Rapid Mismatch Estimation via Neural Network Informed Variational Inference

Mateusz Jaszczuk **Nadia Figueroa**
GRASP Lab, University of Pennsylvania
{jaszczuk, nadiafig}@seas.upenn.edu

Abstract: With robots increasingly operating in human-centric environments, ensuring soft and safe physical interactions, whether with humans, surroundings, or other machines, is essential. While compliant hardware can facilitate such interactions, this work focuses on impedance controllers that allow torque-controlled robots to safely and passively respond to contact while accurately executing tasks. From inverse dynamics to quadratic programming-based controllers, the effectiveness of these methods relies on accurate dynamics models of the robot and the object it manipulates. Any model mismatch results in task failures and unsafe behaviors. Thus, we introduce Rapid Mismatch Estimation (RME), an adaptive, controller-agnostic, probabilistic framework that estimates end-effector dynamics mismatches online, without relying on external force-torque sensors. From the robot's proprioceptive feedback, a Neural Network Model Mismatch Estimator generates a prior for a Variational Inference solver, which rapidly converges to the unknown parameters while quantifying uncertainty. With a real 7-DoF manipulator driven by a state-of-the-art passive impedance controller, RME adapts to sudden changes in mass and center of mass at the end-effector in ~ 400 ms, in static and dynamic settings. We demonstrate RME in a collaborative scenario where a human attaches an unknown basket to the robot's end-effector and dynamically adds/removes heavy items, showcasing fast and safe adaptation to changing dynamics during physical interaction without any external sensory system.

Project Website: https://mateusz-jaszczuk.github.io/rme/

Keywords: Passive Impedance Control, Learning Residual Inverse Dynamics, Model Mismatch Estimation

1 Introduction

Compliant control is fundamental to the development of adaptive and collaborative robots designed to physically interact with uncertain environments, objects of unknown dynamics, and humans with unpredictable intentions and capabilities. Many commercial robots advertise *compliant capabilities*, typically implemented through basic impedance or admittance control laws, rarely possessing the ability of being *passive* to physical perturbations while precise in task execution – an essential property for fluid physical interaction [1, 2]. Humans naturally achieve this balance by leveraging our redundant degrees-of-freedom [3] and mechanical compliance in the neuro-musculoskeletal system [4]. Replicating this behavior in robots requires control strategies that enforce a passive relationship between external forces and motion (i.e., velocity), guaranteeing stability in both free motion and during physical contact [5]. To this end, passivity-based impedance controllers have been extensively developed, guided by energy exchange techniques [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]; demonstrating success in a range of applications including grasping, manipulation, and robotic surgery. A major caveat of these techniques is that convergence and passivity guarantees depend heavily on accurate dynamics models of the robot and the environment it is interacting with. This assumption often breaks down in practice, especially when interacting with *unknown environments*, such as manipulating objects with uncertain mass and inertia. In such cases, gain tuning or adding a

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

PID layer is commonly used to mitigate model mismatch. However, these ad-hoc strategies compromise the passivity of the closed-loop system, resulting in a loss of compliance and safety. Humans, in contrast, exhibit a remarkable ability to interact with objects of unknown dynamics and uncertain environments. For example, when lifting a box with unknown contents, we rely on an initial internal estimate of its inertial properties, and upon sensing the actual load through proprioceptive feedback, we rapidly adjust muscle activation and posture to maintain stable manipulation within our physical capabilities [18, 19]. We perform this rapid adjustment even when we are pushed or mistakenly bump into a wall while carrying the box. This form of rapid adaptation is key to safe and efficient physical interaction. Hence, in this work, we seek to enable robots with the ability to estimate and adapt to dynamic model mismatches, in real-time, which would allow them to maintain passivity while continuing to perform their tasks – just like humans do. In particular, we envision scenarios in which a robot lifts a box of unknown mass, whether empty or filled with heavy books or dynamically being changed by a human, and instantaneously estimates the resulting model mismatch, enabling stable and safe manipulation while remaining passive to external physical contacts.

Related Work The core problem we address in this work is model mismatch, an issue that plagues model-based control. Prior works have addressed this problem, often in the context of exploration and reinforcement learning (RL) frameworks that aim to capture system dynamics accurately. For example, Sun et al. [20] proposed an online learning strategy to estimate the residual dynamics model in legged locomotion based on predictive errors. The episodic learning approach proposed by Khadivar et al. [21] allows for robust learning of Inverse Dynamics (ID), accounting for multiple mismatch factors (internal friction, actuators' nonlinearities, model imperfections, etc.). Similarly, Nguyen-Tuong and Peters [22] used Gaussian Process to improve learning Inverse Dynamics. Koryakovskiy et al. [23] proposed to compensate for model mismatch via a RL policy, learning a compensatory signal by comparing the robot's predicted and observed state transitions. Likewise, Gao et al. [24] proposed improving control policies via quantification of predictive error. Alternatively, Srour et al. [25] proposed trajectory-optimization for manipulation under uncertain payloads. While the aforementioned approaches demonstrate robust performance, they are often sensitive to feedback measurement noise, requiring longer estimation times and thus compromising real-time performance. Moreover, predictive error approaches may misattribute errors caused by infeasible control commands, e.g., due to unmodeled actuator limits or linearization errors, and often rely on conservative assumptions such as smooth dynamic changes starting from zero mismatch [20].

Contribution We introduce Rapid Mismatch Estimation (RME), a probabilistic, controller and motion-planner agnostic framework for real-time quantification of dynamic model mismatch during interaction with unknown objects. RME leverages a neural network (NN) architecture trained in simulation to estimate an initial model mismatch, which is then refined with online variational inference to rapidly infer mismatches in mass and center-of-mass (CoM) at the manipulator's end-effector in $\sim$400 ms. Thus, RME provides online compensation for unknown dynamics and allows the robot to follow a desired motion policy while maintaining the passivity of the system – just like humans do. To the best of our knowledge, no existing method provides a real-time estimation framework that can adapt to abrupt changes in model dynamics arising from interaction with unknown environments. The most related works to ours are by Zhang et al. [26], presenting a system-identification framework, allowing inertial parameters estimation of an unknown payload, however, requiring a 10s-long system identification/calibration action before trajectory execution, and by Jin et al. [27], which introduced an effective exploration strategy for center-of-mass (CoM) estimation; however, it does not operate in real-time or during continuous task execution and depends on force/torque (FT) sensing, which we avoid as it decreases robot's dexterity, reduces allowable payload, and highly increases assembly cost [28].

2 Problem Formulation

Let the following equation define the rigid-body dynamics of an n-DoF manipulator arm [29],

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau_c + \tau_{\text{ext}} \tag{1}$$

where $\ddot{q}, \dot{q}, q \in \mathbb{R}^n$, represent joint accelerations, velocities, and positions, respectively. $M(q) \in \mathbb{R}^{n\times n}$, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$, and $G(q) \in \mathbb{R}^n$ represent the mass/inertia matrix, Coriolis matrix, and

Gravity vector, respectively. $\tau_c \in \mathbb{R}^n$ and $\tau_{\text{ext}} = J(q)^\top F_{\text{ext}} \in \mathbb{R}^n$ represent control and external torques at each joint, respectively, with $J(q) \in \mathbb{R}^{6\times n}$ being the manipulator Jacobian.

Assumption 1. *For robot dynamics (1) we can assume that $M(q)$ and $C(q,\dot{q})$ are uniformly bounded for all q and $C(q,\dot{q})$ linear in $\dot{q}$. Further, we assume $C(q,\dot{q})$ fulfills the following:*

$$\dot{M}(q) = C(q,\dot{q}) + C(q,\dot{q})^T \Rightarrow \dot{M}(q) - 2C(q,\dot{q}) \quad \textit{skew-symmetric} \tag{2}$$

which yields the well-known skew-symmetry condition for revolute joint robots [29].

Assumption 2. *The control torque τ_c in (1) is an impedance controller that renders a passive closed-loop behavior wrt. the environment; i.e., a passive mapping of $(\tau_{ext} \to \dot{q})$ or $(F_{ext} \to \dot{x})$ via,*

$$\dot{S}(q,\dot{q}) \leq \dot{q}^\top \tau_{ext} \quad \textit{or} \quad \dot{S}(x,\dot{x}) \leq \dot{x}^\top F_{ext} \tag{3}$$

with S being the energy storage function including the kinetic energy of the robot (1) and the elastic potential energy from the controller. $\dot{x}, F_{ext} \in \mathbb{R}^6$ is the task-space velocity and external wrench.

Problem Any passive impedance controller τ_c is generally designed so that (2) appears in $\dot{S}$ such that (3) can be guaranteed. This holds for controllers with feedforward/ pre-compensation terms, feedback linearization, inverse dynamics, with or without inertia shaping $M(q)$ [30, 31, 17, 13, 32]. However, any interaction attempt with a heavy unknown object will result in a dynamics model mismatch. More specifically, (1) now includes an additional external torque due to model mismatch,

$$\underbrace{M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)}_{\tau_{\text{nom}}\text{: Nominal Dynamics}} = \tau_c + \tau_{\text{ext}} + \underbrace{J(q)^\top \begin{bmatrix} F_m \\ r_{\text{CoM}\times F_m} \end{bmatrix}}_{\tau_{\text{mm}}\text{: Model Mismatch}} \tag{4}$$

As shown, we assume that τ_{mm} is caused by changing the mass properties of the end-effector (EE), modeled as a point mass added to its current inertia tensor with $F_{\text{m}} = [0, 0, mg]$ being the gravitational force with $g = -9.81m/s^2$ due to unknown mass m and $r_{\text{CoM}} = [r_x, r_y, r_z]$ being the unknown center-of-mass (CoM) relative to the EE. This is a valid assumption as manipulators mostly interact with the environment via the EE. Any other contact can be considered a collision or human-robot interaction, not a dynamics mismatch. Thus, if τ_{mm} is not compensated in τ_c this may lead to instabilities, poor tracking performance and cause unsafe behaviors for the robot and the human.

Goal Thus, we aim to estimate the mismatch parameters $\theta = \{m, r_x, r_y, r_z\}$ online, and explicitly compensate for any model mismatch in real-time by augmenting a passive impedance controller as,

$$\hat{\tau}_c = \tau_c - J(q)^\top \begin{bmatrix} \hat{F}_m(\theta) \\ \hat{r}_{\text{CoM}}(\theta) \times \hat{F}_m(\theta) \end{bmatrix} \quad \text{with} \quad \left\| \begin{bmatrix} \hat{F}_m(\theta) - F_m \\ \hat{r}_{\text{CoM}}(\theta) - r_{\text{CoM}} \end{bmatrix} \right\|_2^2 \to 0 \tag{5}$$

with τ_c being a passive impedance controller designed with the nominal dynamics (1). Our choice of controller is described in Section 3. As the true mismatch parameters F_m, r_{CoM} are unknown, we seek a robust probabilistic estimate of θ by formulating an online Bayesian inference problem, as described in Section 4. Further, we aim to estimate θ using only the robot's proprioceptive feedback (e.g., applied joint torque estimates), without a need for an external FT sensor at the end-effector.

