Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Provably-Safe, Online System Identification

Bohao Zhang[1], Zichang Zhou[1] and Ram Vasudevan[1]

***Abstract*—Precise manipulation tasks require accurate knowledge of payload inertial parameters. Unfortunately, identifying these parameters for unknown payloads while ensuring that the robotic system satisfies its input and state constraints while avoiding collisions with the environment remains a significant challenge. This paper presents an integrated framework that enables robotic manipulators to safely and automatically identify payload parameters while maintaining operational safety guarantees. The framework consists of two synergistic components: an online trajectory planning and control framework that generates provably-safe exciting trajectories for system identification that can be tracked while respecting robot constraints and avoiding obstacles and a robust system identification method that computes rigorous overapproximative bounds on end-effector inertial parameters assuming bounded sensor noise. Experimental validation on a robotic manipulator performing challenging tasks with various unknown payloads demonstrates the framework's effectiveness in establishing accurate parameter bounds while maintaining safety throughout the identification process. The code is available at our project webpage: https://roahmlab.github.io/OnlineSafeSysID/.**

I. Introduction

Robotic arms have widespread applications in cooperative human-robot environments, including manufacturing, package delivery services, and in-home care. These scenarios frequently involve manipulating payloads with uncertain properties while operating under physical constraints. To ensure safe operation, robots must avoid collisions while respecting joint position, velocity, and torque limits. There are three challenges to the safe deployment of robotic arms handling unknown payloads.

The first challenge lies in bridging the gap between safe planning and safe control. Current research in motion planning and manipulation often overlooks controller tracking performance, where model uncertainties can cause deviations between desired and actual trajectories. These deviations may lead to obstacle collisions or torque limit violations. The second challenge involves accurate estimation of model uncertainties. For unknown payloads, online system identification becomes necessary to estimate the range of inertial parameters. However, larger uncertainty ranges necessitate aggressive controller behavior to guarantee tracking performance, resulting in conservative motion planning which can limit performance. Therefore, obtaining tight bounds on model uncertainties is crucial for ensuring satisfactory task performance. The third challenge concerns safety assurances during system identification. While extensive research exists on exciting trajectory design for data collection and identification accuracy, these studies primarily focus on industrial robots without payloads and do not consider operational safety during data collection. Current approaches rarely address obstacle avoidance or torque limit compliance during the identification process. This is particularly important when considering online operation wherein the model parameters of a payload may not be known before it is manipulated.

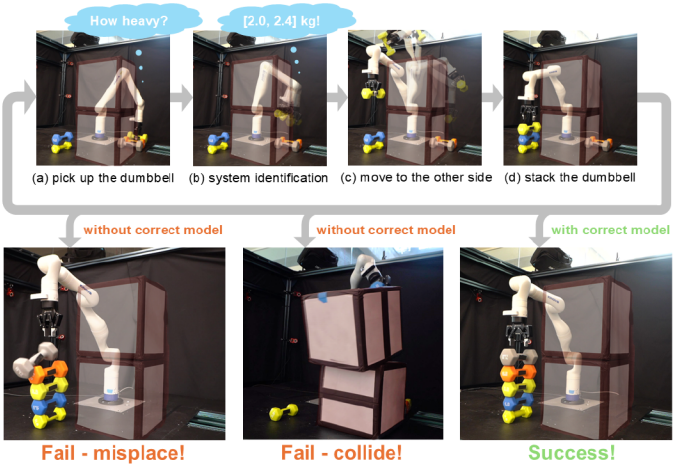

Fig. 1: This figure illustrates how the method proposed in this paper. (a) Initially, the 7 degree-of-freedom Kinova-gen3 robotic arm picks up a series of heavy dumbbells that are close to the design limit of the robot. The inertial parameters of this payload are unknown. (b) The robot then performs online system identification to estimate an interval bound using the method developed in this paper. The interval estimate of the inertial parameters generated by the algorithm is guaranteed to include the true inertial parameters of the dumbbell. Notably, the data used to compute this interval estimate is constructed in a manner that is guaranteed to be collision free while satisfying joint and torque limits. The inertial parameter bound is then used to update the planner and the controller, which allows the robot to (c) safely move all dumbbells to the other side around the obstacles and then (d) stack them vertically in order of increasing weight, which requires high precision. Our experiments illustrate that state-of-the-art methods that do not incorporate such provably overapproximative estimates of the inertial parameters result in a failure to complete the task safely, due to exceeding the torque limits, colliding with obstacles, or misplacing the dumbbells.

This paper presents a provably-safe and real-time system identification framework that addresses these three challenges. As illustrated in Figure 2, the proposed framework comprises two integrated components. Initially, the approach assumes an overapproximated bound on the inertial parameters of the robot end-effector with unknown payloads. A trajectory planning and control framework then generates a provably-safe, locally exciting desired trajectory in real-time based on this initial bound. This exciting trajectory can be tracked in a provably safe manner. Using this collected data, the robust system identification method generates a new, tighter, provably overapproximative bound on the end-effector's inertial parameters. This process iterates continuously, with additional

[1]Robotics Institute, University of Michigan, Ann Arbor, MI <jimzhang, zichang, ramv>@umich.edu.
This work was supported by AFOSR MURI FA9550-23-1-0400.

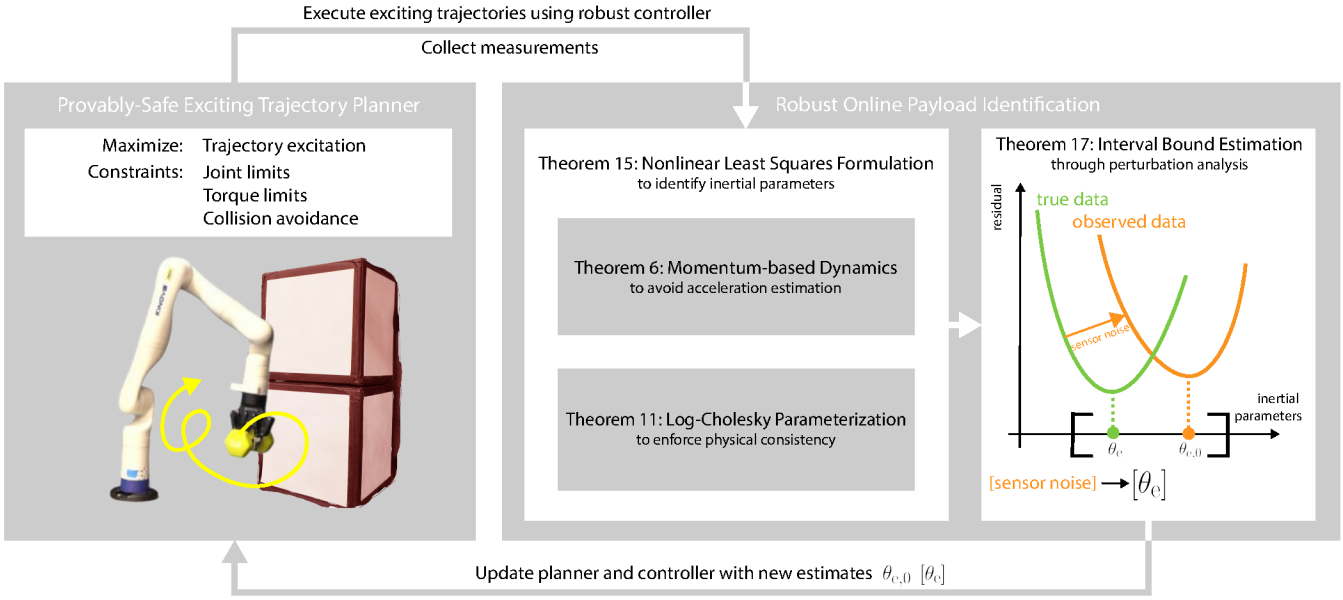

Fig. 2: This figure summarizes the proposed framework. Initially, the approach assumes an overapproximated bound on the inertial parameters of the robot end-effector with unknown payloads. A trajectory planner (Section V) then generates a provably-safe, locally exciting desired trajectory in real-time based on this initial bound. A robust controller, modified from [28, Section VII], tracks this exciting trajectory while collecting robot data including joint positions, velocities, and applied torques. Using this collected data, the robust system identification method (Section IV) generates a new, tighter overapproximated bound on the end-effector's inertial parameters. This process iterates continuously, with additional data enabling more precise parameter estimation and improved planner and controller performance.

data enabling more precise parameter estimation and improved planner and controller performance.

The key contributions of this paper are two-fold: First, a real-time, provably-safe trajectory optimization framework that generates locally exciting trajectories for system identification while respecting robot limits and avoiding obstacles. Second, a robust system identification method that provides an overapproximated bound of the end-effector's inertial parameters given an overapproximated bound of sensor noise.

The remainder of this paper is organized as follows: Section II reviews related work. Section III introduces relevant notation, mathematical objects, and robot kinematics and dynamics. The paper then presents the framework's components in reverse order: Section IV proposes the system identification method based on momentum regressor, sensitivity analysis, and interval arithmetic. Section V details the trajectory optimization problem formulation. Finally, Section VI demonstrates the method's efficacy through real-world experiments and comparisons with state-of-the-art approaches.

II. Related Work

The identification of uncertain payloads attached to the end-effector of robotic manipulators leverages the fundamental property that inertial parameters exhibit linearity in the equations of motion [26], enabling the use of least-squares estimation methods [3] and, more robustly, weighted least-squares techniques [6, 50]. Estimated payload parameters are widely applied in various domains, including precise control [21] and collision detection [14]. Recent advancements have explored alternative formulations of system dynamics based on momentum [31] or energy [36], avoiding estimation of acceleration. Beyond joint encoders, proprioceptive sensors mounted on the end-effector have been employed to estimate payload properties using filtering-based methods [23]. Meanwhile, deep learning approaches have gained traction, leveraging neural networks to learn system dynamics directly from system state data [10, 17] or from pure vision input [22].

Another important direction relevant to both identification and control of unknown systems is adaptive control. Early adaptive controllers aimed to compensate for nonlinear dynamics while decoupling joint interactions [42, 53]. Subsequent theoretical developments leveraged the linear parameterization of robot dynamics and Lyapunov-based stability analysis to ensure convergence guarantees [38]. The introduction of parameter error dynamics through dynamic filtering and torque prediction [39] enabled online parameter adaptation with global stability properties. More recent work further advanced this approach by leveraging momentum-based regressors [30], thereby eliminating the need for acceleration estimation. Despite these advancements, adaptive control still faces important limitations. Performance remains highly dependent on persistent excitation, which affects both the convergence speed and steady-state accuracy. More critically, classical adaptive control methods typically do not offer explicit guarantees on convergence rate for tracking errors. Additionally, they are unable to enforce state and input constraints. Although recent research [25] has attempted to address these issues, the proposed solutions are limited to systems with just a single linear actuators and does not generalize to robots governed by nonlinear rigid body dynamics.

For all of these aforementioned methods, the design of proper experiments for data collection, known as exciting trajectory design, has emerged as a critical aspect of system identification [13, 45, 4, 33, 7]. Early methods focused on minimizing the condition number of the regressor matrix [13], though later work demonstrated greater effectiveness in optimizing the condition number of sub-regressors [45]. Most approaches employ trigonometric series (Fourier series) [19] or B-splines [34] as trajectory bases to facilitate the enforcement of state and velocity limits, while some methods explore deep learning-based techniques to maximize the Fisher information of the payload properties [27]. However, safety guarantees during the identification process are largely unexplored and can be categorized into two main challenges: (1) Generating a safe exciting trajectory and (2) safely tracking a given trajectory. To address the first challenge, state of the art approaches enforce only joint and velocity limits. This may still produce trajectories that violate torque limits. To the author's best knowledge only [11] addresses the challenge of generating an exciting trajectory while satisfying safety constraints. However, [11] employs sampling-based planners to generate collision-free trajectories which are unable to guarantee safety between sampled points. To address the second challenge, state of the art approaches rely on PID controllers [19] or manufacturer-embedded controllers [14]. As a result, trajectory or controller fine-tuning is often required to balance the trade-off between respecting torque limits and achieving low tracking errors.

To overcome these challenges, the concept of closed-loop state/input sensitivity metric was introduced in [15] to evaluate the impact of model uncertainties on robot behavior. By designing an optimized feedforward reference trajectory that minimizes these sensitivity metrics, robustness against model uncertainties is inherently embedded in the reference trajectory itself, eliminating the need for specific control strategies [41]. This approach has been applied to UAVs [15] and quadrotors [8, 48], providing a trajectory planning and control framework that considers state/torque limits and obstacle avoidance. Recent developments have extended this method to robotic manipulators [41]. However, this work only considers a narrow range of payload parameters [41, Section IV] and has not yet been integrated with collision avoidance.

III. Preliminaries

This section establishes the mathematical foundations necessary for developing the safe system identification framework.

A. Interval Arithmetic

To handle uncertainty in robot inertial parameters, this paper employs multidimensional intervals and interval arithmetic. For vectors in $\mathbb{R}^n$, we first establish notation: x_i denotes the i^{th} component of vector x, and for vectors $x, y \in \mathbb{R}^n$, we write $x \preceq y$ when $x_i \leq y_i$ holds for all components i.

