

Multi-Contact Inertial Estimation and Localization in Legged Robots

Sergi Martinez¹ Robert Griffin² Carlos Mastalli^{1,2}

Abstract—Optimal estimation is a promising tool for multi-contact inertial estimation and localization. To harness its advantages in robotics, it is crucial to solve these large and challenging optimization problems efficiently. To tackle this, we (i) develop a multiple-shooting solver that exploits both temporal and parametric structures through a parametrized Riccati recursion. Additionally, we (ii) propose an *inertial local manifold* that ensures its full physical consistency. It also enhances convergence compared to the singularity-free log-Cholesky approach. To handle its singularities, we (iii) introduce a nullspace approach in our optimal estimation solver. We (iv) finally develop the analytical derivatives of contact dynamics for both inertial parametrizations. Our framework can successfully solve estimation problems for complex maneuvers such as brachiation in humanoids. We demonstrate its numerical capabilities across various robotics tasks and its benefits in experimental trials with the Go1 robot.

I. INTRODUCTION

Optimal estimation (OE) [1] emerges as a powerful tool for interpreting observations and accurately estimating a system’s true state, including internal changes like unknown payloads. In the context of robotics, this framework takes into account both proprioceptive and exteroceptive observations [2], [3]. It systematically incorporates considerations such as robot dynamics and nonholonomics [4], balance conditions [5], and kinematic range. Its application in robotics holds promising potentials. For instance, it could empower humanoid robots to manipulate unknown objects or to execute agile maneuvers while carrying an unknown payload (Fig. 1).

Algorithms for optimal estimation solve variations of the following problem:

$$\begin{aligned} \min_{\mathbf{x}_s, \mathbf{w}_s, \boldsymbol{\theta}} & \frac{1}{2} \|\mathbf{x}_0 \ominus \bar{\mathbf{x}}_0\|_{\Sigma_{\mathbf{x}_0}^{-1}}^2 + \frac{1}{2} \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_{\Sigma_{\boldsymbol{\theta}}^{-1}}^2 \\ & + \frac{1}{2} \sum_{k=0}^{N-1} \|\mathbf{w}_k\|_{\Sigma_{\mathbf{w}_N}^{-1}}^2 + \frac{1}{2} \sum_{k=1}^N \|\hat{\mathbf{z}}_k \ominus \mathbf{h}(\mathbf{x}_k; \boldsymbol{\theta} | \hat{\mathbf{u}}_k)\|_{\Sigma_{\mathbf{z}_k}^{-1}}^2 \\ \text{s.t. } & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k; \boldsymbol{\theta} | \hat{\mathbf{u}}_k) \oplus \mathbf{w}_k, \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ represents the system’s state, $\mathbf{w} \in \mathcal{T}_{\mathbf{x}} \mathcal{X} \subseteq \mathbb{R}^{n_x}$ describes its uncertainty, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are static parameters defining the system’s inertial properties, $\hat{\mathbf{u}}_k$ denotes the applied (and known) control commands, $\hat{\mathbf{z}} \in \mathcal{Z} \subseteq \mathbb{R}^{n_z}$ are observations, and \mathbf{x}_0 is the arrival state. Eq. (1) aims at finding the *maximum a-posteriori estimate* $P(\mathbf{x}_s, \mathbf{w}_s, \boldsymbol{\theta} | \hat{\mathbf{z}}_s)$ as the state, parameters, uncertainties, and observations correspond to Gaussian distributions, i.e., $\mathbf{x}_k \sim \mathcal{N}(\bar{\mathbf{x}}_k, \Sigma_{\mathbf{x}_k})$, $\boldsymbol{\theta} \sim \mathcal{N}(\bar{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{w}_k})$, and $\hat{\mathbf{z}}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{z}_k})$, respectively. The term $\frac{1}{2} \|\mathbf{x}_0 \ominus \bar{\mathbf{x}}_0\|_{\Sigma_{\mathbf{x}_0}^{-1}}^2$ describes the *arrival cost*, where \ominus is the *difference operator* used to optimize over state

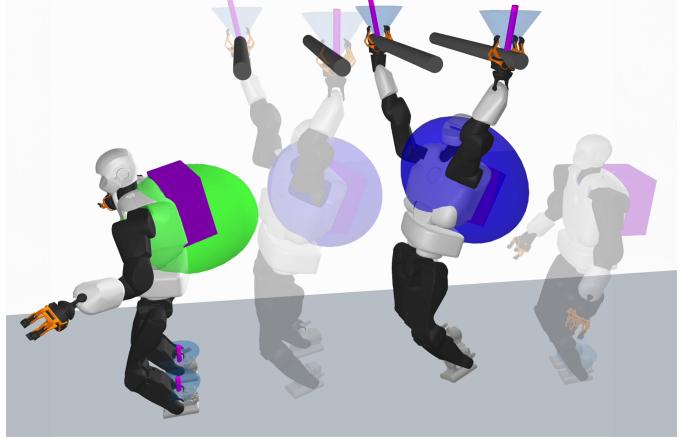


Fig. 1: Talos executing intricate monkey bar maneuvers with an unidentified payload. In the initial stages, our estimator meticulously pinpoints Talos’ localization and estimates its payload’s properties: mass, barycenter, and rotational inertia. The ellipsoid’s dimensions, position, radius, and color serve as visual indicators, representing estimations of the payload’s mass, barycenter, principal components of inertia, and algorithm convergence. To watch the video, click the picture or see <https://youtu.be/J4pqAU708Ew>.

manifolds [6], [7], [8]. Additionally, the term $\frac{1}{2} \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_{\Sigma_{\boldsymbol{\theta}}^{-1}}^2$ specifies the uncertainties in the system’s parameters. Finally, $\mathbf{f} : \mathcal{X} \times \mathbb{R}^{n_\theta} \rightarrow \mathcal{T}_{\mathbf{x}} \mathcal{X}$ and $\mathbf{h} : \mathcal{X} \times \mathbb{R}^{n_\theta} \rightarrow \mathcal{Z}$ are nonlinear functions describing the system dynamics and measurements.

A. Related work

To address optimal estimation in robotics, one can leverage well-established *direct methods* [9], which transcribes Eq. (1) into a nonlinear programming (NLP) problem. Direct methods involve the discretization of both state and uncertainties, followed by optimization using sparse general-purpose NLP software such as SNOPT [10], KNITRO [11], and IPOPT [12]. These software rely on sparse linear solvers such as MA27, MA57, and MA97 (see [13]) to factorize the large Karush-Kuhn-Tucker (KKT) problem. However, a limitation of these linear solvers is their inefficiency in exploiting the Markovian structure of OE problems [14], restricting their applicability in real-time applications, especially in legged robotics. These computational limitations are attributed to their difficulties in utilizing data cache accesses efficiently, resulting in the exclusion of OE strategies. Indeed, recent works focused on factor graphs formulations [15], [16], [17] are restricted to localization approaches only while ignoring the robot’s dynamics. In contrast, our approach considers the robot’s dynamics, thereby reducing drift errors in *proprioceptive localization*.

Alternatively, we can leverage Bellman’s principle of optimality [18] to break the optimal estimation problem into a sequence of smaller subproblems. This approach effectively exploits the Markovian structure, resulting in a differential

¹ Robot Motor Intelligence (RoMI) Lab – Heriot-Watt University, UK.

² IHMC Robotics – Florida Institute for Human & Machine Cognition, US.