3 Preliminaries: Constrained Passive Interaction Controller [13, 32]

Our RME framework is designed to be control-agnostic, and can be used with any passive impedance controller or even on RL trained control policies with passivity properties [33, 34]. In this work, however, we choose a recent extension of the dynamical systems (DS) based passive interaction controller [13] that can guarantee passivity without any inertia shaping, training or FT sensing. In this control paradigm, the desired motion of the robot's end-effector is driven by a DS [35],

$$\dot{x} = f(x) \quad \text{s.t.} \quad \dot{\mathcal{V}}(x) = \nabla\mathcal{V}(x)^\top f(x) \leq 0 \tag{6}$$

which represents a motion policy that is trained (or defined) to be globally asymptotically stable (g.a.s.) wrt. an attractor $x^* \in \mathbb{R}^3 \times SO(3)$ or a task-space trajectory $\{x_t\}_{t=1}^T$ like a limit cycle. The asymptotic stability of $f(x) : \mathbb{R}^3 \times SO(3) \to \mathbb{R}^6$ is enforced from a corresponding Lyapunov function $\mathcal{V}(x) : \mathbb{R}^3 \times SO(3) \to \mathbb{R}$ which can also be learned. For a comprehensive review of

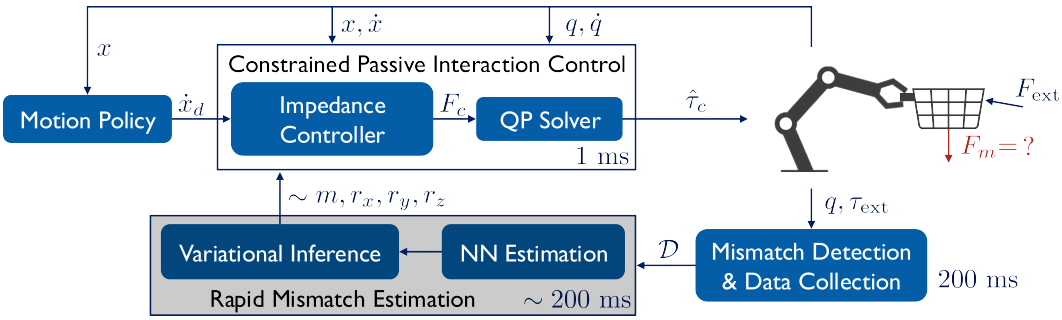

Figure 1: **Rapid Mismatch Estimation (RME) Framework**. The gray block denotes our novel contribution. Note that RME is controller agnostic and can be implemented with any impedance controller. In this work, we choose the constrained passive interaction controller [32].

imitation learning (IL) of DS motion policies, see [35, 36]. To track (6) while guaranteeing closed-loop passivity, we use the following velocity-based passive impedance control law [13],

$$F_c(x) = G_x(q) - D(x)\big(\dot{x} - f(x)\big) \quad \text{with} \quad D(x) \succ 0 \tag{7}$$

with task-space gravity vector $G_x(q) \in \mathbb{R}^6$, damping matrix $D(x) \in \mathbb{R}^{6\times 6}$, and desired velocity from $f(x)$. By aligning $D(x)$ to the direction of $f(x)$ we can generate kinetic energy in the desired direction of motion while dissipating energy in the tangential direction. While this ensures passivity wrt. F_{ext} via (3), it does not ensure that a feasible $\tau_c = J(q)^\top F_c$ satisfies any kinematic constraints, such as joint limits, self-collision, external collisions, and singularities. To address this, Zhang et al. [32] proposed to constrain (7) with a Quadratic Programming (QP) based safety filter, that optimizes τ_c to track $F_c(x)$, subject to the robot dynamics (1) and joint kinematic constraints formulated as exponential control barrier function (CBF) with relative degree-2 [37]. $h_i(q) \geq 0$ denote the invariant sets corresponding to different kinematic constraints and $\mathcal{K}_i = [k_1\ k_2]$ feedback gain vectors that render forward invariance [32]. $J(q)^{-\top}$ is the pseudo-inverse of $J(q)^\top$. We use (8) as the nominal passive impedance controller generating τ_c in (5).

$$\begin{aligned} \min_{\tau_c} \quad & \left\| J(q)^{-\top}\tau_c - F_c(x) \right\|_2^2 \\ \text{s.t} \quad & M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau_c + \tau_{\text{ext}} \\ & \ddot{h}_i(q) \geq -\mathcal{K}_i[h_i(q), \dot{h}_i(q)]^\top \quad \forall i = 1, \dots \end{aligned} \tag{8}$$

Remark 1. *Both the unconstrained (7) and constrained (8) passive interaction controllers are proven to track $f(x)$ and be passive wrt. input-output pair $(F_{ext}, \dot{x})$ using the energy storage function $\mathcal{S} = \frac{1}{2}\dot{x}^\top M_x \dot{x} + \lambda_1 \mathcal{V}(x)$ where M_x is the task-space inertia matrix, λ_1 is the first eigenvalue of $D(x)$ from (7) and $\mathcal{V}(x)$ the Lyapunov function used to shape $f(x)$. These guarantees rely on the satisfaction of condition (2). Thus, neglecting model mismatches not only degrades the tracking performance of $f(x)$ but also compromises the passivity of the closed-loop system.*

4 Rapid Mismatch Estimation

In order to robustly estimate the mismatch parameters $\theta = \{m, r_x, r_y, r_z\}$ while minimizing mismatch errors $\Delta F_m = F_m - \hat{F}_m \to \mathbf{0}$ and $\Delta r_{\text{CoM}} = \|r_{\text{CoM}} - \hat{r}_{\text{CoM}}\|_2^2 \to 0$ we turn to Bayesian inference. Rather than estimating a discrete point estimate of θ, we propose to estimate a posterior distribution of mismatch parameters $p(\theta \mid \mathcal{D})$ given external torque measurements, τ_{ext}, at particular joint configurations q while quantifying prediction uncertainty as follows,

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}, \quad \mathcal{D} = \{\tau_{\text{ext},i} \in \mathbb{R}^n \mid i = 1, \dots, N\}. \tag{9}$$

Mismatch Parameter Estimation as a Bayesian Inference Problem Given a dataset $\mathcal{D}$ of τ_{ext} measurements collected online over a short time-window of length N, we seek to estimate the posterior distribution $p(\theta \mid \mathcal{D})$. We define the likelihood function with the Gaussian Distribution as the probability of observing τ_{ext} on each joint given the mismatched parameters θ, computed using the manipulator arm's Inverse Dynamics (ID) model $f_{\text{ID}}(\cdot)$ with the likelihood noise $\sigma^2_{\text{likelihood}}$ based on the amplitude of noise observed at each joint in the dataset $\mathcal{D}$.

$$p(\mathcal{D}|\theta) = \mathcal{N}(f_{\text{ID}}\left(q, \theta\right), \text{diag}(\sigma^2_{\text{likelihood}})), \quad p(\theta) = \mathcal{N}\left(f_{\text{NN}}(\mathcal{D}), \text{diag}(\sigma^2_{\text{prior}})\right) \tag{10}$$

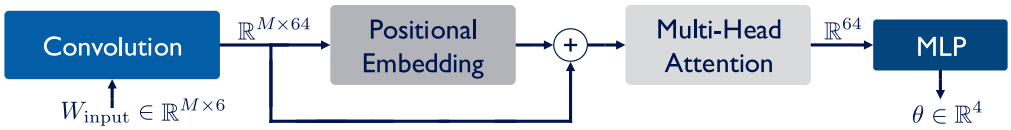

Figure 2: **RME Neural Network Architecture**. In the network, we input a sequence of pseudo-wrenches of dimension $\mathbb{R}^{M\times 64}$, apply a convolution layer, positional embedding, and multi-head attention. Further, we mean pool over the attention scores and apply a sequential Multilayer Perceptron, which performs a final regression to mismatch parameters θ.

This inference problem does not have a unique solution in the short time window, as multiple mismatch parameters will result in similar τ_{ext} observations, making CoM estimation particularly sensitive to the choice of the prior $p(\theta)$. Therefore, we propose to model $p(\theta)$ using the prediction of a NN $f_{\text{NN}}(\cdot)$ as the mean for the Gaussian Distribution, as described in Section 4.1. Notice that, even though we have (10) the true posterior distribution $p(\theta \mid \mathcal{D})$ is still intractable, as the manipulator's dynamics exhibit non-linear dependencies on mismatch parameters θ. We address this by employing a Variational Inference (VI) framework, under which we aim to construct a mean-field approximation to the true posterior [38, 39], guided by the NN $f_{\text{NN}}(\cdot)$ initial solutions.

RME Framework As shown in Figure 1, our framework begins with mismatch detection algorithm that constantly monitors $\|\tau_{\text{ext}}\|_2^2$, distinguishing dynamics mismatch from human-generated perturbations by checking for rapid increases followed by short-term stabilization at a high value, characteristic for rapidly added loads, allowing the RME to run when only the dynamics mismatch is detected. Implementation details of our mismatch detector are provided in Appendix A. Once a mismatch is detected, we record the dataset $\mathcal{D}$ of τ_{ext} measurements over a 200 ms time window while the robot is being controlled by τ_c. $\mathcal{D}$ is then fed into the initial NN mismatch estimator, $f_{\text{NN}}(\cdot)$, which will predict the mean value for the prior as defined in (10). We then run the VI framework to approximate the posterior $p(\theta \mid \mathcal{D})$ and rapidly converge to the parameter estimate θ.

4.1 Neural Network Model Mismatch Estimator Architecture

In this work, we chose to construct and train a NN to guide VI by using it as a meaningful mean prior $p(\theta)$. The NN architecture, depicted in Figure 2, was designed to analyze the M-dimensional sequence of tokens, representing time-series of external pseudo-wrenches, $\hat{W}_{\text{ext}} \in \mathbb{R}^6$, applied to the EE, computed using the external torque measurements at given joint configurations, with damped pseudo-inverse of the Jacobian matrix, where λ is a damping coefficient, as below.

$$\hat{W}_{\text{ext}} = \left(J(q)J(q)^\top + \lambda I\right) J(q)\tau_{\text{ext}},\ W_{\text{input}} = \left\{\hat{W}_{\text{ext,i}} \in \mathbb{R}^6 \mid i = 1, \dots, M\right\} \tag{11}$$

The sequence of inputs to the NN is then $W_{\text{Input}} \in \mathbb{R}^{M\times 6}$, where M inputs are uniformly sampled from N collected data points. To solve the regression problem of the mismatch parameters, maintaining translation-invariance relative to the input sequence,

$$\{q_i \in \mathbb{R}^n,\ \tau_{\text{ext},i} \in \mathbb{R}^n \mid i = 1, \dots, M\} \longrightarrow W_{\text{Inputs}} \xrightarrow{\text{NN}} \{m, r_x, r_y, r_z\} \tag{12}$$

we design an architecture that keeps the forward pass computationally efficient. First, we apply a 1D convolution [40] with kernel size 5 and output dimension 64, to capture local patterns in the data, enhancing translation-invariance. Then, we apply position embedding and multi-head attention [41] with 8 attention heads, to capture global dependencies between all tokens and situate them in the sequence. Further, we apply the mean polling of attention outputs over the sequence length to obtain a vector of dimension $\mathbb{R}^{64}$ that we pass to a sequence of three MLP blocks, each consisting of a linear layer $(64 \to 256)$, ReLU activation function, and a linear layer $(256 \to 64)$. Then, we apply a final linear layer $(64 \to 4)$ to estimate the mean mismatch parameters θ for $p(\theta)$.

