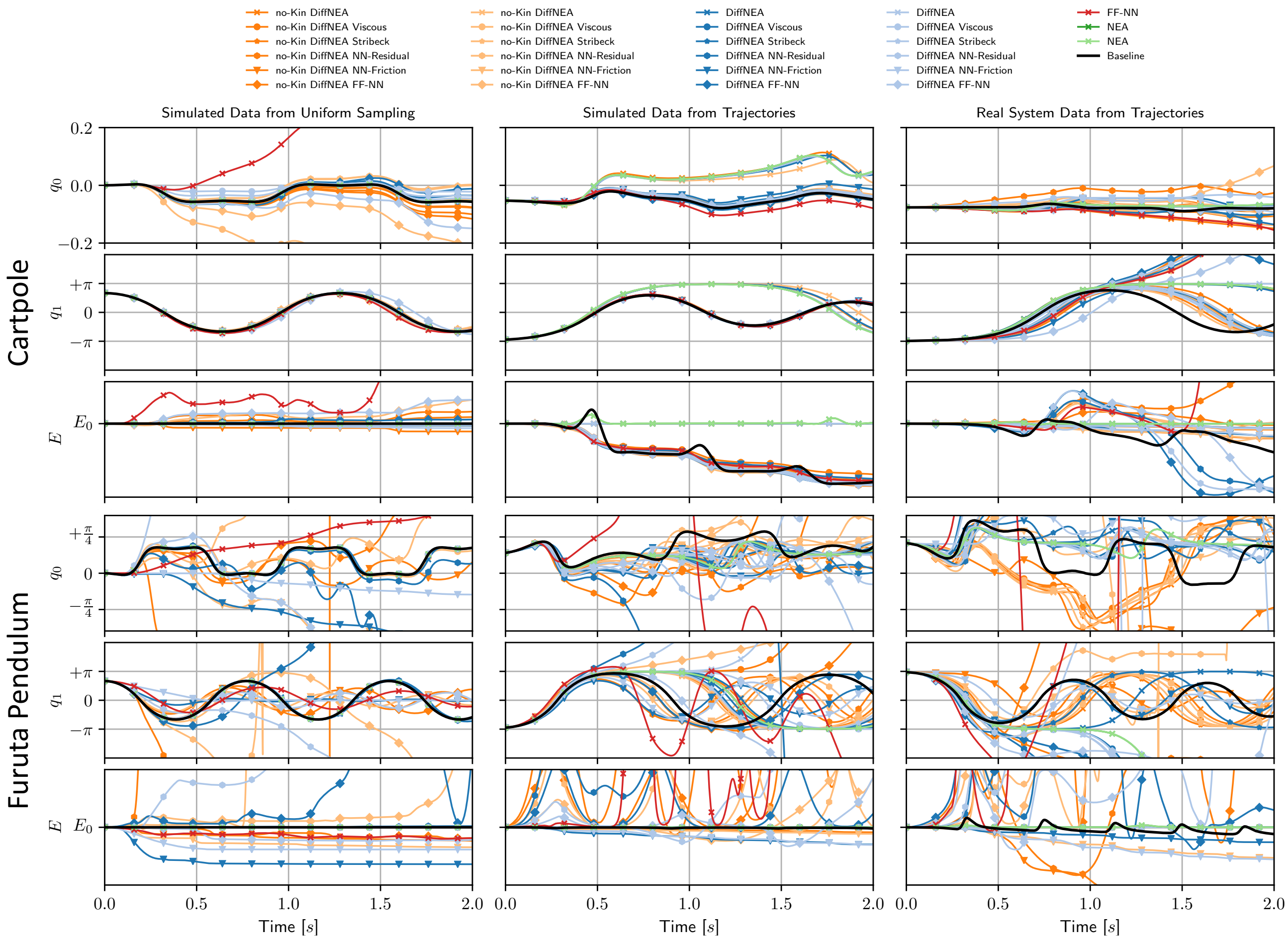

Figure 1. Qualitative model comparison of the 26 different models performing forward roll-outs on the Cartpole and Furuta Pendulum. The roll-outs start from the starting state and are computed with 250Hz sampling frequency and integrated with RK4. The models are trained on three different datasets ranging from uniformly sampled and ideal observations (i.e., Simulated Data from Uniform Sampling) to trajectory data of noisy observation from the physical system.

sampling dataset is generated by sampling joint positions, velocities and torques from a uniform distribution spanning the complete state domain and computing the acceleration with the true analytic forward dynamics. The simulated data from trajectories dataset is generated by simulating the ideal system with viscous friction and small state and action noise. The real system data from trajectories dataset is generated on the physical system by applying an energy controller that repeatedly swings-up the pendulum and lets it fall down without actuation.