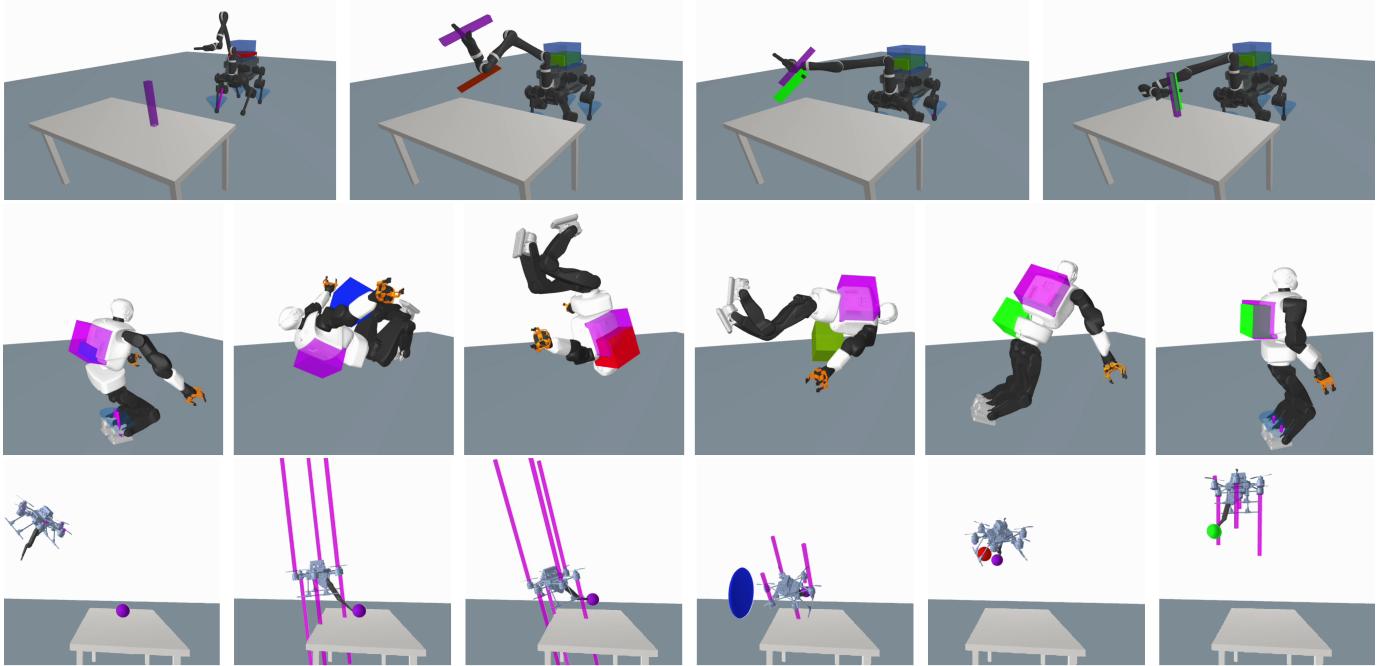


Fig. 2: Snapshots illustrating simulated online multi-contact inertial estimation and localization. (top) ANYmal with a Kinova arm carries an unknown payload while grasping an unknown object. (middle) Talos performs a backflip while simultaneously estimating its backpack. (bottom) Borinot grasping a sphere with a symmetric inertia (only converges thanks to the *null-space* parametrization). Unknown payloads are represented in magenta, and upon convergence of the estimator, the estimated payload is depicted in green. To watch the video, click the picture or see <https://youtu.be/J4pqAU708Ew>.

dynamic programming (DDP) approach for optimal estimation [19]. Additionally, by augmenting the system’s state with its internal parameters, or parametrizing the dynamics, we can simultaneously solve identification and localization problems via DDP [20]. However, these DDP approaches, being *single shooting* algorithms, are prone to numerical instability and require a good initialization [9], both crucial considerations for their application in robotics. These numerical instabilities, arise from enforcing dynamic feasibility, can be mitigated through *feasibility-driven* methods [14], [21] or *multiple shooting* strategies [22], [23] as proposed in model predictive control (MPC) literature. Moreover, DDP approaches for optimal estimation presuppose knowledge of the *arrival state*, a condition not attainable in real-world applications. In contrast, our framework incorporates the arrival state estimation.

To estimate inertial parameters in robotics, two key aspects must be leveraged: (i) the affine relationship between these parameters and the generalized torques, as initially developed in [24], and (ii) the analytical derivatives of rigid body algorithms [25], [26]. The second aspect is particularly relevant to us, as the derivatives of forward dynamics in [25] rely on the inertial matrix method (IMM) outlined in [27]. This is because IMMs involve Cholesky decompositions of the joint-space inertia matrix, limiting their operation to inertial parameters that are *fully physically consistent*.

Conditions for full physical consistency boils down to triangle inequality constraints [28]. However, numerical optimization guarantees inequality feasibilities at convergence, limiting its application to IMMs. Alternatively, these conditions can be embedded in a smooth manifold defined via a log-Cholesky parametrization [29]. This manifold, being singularity-free,

exhibits highly nonlinear geometry. Nevertheless, it is possible to build a smoother manifold by developing an OE solver that handles these singularities. These ideas lead us to a novel *exponential eigenvalue local manifold*, with a better convergence rate, coupled with a nullspace resolution (Fig. 2).

B. Contribution

Our main contribution is an efficient approach for solving hybrid optimal estimation problems in inertial estimation and localization. It relies on four technical contributions:

- (i) a novel smooth manifold with local submersion that guarantees full physical consistency of inertial parameters,
- (ii) an algorithm for analytical derivatives of the hybrid contact dynamics regarding the inertial parameters,
- (iii) a nullspace approach for handling singularities encountered in inertial identification, and
- (iv) a multiple-shooting algorithm for parametrized DDP with hybrid contact dynamics.

Our optimal estimation framework is the first of its class, even when considering the log-Cholesky parametrization (see Fig. 2 and <https://youtu.be/J4pqAU708Ew>). This is because previous work [29] focuses solely on inertial identification. Additionally, it relies on a novel multiple-shooting solver that combines nullspace parametrization to handle the exponential-eigenvalue singularities. Such singularities occur when the principal components of inertia at barycenter are the same, as in a solid sphere or a disk with uniform density.

II. BACKGROUND

In this section, we provide an introduction to contact dynamics, inertial parameters, and the conditions for their full physical consistency.

A. Contact dynamics

The dynamics of rigid body systems, subject to contact constraints at the acceleration level, are governed by:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \boldsymbol{\tau}(\mathbf{u}) - \mathbf{h}(\mathbf{q}, \mathbf{v}) + \mathbf{J}_c(\mathbf{q})^\top \boldsymbol{\lambda}_c, \quad (2)$$

$$\mathbf{J}_c(\mathbf{q})\dot{\mathbf{v}} = -\mathbf{a}_c(\mathbf{q}, \mathbf{v}), \quad (3)$$

where $\mathbf{q} \in \mathcal{Q} \subseteq \mathbb{R}^{n_q}$, $\mathbf{v} \in \mathcal{T}_{\mathbf{q}}\mathcal{Q} \subseteq \mathbb{R}^{n_q}$, $\mathbf{u} \in \mathbb{R}^{n_u}$, and $\boldsymbol{\lambda}_c \in \mathbb{R}^{n_c}$ represent the configuration point, generalized velocity, control commands, and contact forces, respectively. The functions $\mathbf{M} : \mathcal{Q} \rightarrow \mathbb{R}^{n_q \times n_q}$ represents the joint-space inertia matrix, $\boldsymbol{\tau} : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_q}$ denotes the joint-generalized torque, $\mathbf{J}_c : \mathcal{T}_{\mathbf{q}}\mathcal{Q} \rightarrow \mathbb{R}^{n_c \times n_q}$ is the contact Jacobian, and $\mathbf{h} : \mathcal{Q} \times \mathcal{T}_{\mathbf{q}}\mathcal{Q} \rightarrow \mathbb{R}^{n_q}$ is a force vector containing the Coriolis, centrifugal and gravitational terms. In the absence of contacts, free dynamics are governed by Eq. (2) with $\boldsymbol{\lambda}_c = \mathbf{0}$.