We train this NN mismatch estimator in simulation, on a dataset generated by 350 simulations of a manipulator's dynamics experiencing different mismatch parameters. During training, we use Mean-Squared Error loss objective, and apply dropout, with dropout rate $d = 0.1$ after each of the MLP blocks, to prevent network overfitting [42]. We also employ data augmentation to train the NN to be robust to noise and can transfer sim-to-real. Details on the architecture and training are in Appendix G. We provide ablations results on NN impact on the inference in Section 5.

4.2 Variational Inference for Mismatch Estimation Convergence

During runtime, the $f_{\text{NN}}(\cdot)$ trained in the previous sections is used to construct $p(\theta)$ as in (10) to start the VI framework. Since the true posterior distribution $p(\theta \mid \mathcal{D})$ is intractable, we construct a mean-field approximation to the true posterior $p(\theta \mid \mathcal{D}) \approx q_\phi(\theta \mid \mathcal{D})$, parameterized by the following variational parameters ϕ [38, 39],

$$\phi = (\underbrace{\mu_m,\ \mu_x,\ \mu_y,\ \mu_z}_{\mu_\phi},\ \underbrace{\sigma_m^2,\ \sigma_x^2,\ \sigma_y^2,\ \sigma_z^2}_{\sigma_\phi^2}) \tag{13}$$

We formulate an optimization objective for computing the variational parameters ϕ that yield an effective approximation of the true posterior by minimizing the Kullback-Leibler divergence.

$$q_\phi^*(\theta \mid \mathcal{D}) = \arg\min_{q_\phi} \text{KL}\left[q_\phi(\theta \mid \mathcal{D}) \parallel p(\theta \mid \mathcal{D})\right] \tag{14}$$

Under the mean-field approximation, (14) can lead to a tractable optimization objective, by the maximization of the Evidence Lower Bound (ELBO), which is equivalent to the minimization of KL-divergence, that we optimize via stochastic gradient descent [39, 43],

$$\text{ELBO}(\phi) = \mathcal{H}[q_\phi(\theta|\mathcal{D})] + \mathbb{E}_{\theta\sim q_\phi(\theta|\mathcal{D})}[\log p(\mathcal{D},\theta)] \tag{15}$$

with $\mathcal{H}[q_\phi(\theta|\mathcal{D})]$ being the entropy of $q_\phi(\cdot)$.

$$\mathcal{L}(\phi) = -\text{ELBO}(\phi) = -\mathbb{E}_{\theta\sim q_\phi(\theta|\mathcal{D})}[\log p(\mathcal{D}|\theta) + \log p(\theta) - \log q_\phi(\theta|\mathcal{D})] \tag{16}$$

$$\phi^* = \arg\min_{\phi}\ \mathcal{L}(\phi) \tag{17}$$

The expectation of the ELBO loss can be effectively computed with Monte Carlo estimation via the reparameterization trick [39],

$$\theta_i \overset{iid}{\sim} q_\phi(\theta \mid \mathcal{D}), \quad \theta_i = \mu_\phi + \sigma_\phi \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_4) \tag{18}$$

which allows us to evaluate gradients of (17) with automatic differentiation [43]. To estimate ϕ, we use the Adam optimizer [44] with learning rate $\eta = 0.025$ and gradient clipping to prevent gradient explosion in early optimization stages. We run the optimizer until variational parameters stabilize at a given threshold, allowing us to return the optimization result as soon as it converges without performing unnecessary training iterations, accelerating the estimation process. Note that inaccurate NN prediction will not impact convergence, as it only affects the prior $p(\theta)$.

4.3 Preservation of Passivity and Stability of the Closed-Loop System

One of the main motivating factors for developing RME was to achieve a fast enough estimation of θ such that we could preserve the task convergence and passivity of the closed-loop system provided by the passive interactive controllers introduced in Section 3. Note that, when our estimated mismatch parameters are near perfect, $\Delta F_m \approx \mathbf{0}$ and $\Delta r_{\text{CoM}} \approx 0$, and we control the robot dynamics (4) with our proposed augmented controller $\hat{\tau}_c$ as defined in (5) then the mismatch and compensation terms cancel out and we recover the original dynamics (1) whose closed-loop behavior is stable and passive under a control input τ_c defined by (7) or (8) as proven in [13, 35] and [32], respectively.

Proposition 4.1. *Let a robotic manipulator with dynamics (4) be controlled by an augmented passive impedance control law defined in (5) with τ_c computed by (8). Given an imperfect mismatch parameter estimation $\Delta F_m \neq \mathbf{0}$ and $\Delta r_{CoM} \neq 0$ the system will be locally asymptotically stable as unwanted equilibria may arise when $\|\Delta F_{mm} - F_c(x)\| = 0$, with ΔF_{mm} denoting the remaining unknown model mismatch in the EE expressed as a task-space wrench. Yet, the closed-loop system behavior remains passive wrt. input-output port $(F_{ext} + \Delta F_{mm}, \dot{x})$.*

Proof: *Provided in Appendix H.* ■

5 Evaluation

We evaluate our model with physical robot experiments using a 7-DoF Franka Emika torque-controlled manipulator arm with embedded torque sensors. We empirically tuned the standard deviation for the VI prior as $\sigma_{\text{prior}} = [0.5 \quad 0.02 \quad 0.02 \quad 0.05]$. The QP optimization for (8) was implemented using CVXGEN [45]; the controller and RME were evaluated on a workstation with 11th Gen Intel® Core™ i7-11700K @ 3.60 GHz CPU. No GPU acceleration is required.

5.1 Ablations on NN Effect and Hyper-Parameters for VI

We analyze the effect of NN on VI prediction accuracy by comparing inference results achieved with $p(\theta)$ constructed using $\mu_{\text{prior}} = f_{\text{NN}}(\cdot)$ and $\mu_{\text{prior}} = 0$. We ran the RME algorithm on 100 simulation-generated datasets, representing the manipulator's dynamic behavior under different mismatch parameters θ. As shown in Table 1, NN guidance significantly reduces the Mean Squared Error in CoM estimation. Detailed predictions comparison is shown in Appendix C. We further analyze the impact of the estimation interval length on inference accuracy in Appendix F, showing that 200 ms is an optimal length for fast and robust estimation, making the solver more robust to measurement noise while allowing not smooth but abrupt changes, causing noticeable and instantaneous disruption in the trajectory. Since we can solve the estimation problem in $\sim$ 400 ms, we can run this framework sequentially, ensuring that if the estimation was biased by the external factors, like undetected perturbation, it can be corrected by the subsequent prediction.

μ_{prior}	MSE m (kg^2)	MSE r_x (m^2)	MSE r_y (m^2)	MSE r_z (m^2)
0	0.319×10^{-3}	0.594×10^{-3}	0.880×10^{-3}	6.513×10^{-3}
$f_{\text{NN}}(\cdot)$	0.229×10^{-3}	0.366×10^{-3}	0.629×10^{-3}	2.560×10^{-3}

Table 1: Mean Squared Errors for RME estimation of mismatch parameters θ with Gaussian Prior $p(\theta)$ constructed by $\mu_{\text{prior}} = f_{\text{NN}}(\cdot)$ and $\mu_{\text{prior}} = 0$.

5.2 Estimation Results from Static Experiments

To evaluate RME, we extensively tested the model with multiple static experiments, where the physical manipulator, subject to sudden changes in the dynamics model resulting from adding unknown mass to the end-effector, aimed to maintain the target equilibrium position and orientation using the constrained passive impedance controller (CPIC) (8). RME provided rapid and accurate estimation of mismatch parameters θ, as shown in Figure 3, and further in Appendix E, with an average estimation time of 226 ms. As shown in Figure 4, adaptation allowed the robot to quickly converge to the equilibrium position from before applying the mismatch - note that even if the tracking error does not reduce to zero, due to disturbance or under-prediction, RME always provides valid compensation, resulting in trajectory correction.

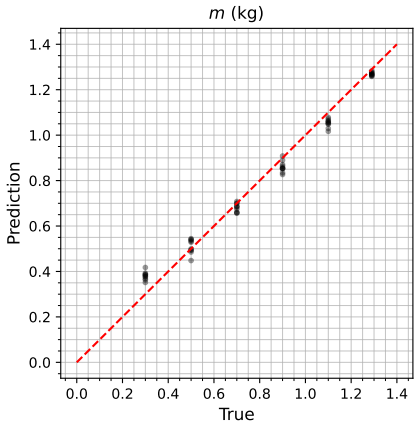

Figure 3: Parity plot for mass predictions over 60 independent experiments. Results show consistent and stable mass estimation using RME.

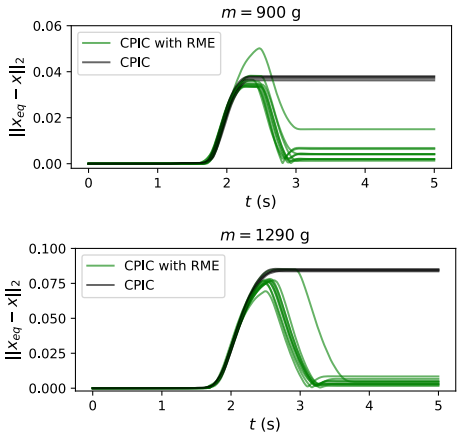

Figure 4: Franka manipulator arm tracking performance subject to mismatch in the dynamics model applied at $t \approx 1.75$ s.

Under this setup, we observed that when the CoM of applied mismatch is close to the z-axis of the end-effector, as in experiments with 1.1 kg and 1.29 kg, the CoM offset causes negligible twist of the end-effector, reducing its observability due to non-linear dependencies of Inverse Dynamics on mismatch parameters θ, shown in results in Appendix E. However, the robot still safely converges to the target state, and CoM can be corrected in subsequent predictions if the twist increases due to a change in orientation.

5.3 Performance on Dynamic Experiments

Tracking a Stable Limit Cycle To test the model in the dynamic scenario, we designed a Dynamical System with a stable limit cycle in the y-z plane, where the manipulator aims to achieve target velocities from the motion planner. As shown in Figure 5, RME rapidly estimates mismatch parameters, allowing the robot to converge to the desired trajectory. Without adaptation, the manipulator reaches the spurious attractor, where the task force from the impedance controller is balanced by the gravitational force acting on the mass mismatch, preventing it from continuing on the given task.