4.3. Models

For the evaluation we compare three different instantiations of the previously described white-box model family with the 6 different actuation models, if applicable.

No-Kin Differential Newton Euler Alg.: The no-Kin DiffNEA model assumes knowledge of the kinematic tree but no knowledge of the kinematics $\boldsymbol{\theta}_K$ or dynamics parameters $\boldsymbol{\theta}_I$. These link parameters are learned from data containing only generalized coordinates using ADAM to minimize the squared loss of the forward dynamics.

Differential Newton Euler Alg.: The DiffNEA model assumes knowledge of the kinematic chain and the kinematics parameters $\boldsymbol{\theta}_K$ and only learns $\boldsymbol{\theta}_I$. These parameters are learnt by minimizing the squared loss of the forward dynamics using ADAM. This approach was introduced by (Sutanto et al., 2020).

Newton Euler Alg.: The NEA model assumes knowledge of the kinematic chain and the kinematics parameters $\boldsymbol{\theta}_K$ and only learns $\boldsymbol{\theta}_I$. These parameters are learnt by linear regression. This model learning approach was initially introduced by (Atkeson et al., 1986). Due to the linear regression this model cannot be augmented with the different actuation models.

Feed-Forward Neural Network (FF-NN) As black-box model learning baseline, we are using a feed-forward neural network trained via ADAM. This network is a continuous time model and predicts the joint acceleration.

4.4. Model Initialization

For the white box models we are differentiating between two initialization strategies, without prior and with prior. Without prior means that the link parameters are initialized randomly. With prior means that the parameters are initialized with the known values given by the manufacturer. This differentiation enables us to evaluate the impact of good initialization for white box models.

5. Experimental Results

The qualitative experimental results are shown in Figure 1. The overall performance depends heavily on the system as well as the dataset. The numerically sensitive conditioning of the Furuta Pendulum causes all models to be worse on all datasets and model classes. Conversely, the magnitude of the physical parameters of the Cartpole make the identification and long-term prediction is simpler. Regarding the datasets, the overall forward prediction performance, with some outliers, decreases with dataset complexity. Regarding the different models, no clear best system identification representation and identification approach from those studied is apparent. One interesting result is that the white-box model approach with *unknown* kinematics (no-Kin DiffNEA) performs comparable to the white-box model with *known* kinematics (DiffNEA, NEA), demonstrating the kinematics and dynamics can be learned jointly. Furthermore, one can observe that the three different model classes, energy-conserving, energy-bounded and energy-unbounded models achieve very different long-term forward prediction.

5.1. Energy Conserving Models

The energy-conserving models, i.e. no-Kin DiffNEA, DiffNEA, NEA and no-actuator model, can only represent energy conserving dynamics. When this prior is correct, as in the simulated uniform dataset, these models perform well. If the prior is not correct, e.g., the large viscous friction of the *simulated* Cartpole, the forward prediction degrades significantly even for simulated data. The identified parameters are physically plausible but unreasonable. While the mass of the cart is about 0.5kg, this model class sets the mass to 1kg for the simulation and 2kg for the physical system. The mass of the about 120g pendulum is set to 340g for the simulation and 3000g for the physical system.

5.2. Energy Bounded Models

The energy-bounded models, i.e., NEA plus Viscous, Stribeck and NN-Friction actuator, guarantee that the energy does not increase without actuation and hence, guarantee to learn a global stable system. This model class contains the best performing models of this benchmark. The learnt models always yield non-diverging trajectories and can model systems with and without friction. The NN-Friction actuator achieves particularly good performance by exploiting its black-box flexibility within the inductive bias. Despite its expressiveness, the model does not overfit and the obtained physical parameters are comparable to the white-box actuator models.