Free dynamics are frequently computed using the articulated body algorithm (ABA) [27]. In contrast, contact dynamics are commonly computed using a Schur-complement approach based on the IMM [30]. Moreover, analytical derivatives of contact dynamics with respect to \mathbf{q} , \mathbf{v} and $\boldsymbol{\tau}$ are computed based on the algorithms described in [25], as explained in [7].

B. Inertial parameters of rigid bodies

The spatial inertia of a rigid body i , encapsulating the body's mass-density field $\rho : \mathbb{R}^3 \rightarrow \mathbb{R}_{\geq 0}$, can be defined using a vector $\boldsymbol{\pi}_i \in \mathbb{R}^{10}$, whose elements are

$$\boldsymbol{\pi}_i = [m \ h_x \ h_y \ h_z \ I_{xx} \ I_{xy} \ I_{yy} \ I_{xz} \ I_{yz} \ I_{zz}]^\top, \quad (4)$$

where $m \in \mathbb{R}$ denotes the body's mass, $\mathbf{h} = [h_x \ h_y \ h_z]^\top = m\mathbf{c} \in \mathbb{R}^3$ represents the first mass moment with \mathbf{c} as its barycenter, and $[I_{xx} \ I_{xy} \ I_{yy} \ I_{xz} \ I_{yz} \ I_{zz}]^\top$ are the elements of its inertia matrix $\mathbf{I} \in \mathbb{R}^{3 \times 3}$. Both barycenter and rotational inertia are expressed in the body-fixed reference frame as introduced in [24]. Moreover, the rotational inertia \mathbf{I} can be expressed at the barycenter using the parallel axis theorem:

$$\mathbf{I}_c = \mathbf{I} - \frac{\mathbf{S}(\mathbf{h})\mathbf{S}(\mathbf{h})^\top}{m}, \quad (5)$$

where $\mathbf{S} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is the skew-symmetric matrix, also known as the Lie algebra of $SO(3)$ (see [8]).

As described in [24], the generalized torques can be expressed as an affine function of the inertial parameters $\boldsymbol{\pi}$, i.e.,

$$\boldsymbol{\tau}(\mathbf{u}) = \mathbf{Y}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}})\boldsymbol{\pi}, \quad (6)$$

where $\mathbf{Y} : \mathcal{Q} \times \mathcal{T}_{\mathbf{q}}\mathcal{Q} \times \mathcal{T}_{\mathbf{q}}\mathcal{Q} \rightarrow \mathbb{R}^{n_q \times 10n_b}$ is the joint-torque regressor matrix with n_b as the number of rigid bodies.

C. Physically-consistent spatial inertia

We say that $\boldsymbol{\pi}_i$ is *fully physically consistent*, or $\boldsymbol{\pi}_i \in \mathcal{I}$, if there exists a finite-value density function ρ . The set $\mathcal{I} \subset \mathbb{R}^{10}$ is characterized by the following inequality constraints:

$$m \geq 0, \quad \mathbf{I}_c \succeq 0, \quad (7)$$

$$D_x < D_y + D_z, \quad D_y < D_x + D_z, \quad D_z < D_x + D_y.$$

Here, $\mathbf{D} = \text{diag}(D_x, D_y, D_z) \in \mathbb{R}^{3 \times 3}$ represents the principal moments of inertia, computed as $\mathbf{I}_c = \mathbf{R}\mathbf{D}\mathbf{R}^\top$ with $\mathbf{R} \in$

$SO(3)$. The condition $\mathbf{I}_c \succeq 0$ ensures positivity in the principal components of inertia, and the triangle inequalities enforce positivity in second moments of mass [28].

The rotational inertia \mathbf{I}_c can be interpreted as a point-mass distribution with covariance Σ_c . When mapping this to its spatial inertia, it results in a 4×4 pseudo-inertia matrix [31] given by:

$$\mathbf{J} = \begin{bmatrix} \Sigma & \mathbf{h} \\ \mathbf{h}^\top & m \end{bmatrix}, \quad \text{where } \Sigma = \frac{1}{2}\text{Tr}(\mathbf{I})\mathbf{1}_3 - \mathbf{I}, \quad (8)$$

$\Sigma = \Sigma_c + m\mathbf{c}\mathbf{c}^\top$, $\text{Tr}(\cdot)$ is the trace operator, and $\mathbf{1}_3$ is a 3×3 identity matrix. The pseudo-inertia matrix $\mathbf{J} \in \mathcal{S}_{++}^4$ must satisfy the condition $\mathbf{J} \succeq 0$ for full physical consistency.

III. PARAMETRIZATION OF INERTIAL PARAMETERS

To map parameters $\boldsymbol{\theta} \in \mathbb{R}^{10}$ to fully physically consistent parameters $\boldsymbol{\pi}_i \in \mathcal{I}$, we can construct a smooth isomorphic mapping $\psi : \mathbb{R}^{10} \rightarrow \mathcal{I}$. Below, we introduce the log-Cholesky parametrization and then proceed to develop our *exponential eigenvalue parametrization*.

A. Log-Cholesky parametrization

Log-Cholesky decompositions are applicable to semidefinite matrices, enabling us to encode the condition $\mathbf{J}_i \succeq 0$ for physical consistency of body i . As suggested in [29], a log-Cholesky parametrization of the pseudo inertia $\mathbf{J} = \mathbf{U}\mathbf{U}^\top$, with

$$\mathbf{U} = e^\alpha \begin{bmatrix} e^{d_1} & s_{12} & s_{13} & t_1 \\ 0 & e^{d_2} & s_{23} & t_2 \\ 0 & 0 & s_{23} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

can be employed to establish an isomorphic map $\boldsymbol{\pi} = \psi(\boldsymbol{\theta}_{\text{chol}})$. Here, $\psi := \text{vech}(\mathbf{U}\mathbf{U}^\top)$, where the vech operator denotes the serialization operation of \mathbf{J} , and

$$\boldsymbol{\theta}_{\text{chol}} = [\alpha \ d_1 \ d_2 \ d_3 \ s_{12} \ s_{23} \ s_{13} \ t_1 \ t_2 \ t_3]^\top \in \mathbb{R}^{10}. \quad (10)$$

B. Exponential eigenvalue parametrization

The singular value decomposition of \mathbf{I}_c can be expressed in terms of its second moments of mass:

$$\mathbf{I}_c = \mathbf{R} \text{diag}(\mathbf{PL}) \mathbf{R}^\top, \quad \text{where } \mathbf{P} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (11)$$

and $\mathbf{L} = [L_x \ L_y \ L_z] \in \mathbb{R}_{\geq 0}^3$ denotes the second moments of mass. Ensuring positivity in L_x , L_y , L_z is equivalent to satisfying both $\mathbf{I}_c \succeq 0$ and the triangle inequalities in Eq. (7). Intuitively, these conditions on L_x , L_y , L_z , and $m \geq 0$ can be embedded using an exponential map. Formally, their manifold constraints are

$$\text{diag}(\mathbf{L}) \cdot \text{diag}(\mathbf{L})^{-1} = \mathbf{1}_3, \quad m \cdot m^{-1} = 1, \quad (12)$$

defining the real numbers as their Lie algebra structure. Therefore, our parametrization is given by

$$m = \exp(\sigma_m), \quad \mathbf{L} = [\exp(\sigma_x) \ \exp(\sigma_y) \ \exp(\sigma_z)]^\top, \quad (13)$$

where σ_m and $(\sigma_x, \sigma_y, \sigma_z)$ denote the mass and rotational inertia parameters, respectively. Moreover, the structure of the

rotational matrix \mathbf{R} is guaranteed by parametrizing it via the Lie algebra of $SO(3)$, i.e.,

$$\mathbf{R} = \text{Exp}(\boldsymbol{\omega}), \quad (14)$$

where $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z] \in \mathbb{R}^3 \cong \mathfrak{so}(3)$, and $\text{Exp} : \mathbb{R}^3 \rightarrow SO(3)$ converts this vector elements to the elements of the $SO(3)$ group. Moreover, we obtain the inertia \mathbf{I} via Eq. (5).