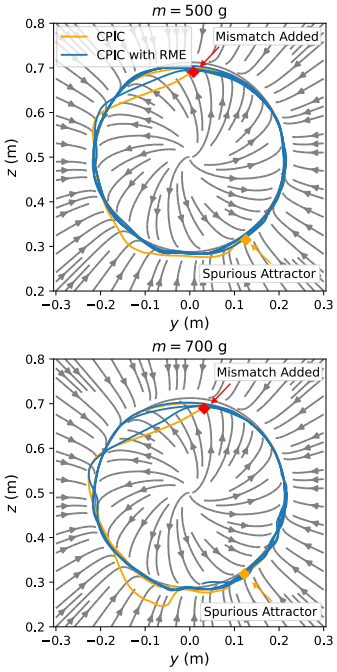

Figure 5: Manipulator tracking a DS with a stable limit cycle in the y-z plane subject to mismatch in the dynamics model.

Sequential Adaptation with Human-Robot Interactions In this experiment, we evaluate the model's capability to perform continuous adaptation while maintaining passivity with respect to human-generated perturbations. We attach the unknown basket to the end effector, sequentially add and remove two unknown objects, and finally, remove the basket, perturbing the robot between these actions. RME provided an accurate estimation of mismatch parameters θ, allowing the robot to maintain target position and orientation, while being passive to human-generated forces (even when the model confused human perturbations with model mismatch, it allowed perturbation, and instantaneously corrected itself after being released). Figure 6 shows the sequential adaptation's experimental setup and mass estimation history. Additional pHRI experiments are provided in Appendix D.

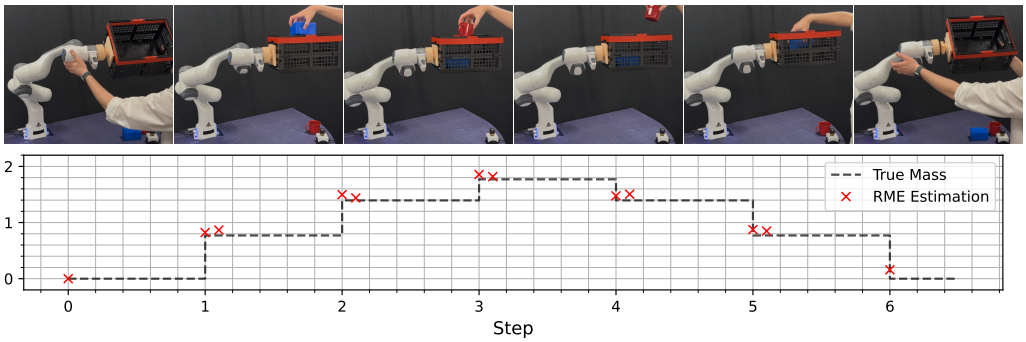

Figure 6: Manipulator adaptation to sequential changes in the dynamics model, while subject to human-generated perturbations. Each step represents modifying the dynamics model of the end effector; RME prediction between two steps denotes immediate correction of the previous θ estimate.

6 Conclusion

We propose a novel adaptation framework capable of estimating mismatch in the dynamics model of the impedance-controlled manipulator in $\sim$ 400 ms. We evaluated our work in a series of experiments, where we abruptly changed the dynamics model of the robot, and showed that our model can provide accurate estimation and compensation, allowing the robot to safely complete the task and stay passive wrt. to human-generated perturbations. In our future work, we plan to utilize uncertainty quantification to develop a robust motion policy that will allow precise estimation of the CoM, addressing the observability issue under certain configurations. We also plan to utilize uncertainty quantification and Neural Network formulation shown in Section 4.1 to develop a more complex mismatch detection condition, allowing more robust mismatch detection. Further, we plan to extend the RME to model more complex dynamic interactions with unknown objects and explicitly account for unmodeled actuator dynamics, improving the framework's applicability and accuracy.

7 Limitations

To enable safe human-robot interactions, it is crucial to disambiguate between human-robot interaction and mismatch in the manipulators' dynamic model, ensuring that the framework will not compensate for human-generated perturbations. We notice that in selected scenarios, our simple activation condition might be unable to distinguish these two factors in such a short time window. Therefore, in our future work, we plan to develop and implement more complex mismatch detection condition, allowing more robust and safe human-robot interaction.

During the evaluation, we observed that when the CoM of applied mismatch is close to the z-axis of the end-effector, its observability is reduced due to non-linear dependencies of Inverse Dynamics on mismatch parameters θ. Although the robot could converge to the desired trajectory in all experiments, and CoM prediction can be corrected in the subsequent prediction, in our future work, we plan to develop an exploration strategy that utilizes prediction uncertainty to allow efficient and accurate CoM estimation. In addition, we observed that when the end-effector is experiencing rapid accelerations along the global z-axis, such as in the stable limit cycle experiment, the mismatch prediction might be biased. Further, unmodeled actuator dynamics, such as joint friction, can be reflected in the τ_{ext} measurements, also biasing the RME predictions. Although in both cases, predictions can be corrected in subsequent estimations, we plan to address these limitations in our future work, further improving the framework's accuracy.

Finally, our model assumes that the detected object is a point mass added to the current inertia tensor of the end-effector. This assumption holds when the manipulator is not rapidly accelerating, which is valid for Human-Robot interaction scenarios; however, inertia estimation can be incorporated into the framework to achieve a more accurate system representation.

Acknowledgments

We thank our anonymous reviewers, who provided thorough and fair feedback that improved the quality of our paper. This work was supported by the National Science Foundation (NSF) Foundational Research in Robotics (FRR) program under NSF CAREER Award Grant No. FRR-2443721.

References

[1] S. Stramigioli. Energy-aware robotics. In M. K. Camlibel, A. A. Julius, R. Pasumarthy, and J. M. Scherpen, editors, *Mathematical Control Theory I*, pages 37–50, Cham, 2015. Springer International Publishing. ISBN 978-3-319-20988-3.

[2] A. Billard. On the mechanical, cognitive and sociable facets of human compliance and their robotic counterparts. *Robotics and Autonomous Systems*, 88:157–164, 2017. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2016.08.030. URL https://www.sciencedirect.com/science/article/pii/S0921889016306303.

[3] R. Sharif Razavian, B. Ghannadi, and J. McPhee. On the relationship between muscle synergies and redundant degrees of freedom in musculoskeletal systems. *Frontiers in Computational Neuroscience*, Volume 13 - 2019, 2019. ISSN 1662-5188. doi:10.3389/fncom.2019.00023. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2019.00023.

[4] G. L. Gottlieb. Muscle compliance: implications for the control of movement. *Exerc Sport Sci Rev*, 24:1–34, 1996.

[5] J. Colgate. Robust control of dynamically interacting systems. *International Journal of Control*, 48(1):65–88, July 1988. ISSN 0020-7179. doi:10.1080/00207178808906161.

[6] N. Hogan. Impedance control: An approach to manipulation. In *1984 American Control Conference*, pages 304–313, 1984. doi:10.23919/ACC.1984.4788393.

[7] S. Stramigioli, C. Melchiorri, and S. Andreotti. A passivity-based control scheme for robotic grasping and manipulation. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 3, pages 2951–2956 vol.3, 1999. doi:10.1109/CDC.1999.831385.

[8] P. Li and R. Horowitz. Passive velocity field control of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 15(4):751–763, 1999. doi:10.1109/70.782030.

[9] Y. Kishi, Z. W. Luo, F. Asano, and S. Hosoe. Passive impedance control with time-varying impedance center. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694)*, volume 3, pages 1207–1212 vol.3, 2003. doi:10.1109/CIRA.2003.1222169.

[10] V. Duindam, S. Stramigioli, and J. Scherpen. Passive compensation of nonlinear robot dynamics. *IEEE Transactions on Robotics and Automation*, 20(3):480–488, 2004. doi:10.1109/TRA.2004.824693.

[11] A. Albu-Schäffer, C. Ott, and G. Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The International Journal of Robotics Research*, 26(1):23–39, 2007. doi:10.1177/0278364907073776.

[12] F. Ferraguti, C. Secchi, and C. Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *2013 IEEE International Conference on Robotics and Automation*, pages 4948–4953, 2013. doi:10.1109/ICRA.2013.6631284.

[13] K. Kronander and A. Billard. Passive interaction control with dynamical systems. *IEEE Robotics and Automation Letters*, 1(1):106–113, 2015.

[14] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi. An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31(5):1073–1088, 2015. doi:10.1109/TRO.2015.2455791.

[15] G. A. Folkertsma and S. Stramigioli. Energy in robotics. *Foundations and Trends® in Robotics*, 6(3):140–210, 2017. ISSN 1935-8253. doi:10.1561/2300000038. URL `http://dx.doi.org/10.1561/2300000038`.

[16] A. Dietrich, X. Wu, K. Bussmann, C. Ott, A. Albu-Schäffer, and S. Stramigioli. Passive hierarchical impedance control via energy tanks. *IEEE Robotics and Automation Letters*, 2 (2):522–529, 2017. doi:10.1109/LRA.2016.2645504.

[17] F. J. Abu-Dakka and M. Saveriano. Variable impedance control and learning—a review. *Frontiers in Robotics and AI*, 7, 2020. ISSN 2296-9144. doi:10.3389/frobt.2020.590681.

[18] B. L. Riemann and S. M. Lephart. The sensorimotor system, part II: The role of proprioception in motor control and functional joint stability. *Journal of Athletic Training*, 37(1):80–84, 2002.

[19] K. Oh, W. Z. Rymer, and J. Choi. The speed of adaptation is dependent on the load type during target reaching by intact human subjects. *Experimental Brain Research*, 239(10):3091–3104, Oct. 2021. ISSN 1432-1106.

[20] Y. Sun, W. L. Ubellacker, W.-L. Ma, X. Zhang, C. Wang, N. V. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. D. Ames. Online learning of unknown dynamics for model-based controllers in legged locomotion. *IEEE Robotics and Automation Letters*, 6(4):8442–8449, 2021. doi:10.1109/LRA.2021.3108510.

[21] F. Khadivar, K. Chatzilygeroudis, and A. Billard. Self-correcting quadratic programming-based robot control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(8): 5236–5247, 2023. doi:10.1109/TSMC.2023.3262954.

[22] D. Nguyen-Tuong and J. Peters. Using model knowledge for learning inverse dynamics. In *2010 IEEE International Conference on Robotics and Automation*, pages 2677–2682, 2010. doi:10.1109/ROBOT.2010.5509858.

[23] I. Koryakovskiy, M. Kudruss, H. Vallery, R. Babuška, and W. Caarls. Model-plant mismatch compensation using reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3): 2471–2477, 2018. doi:10.1109/LRA.2018.2800106.

[24] F. Gao, C. Yu, Y. Wang, and Y. Wu. Neural internal model control: Learning a robust control policy via predictive error feedback. *IEEE Robotics and Automation Letters*, 10(7):6848–6855, 2025. doi:10.1109/LRA.2025.3573169.

[25] A. Srour, A. Franchi, P. R. Giordano, and M. Cognetti. Experimental validation of sensitivity-aware trajectory planning for a redundant robotic manipulator under payload uncertainty. *IEEE Robotics and Automation Letters*, 10(2):1561–1568, 2025. doi:10.1109/LRA.2024.3519857.

[26] B. Zhang, Z. Zhou, and R. Vasudevan. Provably-safe, online system identification. In *Robotics: Science and Systems*, 2025.