5.3. Energy Unbounded Models

The energy-unbounded models, i.e., FF-NN, NEA plus FF-NN and NN-Residual actuator, can potentially learn to increase the system energy during simulation without actuation, which is physically implausible. The benchmark of Figure 1 shows that models of this class learn to pump energy into the system even for perfect sensor measurements (i.e., FF-NN of the Cartpole with simulated uniformly sampled data). For the more challenging trajectory datasets, all black-box and hybrid models learn models that increase in energy during simulation without actuation. Many of these models also generate divergent trajectories during simulation, which leave the training domain.

5.4. Model Initialization

For the the hybrid white-box models the different model initializations were compared. In the simulation experiments no clear difference between models with and without prior initialization is visible. Evaluating the identified physical parameters also yields no clear improvement of the initialization with prior, e.g., even for unreasonably large physical parameters, the identified parameter with a prior was not necessarily smaller. Therefore, we conclude that the initialization with the best known parameters does not necessarily improve model performance when using stochastic gradient descent with ADAM.

6. Conclusion

In this paper, we have described the classical Newton-Euler system identification approach (Atkeson et al., 1986) using the elegant Lie algebra formulation (Kim, 2012) and the differential programming paradigm. We combined this formulation with white- and black-box actuation models for end-to-end learning. In a large-scale benchmark, we compared 26 different models on three datasets and two different systems. The two main conclusions of this benchmark are, (1) the no-Kin DiffNEA model, which learns the kinematics, performs equally well compared to the same DiffNEA with the kinematics given, and (2) models with guaranteed global stability yield the best long term forward simulations.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No #640554 (SKILLS4ROBOTS). Furthermore, this research was also supported by grants from ABB, NVIDIA and the NVIDIA DGX Station.

References

Åström, K. J. and Eykhoff, P. System identification—a survey. *Automatica*, 1971.

Atkeson, C. G., An, C. H., and Hollerbach, J. M. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research (IJRR)*, 1986.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeuRIPS)*, 2018.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *International Conference on machine learning (ICML)*, 2011.

Featherstone, R. *Rigid Body Dynamics Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2007.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeuRIPS)*, 2019.

Gupta, J. K., Menda, K., Manchester, Z., and Kochenderfer, M. J. A general framework for structured learning of mechanical systems. *arXiv preprint arXiv:1902.08705*, 2019.

Handa, A., Bloesch, M., Pătrăucean, V., Stent, S., McCormac, J., and Davison, A. gvnn: Neural network library for geometric computer vision. In *European Conference on Computer Vision (ECCV)*, 2016.

Kim, J. Lie group formulation of articulated rigid body dynamics. Technical report, Carnegie Mellon University, 2012.

Ledezma, F. D. and Haddadin, S. First-order-principles-based constructive network topologies: An application to robot inverse dynamics. In *International Conference on Humanoid Robotics (Humanoids)*, 2017.

Lutter, M. and Peters, J. Deep Lagrangian Networks for end-to-end learning of energy-based control for under-actuated systems. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

Lutter, M., Ritter, C., and Peters, J. Deep Lagrangian Networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Nguyen-Tuong, D. and Peters, J. Model learning for robot control: a survey. *Cognitive Processing*, 2011.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. *Robot modeling and control*. John Wiley & Sons, 2020.

Sutanto, G., Wang, A., Lin, Y., Mukadam, M., Sukhatme, G., Rai, A., and Meier, F. Encoding physical constraints in differentiable Newton-Euler Algorithm. *Learning for Dynamics Control (L4DC)*, 2020.

Ting, J.-A., Mistry, M., Peters, J., Schaal, S., and Nakanishi, J. A bayesian approach to nonlinear parameter identification for rigid body dynamics. In *Robotics: Science and Systems (RSS)*, 2006.

Traversaro, S., Brossette, S., Escande, A., and Nori, F. Identification of fully physical consistent inertial parameters using optimization on manifolds. In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

Wensing, P. M., Kim, S., and Slotine, J.-J. E. Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters (RAL)*, 2017.

Appendix

Articulated Body Algorithm in Lie Algebra

Virtual Physical Parameters

To enable learning via standard gradient descent methods, we carefully parameterize the physical parameters of the algorithm by a set of unrestricted, *virtual* parameters (Ting et al., 2006; Sutanto et al., 2020) to result in physically correct model parameters. These virtual are necessary as not all link parameters are physically plausible, e.g., the link mass has to be positive.