Definition 1 (Multidimensional Interval). *For vectors $\underline{x}, \bar{x} \in \mathbb{R}^n$, a multidimensional interval is the set:*

$$[\underline{x}, \bar{x}] := \{x \in \mathbb{R}^n \mid \underline{x} \preceq x \preceq \bar{x}\}, \tag{1}$$

where $\underline{x}$ and $\bar{x}$ serve as the interval's infimum and supremum, respectively.

The set of all n-dimensional intervals is denoted as $\mathbb{IR}^n$. For analyzing functions over intervals, we introduce interval evaluation:

Definition 2 (Interval Evaluation). *For a function $f : \mathbb{R}^n \to \mathbb{R}^m$ and interval $[\underline{x}, \bar{x}]$, the interval evaluation is defined as:*

$$f([\underline{x}, \bar{x}]) := \{f(x) \mid x \in [\underline{x}, \bar{x}]\}. \tag{2}$$

B. Robotic Manipulator Model and the Environment

1) Dynamics Parameters: For a fully-actuated robotic system with n joints, let $q : [0, \infty) \in \mathcal{Q}$ denote the generalized trajectory, where $\mathcal{Q} \subset \mathbb{R}^n$ represents the robot configuration space. For convenience, we let $N_q = \{1, \ldots, n\}$. The system's behavior is characterized by its *dynamics parameters*, $\theta \in \mathbb{R}^{14n}$, which comprise two distinct components: inertial parameters and friction parameters. The inertial parameters $\theta_{\text{ip}} \in \mathbb{R}^{10n}$ describe the mass properties of each link j:

$$\theta_{\text{ip},j} = [m_j, p_{x,j} \cdot m_j, p_{y,j} \cdot m_j, p_{z,j} \cdot m_j, XX_j, YY_j, ZZ_j, XY_j, YZ_j, XZ_j]^T, \tag{3}$$

where m_j represents mass, $[p_{x,j} \cdot m_j, p_{y,j} \cdot m_j, p_{z,j} \cdot m_j]$ encodes the first moments of mass, and the remaining six components define the inertia tensor. Here, $p_j = [p_{x,j}, p_{y,j}, p_{z,j}]^T$ represents the center of mass for link $j \in \{1, \ldots, n\}$. The friction parameters $\theta_{\text{f}} \in \mathbb{R}^{4n}$ capture joint friction characteristics:

$$\theta_{\text{f},j} = [F_{c_j}, F_{v_j}, I_{a_j}, \beta_j]^T, \tag{4}$$

where these components model static friction, viscous friction, transmission inertia, and bias for each joint j.

This work addresses the identification of end-effector dynamics parameters, including attached payloads, while accounting for uncertainties in the complete robotic system. The dynamic parameters naturally partition into two components: the end-effector parameters $\theta_{\text{e}} \in \mathbb{R}^{10}$ and the remaining robot parameters $\theta_{\text{r}} \in \mathbb{R}^{14n-10}$. In practical applications, our objective extends beyond point estimates to establishing conservative bounds on the end-effector's dynamic parameters that provably contain the true parameters. This challenge is compounded by uncertainties in the base robot's parameters, leading to the following assumption:

Assumption 3 (Dynamics Parameter Bounds). *The robot dynamic parameters excluding the end-effector, θ_r, lie within a known finite interval $[\theta_r]$ containing the true parameters θ_r. Furthermore, a nominal estimate $\theta_{r,0} \in [\theta_r]$ of these parameters is available.*

This assumption aligns with practical robotics applications, as manufacturers typically provide baseline kinematic and inertial parameters for each robot link, including the unloaded end-effector. Friction parameters can be characterized through established methods when the robot operates without a payload, following approaches detailed in [52] and [19]. The interval bounds $[\theta_{\text{r}}]$ can then be constructed by expanding around

the nominal parameters $\theta_{r,0}$ to account for manufacturing uncertainties.

2) Equation of Motion: The *classical equation of motion* takes the form:

$$H(q(t),\theta)\ddot{q}(t) + C(q(t),\dot{q}(t),\theta)\dot{q}(t) + g(q(t),\theta) + F(\dot{q}(t),\ddot{q}(t),\theta) = \tau(t), \quad (5)$$

where $H(q(t),\theta) \in \mathbb{R}^{n\times n}$ is the positive definite *inertia mass matrix*, $C(q(t),\dot{q}(t),\theta) \in \mathbb{R}^{n\times n}$ is the *Coriolis matrix*, $g(q,\theta) \in \mathbb{R}^n$ is the *gravitational force vector*, and $\tau(t) \in \mathbb{R}^n$ is applied joint torques all at time t. The *friction model* $F(\dot{q}(t),\ddot{q}(t),\theta) \in \mathbb{R}^n$, follows [19, (2)]:

$$F(\dot{q}(t),\ddot{q}(t),\theta) = F_c \circ \operatorname{sign}(\dot{q}(t)) + F_v \circ \dot{q}(t) + I_a \circ \ddot{q}(t) + \beta, \quad (6)$$

where $F_c \in \mathbb{R}^n$ is the *static friction coefficients*, $F_v \in \mathbb{R}^n$ is the *viscous friction coefficients*, $I_a \in \mathbb{R}^n$ is the *transmission inertias*, $\beta \in \mathbb{R}^n$ is the *offset/bias terms* of all joints, and $\circ$ denotes the Hadamard product. In addition to this model, we assume that the robot must satisfy several limits.

Assumption 4 (Workspace and Configuration Space)**.** *The robot's j^{th} joint has position and velocity limits given by $q_j \in [q^-_{j,\text{lim}}, q^+_{j,\text{lim}}]$ and $\dot{q}_j \in [\dot{q}^-_{j,\text{lim}}, \dot{q}^+_{j,\text{lim}}]$ for all time and for all $j \in N_q$, respectively. During online operation, the robot has encoders that allow it to measure its joint positions and velocities. The robot's input τ has limits given by $\tau_j(t) \in [\tau^-_{j,\text{lim}}, \tau^+_{j,\text{lim}}]$ for all time and $j \in N_q$.*

The classical equations of motion (5) can be reformulated using a momentum-based approach, which offers practical advantages for system identification:

Theorem 5 (Momentum-Based Dynamics [31, (5)])**.** *For a robotic system with dynamics described by* (5)*, define the system momentum for any time t as:*

$$p(q(t),\dot{q}(t),\theta) = H(q(t),\theta)\dot{q}(t). \quad (7)$$

Then

$$\dot{p}(q(t),\dot{q}(t),\theta) = C^T(q(t),\dot{q}(t),\theta)\dot{q}(t) - g(q(t),\theta) - F(\dot{q}(t),\ddot{q}(t),\theta) + \tau(t), \quad (8)$$

and for any time interval $[t_1, t_2]$ with $0 \le t_1 < t_2$, the change in momentum satisfies:

$$p(q(t_2),\dot{q}(t_2),\theta) - p(q(t_1),\dot{q}(t_1),\theta) = \int_{t_1}^{t_2} (C^T(q(t),\dot{q}(t),\theta)\dot{q}(t) - g(q(t),\theta) - F(\dot{q}(t),\ddot{q}(t),\theta) + \tau(t))dt, \quad (9)$$

This momentum-based dynamics formulation provides two key advantages: it eliminates the need for acceleration measurements, which are often noisy or unavailable in practice, and it expresses the dynamics in an integral form that filters measurement noise.

3) Environment: We begin by defining obstacles:

Assumption 6 (Obstacles)**.** *The transformation between the world frame of the workspace and the base frame of the robot is known, and obstacles are expressed in the base frame of the robot. At any time, the number of obstacles $n_O \in \mathbb{N}$ in the scene is finite ($n_O < \infty$). Let O be the set of all obstacles $\{O_1, O_2, \ldots, O_{n_O}\}$. Each obstacle is bounded and static with respect to time. The manipulator has access to a zonotope overapproximation [18] of each obstacle's volume in workspace.*

The manipulator is *in collision* with an obstacle if $\text{FO}_j(q(t)) \cap O_i \neq \emptyset$ for any $j \in N_q$ or $i \in \{1,\ldots,n_O\}$, where FO_j describes the forward occupancy of the j^{th} link of the manipulator. This work is concerned with planning and control around obstacles while performing system identification, not perception of obstacles. As a result, we have assumed for convenience that the robot is able to sense all obstacles in the workspace.

C. Dynamics Regressors

To enable system identification of the dynamics parameters θ, we introduce two regressor formulations that each expose the linear relationship between system dynamics and these parameters. These regressors provide different perspectives on the system dynamics and serve distinct purposes in controller design and system identification.

Theorem 7 (Standard Dynamics Regressor [12, (12)])**.** *For any time t, the left hand side of the classical equation of motion* (5) *can be expressed as a linear function of the dynamic parameters:*

$$W(q(t),\dot{q}(t),\ddot{q}(t))\theta = \tau(t). \quad (10)$$

We refer to $W(q(t),\dot{q}(t),\ddot{q}(t)) \in \mathbb{R}^{n\times 14n}$ as the standard dynamics regressor [26].

With this linear relationship, a least squares problem can be formulated based on (10) to identify the dynamic parameters [19, (9)]. The momentum-based dynamics (Theorem 5) preserves the similar linear relationship.

Theorem 8 (Momentum-Based Regressors [31, (8)] and [47, Appendix])**.** *The momentum dynamics admit two specialized regressors for any time t:*

I. The system momentum (9) *can be expressed as:*

$$p(q(t),\dot{q}(t),\theta) = W_m(q(t),\dot{q}(t))\theta, \quad (11)$$

where $W_m(q(t),\dot{q}(t)) \in \mathbb{R}^{n\times 14n}$ is the momentum regressor.

II. The momentum derivative (8) *without torque input can be expressed as:*

$$C^T(q(t),\dot{q}(t),\theta)\dot{q}(t) - g(q(t),\theta) - F(\dot{q}(t),\ddot{q}(t),\theta) = W_c(q(t),\dot{q}(t))\theta - I_a \circ \ddot{q}(t), \quad (12)$$

where $W_c(q(t),\dot{q}(t)) \in \mathbb{R}^{n\times 14n}$.

Combining these regressors yields the integral form for $t_2 > t_1$:

$$(W_m(q(t_2),\dot{q}(t_2)) - W_m(q(t_1),\dot{q}(t_1)))\theta = \left(\int_{t_1}^{t_2} W_c(q(t),\dot{q}(t))dt\right)\theta - I_a \circ (\dot{q}(t_2) - \dot{q}(t_1)) + \int_{t_1}^{t_2} \tau(t)dt \quad (13)$$

Similarly, a least squares problem can be formulated using (13) to identify the dynamic parameters [31, (16)]. However, in this instance, one can rely on just position and velocity measurements rather than acceleration measurements. Because this work focuses on identification of the end-effector, we separate the regressors into two parts. In other words, we decompose W_m into two components, $W_{m,r}(q(t),\dot{q}(t)) \in \mathbb{R}^{n\times(14n-10)}$ and $W_{m,e}(q(t),\dot{q}(t)) \in \mathbb{R}^{n\times 10}$, where $W_{m,r}$ corresponds to the columns of W_m associated with θ_r and $W_{m,e}$ corresponds to the columns of W_m associated with θ_e. We similarly decompose W_c into $W_{c,r}$ and $W_{c,e}$.

D. Physical Consistency Constraints and Parameterization

Using the regressors described in the previous subsection, the system identification problem can be formulated as a linear regression task. However, to ensure the physical consistency of the inertial parameters of each link of the robot, additional mathematical constraints must be imposed [40]. These constraints, also known as LMI constraints and defined in Definition 15, require that the pseudo-inertia matrix (31), which encodes the inertial parameters of a robot link, be positive-definite. Consequently, the system identification problem becomes a constrained optimization problem [40, (20)], where physical consistency is explicitly enforced.

An alternative approach to enforcing positive-definiteness is to utilize the Cholesky decomposition of the pseudo-inertia matrix [35]. Specifically, this structure of $\theta_{\mathrm{ip},j}$ can be fully described by

$$\theta_{\mathrm{ip},j} = P(\eta), \quad (14)$$

where the *log-Cholesky parameters* $\eta \in \mathbb{R}^{10}$ are defined as

$$\eta = \begin{bmatrix}\alpha & d_1 & d_2 & d_3 & s_{12} & s_{13} & s_{23} & s_{14} & s_{24} & s_{34}\end{bmatrix}^T, \quad (15)$$

and the mapping $P : \mathbb{R}^{10} \to \mathbb{R}^{10}$ is given by

$$P(\eta) = e^{2\alpha}\begin{bmatrix} s_{14}^2 + s_{24}^2 + s_{34}^2 + 1 \\ s_{14}e^{d_1} \\ s_{14}s_{12} + s_{24}e^{d_2} \\ s_{14}s_{13} + s_{24}s_{23} + s_{34}e^{d_3} \\ s_{12}^2 + s_{13}^2 + s_{23}^2 + e^{2d_2} + e^{2d_3} \\ s_{13}^2 + s_{23}^2 + e^{2d_1} + e^{2d_3} \\ s_{12}^2 + e^{2d_1} + e^{2d_2} \\ -s_{12}e^{d_1} \\ -s_{12}s_{13} - s_{23}e^{d_2} \\ -s_{13}e^{d_1} \end{bmatrix}. \quad (16)$$

Further details regarding this formulation and its properties are provided in Appendix A.