[27] S. Jin, Y. Mo, and W. Yuan. Learning to double guess: An active perception approach for estimating the center of mass of arbitrary objects, 2025. URL https://arxiv.org/abs/2502.02663.

[28] A. Colomé, D. Pardo, G. Alenyà, and C. Torras. External force estimation during compliant robot manipulation. In *2013 IEEE International Conference on Robotics and Automation*, pages 3535–3540, 2013. doi:10.1109/ICRA.2013.6631072.

[29] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2020.

[30] A. Dietrich, X. Wu, K. Bussmann, M. Harder, M. Iskandar, J. Englsberger, C. Ott, and A. Albu-Schäffer. Practical consequences of inertia shaping for interaction and tracking in robot control. *Control Engineering Practice*, 114:104875, 2021. ISSN 0967-0661. doi:https://doi.org/10.1016/j.conengprac.2021.104875. URL https://www.sciencedirect.com/science/article/pii/S0967066121001520.

[31] K. Kronander and A. Billard. Stability considerations for variable impedance control. *IEEE Transactions on Robotics*, 32(5):1298–1305, 2016. doi:10.1109/TRO.2016.2593492.

[32] Z. Zhang, T. Li, and N. Figueroa. Constrained passive interaction control: Leveraging passivity and safety for robot manipulators. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13418–13424, 2024. doi:10.1109/ICRA57147.2024.10610232.

[33] D. Sacerdoti, F. Benzi, and C. Secchi. A reinforcement learning-based control strategy for robust interaction of robotic systems with uncertain environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5788–5794, 2024. doi:10.1109/ICRA57147.2024.10610082.

[34] R. Zanella, F. Califano, C. Secchi, and S. Stramigioli. Learning passive policies. *Springer Proceedings in Advanced Robotics*, 32 SPAR:338 – 343, 2024. doi:10.1007/978-3-031-76424-0_60. Cited by: 0.

[35] A. Billard, S. S. Mirrazavi Salehian, and N. Figueroa. *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach*. MIT Press, Cambridge, USA, 2022. URL https://mitpress.mit.edu/books/learning-adaptive-and-reactive-robot-control.

[36] Y. Hu, F. J. Abu-Dakka, F. Chen, X. Luo, Z. Li, A. Knoll, and W. Ding. Fusion dynamical systems with machine learning in imitation learning: A comprehensive overview. *Information Fusion*, 108:102379, 2024. ISSN 1566-2535. doi:https://doi.org/10.1016/j.inffus.2024.102379.

[37] Q. Nguyen and K. Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328. IEEE, 2016.

[38] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Apr. 2017. ISSN 1537-274X. doi:10.1080/01621459.2017.1285773.

[39] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[40] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[43] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference, 2016. URL https://arxiv.org/abs/1603.00788.

[44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[45] J. Mattingley and S. Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012. doi:10.1007/s11081-011-9176-9. URL https://doi.org/10.1007/s11081-011-9176-9.

[46] S. S. M. Salehian, N. Figueroa, and A. Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37(10):1205–1232, 2018. doi:10.1177/0278364918765952. URL https://doi.org/10.1177/0278364918765952.

Appendix

A Mismatch Detection Algorithm

To ensure that our framework runs only when mismatch is detected, we designed a simple mismatch detection algorithm that constantly monitors $\|\tau_{\text{ext}}\|_2^2$, checking for rapid increases or decrease exceeding activation threshold, which, depending on the experiment, we set between 0.7 and 1.1 Nm, followed by stabilization within 0.2 Nm threshold over 230 ms period, characteristic for applying mismatch to the end effector. Algorithm 1 shows the implementation of the mismatch detection condition. The algorithm analyzes $\|\tau_{\text{ext}}\|_2^2$ time-history over 500 ms of the most recent feedback, monitoring for rapid change of $\|\tau_{\text{ext}}\|_2^2$. Further, it evaluates stabilization condition by computing the mean $\|\tau_{\text{ext}}\|_2^2$ over short time windows within the stabilization interval and compares it with the mean $\|\tau_{\text{ext}}\|_2^2$ at the end of the interval, allowing detection of both adding and removing objects from the end effector. To improve the algorithm's robustness in pHRI, we also analyze measured pseudo-wrenches applied to the end-effector, verifying that the principal applied force acts in the global z-direction, and that forces in x and y directions do not exceed 2.5 N, ensuring that mismatch is caused only by the applied load, not human perturbation. Detection threshold, stabilization threshold, and stabilization interval were found empirically through $\|\tau_{\text{ext}}\|_2^2$ profiles analysis and can be tuned based on the task objective. This simple condition allowed us to detect a mismatch applied to the end-effector and run RME only when all detection conditions were met simultaneously. Further, the algorithm can detect when dynamics mismatch and human perturbations occur simultaneously, as these events exhibit different torque profiles, allowing the detection algorithm to distinguish the two. Hence, RME will be activated once perturbations are no longer present, preserving the controller's passivity. Figure 7 shows an example of algorithm execution. After detecting a mismatch, we proceed to data collection, which is followed by the RME framework estimation.

Algorithm 1 Mismatch Detection Algorithm

function ACTIVATIONCONDITION($\{\|\tau_{\text{ext}}\|_2^2, \hat{F}_{\text{ext}}\}_{i=1}^{N}$) ▷ N=500 steps

rapid change $\leftarrow \left| \|\tau_{\text{ext}}\|_2^2(500) - \|\tau_{\text{ext}}\|_2^2(1) \right| >$ activation threshold

stabilization checks $\leftarrow \left[\frac{1}{20} \sum_{i=431}^{450} \|\tau_{\text{ext}}\|_2^2(i),\ \frac{1}{20} \sum_{i=351}^{370} \|\tau_{\text{ext}}\|_2^2(i),\ \frac{1}{20} \sum_{i=271}^{290} \|\tau_{\text{ext}}\|_2^2(i) \right]$

end stabilization $\leftarrow \frac{1}{10} \sum_{i=491}^{500} \|\tau_{\text{ext}}\|_2^2(i)$

$\overline{F}_{\text{ext}} \leftarrow \left[\frac{1}{50} \sum_{i=451}^{500} \hat{F}_{\text{ext, x}},\ \frac{1}{50} \sum_{i=451}^{500} \hat{F}_{\text{ext, y}},\ \frac{1}{50} \sum_{i=451}^{500} \hat{F}_{\text{ext, z}} \right]$

principal force check $\leftarrow \left(|\overline{F}_{\text{ext, z}}| > |\overline{F}_{\text{ext, x}}|\right) \wedge \left(|\overline{F}_{\text{ext, z}}| > |\overline{F}_{\text{ext, y}}|\right)$

side forces check $\leftarrow \left(|\overline{F}_{\text{ext, x}}| < 2.5\right) \wedge \left(|\overline{F}_{\text{ext, y}}| < 2.5\right)$

force check $\leftarrow$ principal force check $\wedge$ side forces check

stabilization $\leftarrow$ **True**

for all checks $\in$ stabilization checks **do**
  if $|$checks $-$ end stabilization$| \geq$ stabilization threshold **then**
    stabilization $\leftarrow$ **False**
  end if
end for

mismatch detection $\leftarrow$ rapid change $\wedge$ stabilization $\wedge$ force check
return mismatch detection
end function

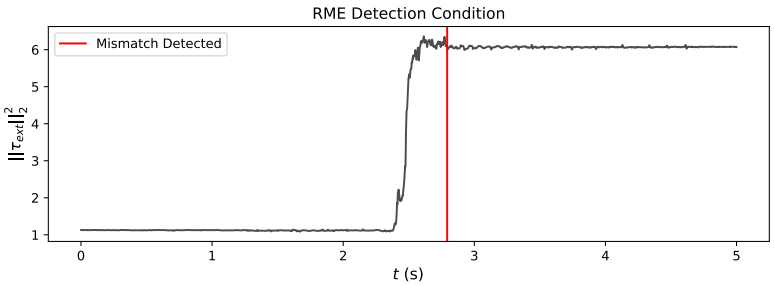

Figure 7: Example of mismatch detection algorithm execution. The algorithm monitors $\|\tau_{\text{ext}}\|_2^2$ and detects a mismatch in dynamics after observing a $\|\tau_{\text{ext}}\|_2^2$ increase followed by stabilization within a given threshold.

B Compensatory Action Formulation

To compensate for the estimated mismatch, we define $\hat{\tau}_c$ (5) as control torque τ_c from the Constrained Passive Interaction Controller with additional mismatch gravity compensation term. In our formulation, we solve the CPIC (8) subject to equality constraints from the Nominal Dynamics model, not corrected by predicted model mismatch. We found that this strategy mitigates potential instabilities due to RME overcompensation, while succeeding in trajectory tracking. This approach also maintains RME controller-agnostic, as the compensation is provided without influencing the nominal controller. However, any impedance controller used with RME can be corrected using the estimation result to improve tracking accuracy further.

C Effect of Neural Network on Inference Accuracy

To examine the importance of the Neural Network in our framework, we conducted an ablation study, comparing inference results achieved with $p(\theta)$ constructed by $\mu_{\text{prior}} = f_{\text{NN}}(\cdot)$ and $\mu_{\text{prior}} = 0$. By running the RME algorithm on 100 simulation-generated datasets, representing the manipulator's dynamic behavior under different mismatch parameters θ, we compare RME estimation results under different prior constructions, as shown in Figure 8. Since Inverse Dynamics is non-linearly dependent on mismatch parameters θ, Neural Network allows us to guide the Inference toward the true center of mass estimation, allowing for more accurate mismatch parameters θ estimation.

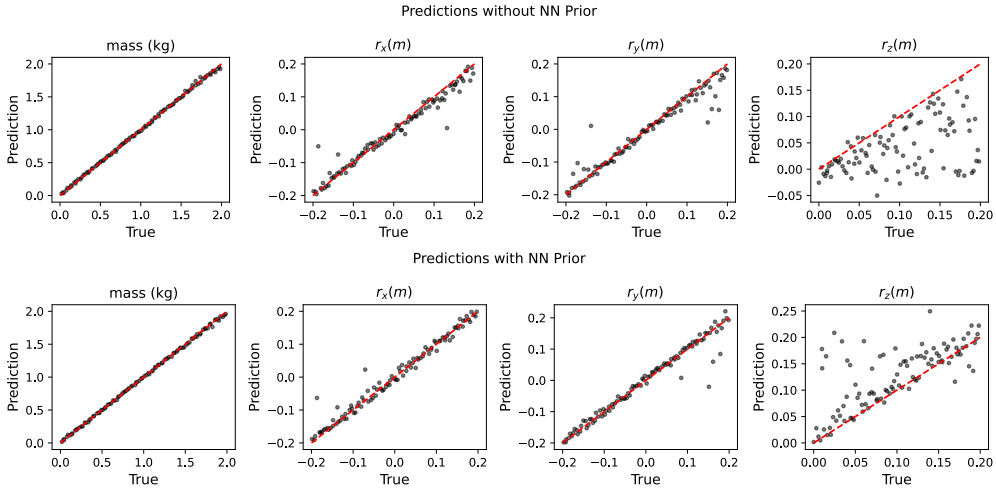

Figure 8: Comparison of parity plots for RME estimation of mismatch parameters θ under different prior constructions; as shown, Neural Network guidance allows for more robust center of mass estimation.