Combining the exponential maps and stacking them into a vector, we construct a map with local submersion $\boldsymbol{\pi} = \psi(\boldsymbol{\theta}_{\text{eval}})$, where

$$\boldsymbol{\theta}_{\text{eval}} = [\sigma_m \ h_x \ h_y \ h_z \ \omega_x \ \omega_y \ \omega_z \ \sigma_x \ \sigma_y \ \sigma_z]^T \in \mathbb{R}^{10}. \quad (15)$$

Finally, we can analytically compute $\frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}}$ for both $\boldsymbol{\theta}_{\text{chol}}$ and $\boldsymbol{\theta}_{\text{eval}}$, as they are smooth and differentiable parametrizations.

IV. MULTI-CONTACT OPTIMAL ESTIMATION

Optimal estimation with hybrid events involves the use of multiphase dynamics and reset maps:

$$\min_{\mathbf{x}_s, \mathbf{w}_s, \boldsymbol{\theta}} \ell_N(\mathbf{x}_N; \boldsymbol{\theta} | \hat{\mathbf{z}}_N) + \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k, \mathbf{w}_k; \boldsymbol{\theta} | \hat{\mathbf{z}}_k) \quad (16)$$

s.t. for $p \in \mathcal{P}$:

for $k \in \{p_0, p_0 + 1, \dots, p_N - 1\}$:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_p(\mathbf{x}_k; \boldsymbol{\theta} | \hat{\mathbf{u}}_k) \oplus \mathbf{w}_k && \text{(phase dyn.)} \\ \mathbf{x}_{p_{N+1}} &= \Delta_p(\mathbf{x}_{p_N}; \boldsymbol{\theta}), && \text{(reset map)} \end{aligned}$$

where $\mathbf{f}_p : \mathcal{X} \times \mathbb{R}^{n_\theta} \rightarrow \mathcal{T}_{\mathbf{x}} \mathcal{X}$ describes the contact dynamics in phase p , and $\Delta_p : \mathcal{X} \times \mathbb{R}^{n_\theta} \rightarrow \mathcal{T}_{\mathbf{x}} \mathcal{X}$ defines its contact-gain transition (modelled via impulse dynamics). Below, we develop their analytical derivatives w.r.t. $\boldsymbol{\theta}$. For their analytical derivatives w.r.t. $\mathbf{x} = (\mathbf{q}, \mathbf{v})$, we refer to [7], [32].

A. Analytical derivatives of parametrized dynamics

In free dynamics, inertial parameters $\boldsymbol{\pi}$ exhibit a linear relationship with the generalized torques. Additionally, the property $\partial \text{FD} = \mathbf{M}^{-1} \partial \text{ID}$ holds, given that the forward dynamics (FD) is the reciprocal of inverse dynamics (ID) [25]. Therefore, by applying the chain rule, we compute the analytical derivatives of forward dynamics with respect to $\boldsymbol{\theta}$ as

$$\frac{\partial \text{FD}}{\partial \boldsymbol{\theta}} = \mathbf{M}(\mathbf{q})^{-1} \frac{\partial \text{ID}}{\partial \boldsymbol{\pi}} \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}} = \mathbf{M}(\mathbf{q})^{-1} \mathbf{Y}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}) \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}}. \quad (17)$$

Moving on to contact dynamics, and similarly to [7], we apply the chain rule to derive the analytical derivatives for both the system's acceleration and the contact forces:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \begin{bmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda}_c \end{bmatrix} = - \begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{J}_c(\mathbf{q})^\top \\ \mathbf{J}_c(\mathbf{q}) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}} \\ \mathbf{0} \end{bmatrix}, \quad (18)$$

where $\frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}} = \mathbf{Y}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}) \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}}$. To enhance efficiency, this matrix inversion is performed via its Schur complement. This complement requires computing \mathbf{M}^{-1} . Similar to free dynamics, we achieve this through the Cholesky decomposition of \mathbf{M} , an approach typically employed in IMMs. Moreover, these matrix inversions are computed for derivatives with respect to \mathbf{q} and \mathbf{v} , allowing for the reuse of computations and leveraging sparsity in Eq. (18).

Similarly, derivatives of impulse dynamics [7] involves inverting the same matrix but with:

$$\begin{aligned} \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{\theta}} &:= \left(\frac{\partial \text{ID}(\mathbf{q}, \mathbf{0}, \mathbf{v}^+ - \mathbf{v})}{\partial \boldsymbol{\pi}} - \frac{\partial \text{ID}(\mathbf{q}, \mathbf{0}, \mathbf{0})}{\partial \boldsymbol{\pi}} \right) \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}}, \\ &= (\mathbf{Y}(\mathbf{q}, \mathbf{0}, \mathbf{v}^+ - \mathbf{v}) - \mathbf{Y}(\mathbf{q}, \mathbf{0}, \mathbf{0})) \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\theta}}, \end{aligned} \quad (19)$$

where $\mathbf{v}^+ \in \mathcal{T}_{\mathbf{q}} \mathcal{Q} \subseteq \mathbb{R}^{n_q}$ is the post impact velocity.

V. DDP WITH PARAMETRIZED DYNAMICS

Handling parametrized dynamics requires to introduce additional decision variables $\boldsymbol{\theta}$. To exploit its *temporal structure*, we begin by examining the optimality conditions through the lens of Bellman, ultimately leading to a parametrized Riccati recursion.