IV. Robust, Online Identification For Payloads

This work focuses on the identification of the inertial parameters of the end-effector link, including the payload using the model and assumptions described in Section III-B. The rest of the section discusses how to accurately estimate $\theta_{\mathrm{e},0}$ with a conservative interval bound $[\theta_\mathrm{e}]$ that includes the true θ_e through a system identification process.

A. System Identification Problem Formulation

This subsection develops the mathematical framework for identifying end-effector parameters from measurement data. The approach leverages momentum-based dynamics and builds toward an optimization problem that ensures physical consistency.

Definition 9 (Measurement Data)**.** *For system identification, we consider N sequential measurements of the robot state, collected at times $t_i \geq 0$ for $i \in 1,\dots,N$ that we refer to as the* measurement data*:*

$$\mathbf{q}_{1:N} = \{q(t_i)\}_{i=1}^N \quad (17)$$

$$\dot{\mathbf{q}}_{1:N} = \{\dot{q}(t_i)\}_{i=1}^N \quad (18)$$

$$\boldsymbol{\tau}_{1:N} = \{\tau(t_i)\}_{i=1}^N. \quad (19)$$

Using this measurement data and the momentum regressor described in Theorem 8, one can prove the following corollary:

Corollary 10 (Momentum-Based Parameter Identification)**.** *Let $h \in \mathbb{N}^+$ denote the forward integration horizon. If one applies Forward Euler Intergration, then the momentum dynamics over interval $[t_i, t_{i+h}]$ yields a linear relationship with respect to the end-effector's dynamics parameters:*

$$Y_{i:i+h}\theta_e = U_{i:i+h}, \quad (20)$$

where the regressor matrix $Y_{i:i+h}$ captures the end-effector dynamics:

$$Y_{i:i+h} = (W_{m,e}(q(t_{i+h}),\dot{q}(t_{i+h})) - W_{m,e}(q(t_i),\dot{q}(t_i))) - \sum_{j=i}^{i+h} W_{c,e}(q(t_j),\dot{q}(t_j))(t_{j+1}-t_j), \quad (21)$$

and $U_{i:i+h}$ describes the rest of the system's dynamics:

$$U_{i:i+h} = -(W_{m,r}(q(t_{i+h}),\dot{q}(t_{i+h})) - W_{m,r}(q(t_i),\dot{q}(t_i)) + \sum_{j=i}^{i+h} W_{c,r}(q(t),\dot{q}(t))(t_{j+1}-t_j))\theta_r - I_a \circ (\dot{q}(t_{i+h}) - \dot{q}(t_i)) + \sum_{j=i}^{i+h} \tau(t_j)(t_{j+1}-t_j). \quad (22)$$

By applying this corollary, one can show that the system identification problem can be formulated as the solution to an optimization problem when there is no uncertainty in the robot dynamics:

Theorem 11 (Uncertainty-Free System Identification)**.** *Let $h \in \mathbb{N}^+$ denote the forward integration horizon, suppose*

that measurement data was generated by the system dynamics while satisfying forward Euler Integration, and suppose that the robot's dynamic parameters excluding the end-effector are known exactly and are equal to $\theta_{r,0}$. Let the observation matrix $\mathbf{Y}$ *be defined as follows:*

$$\mathbf{Y}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}) := \begin{bmatrix} Y_{1:1+h} \\ Y_{1+h:1+2h} \\ \vdots \\ Y_{1+(N_h-1)h:1+N_h h} \end{bmatrix} \in \mathbb{R}^{N_h n \times 10}, \quad (23)$$

and let the observation response $\mathbf{U}$ *be defined as follows:*

$$\mathbf{U}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}, \boldsymbol{\tau}_{1:N}, \theta_r) := \begin{bmatrix} U_{1:1+h} \\ U_{1+h:1+2h} \\ \vdots \\ U_{1+(N_h-1)h:1+N_h h} \end{bmatrix} \in \mathbb{R}^{N_h n}, \quad (24)$$

where $N_h = \lfloor \frac{N}{h} \rfloor$ is the total number of integration steps. Consider the following optimization problem:

$$\min_{\eta_e \in \mathbb{R}^{10}} \|\mathbf{Y}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}) P(\eta_e) - \mathbf{U}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}, \boldsymbol{\tau}_{1:N}, \theta_{r,0})\|, \quad (25)$$

where P is the Log-Cholesky parameterization defined in (16)*. Let η_e^* denote any local minimizer to this optimization problem and let $\theta_e^* = P(\eta_e^*)$, then $\mathbf{Y}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N})\theta_e^* = \mathbf{U}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}, \boldsymbol{\tau}_{1:N}, \theta_{r,0})$.*

Proof: This result follows by first recognizing that in the presence of no uncertainty, one can formulate the system identification problem as:

$$\begin{aligned} \min_{\theta_e \in \mathbb{R}^{10}} \quad & \|\mathbf{Y}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N})\theta_e - \mathbf{U}(\mathbf{q}_{1:N}, \dot{\mathbf{q}}_{1:N}, \boldsymbol{\tau}_{1:N}, \theta_{r,0})\| \\ \text{s.t.} \quad & \text{LMI}(\theta_e) \succ \mathbb{0}_{4\times 4}, \end{aligned} \quad (26)$$

where have applied Definition 15 and Corollary 10. The result then follows by applying Corollary 17. ■

The aforementioned Theorem makes several assumptions that make its practical application challenging. First, as mentioned in Section III-B, usually the robot's dynamic parameters excluding the end-effector are not known exactly. Second, the measurement data may not be noise free. Finally, the measurement data may not be generated by performing Forward Euler Integration. The next subsection describes how we deal with the first two challenges while generating a conservative estimate of the end-effector's dynamics parameters.

B. Interval Bound Estimation via Perturbation Analysis

To construct a conservative bound on the end-effector's dynamics parameters, we begin by making the following assumptions regarding the noise in the measurement:

Assumption 12. *The joint position and velocity measurements are perfectly accurate. The torque measurement is noisy, but its uncertainty can be bounded by an interval vector that we denote by $[\boldsymbol{\tau}_{1:N}] \in \mathbb{IR}^{Nn}$.*

To understand why this assumption is reasonable, recall that the joint position and velocity of robotic arms are usually captured by optical encoders [2]. The measurement error for such sensors arises due to the resolution of the coded disk inside the sensor [24, Section 2.4]. The noise in such sensors is small in modern robots [32, Section IV]. On the other hand, the torque sensor is usually based on electrical and optical techniques [44] that can introduce a non-negligible error. As a result, our system identification techniques focus on dealing with the measurement error of the torque sensor $\boldsymbol{\tau}_{1:N}$ and the uncertainty associated with the dynamics parameters of the robot $\theta_{\text{r},0}$ (excluding the end-effector).

The goal of this subsection is to understand how these various forms of uncertainty impact the system identification process. Using the aforementioned assumption, we simplify our notation to focus on this objective. We concatenate the variables $\boldsymbol{\tau}_{1:N}$ and $\theta_{\text{r},0}$ that affect the estimation and denote this as a measurement vector $\mathbf{m} \in \mathbb{R}^{Nn+14n-10}$. We denote the uncertainty associated with this measurement vector by $[\mathbf{m}] \in \mathbb{IR}^{Nn+14n-10}$, which can be computed by applying Assumptions 3 and 12. Because the focus of this subsection is understanding how uncertainty in this measurement vector impacts the estimation of the end-effector's dynamics parameters, we denote the observation matrix by $\mathbf{Y}$ and the observation response matrix by $\mathbf{U}(\mathbf{m})$.

Using these definitions, we can conservatively bound the end-effector's dynamics parameter estimates that one generates by applying Theorem 11:

Theorem 13. *Let $\eta_e^*(\mathbf{m})$ be any local minimizer to* (25)*. Let $\theta_e^*(\mathbf{m}) = P(\eta_e^*(\mathbf{m}))$, where P is the Log-Cholesky parameterization defined in* (16)*. Then the true dynamics parameters of the end-effector satisfy the following inclusion*

$$\theta_e \in \theta_e^*(\mathbf{m}) + \frac{\partial \theta_e^*}{\partial \mathbf{m}}([\mathbf{m}])([\mathbf{m}] - \mathbf{m}). \quad (27)$$

The proof to Theorem 13 and the formula for $\frac{\partial \theta_e^*}{\partial \mathbf{m}}(\mathbf{m})$ are provided in Appendix C.

Algorithm 1 describes how to perform system identification using Theorem 13 given data and an initial estimate for the dynamics parameters of the robot excluding the end-effector. Note that the output of the algorithm does not require an initial estimate of the dynamics parameters of the end-effector. The algorithm outputs an estimate for the dynamics parameters of the entire robot along with a conservative interval-based bound on the dynamics parameters of the entire robot.

V. Provably-Safe Locally Exciting Trajectory Generation

The objective of this paper is to perform system identification during online operation while ensuring safety. The previous section describes how to apply optimization to the collected data to conservatively identify the dynamics parameters of the end-effector. However, it does not describe how to collect the data in a safe manner. To solve this problem, this paper relies on Autonomous Robust Manipulation via Optimization with Uncertainty-aware Reachability (ARMOUR) [28], which is an optimization-based motion planning and control framework that can ensure the safety of synthesized

Algorithm 1 $(\theta_0, [\theta]) = \textbf{SysID}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \theta_{\mathrm{r},0}, [\theta_\mathrm{r}], h, [\mathbf{m}])$

$\boldsymbol{m} \leftarrow (\boldsymbol{\tau}, \theta_{\mathrm{r},0})$
$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}) \leftarrow$ (23)
$\mathbf{U}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{m}) \leftarrow$ (24)
$\eta_e^*(\boldsymbol{m}) \leftarrow$ (25)
$\theta_e^*(\boldsymbol{m}) \leftarrow P(\eta^*(\boldsymbol{m}))$
$[\theta_\mathrm{e}] \leftarrow \theta_e^*(\mathbf{m}) + \frac{\partial \theta_e^*}{\partial \mathbf{m}}([\mathbf{m}])([\mathbf{m}] - \mathbf{m})$ using Theorem 13
$\theta_0 \leftarrow (\theta_{\mathrm{r},0},\ \theta_e^*(\boldsymbol{m}))$
$[\theta] \leftarrow ([\theta_\mathrm{r}],\ [\theta_\mathrm{e}])$
Return: $(\theta_0,\ [\theta])$

trajectories even in the presence of model uncertainty. This section provides a brief overview of ARMOUR and how it must be modified to design trajectories that can make the identification of the true dynamics parameters of the end-effector occur more rapidly.

A. An Overview of ARMOUR [28]

ARMOUR plans safe trajectories in a receding horizon fashion that minimize a user-specified cost. To accomplish this goal, ARMOUR optimizes over a space of possible desired trajectories that are chosen from a prespecified continuum of trajectories, with each determined by a *trajectory parameter*, $k \in \mathcal{K}$. During each planning iteration, ARMOUR selects a trajectory parameter that can be followed without collisions despite tracking error and model uncertainty while satisfying joint and input limits. To ensure that ARMOUR is capable of planning in real time, each desired trajectory is followed while the robot constructs the next desired trajectory for the subsequent step. Next, we summarize how ARMOUR accomplishes this goal by selecting a feedback controller and a planning time and describing what trajectory optimization problem it solves.

1) Feedback Controller: ARMOUR associates a feedback control input over a compact time interval $T = [0, t_f] \subset \mathbb{R}$ with each trajectory parameter $k \in \mathcal{K}$. Here $\mathcal{K}$ represents a parameterized space that includes a variety of behaviors of the robot. This feedback control input is a function of the nominal inertial parameters θ_0, the interval inertial parameters $[\theta]$, and the state of the robot; however, it cannot be a function of the true inertial parameters of the robot because these are not known. To simplify notation, we denote the feedback control input at time $t \in T$ under trajectory $k \in \mathcal{K}$ by $\tau(t; k)$. Applying this control input to the arm generates an associated trajectory of the arm. These position and velocity trajectories are functions of the true inertial parameters. We denote the position and velocity trajectories at time $t \in T$ under trajectory $k \in \mathcal{K}$ under the true inertia model parameters $\theta \in [\theta]$ by $q(t; k, \theta)$ and $\dot{q}(t; k, \theta)$, respectively.

2) Timing: Because ARMOUR performs receding horizon planning, we assume without loss of generality that the control input and trajectory begin at time $t = 0$ and end at a fixed time t_f. To ensure real-time operation, ARMOUR identifies a new trajectory parameter within a fixed planning time of t_p seconds, where $t_p < t_f$. ARMOUR must select a new trajectory parameter before completing its tracking of the previously identified desired trajectory. If no new trajectory parameter is found in time, ARMOUR defaults to a braking maneuver that brings the robot to a stop at time $t = t_f$.