D Pick and Place with Human-Robot Interactions

In this experiment, we evaluate model performance in more complex continuous adaptation scenarios by performing a pick and place task with a passive velocity-based inverse kinematics (PVIK) controller [46], utilizing the qb SoftHand2 Research end effector and the OptiTrack motion capture system to track the basket. In the first test, Figure 9, the robot aims to pick up an unknown object (700 grams) and place it on top of the box, maintaining passivity with respect to human-generated perturbations. During the process, we add 500 grams to the basket. RME allows the robot to track the desired trajectory by rapidly adapting to unknown dynamics, while a controller without RME fails to pick up a heavy object. In the second test, Figure 10, the robot aims to track and intercept the basket (1200 grams) from a human and place it on top of the box, which is only possible with the use of RME. In both experiments, the controller with RME augmentation preserves stability and passivity of the closed-loop system.

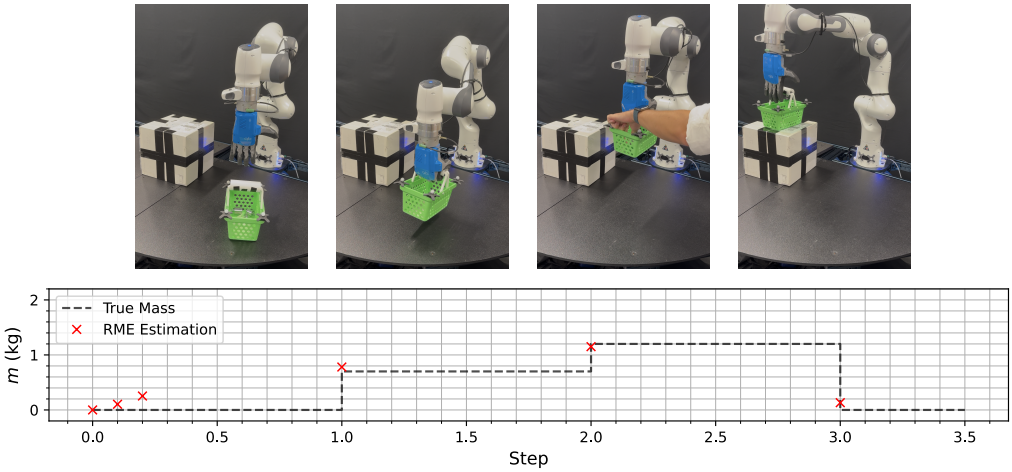

Figure 9: Manipulator adaptation to sequential changes in the dynamics model while interacting with unknown, heavy objects. Each step represents modifying the dynamics model of the end effector; at Step 1, the manipulator picks up an unknown 700-gram basket, at Step 2, the user adds 500 grams to the basket, and at Step 3, the robot places the basket on top of the box. RME prediction between two steps denotes immediate correction of the previous θ estimate.

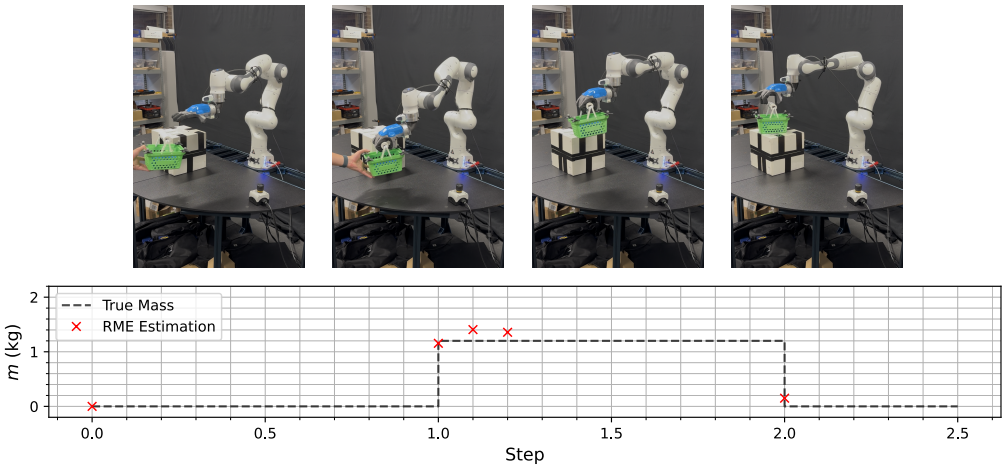

Figure 10: Manipulator intercepting basket of unknown mass (1200 grams) from a user and placing it on top of the box. Using RME, the robot can safely adapt and finish the given task, while maintaining passivity wrt. human-generated perturbations, ensuring safe pHRI. RME prediction between two steps denotes immediate correction of the previous θ estimate.

E RME Estimation Results from Static Experiments

To perform the evaluation presented in Section 5, we designed two end effectors as baskets attached to the manipulator, as shown in Figure 11. The white end-effector was 3D printed and served as the base for static experiments and limit cycle tracking. Its construction allowed us to place heavy objects at precise positions, providing reliable ground truth information for RME evaluation. Further, we designed a 3D printed connector to a commercially available basket to test Sequential Adaptation with Human-Robot Interactions. This setup allowed us to verify RME adaptation capabilities with everyday objects, ensuring our model is generalizable to various tasks.

Table 2 shows RME estimation results over 60 independent experiments, where the physical manipulator, subject to sudden changes in the dynamics model resulting from adding unknown mass to the end-effector, aimed to maintain target equilibrium position and orientation, as described in Section 5.2. Estimation results are mean predictions and standard deviations from 10 experiments per Applied Mismatch. The average RME estimation time over 60 experiments was 226 ms.

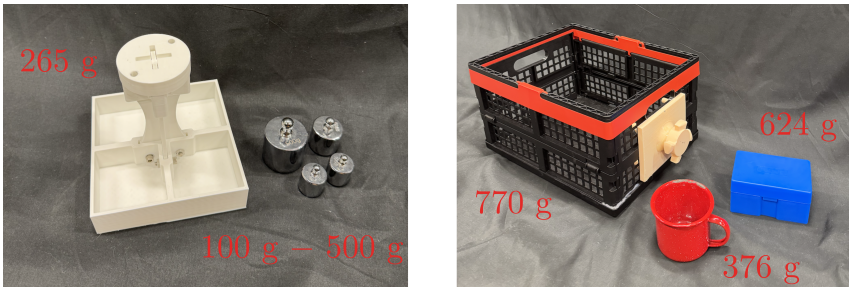

Figure 11: End-effectors and unknown objects used in the RME evaluation. The left image shows the 3D-printed basket used for static experiments and limit cycle tracking. The right image depicts a commercially available basket with a 3D printed connector, used in a Sequential Adaptation with Human-Robot interactions experiment.

Applied Mismatch				**Mean Prediction**			
m (kg)	r_x (m)	r_y (m)	r_z (m)	m (kg)	r_x (m)	r_y (m)	r_z (m)
0.300	0.03	0.00	0.13	0.380 ±0.018	-0.04±0.03	0.02±0.01	0.13±0.04
0.500	0.06	0.00	0.13	0.511±0.031	0.04±0.04	0.01±0.01	0.15±0.02
0.700	0.06	0.00	0.13	0.681±0.019	0.07±0.03	0.00±0.01	0.14±0.02
0.900	0.05	-0.03	0.13	0.861±0.024	0.07±0.01	-0.02±0.01	0.11±0.03
1.100	0.05	0.02	0.13	1.053±0.018	0.07±0.01	0.00±0.01	0.00±0.04
1.290	0.05	0.01	0.13	1.269±0.008	0.05±0.00	0.00±0.01	0.00±0.06

Table 2: Framework mean estimation results and standard deviations over 60 independent experiments. At each trial, we recorded the robot's proprioceptive feedback over a 200-ms-long time window, followed by the model estimation, which took 226 ms on average. In each experiment, RME prediction allowed the robot to converge back to the desired trajectory.

As mentioned in Section 5.2, the manipulator's goal is to maintain target equilibrium position and orientation when interacting wth unknown objects from the environment. Examples of such adaptations are depicted in Figure 12. As shown, the robot controlled with CPIC without the estimation framework deviates from the target position and orientation as it is pulled by the additional mass attached to the end-effector. RME allows the manipulator to correct its position and converge to the equilibrium point from before applying the mismatch.

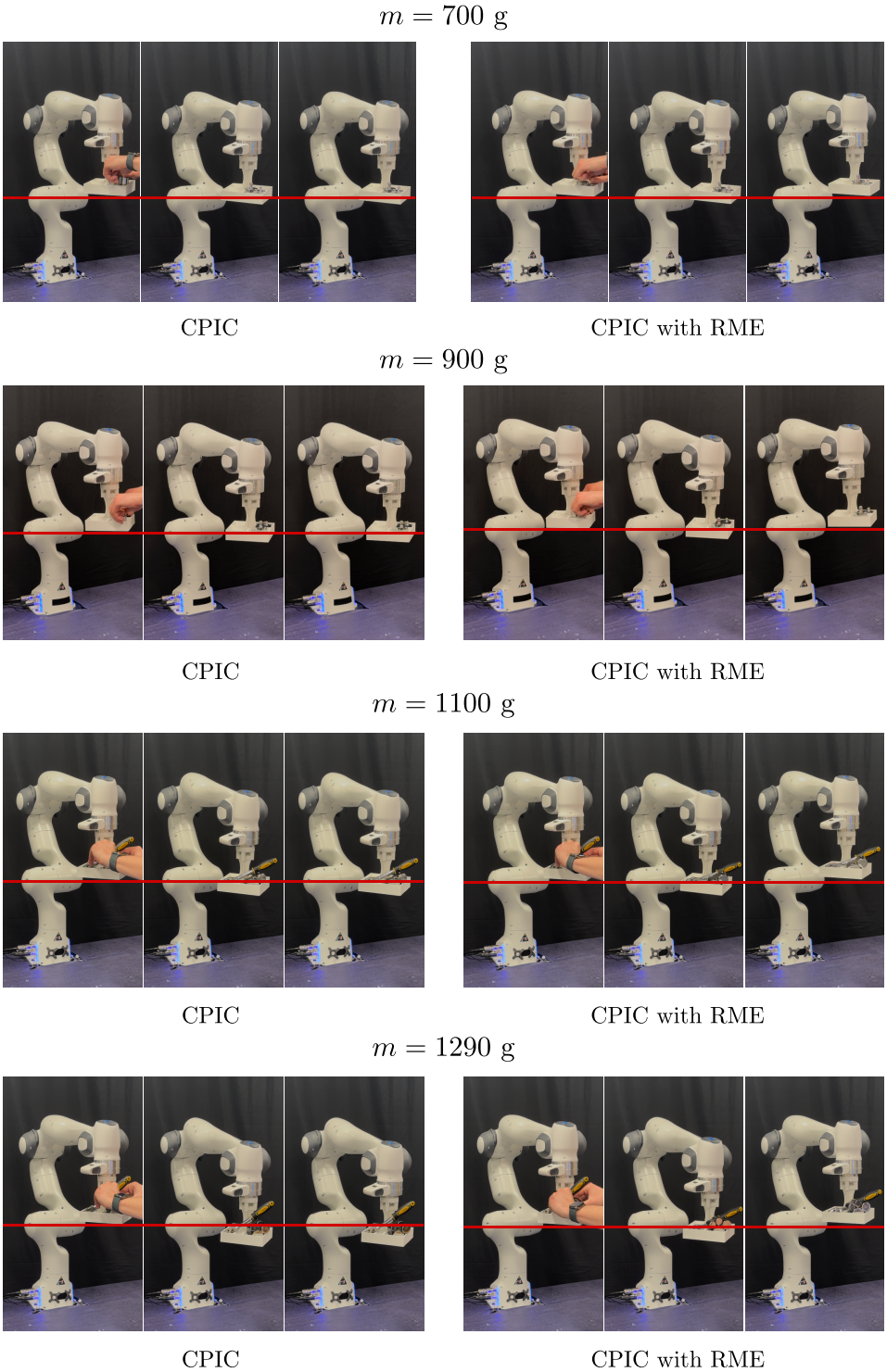

Figure 12: Comparison of manipulators' behavior subject to unknown mismatch in dynamics model ranging from 700 g to 1290 g, when controlled with Constrained Passive Interaction Controller (CPIC), and CPIC with RME adaptation. Red lines indicate the manipulator's equilibrium position before experiencing the mismatch. As shown, for all weights used in the experiment, RME allows the robot to correct its position and orientation, resulting in convergence to the equilibrium position; CPIC-controlled manipulators deviate from the equilibrium state and fail to converge to the goal.