A. Optimality conditions

By examining the Bellman equation of Eq. (16), specifically,

$$\begin{aligned} \mathcal{V}(\mathbf{x}; \boldsymbol{\theta} | \hat{\mathbf{u}}, \hat{\mathbf{z}}) &= \min_{\mathbf{x}', \mathbf{x}, \mathbf{w}; \boldsymbol{\theta}} \ell(\mathbf{x}, \mathbf{w}; \boldsymbol{\theta} | \hat{\mathbf{z}}) + \mathcal{V}'(\mathbf{x}'; \boldsymbol{\theta} | \hat{\mathbf{u}}, \hat{\mathbf{z}}) \\ &\quad \text{s.t. } \mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{w}; \boldsymbol{\theta} | \hat{\mathbf{u}}), \end{aligned} \quad (20)$$

we break the optimal estimation problem into a sequence of subproblems. The KKT point for each subproblem can be efficiently determined using the Newton method, yielding the following linear system of equations in each iteration:

$$\begin{bmatrix} \mathcal{L}_{\mathbf{xx}} & \mathcal{L}_{\mathbf{xw}} & \mathcal{L}_{\mathbf{x}\boldsymbol{\theta}} & \mathbf{f}_x^\top \\ \mathcal{L}_{\mathbf{xw}}^\top & \mathcal{L}_{\mathbf{ww}} & \mathcal{L}_{\mathbf{w}\boldsymbol{\theta}} & \mathbf{f}_w^\top \\ \mathcal{L}_{\mathbf{x}\boldsymbol{\theta}}^\top & \mathcal{L}_{\mathbf{w}\boldsymbol{\theta}}^\top & \mathcal{L}_{\boldsymbol{\theta}\boldsymbol{\theta}} & \mathbf{f}_{\boldsymbol{\theta}}^\top \\ \mathbf{f}_x & \mathbf{f}_w & \mathbf{f}_{\boldsymbol{\theta}} & -\mathbf{I} \\ \mathcal{V}'_{\mathbf{xx}} & \mathcal{V}'_{\mathbf{xw}} & -\mathbf{I} & \mathcal{V}'_{\mathbf{xx}}^\top \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \mathbf{w} \\ \delta \boldsymbol{\theta} \\ \boldsymbol{\xi}^+ \\ \delta \mathbf{x}' \end{bmatrix} = - \begin{bmatrix} \ell_x \\ \ell_w \\ \ell_{\boldsymbol{\theta}} + \mathcal{V}'_{\boldsymbol{\theta}} \\ \bar{\mathbf{f}} \\ \mathcal{V}'_x \end{bmatrix} \quad (21)$$

with:

$$\begin{aligned} \boldsymbol{\xi}^+ &:= \boldsymbol{\xi} + \delta \boldsymbol{\xi}, & \bar{\mathbf{f}} &:= \mathbf{f}(\mathbf{x}, \mathbf{w}; \boldsymbol{\theta}) \ominus \mathbf{x}', \\ \mathcal{L}_{\mathbf{xx}} &:= \ell_{\mathbf{xx}} + \mathcal{V}'_x \cdot \mathbf{f}_{\mathbf{xx}}, & \mathcal{L}_{\mathbf{xw}} &:= \ell_{\mathbf{xw}} + \mathcal{V}'_x \cdot \mathbf{f}_{\mathbf{xw}}, \\ \mathcal{L}_{\mathbf{x}\boldsymbol{\theta}} &:= \ell_{\mathbf{x}\boldsymbol{\theta}} + \mathcal{V}'_x \cdot \mathbf{f}_{\mathbf{x}\boldsymbol{\theta}}, & \mathcal{L}_{\mathbf{ww}} &:= \ell_{\mathbf{ww}} + \mathcal{V}'_x \cdot \mathbf{f}_{\mathbf{ww}}, \\ \mathcal{L}_{\mathbf{w}\boldsymbol{\theta}} &:= \ell_{\mathbf{w}\boldsymbol{\theta}} + \mathcal{V}'_x \cdot \mathbf{f}_{\mathbf{w}\boldsymbol{\theta}}, & \mathcal{L}_{\boldsymbol{\theta}\boldsymbol{\theta}} &:= \ell_{\boldsymbol{\theta}\boldsymbol{\theta}} + \mathcal{V}'_{\boldsymbol{\theta}\boldsymbol{\theta}} + \mathcal{V}'_x \cdot \mathbf{f}_{\boldsymbol{\theta}\boldsymbol{\theta}}. \end{aligned}$$

Here, ℓ_p , \mathbf{f}_p , are the first derivative of the cost and the system dynamics with respect to \mathbf{p} , with \mathbf{p} a hypothetical decision variable that represents \mathbf{x} , \mathbf{w} or $\boldsymbol{\theta}$; ℓ_{pp} , \mathbf{f}_{pp} are the second derivatives; \mathcal{V}'_p is the gradient of the value function; \mathcal{V}'_{pp} is the Hessian of the value function; $\bar{\mathbf{f}}$ describe the gaps or infeasibility in the dynamics; $\delta \mathbf{x}$, $\delta \mathbf{w}$, $\delta \boldsymbol{\theta}$, $\delta \mathbf{x}'$ and $\delta \boldsymbol{\xi}$ provides the search direction computed for the primal and dual variables, respectively; and the prime superscript is used to refer to the next node.

Similar to optimal control [14, Section 2.2], we observe the presence of the Markovian structure, leading to $\boldsymbol{\xi}^+ = \mathcal{V}'_x + [\mathcal{V}'_{\mathbf{xx}} \ \mathcal{V}'_{\mathbf{x}\boldsymbol{\theta}}] \begin{bmatrix} \delta \mathbf{x}' \\ \delta \boldsymbol{\theta} \end{bmatrix}$. This establishes a connection between the KKT conditions with the Pontryagin's maximum principle [18] because $\boldsymbol{\xi}^+$ assumes the role of the *next costate*. This connection becomes evident when augmenting the system's state with its internal parameters, contributing to the

condensation of the system of equations in Eq. (21) into:

$$\begin{bmatrix} \mathbf{Q}_{xx} & \mathbf{Q}_{xw} & \mathbf{Q}_{x\theta} \\ \mathbf{Q}_{xw}^T & \mathbf{Q}_{ww} & \mathbf{Q}_{w\theta} \\ \mathbf{Q}_{x\theta}^T & \mathbf{Q}_{w\theta}^T & \mathbf{Q}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta w \\ \delta \theta \end{bmatrix} = - \begin{bmatrix} \mathbf{Q}_x \\ \mathbf{Q}_w \\ \mathbf{Q}_\theta \end{bmatrix}, \quad (22)$$

where the \mathbf{Q} 's terms represent the local approximation of the *action-value function* whose expressions are:

$$\begin{aligned} \mathbf{Q}_{xx} &= \mathcal{L}_{xx} + \mathbf{f}_x^T \mathcal{V}'_{xx} \mathbf{f}_x, & \mathbf{Q}_{w\theta} &= \mathcal{L}_{w\theta} + \mathbf{f}_w^T (\mathcal{V}'_{x\theta} + \mathcal{V}'_{xx} \mathbf{f}_\theta), \\ \mathbf{Q}_{xw} &= \mathcal{L}_{xw} + \mathbf{f}_x^T \mathcal{V}'_{xx} \mathbf{f}_w, & \mathbf{Q}_{x\theta} &= \mathcal{L}_{x\theta} + \mathbf{f}_x^T (\mathcal{V}'_{x\theta} + \mathcal{V}'_{xx} \mathbf{f}_\theta), \\ \mathbf{Q}_{ww} &= \mathcal{L}_{ww} + \mathbf{f}_w^T \mathcal{V}'_{xx} \mathbf{f}_w, & \mathbf{Q}_{\theta\theta} &= \mathcal{L}_{\theta\theta} + \mathbf{f}_\theta^T (2\mathcal{V}'_{x\theta} + \mathcal{V}'_{xx} \mathbf{f}_\theta), \\ \mathbf{Q}_x &= \ell_x + \mathbf{f}_x^T \mathcal{V}_x^+, & \mathbf{Q}_\theta &= \ell_\theta + \mathcal{V}_\theta^+ + \mathbf{f}_\theta^T \mathcal{V}_x^+, \\ \mathbf{Q}_w &= \ell_w + \mathbf{f}_w^T \mathcal{V}_w^+ \end{aligned} \quad (23)$$

with $\mathcal{V}_x^+ := \mathcal{V}'_x + \mathcal{V}'_{xx} \bar{\mathbf{f}}$ and $\mathcal{V}_\theta^+ := \mathcal{V}'_\theta + \mathcal{V}'_{x\theta} \bar{\mathbf{f}}$ as the gradients of the value function after the deflection produced by the dynamics infeasibility $\bar{\mathbf{f}}$ (see [7]). To exploit the *parametric structure*, we then compute the estimation policy and value function as functions of the parameters and arrival state.