3) Online Trajectory Optimization: During each receding-horizon planning iteration, ARMOUR generates a trajectory by solving a tractable representation of the following nonlinear optimization:

$$\begin{aligned}
\min_{k \in K} \quad & \texttt{cost}(k) && (28)\\
& q_j(t; k, \theta) \in [q_{j,\mathrm{lim}}^-, q_{j,\mathrm{lim}}^+] && \forall t \in T, \theta \in [\theta], j \in N_q\\
& \dot{q}_j(t; k, \theta) \in [\dot{q}_{j,\mathrm{lim}}^-, \dot{q}_{j,\mathrm{lim}}^+] && \forall t \in T, \theta \in [\theta], j \in N_q\\
& \tau_j(t; k) \in [\tau_{j,\mathrm{lim}}^-, \tau_{j,\mathrm{lim}}^+] && \forall t \in T, \theta \in [\theta], j \in N_q\\
& \mathrm{FO}_j(q(t; k, \theta)) \bigcap O = \emptyset && \forall t \in T, \theta \in [\theta], j \in N_q
\end{aligned}$$

The cost function (28) specifies a user-defined objective, such as bringing the robot close to some desired goal. Each of the constraints guarantee the safety of any feasible trajectory parameter as we describe next. The trajectory must be executable by the robot, which means the trajectory must not violate the robot's joint position, velocity, or input limits (i.e., the first three constraints in the optimization problem, respectively). These constraints must be satisfied for each joint over the entire planning horizon despite model uncertainty; The robot must not collide with any obstacles in the environment (i.e., the last constraint in the optimization problem). ARMOUR solves this optimization problem in a provably safe fashion in real-time despite model uncertainty [28, Lemma 22].

We make one final observation about ARMOUR. Recall that during receding horizon planning, ARMOUR must select a new trajectory parameter before completing its tracking of the previously identified desired trajectory. Note that to do this, naively one may assume that one needs to know the future state of the manipulator to compute a trajectory parameter using ARMOUR. Because there is uncertainty in the system model, knowing the future state of the robot perfectly is untenable. Fortunately, ARMOUR does not require knowledge of the future state of the manipulator to ensure that the robot stays persistently safe. In fact, to ensure that the manipulator is persistently safe using ARMOUR in a receding horizon fashion, one only needs to know the previously computed ARMOUR trajectory parameter [28, Remark 12].

In the remainder of the paper, `ARMOUR` represents a function that solves the optimization problem (28), denoted as:

$$k^* = \texttt{ARMOUR}(k_p, O, \texttt{cost}, t_p, T, \theta_0, [\theta]), \qquad (29)$$

where k^* represents the optimal trajectory parameters, and the inputs specify the previously computed trajectory parameter k_p, obstacle set O, objective function `cost`, planning time t_p, trajectory duration T, nominal dynamics parameters θ_0, and dynamics parameter bounds $[\theta]$. Note during the first planning iteration, one can just pass in the actual initial state of the robot instead of k_p.

B. Modify Cost Function for Locally Exciting Trajectories

This subsection describes how to design ARMOUR's cost function to generate exciting trajectories which can expedite the rate at which the dynamics parameters of the end-effector are identified. To appreciate why this is important, note that large model uncertainty may require an overly conservative representation of the forward occupancy while necessitating larger control effort to ensure safety. This could impede completing subsequent planning tasks. We illustrate these challenges with poor system identification in the experiments as is described Section VI.

By looking at (27), one can determine that the size of $[\theta_{\mathrm{e}}]$ depends on the measurements uncertainties. Besides, the size of $[\theta_{\mathrm{e}}]$ also depends on the inverse of matrix $\frac{\partial^2 J}{\partial \eta_e^2}(\mathbf{m}, \eta_e^*(\mathbf{m}))$, which depends on the matrix $\mathbf{Y}^T\mathbf{Y}$, as shown in (42) and (43). Unfortunately, in certain cases $\mathbf{Y}^T\mathbf{Y}$ may be close to singular, which yields a large bound for $[\theta_{\mathrm{e}}]$. As a result, we would like to generate trajectories for the robot to follow that minimize the condition number of $\mathbf{Y}^T\mathbf{Y}$. Such desired trajectories are usually referred to as *exciting trajectories*. To be more specific, we consider the 2-norm condition number of $\mathbf{Y}^T\mathbf{Y}$, which is essentially the ratio between the largest and the smallest singular values of $\mathbf{Y}^T\mathbf{Y}$ [5, (5)].

However, recalling the integral nature of the momentum regressor $\mathbf{Y}$ described in (21), it is computationally expensive to optimize the condition number of $\mathbf{Y}^T\mathbf{Y}$ because the gradient of the regressor matrix is required. In this work, we instead minimize the condition number of the portion of the Standard Dynamics Regressor (Theorem 7) associated with the dynamics parameters of the end-effector. Note that this is the approach taken by most prior work to generate exciting trajectories [13, 45, 4, 33, 7].

To be specific, consider the following regressor matrix:

$$\mathbf{W}(k) := \begin{bmatrix} W_e(q(t_1;k), \dot{q}(t_1;k), \ddot{q}(t_1;k)) \\ \vdots \\ W_e(q(t_{N_s};k), \dot{q}(t_{N_s};k), \ddot{q}(t_{N_s};k)) \end{bmatrix}, \quad (30)$$

where W_e is the collection of columns corresponding to the dynamics parameters of the end-effector within (10), $t_i \geq 0$ for $i \in \{1, \ldots, N_s\}$ are the time instances of each sample of the desired trajectory, and N_s is the number of samples. To compute locally exciting trajectories, we set the cost function in ARMOUR equal to the condition number of $\mathbf{W}(k)$.

To see why this works, recall that the original robot dynamics (5) is equivalent to the momentum version of the dynamics (13) by taking time derivatives on both sides. Hence, $\mathbf{W}$ can be thought of as approximating the time derivative of $\mathbf{Y}$, as they contribute linearly to (5) and (13), respectively. Unfortunately, it is non-trivial to analytically prove that minimizing the condition numbers of these two regressor matrices is equivalent. However, during the experiments described in Appendix D, we illustrate that minimizing the condition numbers of these two regressor matrices generates approximately similar behavior.

C. Provably-safe Online System Identification Pipeline

Algorithm 2 Provably-Safe Online System Identification

1: **Require:** $q_0 \in \mathcal{Q}$, O, $t_p > 0$, $[\theta]$, $\theta_0 \in [\theta]$, $h \in \mathbb{N}^+$, $[\mathbf{m}]$
2: **Initialize:** $j = 0$, $t_j = 0$, $\boldsymbol{\tau} = \{\}$, $\boldsymbol{q} = \{\}$, $\dot{\boldsymbol{q}} = \{\}$, and $k_0^* = \texttt{ARMOUR}(q_0, O, \texttt{cost}, t_p, T, \theta_0, [\theta])$
3: **If:** $k_0^* = \texttt{NaN}$, **then** break
4: **while** 1 **do**
5: // Line 6 executes simultaneously with Lines 7 – 10 //
6: **Apply** $\tau(t; k_j^*)$ to robot for $t \in [t_j, t_j + t_p]$ and append input, position, and velocity of robot into $\boldsymbol{\tau}$, $\boldsymbol{q}$, and $\dot{\boldsymbol{q}}$, respectively
7: $k_{j+1}^* = \texttt{ARMOUR}(k_j^*, O, \texttt{cost}, t_p, T, \theta_0, [\theta])$
8: $(\theta_0, [\theta]) = \textbf{SysID}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \theta_{\mathrm{r},0}, [\theta_{\mathrm{r}}], h, [\mathbf{m}])$
9: **If** $k_{j+1}^* = \texttt{NaN}$, **then** break
10: **Else** $t_{j+1} \leftarrow t_j + t_p$ and $j \leftarrow j + 1$
11: **end while**
12: **Apply** $\tau(t; k_j^*)$ to robot for $t \in [t_j, t_j + t_p]$

This subsection describes how to combine ARMOUR with the cost function described in the previous subsection with the system identification procedure described in Theorem 13 to create the provably-safe online system identification algorithm described in Algorithm 2.

To begin, we assume that the robot has picked up a payload, which is then rigidly attached to the end-effector of the robot. We assume that the robot starts at rest from a known initial state, q_0, given known obstacle configurations, O. A user then specifies a planning time horizon, t_p, provides an estimate of the nominal dynamics parameters of the robot, θ_0 along with a conservative interval based bound on the dynamics parameters of the robot, $[\theta]$, selects a forward integration horizon $h \in \mathbb{N}^+$, and specifies a conservative error bound for the measurement vector, $[\boldsymbol{m}]$. Before performing any system identification, ARMOUR first tries to compute an exciting trajectory (Line 2). If no feasible solution is found, then NaN is returned and the robot does not move (Line 3).

On the other hand if a feasible solution can be found, the while loop begins. At this point, the previously computed optimal trajectory is tracked by ARMOUR's feedback controller and the behavior of the robot is saved in the measurement varaibles (Line 6). While this is happening the next trajectory to be tracked is computed by ARMOUR (Line 7) and then system identification is performed using Algorithm 1 (Line 8). This is then repeated, unless ARMOUR is not able to find a feasible solution at which point ARMOUR uses the braking maneuver whose safety was verified in a previous planning iteration to bring the robot to a stop (Line 12). In practice, the while loop is terminated if the the size of the interval estimate does not increase during subsequent iterations.

One can prove that Algorithm 2 generates provably safe behavior while performing system identification. However, before proving that result, we make one observation. Recall that Theorem 13 allows us to estimate conservatively bound the dynamics parameters of the end-effector without starting from an initial conservative estimate of these parameters.

Unfortunately, to collect data to perform system identification using ARMOUR, we must rely on ARMOUR's feedback controller. This controller requires a conservative estimate of the dynamics parameters of the overall robot to ensure safe behavior. However, note that generating a conservative overapproximation of the range of the initial parameters of the end-effector with payload is not challenging. For example, the range for mass could be set from 0 to the maximum payload weight defined by the manufacturer. The center of mass can also be bounded within the geometry of the end-effector and the payload. The range of inertia can also be overapproximated given the range of mass and the center of mass.

Finally by applying [28, Lemma 22] and Theorem 13 one can prove the following result:

Lemma 14 (Algorithm 2 Generates Safe Motion)**.** *Suppose q_0 is collision free and $[\theta]$ is an overapproximation that includes the true dynamics parameters of the robot, then the robot motion generated by Algorithm 2 satisfies all the limits of the robot while staying collision-free.*

VI. Experiments & Comparisons

This section describes experiments that we conducted to evaluate the performance of the proposed algorithm. Our implementation has been open-sourced[1].

A. Robotic Platform

We performed hardware experiments on Kinova-gen3, which is a 7 degree-of-freedom robotic arm. Encoders and torque sensors are equipped on all joints. The joint encoder position resolutions are 0.02 degrees for the first four joints and 0.011 degrees for the last three. Consequently, this work ignores the joint encoder measurement error and focuses only on the torque sensor (Assumption 12). The frequency of control and data collection on Kinova-gen3 is not constant, but around 3.5-4 kHz. The robot end-effector (the last link and the gripper) weighs about 1.28 kg. The maximum payload weight of Kinova-gen3 is 2 kg for full-range continuous motion and 4 kg for mid-range continuous motion. For safety reasons, in the hardware experiment, we restrict the payload weight to less than 4 kg. All elements of Algorithm 2 are implemented in C++ and executed on a desktop computer with an AMD Ryzen Threadripper 7980X CPU and 256 GB RAM.

B. Experiment Settings

We evaluate our method across three hardware experiments:

(a) Five dumbbells are placed on one side of the robot. The robot must place the lightest dumbbell on a 3D-printed platform, which is positioned at 0.25m in front of the robot, while stacking the remaining dumbbells vertically in ascending order of weight, all while avoiding obstacles in the environment, particularly with one low obstacle in the way, as illustrated at the top of Fig. 3.

(b) The same stacking task is repeated with the platform positioned at 0.50m in front of the robot, particularly with one high obstacle in the way, as illustrated at the bottom of Fig. 3.

(c) The robot must pick up the heaviest dumbbell (8lb) and then follow a trajectory while avoiding obstacles in a more challenging setting, as illustrated in Fig. 4.

Throughout these experiments, we aim to demonstrate that our provably-safe online system identification framework is able to accomplish a variety of complex manipulation tasks with a higher success rate because it is able to provide the robust controller with a more accurate model estimate while guaranteeing safety during the system identification process. Experiments (a) and (b) evaluate the accuracy of each method while performing manipulation of a variety of unknown payloads. Experiment (c) evaluates the accuracy of trajectory tracking for collision avoidance. Experiment (c) is the most challenging task because the desired trajectory is close to the obstacles to minimize the path length. The dumbbells weigh 4, 5, 6, 7, and 8lb (about 1.81, 2.27, 2.72, 3.18, and 3.63kg, respectively). The robot has no access to their inertial parameters in prior.

Inertial parameters	nominal $\theta_{\mathrm{e},0}$	interval $[\theta_{\mathrm{e}}]$
m (kg)	3.2	[1.2, 5.2]
$p_x \cdot m$ (kg·m)	0.0	[-0.4, 0.4]
$p_y \cdot m$ (kg·m)	0.0	[-0.4, 0.4]
$p_z \cdot m$ (kg·m)	-0.5	[-1.0, -0.1]
XX (kg·m^2)	0.0	[-0.2, 0.2]
YY (kg·m^2)	0.0	[-0.2, 0.2]
ZZ (kg·m^2)	0.0	[-0.2, 0.2]
XY (kg·m^2)	0.0	[-0.2, 0.2]
YZ (kg·m^2)	0.0	[-0.2, 0.2]
XZ (kg·m^2)	0.0	[-0.2, 0.2]

TABLE I: The conservative nominal parameters and interval uncertainties of the end-effector assigned to the planner and the controller at the beginning of the identification phase, which serve as initial estimates for Algorithm 2 before identifying each dumbbell. The interval $[\theta_{\mathrm{e}}]$ covers the inertial parameters of the end-effector with any of the 5 dumbbells, while the nominal $\theta_{\mathrm{e},0}$ lies around the middle of $[\theta_{\mathrm{e}}]$.

