

Additional Related Work

We noticed that recent work by B. Zhang, Z. Zhou, and R. Vasudevan, accepted to Robotics: Science and Systems 2025 Conference was posted on Arxiv on April 30, 2025 – exactly at the time of Rapid Mismatch Estimation manuscript submitted to CoRL 2025. This recent related work proposed an “online” system identification framework that allows the identification of inertia parameters of unknown objects attached to the manipulator’s end-effector. While the problem they are tackling is similar to ours, similar to prior work, it still requires performing a system identification/calibration action, typically around 10s, before execution of the trajectory tracking task the wish the robot to perform. We, on the other hand, offer an approach that performs the model mismatch estimation online, in real-time and while the robot is performing a task of tracking a passive DS motion policy. During evaluation, we showed that RME can be employed during task execution with very short computation time (400 ms), allowing the manipulator to estimate and adapt to unknown dynamics online and in real-time, while maintaining passivity with respect to human-generated perturbation.

If our manuscript is accepted, we will discuss the work by Zhang et al. in the **Related Work** section in future manuscript revisions.

Reference

B. Zhang, Z. Zhou, and R. Vasudevan. Provably-safe, online system identification, 2025. URL <https://arxiv.org/abs/2504.21486>. <https://roahmlab.github.io/OnlineSafeSysID/>

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 mismatch detection condition. 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 mean $\|\tau_{\text{ext}}\|_2^2$ over short time windows within stabilization interval, and compare them with mean $\|\tau_{\text{ext}}\|_2^2$ at the end of the interval. This approach allows us to detect both adding and removing objects from the end effector. 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 mismatch applied to the end-effector, and run RME only when both detection conditions were met simultaneously. 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.

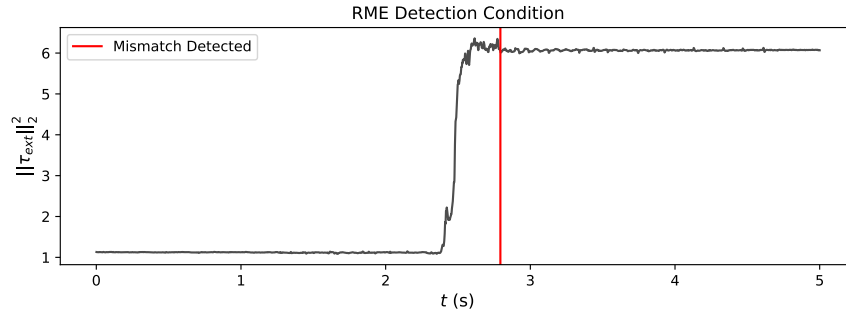


Figure 7: Example of mismatch detection algorithm execution. The algorithm monitors $\|\tau_{\text{ext}}\|_2^2$ and detects a mismatch in dynamics after observing a $\|\tau_{\text{ext}}\|_2^2$ increase followed by stabilization within a given threshold.

Algorithm 1 Mismatch Detection Algorithm

```
function ACTIVATIONCONDITION( $\{\|\tau_{\text{ext}}\|_2^2\}_{i=1}^N$ ) ▷ N=500 steps  
  rapid change  $\leftarrow \left| \|\tau_{\text{ext}}\|_2^2(500) - \|\tau_{\text{ext}}\|_2^2(1) \right| > \text{activation threshold}$   
  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  $\leftarrow \frac{1}{10} \sum_{i=491}^{500} \|\tau_{\text{ext}}\|_2^2(i)$   
  stabilization  $\leftarrow \mathbf{True}$   
  for all checks  $\in$  stabilization checks do  
    if  $|\text{checks} - \text{end stabilization}| \geq \text{stabilization threshold}$  then  
      stabilization  $\leftarrow \mathbf{False}$   
    end if  
  end for  
  mismatch detection  $\leftarrow \text{rapid change} \wedge \text{stabilization}$   
  return mismatch detection  
end function
```

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 further improve tracking accuracy.

C 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 8. 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.

As mentioned in Section 5.2, the manipulator’s goal is to maintain target equilibrium position and orientation when interacting with unknown objects from the environment. Examples of such adaptations are depicted in Figure 9. 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.