B. Policy and value function

The system of linear equations in Eq. (22) can be interpreted as a quadratic program that locally minimizes the Bellman equation Eq. (20), i.e.,

$$\begin{aligned} \delta \mathcal{V}(\delta x; \delta \theta | \hat{\mathbf{u}}, \hat{\mathbf{z}}) &\simeq \\ \min_{\delta w} \frac{1}{2} \left[\begin{array}{c} 1 \\ \delta x \\ \delta w \\ \delta \theta \end{array} \right]^T \begin{bmatrix} \mathbf{Q}_x & \mathbf{Q}_x^T & \mathbf{Q}_w^T & \mathbf{Q}_\theta^T \\ \mathbf{Q}_x & \mathbf{Q}_{xx} & \mathbf{Q}_{xw} & \mathbf{Q}_{x\theta} \\ \mathbf{Q}_w & \mathbf{Q}_{xw}^T & \mathbf{Q}_{ww} & \mathbf{Q}_{w\theta} \\ \mathbf{Q}_\theta & \mathbf{Q}_{x\theta}^T & \mathbf{Q}_{w\theta}^T & \mathbf{Q}_{\theta\theta} \end{bmatrix} \begin{bmatrix} 1 \\ \delta x \\ \delta w \\ \delta \theta \end{bmatrix}. \end{aligned} \quad (24)$$

This leads to what we call the *DDP approach* for optimal estimation. The term is chosen because it computes the changes in the *estimation policy* δw as a function of changes in the system's state δx and its internal parameters $\delta \theta$ given observations $(\hat{\mathbf{u}}, \hat{\mathbf{z}})$, i.e.,

$$\delta w = -\mathbf{k} - \mathbf{K} \delta x - \mathbf{P} \delta \theta, \quad (25)$$

where $\mathbf{k} = \mathbf{Q}_{ww}^{-1} \mathbf{Q}_w$, $\mathbf{K} = \mathbf{Q}_{ww}^{-1} \mathbf{Q}_{xw}^T$ and $\mathbf{P} = \mathbf{Q}_{ww}^{-1} \mathbf{Q}_{w\theta}$ are the feed-forward and feedback terms. In this approach, when plugging the changes in the estimation into Eq. (24), we obtain a quadratic approximation of the *optimal arrival cost* (a.k.a. value function) as follows

$$\begin{aligned} \delta \mathcal{V}(\delta x; \delta \theta | \hat{\mathbf{u}}, \hat{\mathbf{z}}) &\simeq \Delta \mathcal{V}_1 + \frac{\Delta \mathcal{V}_2}{2} \\ &+ \frac{1}{2} \left[\begin{array}{c} \delta x \\ \delta \theta \end{array} \right]^T \begin{bmatrix} \mathcal{V}_{xx} & \mathcal{V}_{x\theta} \\ \mathcal{V}_{x\theta}^T & \mathcal{V}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta \theta \end{bmatrix} + \begin{bmatrix} \mathcal{V}_x \\ \mathcal{V}_\theta \end{bmatrix}^T \begin{bmatrix} \delta x \\ \delta \theta \end{bmatrix}, \end{aligned} \quad (26)$$

where

$$\begin{aligned} \Delta \mathcal{V}_1 &= -\mathbf{k}^T \mathbf{Q}_w, & \Delta \mathcal{V}_2 &= \mathbf{k}^T \mathbf{Q}_{ww} \mathbf{k}, \\ \mathcal{V}_x &= \mathbf{Q}_x - \mathbf{Q}_{xw} \mathbf{k}, & \mathcal{V}_\theta &= \mathbf{Q}_\theta - \mathbf{Q}_{w\theta}^T \mathbf{k}, \\ \mathcal{V}_{xx} &= \mathbf{Q}_{xx} - \mathbf{Q}_{xw} \mathbf{K}, & \mathcal{V}_{\theta\theta} &= \mathbf{Q}_{\theta\theta} - \mathbf{Q}_{w\theta}^T \mathbf{P}, \\ \mathcal{V}_{x\theta} &= \mathbf{Q}_{x\theta} - \mathbf{Q}_{xw} \mathbf{P}, \end{aligned} \quad (27)$$

and $\mathbf{Q}_{xw} \mathbf{P} = \mathbf{K}^T \mathbf{Q}_{w\theta}$.

The derivations of changes in the optimal policy and value function are expressed in terms of the search direction for the

system's parameters $\delta \theta$ and the arrival state δx_0 . To compute the search directions $\delta \theta$ and δx_0 , we analyze the conditions of optimality for the initial node as follows.

C. Arrival state and parameters

By substituting Eq. (25) into the condensed KKT equations of the initial or arrival node in Eq. (22), we obtain:

$$\begin{bmatrix} \mathcal{V}_{xx}^\bullet & \mathcal{V}_{x\theta}^\bullet \\ \mathcal{V}_{x\theta}^{\bullet T} & \mathcal{V}_{\theta\theta}^\bullet \end{bmatrix} \begin{bmatrix} \delta x_0 \\ \delta \theta \end{bmatrix} = - \begin{bmatrix} \mathcal{V}_x^\bullet \\ \mathcal{V}_\theta^\bullet \end{bmatrix}, \quad (28)$$

where the *bullet* superscript is used to refer to the arrival node (e.g., $\delta x_0 := \delta x^\bullet$).

1) *Schur-complement*: We can compute both $\delta \theta$ and δx_0 by factorizing Eq. (28) via the Schur-complement approach. This results in the following expressions:

$$\delta \theta = -\mathbf{k}_\theta - \mathbf{K}_\theta \delta x_0, \quad \delta x_0 := -\mathcal{V}_{xx_0}^{-1} \mathcal{V}_{x_0}, \quad (29)$$

where $\mathbf{k}_\theta = \mathcal{V}_{\theta\theta}^{\bullet -1} \mathcal{V}_\theta^\bullet$, $\mathbf{K}_\theta = \mathcal{V}_{\theta\theta}^{\bullet -1} \mathcal{V}_{x\theta}^{\bullet T}$ are the feed-forward and feedback terms of the parameters update,

$$\mathcal{V}_{x_0} = \mathcal{V}_x^\bullet - \mathcal{V}_{x\theta}^\bullet \mathbf{k}_\theta, \quad \mathcal{V}_{xx_0} = \mathcal{V}_{xx}^\bullet - \mathcal{V}_{x\theta}^\bullet \mathbf{K}_\theta, \quad (30)$$

are the derivatives of the value function associated to arrival node after updating the system's parameters, and \mathcal{V}_{xx_0} is of the Schur complement of Eq. (28). Now, we can rewrite the changes of the estimation policy as $\delta w = -\bar{\mathbf{k}} - \mathbf{K} \delta x$, where $\bar{\mathbf{k}} = \mathbf{k} + \mathbf{P} \delta \theta$ represents the entire feed-forward term.