C. Comparisons

For the hardware experiments, we evaluate a variety of comparison methods, summarized in TABLE II. These comparisons fall into two main categories.

The first category consists of methods that execute the task directly without an additional identification phase ("wrong", "conservative", "adap-1/2", "grav-pid"). Among these, "wrong" uses the same robust controller as our method, but assumes no payload is attached to the end-effector, while "conservative" uses a conservative estimate in TABLE I that encompasses all possible dumbbells.

The second category includes methods that perform identification of the payload's inertial parameters before executing the task ("ours", "random", "adap-1/2-excit", "grav-pid-ours/excit"). The "random" baseline serves as an ablation study, where identification is conducted using randomly chosen trajectories rather than the most exciting ones. The "grav-pid-excit" method utilizes a classical approach [5] based on Fourier series to generate exciting trajectories, considering

[1]https://github.com/roahmlab/OnlineSafeSysID

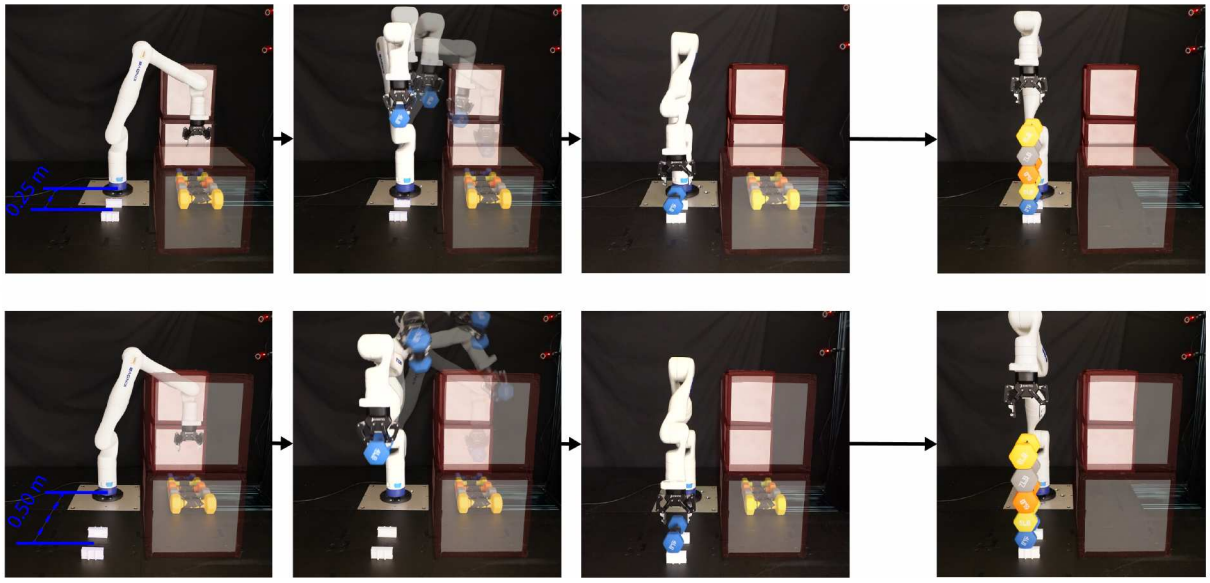

Fig. 3: This figure illustrates a complex pick-and-place task used in the hardware experiment. Five dumbbells are placed on one side of the robot, whose inertial parameters are unknown to it (first image in both rows). The robot is required to move each dumbbell around the obstacles, place the lightest dumbbell on a 3D-printed platform in front of it, and stack the remaining dumbbells vertically in ascending order of weight (fourth image in both rows). We design two experiments with different settings: (a) The 3D-printed platform is positioned 0.25 m from the robot, with one low obstacle in the way, as shown in the images in the first row. (b) The 3D-printed platform is positioned 0.50 m from the robot, with one high obstacle in the way, as shown in the images in the second row.

methods	controller	robot model	robot model uncertainties	exciting trajectories
ours	**ARMOUR robust [28]**	θ_0 **from Algorithm. 1**	$[\theta]$ **from Algorithm. 1**	**Algorithm. 2**
wrong	ARMOUR robust [28]	assume no payload at end-effector	0%	N/A
conservative	ARMOUR robust [28]	nominal in TABLE I	interval in TABLE I	N/A
random	ARMOUR robust [28]	θ_0 from Algorithm. 1	$[\theta]$ from Algorithm. 1	random
adap-1	adaptive [38]	identified by adaptive controller	N/A	N/A
adap-1-excit	adaptive [38]	identified by adaptive controller	N/A	Algorithm. 2
adap-2	adaptive [1]	identified by adaptive controller	N/A	N/A
adap-2-excit	adaptive [1]	identified by adaptive controller	N/A	Algorithm. 2
grav-pid	gravity compensated PID [37, (6.19)]	assume no payload at end-effector	N/A	N/A
grav-pid-ours	gravity compensated PID [37, (6.19)]	θ_0 from Algorithm. 1	N/A	Algorithm. 2
grav-pid-excit	gravity compensated PID [37, (6.19)]	θ_0 from Algorithm. 1	N/A	[5]

TABLE II: All comparisons evaluated in the hardware experiments, categorized by the controller type, the robot model used in the controller, the robot model uncertainties used in the controller, and whether exciting trajectories were used for system identification prior to task execution. Methods marked "N/A" in the "exciting trajectories" column skip the identification phase and directly execute the task.

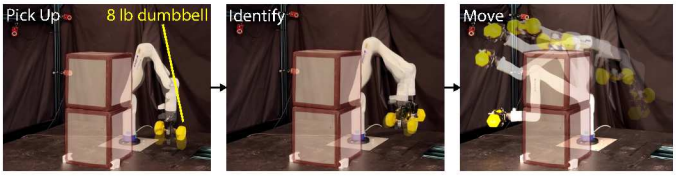

Fig. 4: An illustration of the third real-world experiment. The robot is required to pick up and perform system identification with an 8lb dumbbell and then follow a trajectory to avoid obstacles.

only velocity and acceleration limits. However, these constraints may be insufficient for ensuring safety in terms of collision avoidance or torque limits when handling unknown payloads. "adap-1/2-excit" applies the adaptive controllers from [38, 1] while following the same exciting trajectories as "ours" before executing the task, allowing the adaptive controllers additional time to estimate the end-effector inertial parameters.

D. Implementation Details

1) System Identification Implementation: The system identification solver (Algorithm 1) is also implemented using Ipopt [46]. The analytical gradient and analytical hessian are also provided to improve computational efficiency. The forward integration horizon h is chosen to be 400, which implies a forward integration time of 100-120 ms on Kinova-gen3. The maximum amount of uncertainties over the measurements $[\delta \mathbf{m}]$ is set to 5% maximum uncertainty for both robot dynamics parameters $[\theta_{\mathrm{r}}]$ (without end-effector) and 2.5% for applied torque measurement $\boldsymbol{\tau}$.

2) Exciting Trajectory Planner Implementation: The trajectory planner (28) is implemented using Ipopt [46], an open source interior point optimizer for nonlinear programming. The duration of the exciting trajectories t_f is chosen to be 3.0 s. The total computation time of the trajectory planner is limited to $t_p = 1.5$ s. $N_s = 128$ samples are uniformly distributed along the trajectory to formulate the inverse dynamics

regressor $\mathbf{W}$ (30). We quit the while loop in Algorithm 2 after generating 4 exciting trajectories, implying a total time of $(4-1)\times 1.5+3.0=7.5$ s during the identification phase.

Algorithm 2 requires initial estimates of the nominal parameters $\theta_{\mathrm{e},0}$ and uncertainties $[\theta_{\mathrm{e}}]$ of the end-effector inertial parameters. TABLE I reports the initial nominal parameters $\theta_{\mathrm{e},0}$ and the initial interval uncertainties $[\theta_{\mathrm{e}}]$ at the beginning of the identification phase. We select a range for mass that includes both the robot end-effector's mass and any dumbbell's mass. Assuming the robot consistently picks up the dumbbell at its center, the center of mass of the end-effector remains near the z-axis of its inertial frame. Thus, we select a relatively narrow range centered on 0 for $p_x \cdot m$ and $p_y \cdot m$. For other inertia parameters, we determine the ground truth using CAD estimation and select a range that includes the results of all dumbbells. Note that the distribution that we use in TABLE I is much wider than that of the existing work [41, Section IV].

For the "random" comparison, we run a different version of Algorithm 2. Instead of optimizing for the most exciting trajectories in line 7, we randomly sample the trajectory parameter k^*_{j+1} in the parameter space $\mathcal{K}$.

3) Trajectory for Executing Tasks: For all three experiments, each compared method (including our own) is required to follow a pre-defined trajectory to move the payload at a user-specified terminal location. We compute this trajectory offline by minimizing the jerk while avoiding obstacles and satisfying joint and torque limits. This is done by using RAPTOR [51], an open-source trajectory optimization toolbox, with the estimated 8lb dumbbell model attached to the end-effector. The fast trajectory spans 2s and consists of two 1-second piecewise-continuous degree-5 Bezier curves.

E. Results

We repeat all the experiments 5 times using the method proposed in this paper and each of the aforementioned comparison methods in TABLE II. The results of all the experiments were consistent across all 5 trials and are summarized in TABLE III. Only the method proposed in the paper ("ours") is able to successfully complete all three experiments. For safety reasons, we only ran "grav-pid-excit" in simulation because, in contrast to the trajectories generated by our approach, the trajectories from [7] violate safety constraints and torque limits, even with the lightest dumbbell. The tracking performance and the commanded torque inputs of placing the dumbbells after the identification phase are illustrated in Figure 11 for Experiment (a) and Figure 12 for Experiment (b) in Appendix E.

Figure 5 shows the evolution of the estimated end-effector mass over time as the robot picks up and places the 4lb dumbbell in Experiment (b), providing further insight into the performance of methods that perform system identification of the payload's inertial parameters before executing the task. For both "ours" and "random", inertial parameter estimates are updated only after completing each trajectory during the identification phase (line 8 in Algorithm 2), due to the receding horizon nature of their strategy. In contrast, "adap-1" and "adap-1-excit" continuously update their parameter estimates

Methods	Experiment (a)	Experiment (b)	Experiment (c)
ours	**success**	**success**	**success**
wrong	fail at 8lb	fail at 8lb	collide
conservative	fail at 8lb	fail at 8lb	collide
random	success	fail at 8lb	collide
adap-1	success	fail at 8lb	collide
adap-1-excit	success	success	collide
adap-2	fail at 8lb	fail at 4lb	collide
adap-2-excit	fail at 8lb	fail at 4lb	collide
grav-pid	success	fail at 6lb	collide
grav-pid-ours	success	success	collide
grav-pid-excit	fail at 4lb	fail at 4lb	collide

TABLE III: Results of our method and baseline comparisons across three hardware experiments. Each experiment was repeated five times and returned consistent results.

from the very beginning of the experiment. Notably, "ours" is more accurate while converging faster than all tested methods. It does this before the execution of the move-and-place task, which results in a higher success rate across all experiments. Without overburdening the readers, the results for "adap-2" and "adap-2-excit" are omitted from this figure.

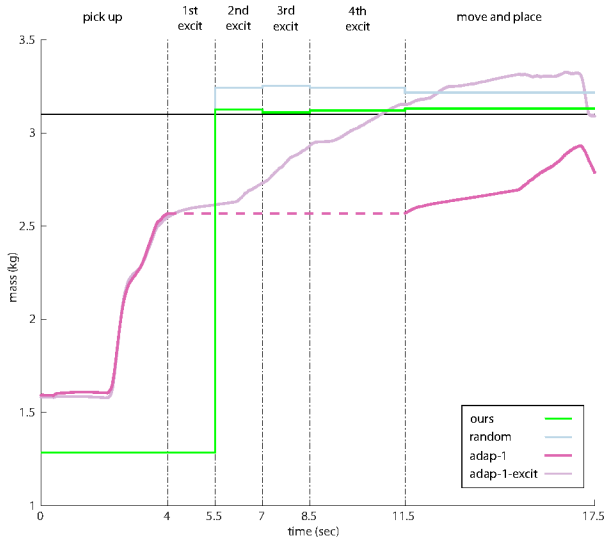

Fig. 5: This figure illustrates the evolution of the estimated end-effector mass over time as the robot picks up and places the 4lb dumbbell in Experiment (b). For both "ours" and "random", we only plot the nominal estimate $\theta_{\mathrm{e},0}$. For both "ours" and "adap-1-excit", an additional identification phase (from 4 to 11.5 seconds) is included, during which the robot tracks four exciting trajectories from Algorithm 2 to identify the end-effector inertial parameters. In contrast, "random" follows four randomly generated trajectories during this phase instead of exciting ones. For "adap-1", since the identification phase is skipped entirely and the robot directly executes the task (i.e., moving and stacking the dumbbells onto the platform), the identification phase (4-11.5 second) is denoted as a dotted line. We observe that "ours" achieves more accurate estimation of the end-effector parameters than random", demonstrating the effectiveness of using exciting trajectories. Additionally, "adap-1-excit" benefits from both extended identification time and proper data excitation, resulting in improved estimation accuracy and thus, a higher success rate in the experiments compared to "adap-1".