F Effect of Data Collection Interval Length on Inference Accuracy

To choose the data collection interval for the Neural Network and Variational Inference, we conducted a comparative analysis to examine model performance under different collection intervals. We evaluated the study with 11 datasets from real robot experiments to ensure the model can account for measurement noise and external factors not present in the simulation environment. As shown in Table 3, the RME model achieves lower MSE values with increased collection intervals. Although we noticed that a longer collection interval might allow us to capture more information-rich feedback, based on this study, we decided to set the collection interval to 200 ms, as it allows us to achieve high estimation accuracy while shortening RME execution time. The collection interval length can be tuned based on the particular manipulator and task specifications.

Collection Interval	MSE m (kg^2)	MSE r_x (m^2)	MSE r_y (m^2)	MSE r_z (m^2)
50 ms	2.480 $\times 10^{-3}$	1.607 $\times 10^{-3}$	0.574 $\times 10^{-3}$	8.935$\times 10^{-3}$
100 ms	2.436$\times 10^{-3}$	0.909 $\times 10^{-3}$	0.468$\times 10^{-3}$	4.827$\times 10^{-3}$
200 ms	2.467$\times 10^{-3}$	0.86 $\times 10^{-3}$	0.396$\times 10^{-3}$	3.88$\times 10^{-3}$
300 ms	2.081 $\times 10^{-3}$	0.979 $\times 10^{-3}$	0.477 $\times 10^{-3}$	1.25$\times 10^{-3}$

Table 3: Comparison of RME Mean Squared Errors for θ parameters for estimation with different data collection intervals.

G Neural Network Architecture

The summary of the Neural network architecture described in Section 4.1 is shown in Table 4. As input, we choose a sequence of $M = 20$ pseudo-wrenches $\hat{W}_{\text{ext}}$ calculated using the manipulator's proprioceptive feedback, uniformly sampled from N collected data points over a 200 ms time window. Decreasing input dimensionality ($N = 200 \rightarrow M = 20$) reduces model complexity without negatively impacting model performance.

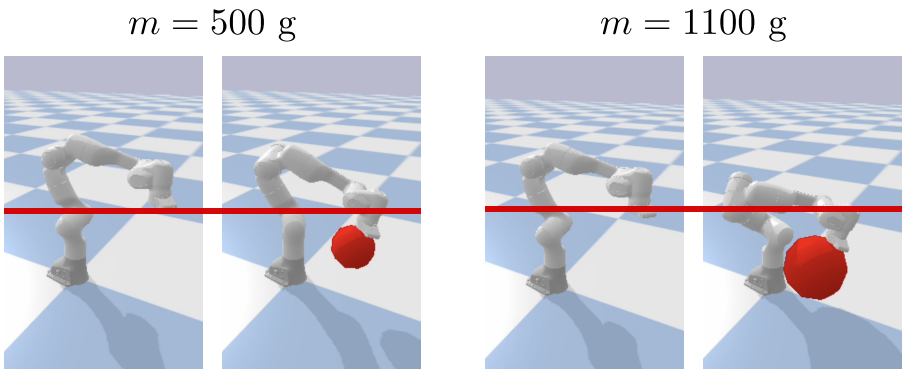

Figure 13: Computer simulations of the manipulator's dynamics subject to an unknown mass attached to the end effector. The figure compares the manipulator's equilibrium position before applying the mismatch (red line) to its position shortly after applying unknown masses of 500 grams and 1100 grams (red spheres), shown in the left and right figures, respectively.

To train the Neural Network, we used a dataset generated from 350 computer simulations of a manipulator's dynamics experiencing different mismatch parameters θ, as shown in Figure 13. This allowed us to train the NN without performing multiple experiments on the physical robot, making the framework easily generalizable to other manipulator arms. We further augmented the dataset by extracting input time series beginning at different time steps. This provided us with 1050 training inputs, which we split into 80 - 20 training-validation datasets. We tested the network performance

with a separate dataset representing time-translated data relative to the training dataset; the NN made accurate predictions of mismatch parameters, which is essential, assuming we cannot guarantee the exact positioning of the extracted time series of data. Figure 14 shows the training and validation loss history over the training iterations. As shown in the plot, both loss curves converge, suggesting that the applied dropout prevented the model from overfitting.

Parameter	Value
Input Sequence Length (M)	20
Input Dimension	$M \times 6$
Convolution Kernel Size	5
Convolution Output Dimension	$M \times 64$
Number of Attention Heads	8
MLP Hidden Dimension	256
MLP Activation Function	ReLU
MLP Sequential Blocks	3
Dropout Rate	0.1
Output Dimension	4
Total Number of Parameters	53,252
Train-Validation Split	80 – 20
Training Dataset Size	$M \times 6 \times 840$
Validation Dataset Size	$M \times 6 \times 210$
Learning Rate	0.0001
Training Iterations	50,000
Training Time	1h 0m 26s

Table 4: Summary of the Neural Network architecture.

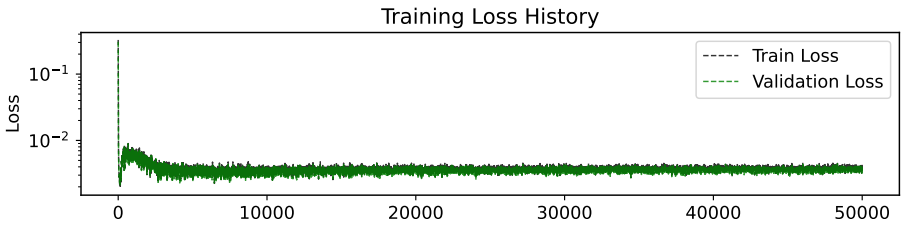

Figure 14: Neural Network train and validation loss history.

H Proof of Proposition 4.1

Let us re-state the proposition that must be proven in this section:

Proposition 4.1 Let a robotic manipulator with dynamics (4) be controlled by an augmented passive impedance control law defined in (5) with τ_c computed by (8). Given an imperfect mismatch parameter estimation $\Delta F_m \neq 0$ and $\Delta r_{\text{CoM}} \neq 0$ the system will be locally asymptotically stable as unwanted equilibria may arise when $\|\Delta F_{\text{mm}} - F_c(x)\| = 0$, with ΔF_{mm} denoting the remaining unknown model mismatch in the EE expressed as a task-space wrench. Yet, the closed-loop system behavior remains passive wrt. input-output port $(F_{\text{ext}} + \Delta F_{\text{mm}}, \dot{x})$. ■

Proof. Without loss of generality, assuming a single constraint exists, we can convert the CPIC optimization problem (8) enacting the E-CBF kinematic constraint to [32]:

$$\min_{\tau_c} \quad \left\|J(q)^{-\top}\tau_c - F_c(x)\right\|_2^2 \quad \text{s.t} \quad \begin{cases} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau_c + \tau_{\text{ext}} \\ \nabla_q h(q)^\top \ddot{q} \geq b(q,\dot{q}) \end{cases} \tag{19}$$

given $\mathcal{K} = [k_1 \; k_2]$ and $b(q,\dot{q}) = -k_1 h(q) - k_2 \nabla_q h(q)^\top \dot{q} - \dot{q}^\top \nabla_q^2 h(q)\dot{q}$. In [32], the authors mention that if an optimal τ_c^* for (19) can be found in the feasible set, i.e., when $J(q)^{-\top}\tau_c = F_c(x)$ or $F_c \notin \mathcal{N}(J(q)^{-\top})$ then the closed-loop behavior of the robot is *passive when feasible*. Next, we prove this statement in a more rigorous fashion. Let us rewrite the equality constraint in (19) as:

$$\ddot{q} = M(q)^{-1}(\tau_c + \rho(q,\dot{q})) \;\text{ with }\; \rho(q,\dot{q}) := \tau_{\text{ext}} - C(q,\dot{q})\dot{q} - G(q) \tag{20}$$

Via (20) the CPIC controller (19) can be expressed as:

$$\min_{\tau_c} \quad \left\|J(q)^{-\top}\tau_c - F_c(x)\right\|_2^2 \quad \text{s.t.} \;\; \nabla_q h(q)^\top \left(M(q)^{-1}(\tau_c + \rho(q,\dot{q}))\right) \geq b(q) \tag{21}$$

which can be re-written in terms of τ_c as follows:

$$\min_{\tau_c} \quad \frac{1}{2}\tau_c^\top Q \tau_c + \tau_c^\top p + c \quad \text{s.t.} \quad A\tau_c \geq \nu \tag{22}$$

where $c = ||F_c(x)||_2^2$ and we define the following variables:

$$\begin{aligned} Q &:= 2J(q)^{-1}J(q)^{-\top} \qquad & A &:= \nabla_q h(q)^\top M(q)^{-1} \\ p &:= -2J(q)^{-1}F_c(x) \qquad & \nu &:= b(q) - A\rho(q,\dot{q}) \end{aligned} \tag{23}$$

This shows that the CPIC controller (19) is reduced to a standard constrained QP, which is always feasible. Hence, we can express the optimal solution τ_c^* as an analytic closed-form expression via KKT conditions. The KKT system for the *active inequality* constraints becomes:

$$\begin{bmatrix} Q & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \tau_c^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -p \\ \nu \end{bmatrix} \tag{24}$$

with λ^* being the optimal solution of Lagrange multipliers corresponding to the inequality constraints $A\tau_c \geq \nu$ in the Lagrangian of (22) which is $\mathcal{L}(\tau_c, \lambda) = \frac{1}{2}\tau_c^\top Q \tau_c + \tau_c^\top p - \lambda^\top (A\tau_c - \nu)$.