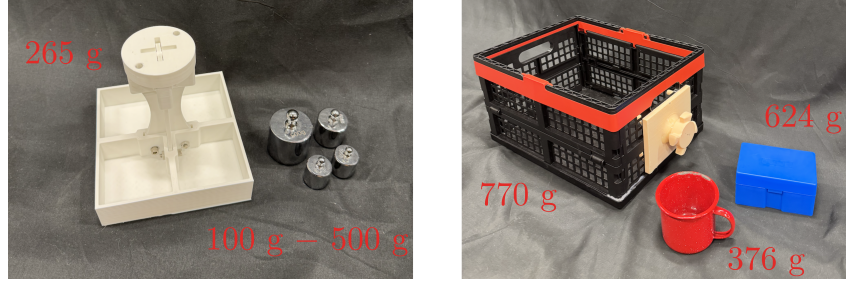


Figure 8: End-effectors and unknown objects used in the RME evaluation. The left image shows the 3D printed basket used for static experiments and limit cycle tracking. The right image depicts a commercially available basket with a 3D printed connector, utilized to conduct a Sequential Adaptation with 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.

498 D Effect of Neural Network on Inference Accuracy

499 To examine the importance of the Neural Network in our framework, we conducted an ablation
500 study, comparing inference results achieved with $p(\theta)$ constructed by $\mu_{\text{prior}} = f_{\text{NN}}(\cdot)$ and $\mu_{\text{prior}} = 0$.
501 By running the RME algorithm on 100 simulation-generated datasets, representing the manipula-
502 tor’s dynamic behavior under different mismatch parameters θ , we compare RME estimation results
503 under different prior constructions, as shown in Figure 10. Since Inverse Dynamics is non-linearly
504 dependent on mismatch parameters θ , Neural Network allows us to guide the Inference toward the
505 true center of mass estimation, allowing for more accurate mismatch parameters θ estimation.