Figure 6 reports the interval bound estimates after identifying the inertial parameters of the end-effector while manipulating the 4lb dumbbell. Because both our method and "random" utilize Algorithm 1 they are guaranteed to generate conservative interval estimates for all relevant inertial param-

eters, which empirically validates Theorem 13. In particular, given a conservative measurement noise bound, (27) can always generate overapproximated bounds that include the true inertial parameters θ_e. In addition, note that the interval bounds generated using the proposed method are smaller than the ones generated by selecting random trajectories for almost all inertial parameters, which illustrates that the cost function described in (30) is able to generate exciting trajectories. The identification results of 5, 6, 7, and 8lb dumbbells can be found in Appendix E. We also report the condition numbers of the observation matrix $\mathbf{Y}$ corresponding to these results in TABLE V.

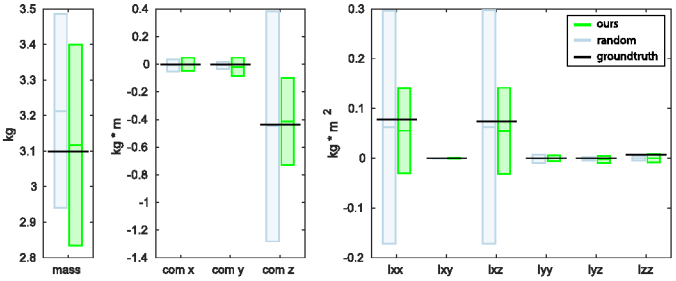

Fig. 6: This figure illustrates the interval bound estimates of the 10 inertial parameters of the end-effector along with the 4lb dumbbell after the identification phase. Note that by following the most exciting trajectories, our method is able to achieve more accurate results and tighter interval estimates, compared to "random", which does not optimize for the most exciting trajectories.

VII. Limitations

There exist several limitations in both the exciting trajectory planner and the robust system identification in our framework. While most of the related work parameterize exciting trajectories as a Fourier series [5, (50)], the trajectory planner in our framework, which is based on ARMOUR [28], parameterizes the trajectories as degree-5 Bezier curves and plans for receding horizons. Although inheriting the safety guarantees properties from ARMOUR (Lemma 14), the generated trajectories can only capture locally exciting features, generally resulting in a regressor matrix with a larger condition number than the Fourier based methods.

In addition, the approach developed in this paper has assumed that the noise in the robot manipulator system is dominated by the torque measurements. In addition, the proposed approach assumes that the data are generated by forward Euler integration. Though this could be extended to more accurate forms of numerical integration, these methods would still potentially just be approximations to the true system dynamics.

The Log-Cholesky parameterization P (14) transforms the traditional SDP formulation (26) into an unconstrained convex problem (25), thereby facilitating perturbation analysis for the estimation of the interval bound of inertial parameters. However, introducing exponentials within P to enforce positive diagonal entries can also introduce numerical issues when computing the inverse of the second-order Hessian matrix in (42) and, as a consequence, complicates the process of obtaining tighter interval bounds during online estimation.

Finally, our framework requires system identification to be performed prior to task execution, which may increase overall task completion time. According to our simulation experiments, trajectories that attempt to complete the task directly may lack sufficient excitation, resulting in larger interval bounds on the estimated parameters. These conservative estimates, in turn, lead to more conservative trajectory planning to ensure safety, ultimately prolonging task execution.

VIII. Conclusion

This paper presents a provably-safe, online system identification framework for robotic arms manipulating heavy payloads with inertial uncertainties. By combining provably-safe trajectory optimization and robust system identification based on perturbation analysis, the framework ensures safety while refining inertial parameter estimates of the end-effector online, leading to improved planning and control performance. Hardware experiments demonstrate its effectiveness in handling heavy payloads under significant uncertainty, ensuring precise and safe operation.

References

[1] Mohamed Abdelwahab, Giulio Giacomuzzo, Alberto Dalla Libera, and Ruggero Carli. Adaptive robust controller for handling unknown uncertainty of robotic manipulators. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 2992–2997. IEEE, 2024.

[2] Haider AF Almurib, Haidar Fadhil Al-Qrimli, and Nandha Kumar. A review of application industrial robotic design. In *2011 Ninth International Conference on ICT and Knowledge Engineering*, pages 105–112. IEEE, 2012.

[3] Chae An, Christopher Atkeson, and John Hollerbach. Estimation of inertial parameters of rigid body links of manipulators. In *1985 24th IEEE Conference on Decision and Control*. IEEE, December 1985. doi: 10.1109/cdc.1985.268648.

[4] Ko Ayusawa, Antoine Rioux, Eiichi Yoshida, Gentiane Venture, and Maxime Gautier. Generating persistently exciting trajectory based on condition number optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017. doi: 10.1109/icra.2017.7989770.

[5] Ko Ayusawa, Antoine Rioux, Eiichi Yoshida, Gentiane Venture, and Maxime Gautier. Generating persistently exciting trajectory based on condition number optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6518–6524, 2017. doi: 10.1109/ICRA.2017.7989770.

[6] A. Bahloul, S. Tliba, and Y. Chitour. Dynamic parameters identification of an industrial robot with and without payload. *IFAC-PapersOnLine*, 51(15):443–448, 2018. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2018.09.185. URL https://www.sciencedirect.com/science/article/pii/S2405896318318524. 18th IFAC Symposium on System Identification SYSID 2018.

[7] Vincent Bonnet, Philippe Fraisse, Andre Crosnier, Maxime Gautier, Alejandro Gonzalez, and Gentiane Venture. Optimal exciting dance for identifying inertial parameters of an anthropomorphic structure. *IEEE Transactions on Robotics*, 32(4):823–836, August 2016. ISSN 1941-0468. doi: 10.1109/tro.2016.2583062.

[8] Pascal Brault, Quentin Delamare, and Paolo Robuffo Giordano. Robust trajectory planning with parametric uncertainties. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11095–11101, 2021. doi: 10.1109/ICRA48506.2021.9561118.

[9] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen.

Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[10] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

[11] Saverio Farsoni, Federica Ferraguti, and Marcello Bonfè. Safety-oriented robot payload identification using collision-free path planning and decoupling motions. *Robotics and Computer-Integrated Manufacturing*, 59:189–200, 2019.

[12] M. Gautier and W. Khalil. A direct determination of minimum inertial parameters of robots. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1682–1687 vol.3, 1988. doi: 10.1109/ROBOT.1988.12308.

[13] M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 494–499 vol.1, 1991. doi: 10.1109/CDC.1991.261353.

[14] Claudio Gaz and Alessandro De Luca. Payload estimation based on identified coefficients of robot dynamics — with an application to collision detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3033–3040, 2017. doi: 10.1109/IROS.2017.8206142.

[15] Paolo Robuffo Giordano, Quentin Delamare, and Antonio Franchi. Trajectory generation for minimum closed-loop state sensitivity. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 286–293, 2018. doi: 10.1109/ICRA.2018.8460546.

[16] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[17] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

[18] Leonidas J Guibas, An Thanh Nguyen, and Li Zhang. Zonotopes as bounding volumes. In *SODA*, volume 3, pages 803–812, 2003.

[19] Yong Han, Jianhua Wu, Chao Liu, and Zhenhua Xiong. An iterative approach for accurate dynamic model identification of industrial robots. *IEEE Transactions on Robotics*, 36(5):1577–1594, October 2020. ISSN 1941-0468. doi: 10.1109/tro.2020.2990368.

[20] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[21] Jinfei Hu, Chen Li, Zheng Chen, and Bin Yao. Precision motion control of a 6-dofs industrial robot with accurate payload estimation. *IEEE/ASME Transactions on Mechatronics*, 25(4):1821–1829, 2020. doi: 10.1109/TMECH.2020.2994231.

[22] K Niranjan Kumar, Irfan Essa, Sehoon Ha, and C Karen Liu. Estimating mass distribution of articulated objects using non-prehensile manipulation. *arXiv preprint arXiv:1907.03964*, 2019.

[23] Alexander Kurdas, Mazin Hamad, Jonathan Vorndamme, Nico Mansfeld, Saeed Abdolshah, and Sami Haddadin. Online payload identification for tactile robots using the momentum observer. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5953–5959. IEEE, 2022.

[24] Peng Li and Xiangpeng Liu. Common sensors in industrial robots: A review. *Journal of Physics: Conference Series*, 1267 (1):012036, jul 2019. doi: 10.1088/1742-6596/1267/1/012036. URL https://dx.doi.org/10.1088/1742-6596/1267/1/012036.

[25] Yingqiang Liu, Zheng Chen, and Bin Yao. Online optimization-based time-optimal adaptive robust control of linear motors with input and state constraints. *IEEE/ASME Transactions on Mechatronics*, 29(4):3157–3165, 2024. doi: 10.1109/TMECH.2024.3404821.

[26] Hirokazu Mayeda, Koji Yoshida, and Koichi Osuka. Base parameters of manipulator dynamic models. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1367–1372. IEEE, 1988.

[27] Marius Memmel, Andrew Wagenmaker, Chuning Zhu, Patrick Yin, Dieter Fox, and Abhishek Gupta. Asid: Active exploration for system identification in robotic manipulation. *arXiv preprint arXiv:2404.12308*, 2024.

[28] Jonathan Michaux, Patrick Holmes, Bohao Zhang, Che Chen, Baiyue Wang, Shrey Sahgal, Tiancheng Zhang, Sidhartha Dey, Shreyas Kousik, and Ram Vasudevan. Can't touch this: Real-time, safe motion planning and control for manipulators under uncertainty, 2023.

[29] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to interval analysis*. SIAM, 2009.

[30] Simeon Nedelchev, Lev Kozlov, Ramil R Khusainov, and Igor Gaponov. Enhanced adaptive control over robotic systems via generalized momentum dynamic extensions. *Russian Journal of Nonlinear Dynamics*, 19(4):633–646, 2023.

[31] Kyongho Park and Youngjin Choi. System identification method for robotic manipulator based on dynamic momentum regressor. In *2016 12th IEEE International Conference on Control and Automation (ICCA)*, pages 755–760, 2016. doi: 10.1109/ICCA.2016.7505369.

[32] Sarbajit Paul, Junghwan Chang, John Edward Fletcher, and Subhas Mukhopadhyay. A novel high-resolution optical encoder with axially stacked coded disk for modular joints: Physical modeling and experimental validation. *IEEE Sensors Journal*, 18(14):6001–6008, 2018.

[33] C. Presse and M. Gautier. New criteria of exciting trajectories for robot identification. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 907–912 vol.3, 1993. doi: 10.1109/ROBOT.1993.292259.

[34] Wolfgang Rackl, Roberto Lampariello, and Gerd Hirzinger. Robot excitation trajectories for dynamic parameter estimation using optimized b-splines. In *2012 IEEE International Conference on Robotics and Automation*, pages 2042–2047, 2012. doi: 10.1109/ICRA.2012.6225279.

[35] Caleb Rucker and Patrick M. Wensing. Smooth parameterization of rigid-body inertia. *IEEE Robotics and Automation Letters*, 7(2):2771–2778, 2022. doi: 10.1109/LRA.2022.3144517.

[36] Adelia Sabirova, Simeon Nedelchev, and Igor Gaponov. Parameter identification in mechanical systems with energy-based regressor: Preliminary study. In *2021 International Conference" Nonlinearity, Information and Robotics"(NIR)*, pages 1–6. IEEE, 2021.

[37] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[38] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987. doi: 10.1177/027836498700600303. URL https://doi.org/10.1177/027836498700600303.

[39] Jean-Jacques E Slotine and Weiping Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519, 1989.

[40] Cristovao D. Sousa and Rui Cortesao. Inertia tensor properties in robot dynamics identification: A linear matrix inequality approach. *IEEE/ASME Transactions on Mechatronics*, 24(1):406–411, February 2019. ISSN 1941-014X. doi: 10.1109/tmech.2019.2891177.

[41] Ali Srour, Antonio Franchi, Paolo Robuffo Giordano, and Marco Cognetti. Experimental validation of sensitivity-aware trajectory planning for a redundant robotic manipulator under payload uncertainty. *IEEE Robotics and Automation Letters*, 2024.

[42] M Tomizuka and R Horowitz. Model reference adaptive control of mechanical manipulators. In *Adaptive Systems in Control and Signal Processing 1983*, pages 27–32. Elsevier, 1984.

[43] Kevin Tracy and Zachary Manchester. On the differentiability of the primal-dual interior-point method. *arXiv preprint arXiv:2406.11749*, 2024.

[44] Dzmitry Tsetserukou and Susumu Tachi. Torque sensors for robot joint control. *Sensors, Focus on Tactile, Force and Stress Sensors*, pages 15–36, 2008.

[45] G. Venture, K. Ayusawa, and Y. Nakamura. A numerical method for choosing motions with optimal excitation properties for identification of biped dynamics - an application to human. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, May 2009. doi: 10.1109/robot.2009.5152264.