We will next derive the closed-form expression in the following two scenarios.

When the robot is far from violating the kinematic constraints, this means that the E-CBF inequality $\nabla_q h(q)^\top \ddot{q} - b(q,\dot{q}) \geq 0$ <u>is satisfied</u>, in such cases the inequality constraint is *inactive* and thus the optimal solution to (22) (via (24)) becomes:

$$\begin{aligned} \tau_c^* &= -Q^{-1}p \\ &= \left(2J(q)^{-1}J(q)^{-\top}\right)^{-1} 2J(q)^{-1}F_c(x) \\ &= J(q)^\top J(q) J(q)^{-1} F_c(x) \qquad \text{(assuming } J(q) \text{ is full rank)} \\ &= J(q)^\top F_c(x). \end{aligned} \tag{25}$$

which reduces to the solution of the original passive interaction controller

When the robot is close to violating (or in violation) of the kinematic constraints, this means that the E-CBF inequality $\nabla_q h(q)^\top \ddot{q} - b(q, \dot{q}) \geq 0$ is <u>not satisfied</u>, in such cases an optimal QP solution, τ_c^* and λ^* can be expressed by the analytical solution of the KKT system (24) as below:

$$\begin{aligned} \tau_c^* &= Q^{-1}\left(-p - A^\top \lambda^*\right) \\ &= J(q)^\top F_c(x) - Q^{-1} A^\top \lambda^* \end{aligned} \tag{26}$$

assuming $J(q)$ is full rank and with:

$$\lambda^* = \left(AQ^{-1}A^\top\right)^{-1}\left(AQ^{-1}p + \nu\right) \tag{27}$$

shows that the constrained solution is a correction to the unconstrained torque estimate $J(q)^\top F_c(x)$ from (25). During execution, the solution of the CPIC (19) will either be (25) or (26).

Thus, we can now analyze the closed-loop stability and passivity of these two solutions when plugged into the robot dynamics (4) via our augmented controller with RME compensation (5) as,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau_c^* - J(q)^\top (F_{ext} + \Delta F_{mm}), \tag{28}$$

with ΔF_{mm} representing the estimation error of the true unknown model mismatch:

$$\Delta F_{mm} = \begin{bmatrix} \hat{F}_m(\theta) \\ \hat{r}_{\text{CoM}}(\theta) \times \hat{F}_m(\theta) \end{bmatrix} - \begin{bmatrix} F_m \\ r_{\text{CoM} \times F_m} \end{bmatrix} \tag{29}$$

As our control and RME compensation is defined in task-space we convert (28) to,

$$M_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} + G_x(q) = J(q)^{-\top}\tau_c^* + (F_{ext} + \Delta F_{mm}), \tag{30}$$

with the gravity vector being mapped to task-space by $G_x(q) = J(q)^{-\top}G(q)$, and $M_x(q) = J(q)^{-\top}M(q)J(q)^{-1}$, $C_x(q, \dot{q}) = J(q)^{-\top}C(q, \dot{q})J^{-1} - J(q)^{-\top}M(q)J(q)^{-1}\dot{J}(q)J(q)^{-1}$. Similarly, the joint-space velocities are mapped to task-space as $\dot{x} = J(q)\dot{q}$ with $\dot{x} \in \mathbb{R}^d$ and joint-space accelerations $\ddot{q}$ mapped to task-space as follows $\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$.

Closed-loop behavior when τ_c^* solved as (25) Let us now analyze the closed-loop behavior of (30) when the robot is not close to a kinematic constraint; i.e., τ_c^* is expressed as (25), which reduces to:

$$\begin{aligned} M_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} + G_x(q) &= J(q)^{-\top}J(q)^\top \underbrace{F_c(x)}_{(7)} + (F_{ext} + \Delta F_{mm}) \\ M_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} &= -D(x)\big(\dot{x} - f(x)\big) + (F_{ext} + \Delta F_{mm}) \end{aligned} \tag{31}$$

The damping function $D(x)$ can be designed to dissipate energy in orthogonal directions to $f(x)$ formulated as $D(x) = V(x)\Lambda(x)V(x)^T$ with (for a 2D example):

$$V(x) = \begin{bmatrix} e_1(x) & e_2(x) \end{bmatrix}, \quad e_1(x) = \frac{f(x)}{\|f(x)\|}, \quad e_1(x)^T e_2(x) = 0, \quad \Lambda(x) = \begin{bmatrix} \lambda_1(x) & 0 \\ 0 & \lambda_2(x) \end{bmatrix} \tag{32}$$

with the eigenvalues $\lambda_i(x) \geq 0$ setting the impedance in orthogonal directions of $Q(x)$ basis, as originally presented [13]. These $D(x)$ design choices for the controller simplify it to:

$$M_x(q)\ddot{x} + \left(C_x(q, \dot{q}) + D(x)\right)\dot{x} - \lambda_1 f(x) = (F_{ext} + \Delta F_{mm}) \tag{33}$$

The stability and passivity of the robot's closed-loop dynamics with RME (33) can be proven by following (3) with the following energy storage function:

$$S(x, \dot{x}) = \frac{1}{2}\dot{x}^T M_x \dot{x} + \lambda_1 \mathcal{V}(x) \tag{34}$$

that includes the kinetic energy of the robot and the potential energy injected by the controller term. The latter depends solely on the eigenvalue λ_1 and the Lyapunov function $\mathcal{V}(x)$ used to ensure asymptotic stability of $f(x)$, which for this analysis we assume to be conservative $f(x) = -\nabla\mathcal{V}(x)$.

The time-derivative of the energy storage function is,

$$
\begin{aligned}
\dot{S}(x,\dot{x}) &= \dot{x}^T M_x \ddot{x} + \frac{1}{2}\dot{x}^T \dot{M}_x \dot{x} + \lambda_1 \nabla \mathcal{V}^T \dot{x} \\
&= \dot{x}^T \left((F_{ext} + \Delta F_{mm}) - (C_x + D(x))\dot{x} + \lambda_1 f(x) \right) + \frac{1}{2}\dot{x}^T \dot{M}_x \dot{x} + \lambda_1 \nabla \mathcal{V}(x) \\
&= \dot{x}^T (F_{ext} + \Delta F_{mm}) - \dot{x}^T D(x)\dot{x} + \frac{1}{2}\dot{x}^T \underbrace{(\dot{M}_x - 2C_x)}_{=0 \ \ (2)} \dot{x} + \lambda_1 \left(f(x) + \nabla \mathcal{V}(x) \right) \\
&= \dot{x}^T (F_{ext} + \Delta F_{mm}) - \dot{x}^T D(x)\dot{x} \\
&\leq \dot{x}^T (F_{ext} + \Delta F_{mm})
\end{aligned}
\tag{35}
$$

Thus, the following conditions are guaranteed,

$$
\begin{cases}
\dot{S} \leq 0 & \Delta F_{mm} = \mathbf{0} \ \ \& \ \ F_{ext} = \mathbf{0} \quad \text{(stable)} \\
\dot{S} \leq \Delta F_{mm}^\top \dot{x} & \Delta F_{mm} \neq \mathbf{0} \ \ \& \ \ F_{ext} = \mathbf{0} \quad \text{(passive)} \\
\dot{S} \leq (F_{ext} + \Delta F_{mm})^\top \dot{x} & \Delta F_{mm} \neq \mathbf{0} \ \ \& \ \ F_{ext} \neq \mathbf{0} \quad \text{(passive)}
\end{cases}
\tag{36}
$$

Which means that, when there is no external force $F_{ext} = \mathbf{0}$ then, if RME provides perfect estimates then we can guarantee that the robot will track the DS $f(x)$ and converge to its target x^* or trajectory $x(t)^*$. Nevertheless, if the estimate is incorrect, the system will remain passive wrt. the estimation error $\Delta F_{mm} \neq \mathbf{0}$, which effectively reshapes the desired $f(x)$, potentially creating a spurious attractor when, $\dot{x}^T(\Delta F_{mm}) = \dot{x}^T D(x)\dot{x}$. Finally, when an external force exists then the system is passive wrt. the input-output port $(F_{ext} + \Delta F_{mm}, \dot{x})$.

Closed-loop behavior when τ_c^* solved as (26). Following the same procedure, we can write the closed-loop dynamics of the robot when the constraints are *active*, while also considering the RME compensation error as follows,

$$
\begin{aligned}
M_x(q)\ddot{x} + C_x(q,\dot{q})\dot{x} + G_x(q) &= \underbrace{F_c(x)}_{(7)} - \underbrace{(J(q)^{-\top} Q^{-1} A^\top \lambda^*)}_{F_{\text{ecbf}}} + (F_{ext} + \Delta F_{mm}) \\
M_x(q)\ddot{x} + C_x(q,\dot{q})\dot{x} &= -D(x)\big(\dot{x} - f(x)\big) + (F_{\text{ecbf}} + F_{ext} + \Delta F_{mm}) \\
M_x(q)\ddot{x} + \left(C_x(q,\dot{q}) + D(x)\right)\dot{x} - \lambda_1 f(x) &= (-F_{\text{ecbf}} + F_{ext} + \Delta F_{mm})
\end{aligned}
\tag{37}
$$

Note that $F_{\text{ecbf}} \in \mathbb{R}^6$ is the pseudo-wrench the robot experiences when a kinematic constraint in joint-space is being activated; i.e., the robot will stiffen up near a constraint boundary. Following (34) and (35), the time-derivative of the energy storage function for the closed-loop system (37) is,

$$
\begin{aligned}
\dot{S}(x,\dot{x}) &= \dot{x}^T M_x \ddot{x} + \frac{1}{2}\dot{x}^T \dot{M}_x \dot{x} + \lambda_1 \nabla \mathcal{V}^T \dot{x} \\
&= \dot{x}^T (F_{ext} + \Delta F_{mm}) - \dot{x}^T D(x)\dot{x} - \dot{x}^T \underbrace{(J(q)^{-\top} Q^{-1} A^\top \lambda^*)}_{F_{\text{ecbf}}}.
\end{aligned}
\tag{38}
$$

Now we have an indefinite term in our power equation corresponding to the effect of enforcing a kinematic constraints at the joint-level by the E-CBF. Nevertheless, we can easily understand the behavior of this term. When the task-space motion of the robot $\dot{x}$ is feasible, either driven by $f(x)$ via $F_c(x)$ or an external force F_{ext} or estimation error ΔF_{mm} the E-CBF inequality $\nabla_q h(q)^\top \ddot{q} - b(q,\dot{q}) \geq 0$ <u>is satisfied</u>, then $\dot{x}^\top F_{\text{ecbf}} \geq 0$ preserving stability and passivity as in (36).

Conversely, if the E-CBF inequality $\nabla_q h(q)^\top \ddot{q} - b(q,\dot{q}) \geq 0$ <u>is not satisfied</u> the QP will enforce it, stiffening up in the opposite direction of $\dot{x}$ and generating an opposite force rendering $\dot{x}^\top F_{\text{ecbf}} < 0$. This is the only case when passivity is lost, yet it is a desired behavior, as we seek to enforce the constraints. This property is referred to in [32] as *passive when feasible*. □