2) *Nullspace*: Additionally, we can parametrize the search direction for the parameters as $\delta \theta = \mathbf{Y} \delta \theta_y + \mathbf{Z} \delta \theta_z$. Here, $\mathbf{Z} \in \mathbb{R}^{n_\theta \times n_z}$ is the nullspace basis of $\mathcal{V}_{\theta\theta}^\bullet$ and $\mathbf{Y} \in \mathbb{R}^{n_\theta \times n_y}$ is its orthogonal matrix. Then, by substituting this parametrization into the second term of Eq. (29) and observing that $\mathcal{V}_{\theta\theta}^\bullet \mathbf{Z} = \mathbf{0}$, we obtain

$$\mathbf{Y}^T \mathcal{V}_{\theta\theta}^\bullet \mathbf{Y} \delta \theta_y = -\overbrace{\mathbf{Y}^T \mathcal{V}_\theta^\bullet}^{\mathbf{k}_{\theta y}} - \overbrace{\mathbf{Y}^T \mathcal{V}_{x\theta}^{\bullet T}}^{\mathbf{K}_{\theta y}} \delta x_0, \quad (31)$$

where we pre-multiply by \mathbf{Y}^T to ensure the squareness of $\mathcal{V}_{\theta\theta}^\bullet \mathbf{Y}$. Eq. (31) is then solved efficiently using a Cholesky decomposition. Finally, we recover the search direction for the parameters as $\delta \theta = \mathbf{Y} \delta \theta_y$. Note that we compute δx_0 as in the second term of Eq. (29).

D. Forward rollouts

The search directions obtained from Eqs. (25), (29) and (31) provide updates for the system's uncertainties, parameters, and arrival state. To determine the step length along these directions, we perform a forward rollout of the disturbed system's dynamics and evaluate the associated costs with the *a-posteriori estimate*. Below, we develop two distinct approaches inspired by optimal control literature, known as feasibility-driven and multiple shooting. Let's start by describing how we update the arrival state and parameters first.

1) *Arrival state and parameters update*: We follow a line-search approach for updating the arrival state and system's parameters, i.e.,

$$\mathbf{x}_0^+ = \mathbf{x}_0 \oplus \alpha \delta \mathbf{x}_0, \quad \theta^+ = \theta + \alpha \delta \theta, \quad (32)$$

where $\alpha \in (0, 1]$ denotes the step length and the *plus* superscript is used to denote the next guess under evaluation. Next, we begin by describing our feasibility-driven approach.

2) *Feasibility-driven rollout*: Building upon classical line-search procedures (see [33, Chapter 3]), this approach tries the search direction along a *feasibility-driven* nonlinear rollout that preserves the gaps in the linear-dynamics sense [7]:

$$\begin{aligned}\bar{\mathbf{f}}_{k+1}^+ &= (\alpha - 1)\bar{\mathbf{f}}_{k+1}, \\ \mathbf{w}_k^+ &= \mathbf{w}_k - \alpha\bar{\mathbf{k}}_k - \mathbf{K}_k\delta\mathbf{x}_k^+, \\ \mathbf{x}_{k+1}^+ &= \mathbf{f}(\mathbf{x}_k^+, \mathbf{w}_k^+; \boldsymbol{\theta}^+) \oplus \bar{\mathbf{f}}_{k+1}^+, \end{aligned}\quad (33)$$

for $k = \{0, \dots, N\} \in \mathbb{N}$ and $\delta\mathbf{x}_k^+ = \mathbf{x}_k^+ \ominus \mathbf{x}_k$, where we employ backtracking procedure, and \mathbf{x}_k^+ , \mathbf{w}_k^+ represent the new guesses, for the states and uncertainties, under evaluation.

3) *Multiple shooting rollout*: Taking inspiration from multiple shooting approaches in optimal control, this approach tries a linear rollout and updates the gaps based on nonlinear shoots, i.e.,

$$\begin{aligned}\mathbf{w}_k^+ &= \mathbf{w}_k + \alpha\delta\mathbf{w}_k, \\ \mathbf{x}_{k+1}^+ &= \mathbf{x}_{k+1} \oplus \alpha\delta\mathbf{x}_{k+1}, \\ \bar{\mathbf{f}}_{k+1}^+ &= \mathbf{f}(\mathbf{x}_k^+, \mathbf{w}_k^+; \boldsymbol{\theta}_k^+) \ominus \mathbf{x}_{k+1}^+, \end{aligned}\quad (34)$$

for $k = \{0, \dots, N\} \in \mathbb{N}$. Compared to the feasibility-driven rollout in Eq. (33), multiple shooting rollouts can be broken into two parallel for loops: one for the update of the uncertainties and state, and another one for the update of the dynamics infeasibilities and costs. Below, we describe how we evaluate the goodness of a given step length α .

E. Merit function

Our merit function computes the expected cost reduction as a local approximation of the value function while being aware of the dynamics infeasibilities, i.e.,

$$\Delta\ell(\alpha) = \alpha \left(\Delta\mathcal{V}_1^\theta + \frac{1}{2}\alpha\Delta\mathcal{V}_2^\theta + \sum_{k=0}^{N-1} \left(\Delta\ell_{1_k} + \frac{1}{2}\alpha\Delta\ell_{2_k} \right) \right), \quad (35)$$

where, by considering that derivatives of the value function are deflected by the dynamics gaps, we have:

$$\Delta\ell_{1_k} = \Delta\mathcal{V}_{1_k} + \bar{\mathbf{f}}_k^\top \mathcal{V}_{\mathbf{x}_k}, \quad \Delta\ell_{2_k} = \Delta\mathcal{V}_{2_k} + \bar{\mathbf{f}}_k^\top \mathcal{V}_{\mathbf{x}\mathbf{x}_k} \bar{\mathbf{f}}_k, \quad (36)$$

with $\bar{\mathbf{f}}_0 = \delta\mathbf{x}_0$, $\Delta\mathcal{V}_1^\theta = \delta\boldsymbol{\theta}^\top \mathcal{V}_{\boldsymbol{\theta}}^\bullet$, and $\Delta\mathcal{V}_2^\theta = 2\bar{\mathbf{f}}_0^\top \mathcal{V}_{\boldsymbol{\theta}\boldsymbol{\theta}}^\bullet \delta\boldsymbol{\theta} + \delta\boldsymbol{\theta}^\top \mathcal{V}_{\boldsymbol{\theta}\boldsymbol{\theta}}^\bullet \delta\boldsymbol{\theta}$. It is noteworthy that Eq. (35) simplifies to the expected improvement formulation presented in [20], which employs a single shooting approach for optimal estimation.

We then compute a merit function of the form:

$$\phi(\mathbf{x}_s, \mathbf{w}_s, \boldsymbol{\theta}; \nu) = \phi(\mathbf{x}_s, \mathbf{w}_s, \boldsymbol{\theta}; \nu) = \ell(\mathbf{x}_N; \boldsymbol{\theta}) + \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, \mathbf{w}_k; \boldsymbol{\theta}) + \nu\epsilon(\mathbf{x}_k, \mathbf{w}_k; \boldsymbol{\theta}), \quad (37)$$

with $\epsilon(\mathbf{x}_k, \mathbf{w}_k; \boldsymbol{\theta}) := \|\bar{\mathbf{f}}_k\|_1$ measures the dynamics infeasibility of the current guess and

$$\nu \leftarrow \max \left(\beta^\nu \nu, \frac{\Delta\ell(1)}{(1-\rho) \sum_{k=0}^{N-1} \epsilon(\mathbf{x}_k, \mathbf{w}_k; \boldsymbol{\theta})} \right), \quad (38)$$

is a penalty parameter that balances optimality and feasibility, where $0 < \rho < 1$ is a tunable hyper-parameter and β^ν is

a factor used to reduce ν . The updating rule in Eq. (38) is inspired by [34], [21]. Finally, we utilize (1) a nonmonotone step acceptance strategy based on Armijo conditions and (2) a Levenberg-Marquardt scheme, similar to [21].