[46] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[47] Hanlei Wang. Recursive composite adaptation for robot manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 135(2):021010, 11 2012. ISSN 0022-0434. doi: 10.1115/1.4007557. URL https://doi.org/10.1115/1.4007557.

[48] Simon Wasiela, Paolo Robuffo Giordano, Juan Cortés, and Thierry Siméon. A sensitivity-aware motion planner (samp) to generate intrinsically-robust trajectories. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12707–12713, 2023. doi: 10.1109/ICRA48891.2023.10160576.

[49] Patrick M. Wensing, Sangbae Kim, and Jean-Jacques E. Slotine. Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters*, 3(1):60–67, 2018. doi: 10.1109/LRA.2017.2729659.

[50] Tian Xu, Jizhuang Fan, Qianqian Fang, Yanhe Zhu, and Jie Zhao. An accurate identification method based on double weighting for inertial parameters of robot payloads. *Robotica*, 40(12):4358–4374, July 2022. ISSN 1469-8668. doi: 10.1017/s0263574722000960.

[51] Bohao Zhang and Ram Vasudevan. Rapid and robust trajectory optimization for humanoids. *arXiv preprint arXiv:2409.00303*, 2024.

[52] Bohao Zhang, Daniel Haugk, and Ram Vasudevan. System identification for constrained robots. *arXiv preprint arXiv:2408.08830*, 2024.

[53] Dan Zhang and Bin Wei. A review on model reference adaptive control of robotic manipulators. *Annual Reviews in Control*, 43: 188–198, 2017.

APPENDIX A
PHYSICAL CONSISTENCY CONSTRAINTS

Using the regressors described in the previous subsection, one can cast the system identification problem as a linear regression problem. However, physical consistency of robot link parameters requires satisfying additional mathematical constraints. This subsection presents two approaches for ensuring these constraints: Linear Matrix Inequalities (LMI) and log-Cholesky parameterization.

Definition 15 (Physical Consistency LMI, [40]). *The inertial parameters of the j^{th} link $\theta_{ip,j}$ are* physically consistent *if they satisfy:*

$$LMI(\theta_{ip,j}) := \begin{bmatrix} \left(\frac{tr(I_j)}{2}\mathbb{1}_{3\times 3} - I_j\right) & p_j m_j \\ p_j^T m_j & m_j \end{bmatrix} \succ \mathbb{0}_{4\times 4}, \quad (31)$$

where $LMI(\theta_{ip,j})$ is defined as pseudo-inertia matrix *for each link $j \in \{1, \ldots, n\}$, tr denotes the trace operator, $\mathbb{1}_{3\times 3}$ represents the 3-by-3 identify matrix, and I_j represents the* inertia tensor*:*

$$I_j := \begin{bmatrix} XX_j & XY_j & XZ_j \\ XY_j & YY_j & YZ_j \\ XZ_j & YZ_j & ZZ_j \end{bmatrix}, \quad (32)$$

One can cast the system identification problem as trying to minimize a linear cost function while satisfying LMI constraints associated with physical consistency. While these LMI constraints can be incorporated into convex optimization problems [49, (25)], they require semi-definite programming (SDP) solvers and involve relaxing the LMI (31) to positive semidefiniteness rather than strict positive definiteness. As a result, a computed solution may not be physically consistent because the LMI may just be positive semidefinite rather than positive definite. An alternative approach leverages the Cholesky decomposition to ensure positive-definiteness:

Theorem 16 (Log-Cholesky Parameterization, [20, Corollary 7.2.9]). *Every physically consistent set of inertial parameters can be uniquely parameterized through a log-Cholesky decomposition:*

$$LMI(\theta_{ip,j}) = MM^T, \quad (33)$$

where M takes the form of an upper triangular matrix with positive diagonal entries:

$$M := e^{\alpha} \begin{bmatrix} e^{d_1} & s_{12} & s_{13} & s_{14} \\ 0 & e^{d_2} & s_{23} & s_{24} \\ 0 & 0 & e^{d_3} & s_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

This structure is completely characterized by the log-Cholesky parameters $\eta \in \mathbb{R}^{10}$*:*

$$\eta = \begin{bmatrix} \alpha & d_1 & d_2 & d_3 & s_{12} & s_{13} & s_{23} & s_{14} & s_{24} & s_{34} \end{bmatrix}^T. \quad (35)$$

To enforce positive definiteness and thereby enforce physical consistency of the links, one could apply the log-Cholesky decomposition (14). However, this would transform the convex system identification problem into a nonlinear optimization problem that may have spurious local minima. Fortunately, the following corollary resolves that shortcoming:

Corollary 17 (Parameter Recovery, [35, Section V.A]). *The function $P : \mathbb{R}^{10} \to \mathbb{R}^{10}$ defined in* (16) *that transforms log-Cholesky parameters to inertial parameters is a diffeomorphism.*

This diffeomorphism result has significant practical implications for system identification. As described earlier, the log-Cholesky parameterization offers an approach to enforce physical consistency through its structure. While this transformation introduces nonlinear relationships between the parameters, the diffeomorphic nature of the mapping ensures that this nonlinearity does not create spurious local minima in the optimization landscape. This mathematical guarantee allows us to work with the nonlinear parameterization confidently, knowing that any local minimum we find in the transformed

space corresponds uniquely to a global minimum in the original parameter space.

APPENDIX B
PERTURBATION ANALYSIS OF CONVEX OPTIMIZATION PROBLEMS

The objective of this work is to understand how bounded noise in the data influences the dynamics parameter estimates. To analyze how measurement noise affects parameter estimation in system identification, we draw upon perturbation analysis of optimization problems:

Theorem 18 (Sensitivity of Optimal Solutions, [43]). *Consider a parametric optimization problem:*

$$\min_{x\in\mathcal{X}} f(x,y), \tag{36}$$

where x represents the decision variable and y denotes problem parameters, with $\mathcal{X}\subset\mathbb{R}^n$ and $\mathcal{Y}\subset\mathbb{R}^m$. If:

I. $f:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ is continuously differentiable on both $\mathcal{X}$ and $\mathcal{Y}$

II. f is geodesically convex on $\mathcal{X}$ and convex on $\mathcal{Y}$

then the sensitivity of the optimal solution to parameter perturbations is given by

$$\frac{\partial x}{\partial y} = -(\frac{\partial^2 f}{\partial x^2}(x,y))^{-1}\frac{\partial^2 f}{\partial x\partial y}(x,y). \tag{37}$$

This sensitivity analysis, discussed extensively in [16], provides the foundation for bounding optimal solutions when problem parameters are subject to bounded perturbations. Specifically, given bounds on parameter perturbations, we can establish corresponding bounds on variations in optimal solutions.

APPENDIX C
PROOF OF THEOREM 13

Proof: To prove (27), first note that $\theta_e\in\theta_e^*([\mathbf{m}])$. Hence one can apply the Mean Value Form from Interval Analysis [29, (6.25)] to compute the right-hand side of (27), which describes how to overapproximate $\theta_e^*([\mathbf{m}])$ by applying the Mean Value Theorem over interval arguments. Required by (27), we then derive the full expression of $\frac{\partial\theta_e^*}{\partial\mathbf{m}}(\mathbf{m})$.

Let $\eta_e^*(\mathbf{m})$ be any local minimizer to (25):

$$\eta_e^*(\mathbf{m}) = \underset{\eta_e\in\mathbb{R}^{10}}{\arg\min}\|\mathbf{Y}P(\eta_e)-\mathbf{U}(\mathbf{m})\|. \tag{38}$$

Note that this is also equivalent to the following form

$$\eta_e^*(\mathbf{m}) = \underset{\eta_e\in\mathbb{R}^{10}}{\arg\min} J(\mathbf{m},\eta_e), \tag{39}$$

where

$$J(\mathbf{m},\eta_e) = \frac{1}{2}(\mathbf{Y}P(\eta_e)-\mathbf{U}(\mathbf{m}))^T(\mathbf{Y}P(\eta_e)-\mathbf{U}(\mathbf{m})). \tag{40}$$

The first-order optimality condition of the optimization problem above is as follows:

$$\frac{\partial J}{\partial\eta_e}(\mathbf{m},\eta_e^*(\mathbf{m})) = (\mathbf{Y}P(\eta_e^*)-\mathbf{U}(\mathbf{m}))^T\mathbf{Y}\frac{\partial P}{\partial\eta_e}(\eta_e^*) = 0. \tag{41}$$

Treating this as an implicit equation and apply (37), we can get that

$$\frac{\partial\eta_e^*}{\partial\mathbf{m}}(\mathbf{m}) = -\left(\frac{\partial^2 J}{\partial\eta_e^2}(\mathbf{m},\eta_e^*(\mathbf{m}))\right)^{-1}\cdot\frac{\partial^2 J}{\partial\mathbf{m}\partial\eta_e}(\mathbf{m},\eta_e^*(\mathbf{m})), \tag{42}$$

where

$$\frac{\partial^2 J}{\partial\eta_e^2}(\mathbf{m},\eta_e^*(\mathbf{m})) = \frac{\partial P^T}{\partial\eta_e}(\eta_e^*(\mathbf{m}))\mathbf{Y}^T\mathbf{Y}\frac{\partial P}{\partial\eta_e}(\eta_e^*(\mathbf{m}))+ \\ +(\mathbf{Y}P(\eta_e^*(\mathbf{m}))-\mathbf{U}(\mathbf{m}))^T\mathbf{Y}\frac{\partial^2 P}{\partial\eta_e^2}(\eta_e^*(\mathbf{m})), \tag{43}$$

and

$$\frac{\partial^2 J}{\partial\mathbf{m}\partial\eta_e}(\mathbf{m},\eta_e^*(\mathbf{m})) = -\frac{\partial\mathbf{U}^T}{\partial\mathbf{m}}(\mathbf{m})\mathbf{Y}\frac{\partial P}{\partial\eta_e}(\eta_e^*(\mathbf{m})). \tag{44}$$

Given $\theta_e^*(\mathbf{m}) = P(\eta_e^*(\mathbf{m}))$, we can then apply the chain rule:

$$\frac{\partial\theta_e^*}{\partial\mathbf{m}}(\mathbf{m}) = \frac{\partial P}{\partial\eta_e}(\eta_e^*(\mathbf{m}))\frac{\partial\eta_e^*}{\partial\mathbf{m}}(\mathbf{m}), \tag{45}$$

which can be further expanded using (42)

$$\frac{\partial\theta_e^*}{\partial\mathbf{m}}(\mathbf{m}) = -\frac{\partial P}{\partial\eta_e}(\eta_e^*(\mathbf{m}))\left(\frac{\partial^2 J}{\partial\eta_e^2}(\mathbf{m},\eta_e^*(\mathbf{m}))\right)^{-1} \\ \cdot\frac{\partial^2 J}{\partial\mathbf{m}\partial\eta_e}(\mathbf{m},\eta_e^*(\mathbf{m})). \tag{46}$$

■

APPENDIX D
CONDITION NUMBER OF INVERSE DYNAMICS REGRESSOR (30) AND MOMENTUM REGRESSOR (23)

In Section V-B, we discuss minimizing the condition number of the standard dynamics regressor $\mathbf{W}$ instead of the momentum regressor $\mathbf{Y}$ to generate exciting trajectories for system identification. Since the original robot dynamics in (5) are equivalent to the momentum-based formulation in (13) via time differentiation, $\mathbf{W}$ can be viewed as an approximation of the time derivative of $\mathbf{Y}$, as both regressors contribute linearly to (5) and (13), respectively. Here, we numerically demonstrate that the condition numbers of these two regressors are positively correlated.

To do so, we parameterize a 10-second trajectory using a Fourier series with randomly assigned coefficients. We sample the trajectory at a uniform time interval of 1ms, which indicates 10000 instances in total along the trajectory. We evaluate both regressors at each time instance, and construct observation matrices from the collected data. We then compute the condition numbers of these observation matrices for both the standard dynamics and momentum regressors across eight different values of the forward integration horizon h. The positive correlation between the condition numbers of $\mathbf{W}$ and $\mathbf{Y}$ is quantified using Pearson correlation coefficients [9], as shown in TABLE IV. As h increases, this correlation weakens due to the accumulation of integration errors. These results suggest that, with an appropriately chosen h, minimizing the condition number of the standard dynamics regressor $\mathbf{W}$ can

effectively generate exciting trajectories that also reduce the condition number of the momentum regressor $\mathbf{Y}$.

forward integration horizon h	Pearson correlation coefficients
1	1.0000
5	1.0000
10	0.9994
20	0.9934
50	0.9540
100	0.6709
200	0.3537
500	0.2789

TABLE IV: The Pearson correlation coefficients between the condition number of the standard dynamics regressor $\mathbf{W}$ and that of the momentum regressor $\mathbf{Y}$ are reported for different forward integration horizons h. When h is small, the Pearson correlation coefficient is close to 1, indicating a strong positive correlation between the two condition numbers. As h increases, this positive correlation gradually decreases.