Kinematics

The transformation $\boldsymbol{T}(q)$ between two links depends on link dimension, the joint position and the joint constraint connecting the two links. We decompose this transformation $\boldsymbol{T}(q){=}\boldsymbol{T}_O\boldsymbol{T}_q(q)$ into a fixed transform $\boldsymbol{T}_O$ and variable transform $\boldsymbol{T}_q(q_i)$. The fixed transform describes the the distance and rotation between two joints and is parametrized by the translation vector $\boldsymbol{p}_k$ and the RPY Euler angles

Algorithm 1 Forward dynamics with Articulated Rigid Body for a kinematic tree in terms of Lie algebra.

```
Input: $\boldsymbol{q}_{1:n}$, $\dot{\boldsymbol{q}}_{1:n}$, $\boldsymbol{u}_{1:n}$
Output: $\ddot{q}_{0:n}$, $\bar{\boldsymbol{a}}_{0:n}$, $\bar{\boldsymbol{f}}_{0:n}$
for $i = 1$ to $n$ do
    // Forward Kinematics
    $\bar{\boldsymbol{v}}_i = Ad_{T^{-1}_{\lambda,i}} \bar{\boldsymbol{v}}_\lambda + \boldsymbol{s}_i \dot{q}_i$
    $\bar{\boldsymbol{\eta}}_i = ad_{\bar{\boldsymbol{v}}_i} \boldsymbol{s}_i \, \dot{q}_i$
end for
for $i = n$ to $1$ do
    // Compute lumped inertia
    $\bar{\boldsymbol{M}}_{i:n} = \bar{\boldsymbol{M}}_i + \sum_{k \in \mu} Ad^*_{T^{-1}_{i,k}} \bar{\Pi}_k Ad_{T^{-1}_{i,k}}$
    // Compute bias forces
    $\bar{\boldsymbol{f}}_{b,i} = -ad^*_{\bar{\boldsymbol{v}}_i} \bar{\boldsymbol{M}}_{i:n} \bar{\boldsymbol{v}}_i + \sum_{k \in \mu} Ad^*_{T^{-1}_{i,k}} \left( \bar{\boldsymbol{f}}_{b,k} + \bar{\boldsymbol{\beta}}_k \right)$
    $\bar{\Psi}_i = \left( \boldsymbol{s}_i^T \, \bar{\boldsymbol{M}}_{i:n} \, \boldsymbol{s}_i \right)^{-1}$
    $\bar{\boldsymbol{\Pi}}_i = \bar{\boldsymbol{M}}_{i:n} - \bar{\boldsymbol{M}}_{i:n} \boldsymbol{s}_i \bar{\Psi}_i \boldsymbol{s}_i^T \bar{\boldsymbol{M}}_{i:n}$
    $\bar{\boldsymbol{\beta}}_i = \bar{\boldsymbol{M}}_{i:n} \left( \bar{\boldsymbol{\eta}}_i + \boldsymbol{s}_i \bar{\Psi}_i \left( u_i - \boldsymbol{s}_i^T \left( \bar{\boldsymbol{M}}_{i:n} \bar{\boldsymbol{\eta}}_i + \bar{\boldsymbol{f}}_{b,i} \right) \right) \right)$
end for
for $i = 1$ to $n$ do
    // Newton Euler Equations
    $\ddot{q}_i = \bar{\Psi}_i \left( u_i - \boldsymbol{s}_i^T \left( \bar{\boldsymbol{M}}_{i:n} \left( Ad_{T^{-1}_{\lambda,i}} \bar{\boldsymbol{a}}_\lambda + \bar{\eta}_i \right) - \bar{\boldsymbol{f}}_{b,i} \right) \right)$
    $\bar{\boldsymbol{a}}_i = Ad_{T^{-1}_{\lambda,i}} \bar{\boldsymbol{a}}_\lambda + \boldsymbol{s}_i \ddot{q}_i + \bar{\boldsymbol{\eta}}_i$
    $\bar{\boldsymbol{f}}_i = \bar{\boldsymbol{M}}_{i:n} \bar{\boldsymbol{a}}_i + \bar{\boldsymbol{f}}_{b,i}$
end for
```