506 E Effect of Data Collection Interval Length on Inference Accuracy

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

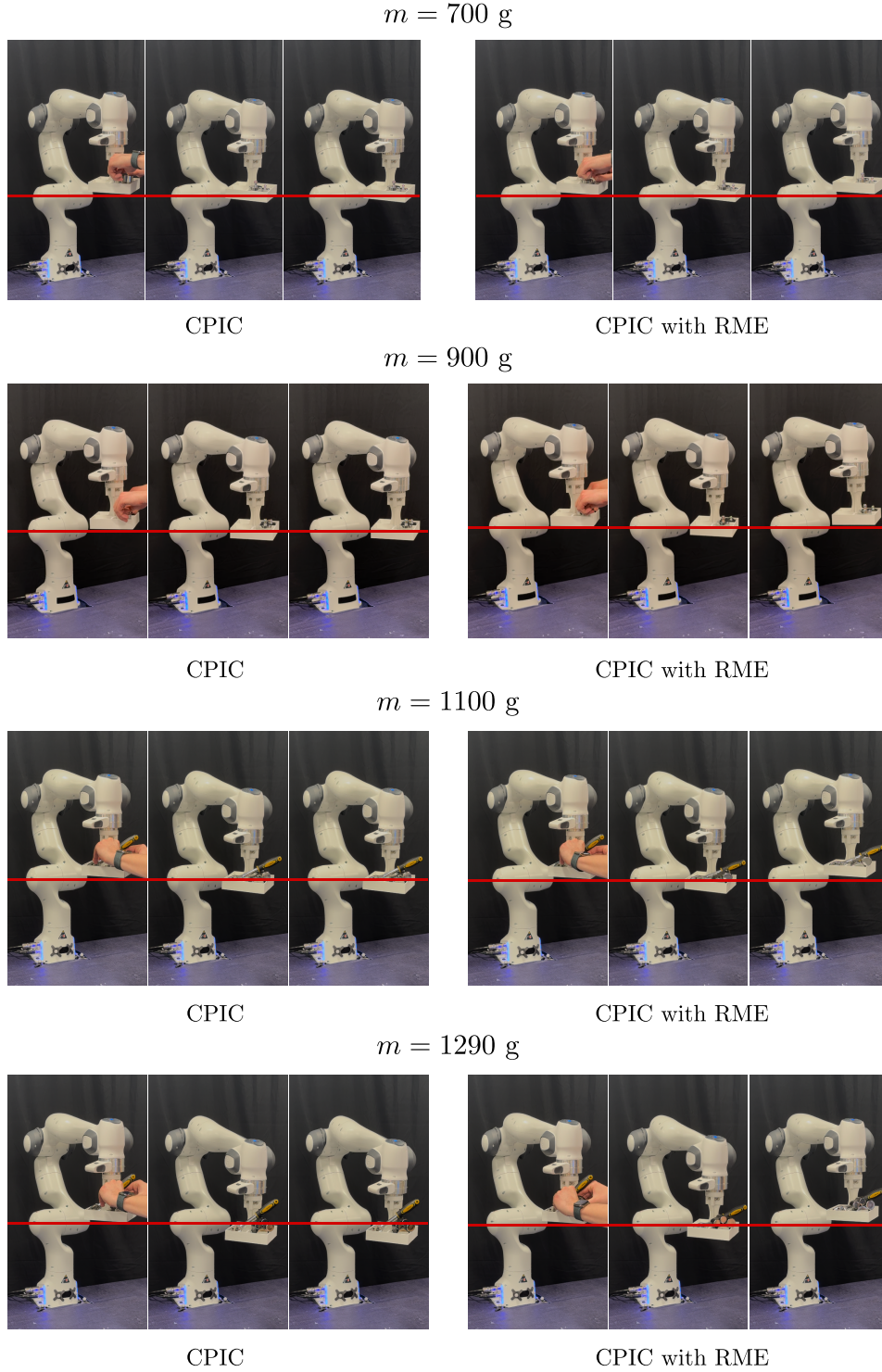


Figure 9: Comparison of manipulators' behavior subject to unknown mismatch in dynamics model ranging from 700 g to 1290 g, when controlled with Constrained Passive Interaction Controller (CPIC), and CPIC with RME adaptation. Red lines indicate the manipulator's equilibrium position before experiencing 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.

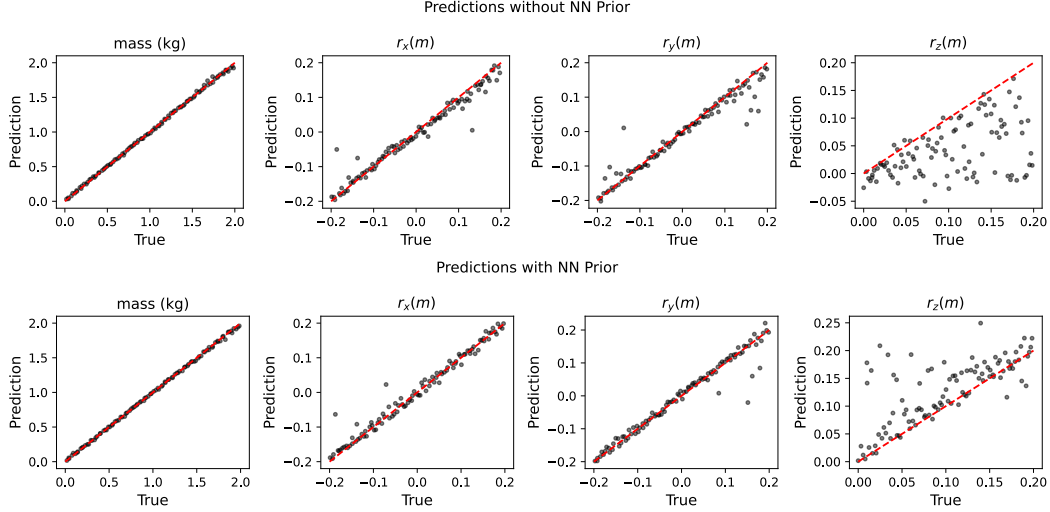


Figure 10: Comparison of parity plots for RME estimation of mismatch parameters θ under different prior constructions; as shown, Neural Network guidance allows for more robust center of mass estimation.

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

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

516 F Neural Network Architecture

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

522 To train the Neural Network, we used a dataset generated from 350 computer simulations of a
 523 manipulator's dynamics experiencing different mismatch parameters θ , as shown in Figure 11. This
 524 allowed us to train the NN without performing multiple experiments on the physical robot, making
 525 the framework easily generalizable to other manipulator arms. We further augmented the dataset by
 526 extracting input time series beginning at different time steps. This provided us with 1050 training
 527 inputs, which we split into 80 - 20 training-validation datasets. We tested the network performance
 528 with a separate dataset representing time-translated data relative to the training dataset; the NN made
 529 accurate predictions of mismatch parameters, which is essential, assuming we cannot guarantee the
 530 exact positioning of the extracted time series of data. Figure 12 shows the training and validation
 531 loss history over the training iterations. As shown in the plot, both loss curves converge, suggesting
 532 that the applied dropout prevented the model from overfitting.