APPENDIX E
FULL SYSTEM IDENTIFICATION RESULTS

This section reports the full system identification results of the inertial parameters of the end-effector along with the 5, 6, 7, and 8lb dumbbells. The final system identification results can be found in Figures 7, 8, 9, and 10. We also report the condition numbers of the observation matrix $\mathbf{Y}$ corresponding to these results in TABLE V. Because both our method and "Random" utilize Algorithm 1, they are guaranteed to generate conservative interval estimates for all relevant inertial parameters, which empirically validates Theorem 13. In addition note that the interval bounds generated using the proposed method are smaller than the ones generated by selecting random trajectories for almost all inertial parameters, which illustrates that cost function described in (30) is able to generate exciting trajectories.

dumbbell	ours	random
4lb	**274.030**	479.573
5lb	**282.840**	418.078
6lb	**250.704**	373.448
7lb	**258.517**	312.352
8lb	**233.539**	413.381

TABLE V: This table reports the 2-norm condition number of $\mathbf{Y}$ based on data collected during the entire 7.5-second identification phase. By optimizing for the most exciting trajectories, our method produces an observation matrix $\mathbf{Y}$ with a significantly smaller condition number compared to the "random" baseline, which does not optimize for excitation. As a result, our method achieves more accurate parameter estimation and generates tighter interval bounds, as demonstrated in the figures on the right column.

APPENDIX F
FULL TRACKING ERROR & COMMANDED TORQUE PLOTS

This section reports the tracking error and the commanded torque when the robot moves each of the five dumbbells around the obstacles by following a precomputed trajectory and stacks all dumbbells vertically on a 3D-printed platform during the second phase of the second experiment. Figure 11 and 12 include the results of our method and all the comparisons listed in TABLE II.

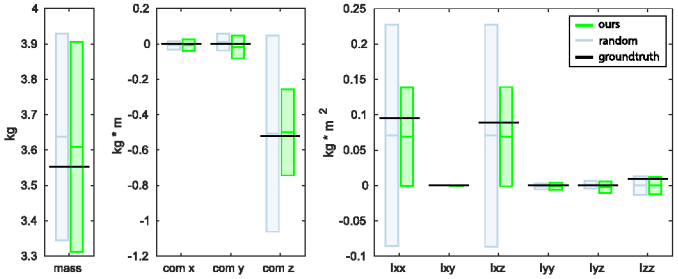

Fig. 7: This figures illustrates the interval bound estimates of the 10 inertial parameters of the end-effector along with the 5lb dumbbell after the identification phase. Note that by following the most exciting trajectories, our method is able to achieve more accurate results and tighter interval estimates, compared to "random", which does not optimize for excitation.

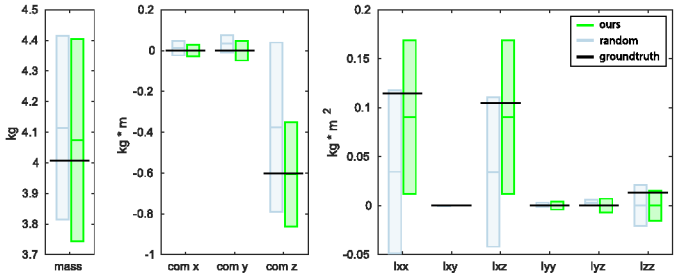

Fig. 8: This figures illustrates the interval bound estimates of the 10 inertial parameters of the end-effector along with the 6lb dumbbell after the identification phase. Note that by following the most exciting trajectories, our method is able to achieve more accurate results and tighter interval estimates, compared to "random", which does not optimize for excitation.

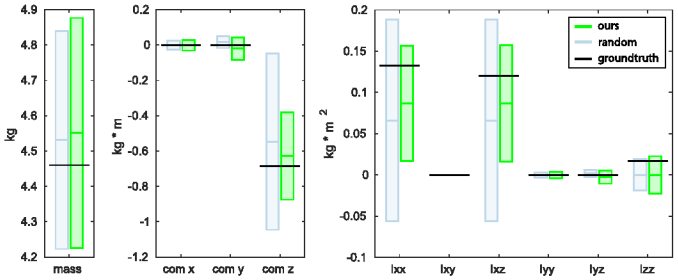

Fig. 9: This figures illustrates the interval bound estimates of the 10 inertial parameters of the end-effector along with the 7lb dumbbell after the identification phase. Note that by following the most exciting trajectories, our method is able to achieve more accurate results and tighter interval estimates, compared to "random", which does not optimize for excitation.

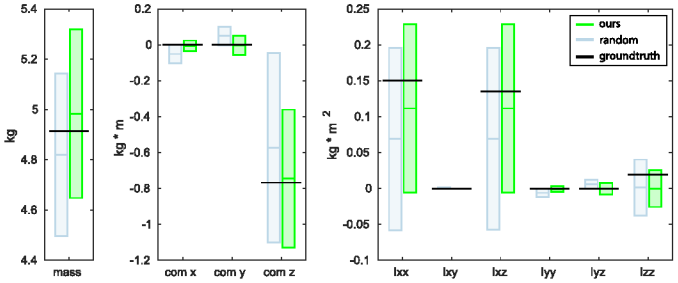

Fig. 10: This figures illustrates the interval bound estimates of the 10 inertial parameters of the end-effector along with the 8lb dumbbell after the identification phase. Note that by following the most exciting trajectories, our method is able to achieve more accurate results and tighter interval estimates, compared to "random", which does not optimize for excitation.

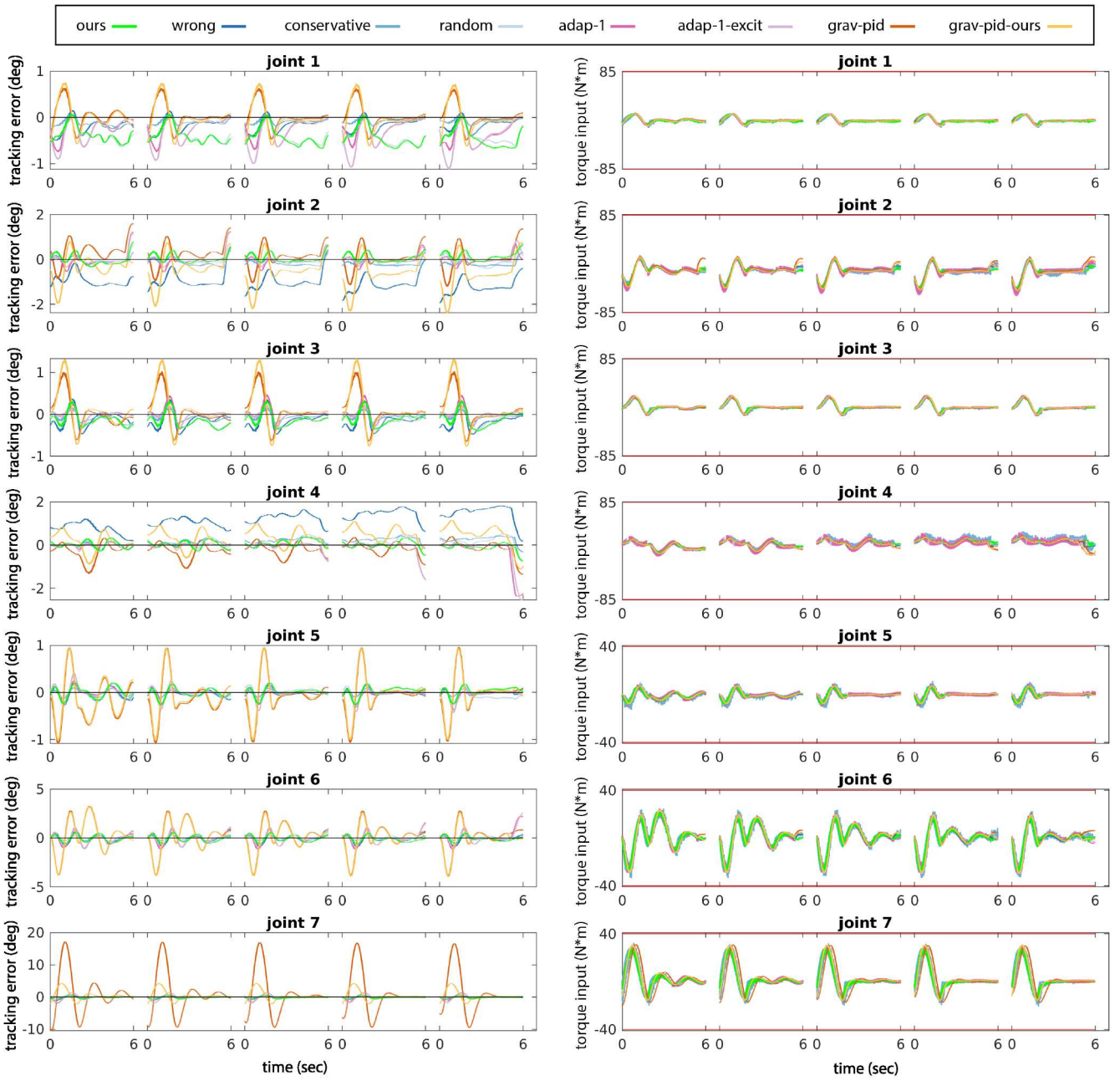

Fig. 11: This figure illustrates the tracking error of our method and all the comparisons on the left and the commanded torque on the right, while moving and stacking each of the five dumbbells on the 3D-printed platform located 0.25 m in front of the robot in Experiment (a).

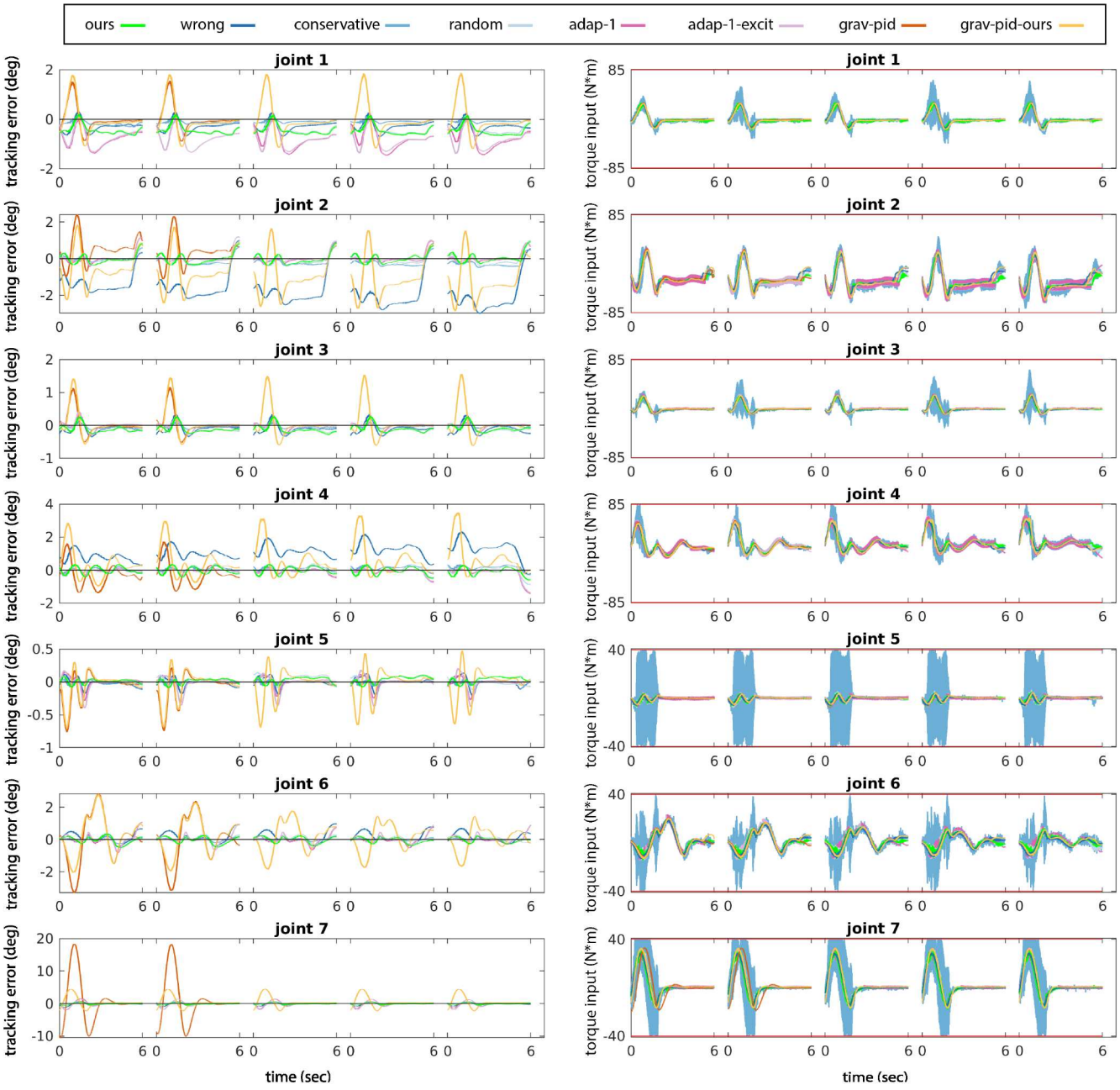

Fig. 12: This figure illustrates the tracking error of our method and all the comparisons on the left and the commanded torque on the right, while moving and stacking each of the five dumbbells on the 3D-printed platform located 0.50 m in front of the robot in Experiment (b).