$\boldsymbol{\theta}_R{=}[\phi_x, \phi_y, \phi_z]^T$. The transformation is then described by

$$\boldsymbol{T}_O = \begin{bmatrix} \boldsymbol{R}_z(\phi_z) \boldsymbol{R}_y(\phi_y) \boldsymbol{R}_x(\phi_x) & \boldsymbol{p}_k \\ 0 & 1 \end{bmatrix} \tag{4}$$

where $\boldsymbol{R}_a(\phi)$ denotes the rotation matrix corresponding to the rotation by ϕ about axis a using the right-hand rule. Note that the rotation matrices about the elementary axis only depend on $\boldsymbol{\theta}_R$ through arguments to trigonometric functions. Due to the periodic nature of those functions we obtain a desired unrestricted parameterization. The variable transform $\boldsymbol{T}_q(q_i)$ describes the joint constraint and joint configuration. For all joints we assume that the variable link axis is aligned with the z-axis. Hence, the transformation matrix and joint motion vector for revolute joints ($\boldsymbol{T}_{q_r}, \boldsymbol{s}_r$) and prismatic joints ($\boldsymbol{T}_{q_p}, \boldsymbol{s}_p$) is described by

$$\boldsymbol{T}_{q_r} = \begin{bmatrix} \boldsymbol{R}_z(q) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix}, \qquad \boldsymbol{T}_{q_p} = \begin{bmatrix} \boldsymbol{0} & q\,\boldsymbol{e}_z \\ \boldsymbol{0} & 1 \end{bmatrix},$$

$$\boldsymbol{s}_r = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{e}_z \end{bmatrix}, \qquad \boldsymbol{s}_p = \begin{bmatrix} \boldsymbol{e}_z & \boldsymbol{0} \end{bmatrix},$$

with the z axis unit vector $\boldsymbol{e}_z$. Technically, one can also use fixed joints with $\boldsymbol{T}_q{=}\mathbf{I}$ but this simply yields an overparameterized model and is not necessary for describing the system dynamics. The complete kinematics parameters of a link are summarized as $\boldsymbol{\theta}_K{=}\{\boldsymbol{\theta}_R, \boldsymbol{p}_k\}$

Inertias

For physical correctness, the diagonal rotational inertia $\boldsymbol{J}_p{=}\text{diag}([J_x, J_y, J_z])$ at the body's CoM and around principal axes must be positive definite and need to conform with the triangle inequalities (Traversaro et al., 2016), i.e.,

$$J_x \leq J_y + J_z, \quad J_y \leq J_x + J_z, \quad J_z \leq J_x + J_y \ .$$

To allow an unbounded parameterization of the inertia matrix, we introduce the parameter vector $\boldsymbol{\theta}_L{=}[\theta_{\sqrt{L_1}}, \theta_{\sqrt{L_2}}, \theta_{\sqrt{L_3}}]^\intercal$, where L_i represents the central second moments of mass of the density describing the mass distribution of the rigid body with respect to a principal axis frame. Then rotational inertia is described by

$$\boldsymbol{J}_p = \text{diag}(\theta^2_{\sqrt{L_2}} + \theta^2_{\sqrt{L_3}},\ \theta^2_{\sqrt{L_1}} + \theta^2_{\sqrt{L_3}},\ \theta^2_{\sqrt{L_1}} + \theta^2_{\sqrt{L_2}}).$$

The rotational inertia is then mapped to the link coordinate frame using the parallel axis theorem described by

$$\boldsymbol{J} = \boldsymbol{R}_J \boldsymbol{J}_p \boldsymbol{R}_J^\intercal + m[\boldsymbol{p}_m][\boldsymbol{p}_m] \tag{5}$$

with the link mass m and the translation $\boldsymbol{p}_m$ from the coordinate from to the CoM. The fixed affine transformation uses the same parameterization as described in 6. The mass of the rigid body is parameterized by $\theta_{\sqrt{m}}$ where $m{=}\theta^2_{\sqrt{m}}$. Given the dynamics parameters $\boldsymbol{\theta}_I{=}\{\boldsymbol{\theta}_L, \theta_{\sqrt{m}}, \boldsymbol{\theta}_J, \boldsymbol{p}_m\}$ for each link, the inertia in the desired frame using as well as generalized inertia $\bar{\boldsymbol{M}}$ can be computed.