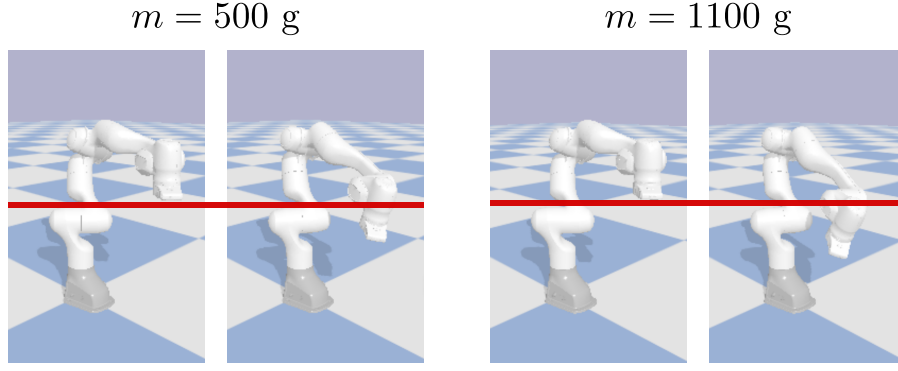


Figure 11: Computer simulations of manipulator's dynamics subject to unknown mass attached to the end effector. Figure compares manipulator's equilibrium position before applying the mismatch (red line) to its position shortly after applying unknown masses of 500 grams and 1100 grams, shown in the left and right figures respectively.

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.

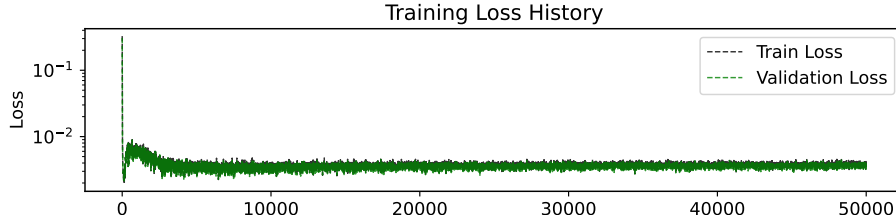


Figure 12: Neural Network train and validation loss history.

G 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 [26]:

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

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 [26], 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:

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

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

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

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

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

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

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

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:

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

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 $\nabla_q h(q)^\top \ddot{q} - b(q, \dot{q}) \geq 0$ is satisfied, in such cases the inequality constraint is *inactive* and thus the optimal solution to (22) (via (24)) becomes:

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

which reduces to solution of the original passive interaction controller

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

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

563 assuming $J(q)$ if full rank and with:

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

564 which shows that that the constrained solution is a correction to the unconstrained torque estimate
 565 $J(q)^\top F_c(x)$ from (25). During execution, the solution of the CPIC (19) will either be (25) or (26).

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

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

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

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

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

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

570 with the gravity vector being mapped to task-space by $G_x(q) = J(q)^{-\top} G(q)$, and $M_x(q) =$
 571 $J(q)^{-\top} M(q) J(q)^{-1}$, $C_x(q, \dot{q}) = J(q)^{-\top} C(q, \dot{q}) J(q)^{-1} - J(q)^{-\top} M(q) J(q)^{-1} \dot{J}(q) J(q)^{-1}$. Sim-
 572 ilarly, 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
 573 accelerations \ddot{q} mapped to task-space as follows $\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$.

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

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

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

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

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

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

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

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

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

585 The time-derivative of the energy storage function is,

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

586 Thus, the following conditions are guaranteed,

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

587 Which means that, when there is no external force $F_{ext} = 0$ then, if RME provides perfect estimates
588 then we can guarantee that the robot will track the DS $f(x)$ and converge to its target x^* or trajectory
589 $x(t)^*$. Nevertheless, if the estimate is incorrect, the system will remain passive wrt. the estimation
590 error $\Delta F_{mm} \neq 0$, which effectively reshapes the desired $f(x)$, potentially creating a spurious at-
591 tractors when, $\dot{x}^T (\Delta F_{mm}) = \dot{x}^T D(x)\dot{x}$ Finally, when an external force exists then the system is
592 passive wrt. the input-output port $(F_{ext} + \Delta F_{mm}, \dot{x})$.

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

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

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

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

599 Now, we have an indefinite term in our power equation corresponding to the effect of enforcing
600 a kinematic constraints at the joint-level by the E-CBF. Nevertheless, we can easily understand the
601 behavior of this term. When the task-space motion of the robot \dot{x} is feasible, either driven by $f(x)$ via
602 $F_c(x)$ or an external force F_{ext} or estimation error ΔF_{mm} then the E-CBF $\nabla_q h(q)^{\top} \ddot{q} - b(q, \dot{q}) \geq$
603 0 is satisfied and $\dot{x}^{\top} F_{cbf} \geq 0$ and the conditions of stability and passivity defined in (36) hold.
604 Conversely, if $\nabla_q h(q)^{\top} \ddot{q} - b(q, \dot{q}) \geq 0$ is not satisfied the QP will enforce it, stiffening up in the
605 opposite direction of \dot{x} and generating an opposite force rendering $\dot{x}^{\top} F_{cbf} < 0$. Such is the only case
606 when passivity is lost in our system, yet it is a desired behavior, as we seek to enforce the constraints.
607 This property is referred to in the main manuscript and in [26] as *passive when feasible*. \square