VI. RESULTS

We evaluate our multi-contact inertial estimation and localization framework through the following steps. First, we compare the numerical performance of both inertial parametrizations. Second, we analyze the effect of shooting approaches in optimal estimation. Third, we experimentally validate the advantages of updating inertial parameters for carrying unknown payloads. Finally, we show the importance of using hybrid dynamics for localization in legged robots. Additionally, we refer the reader to the video, where the performance of the null-space is evaluated.

A. Log-Cholesky vs Exponential Eigenvalue

Our numerical performance analysis encompasses four diverse robotics systems: the Kinova arm, an asymmetric quadrotor, the ANYmal quadruped, and the Talos biped. In each scenario, we generated dynamically-consistent trajectories. This simulated encoders and IMU data (linear acceleration, gyroscopic velocities, and orientations) were integrated into the observation model, denote as $\hat{\mathbf{z}} = \mathbf{h}(\mathbf{x}; \boldsymbol{\theta} | \hat{\mathbf{u}})$. Covariances for each observation, including the applied control $\hat{\mathbf{u}}$ for prediction, were meticulously considered. To ensure a fair comparison between the inertial parameterizations, we omitted the parameter regularization term $\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_{\Sigma_{\boldsymbol{\theta}}^{-1}}^2$. Moreover, we initialized the optimal estimators' inertial parameters for all the bodies with the same error of 70%. Finally, we configured the estimators with a feasibility-driven rollout. In each scenario illustrated in Fig. 3, both parametrization converge to the same estimated trajectory, resulting in the same estimation error. Nevertheless, the exponential eigenvalue parametrization demonstrated superior performance, characterized by faster

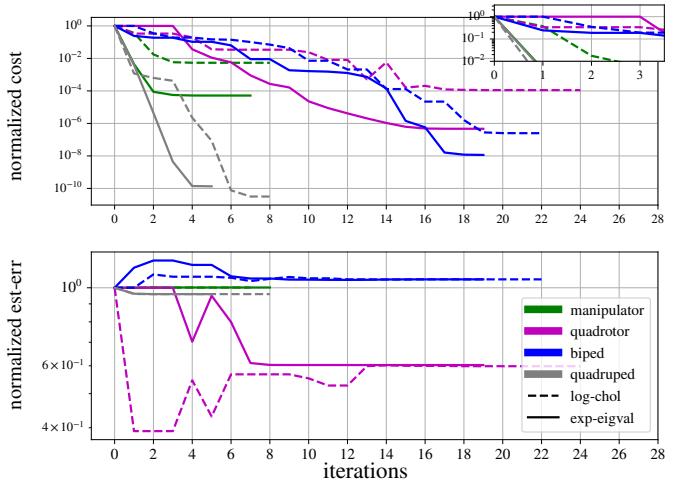


Fig. 3: Top: Local convergence for both parametrizations, showing better convergence for the *exponential-eigenvalue* (exp-eigval) parametrization in all cases. Bottom: Evolution of the estimation error, computed as the ℓ_1 -norm of difference between the optimized trajectory and the nominal one. Both parametrizations converge to the same local minima.

TABLE I: Rollout’s local convergence computed from 100 randomly initial guesses on three robotics systems.

		Iterations	Cost [$\cdot 10^{-1}$]	Error [ℓ_∞ -norm]
<i>Kinova</i>	single	39.7 ± 11.2	2.78 ± 0.89	3.11 ± 0.56
	feasible	44.7 ± 11.6	1.16 ± 0.42	3.12 ± 0.01
	multiple	99.4 ± 71.5	0.61 ± 0.42	2.67 ± 0.06
<i>Quadrotor</i>	single	140.9 ± 65.1	0.12 ± 0.04	10.74 ± 0.0023
	feasible	150.1 ± 59.3	0.26 ± 0.11	10.74 ± 0.0013
	multiple	16.9 ± 19.1	0.07 ± 1.31	10.74 ± 0.0002
<i>ANYmal</i>	single	X	X	X
	feasible	198.3 ± 16.6	0.22 ± 0.08	6.12 ± 0.92
	multiple	14.1 ± 14.8	0.01 ± 1.65	6.02 ± 0.22

Xalgorithm does not find a solution within 200 iterations.

convergence rates and often lower cost values. This can be attributed to its lower degree of nonlinearity compared to the log-Cholesky parametrization. Furthermore, the exponential eigenvalue parametrization offers a direct physical interpretation, proving advantageous, especially when partial knowledge of the rotational inertial is available. This interpretability facilitated easy tuning of lower covariances in known inertial parameters, contributing to enhanced precision and applicability in practical trials with the Go1 robot.

B. Numerical effect of rollout strategies

To evaluate the numerical impact of our feasibility-driven and multiple shooting rollouts, we randomly generated 100 initial guess and compared their convergence, cost and estimation error against single shooting approaches. In Table I, we observe that multiple shooting rollouts reduced the costs or estimation errors across all cases. Moreover, single shooting rollouts have a reduced basin of attraction to local minima and can encounter difficulties in convergence for challenging cases.

C. Validation on the Go1 robot

We experimentally validated our multi-contact inertial estimation and localization in trials with the Go1 robot. To showcase its capability in inertial estimation, we added an unknown payload of 7.2 kg to the robot’s torso, as depicted in Fig. 4. The observations recorded when the Go1’s predictive controller was unaware of the additional payload (Fig. 4-top). We integrated a variety of observation sources with a step integration interval of 1 ms. First, leveraging data from Go1’s encoders, we included joint positions $\mathbf{q}_j \in \mathbb{R}^{12}$ and numerically differentiated them to account for joint velocities $\mathbf{v}_j \in \mathbb{R}^{12}$. Second, we derived Go1’s torso rotation from the IMU’s sensor fusion algorithm. This was incorporated with the



Fig. 4: Go1 performing four walking gaits while carrying an unknown payload. In the top snapshots, Go1 struggled to maintain its posture due to model mismatch in the MPC. In the bottom snapshots, after correctly estimating the robot’s payload, Go1 confidently maintained its nominal posture.

following observation model $\|\hat{\mathbf{R}}_b \ominus \mathbf{R}_b\|_{\Sigma_R}^2$. Here, $\hat{\mathbf{R}}_b \ominus \mathbf{R}_b$ denotes the SO(3) inverse composition between the measured body orientation $\hat{\mathbf{R}}_b$ and the estimated one \mathbf{R}_b and $\Sigma_R \in S(3)$ represents the covariance estimated by the sensor fusion algorithm. Third, we pre-integrated the IMU’s linear acceleration to obtain the torso’s linear velocities. This observation also encompasses the IMU’s angular velocities, both of them are expressed in local coordinates for efficiency reasons. Lastly, we incorporated the torque commands $\hat{\mathbf{u}} \in \mathbb{R}^{12}$ applied by the state-feedback controller and measured contact sequences.

1) *Estimating an unknown payload:* The optimal estimator ran offline, estimating the inertial parameters after 27 iterations. Within the estimation horizon of 14 s, it accurately determined the total payload inertia, equivalent to a mass of 7.364 kg. Subsequently, we updated the inertial parameters of Go1’s torso in its full-dynamics MPC. The impact of this update is vividly illustrated in Fig. 4-bottom. Concretely, it enabled the Go1 robot to maintain a stable posture during walking, emphasizing the significance of correcting model mismatches even for a fast MPC running at 50 Hz.

The payload estimation significantly influences the tracking capabilities of the swing foot, as depicted in Fig. 5. A comparison between swing trajectories with and without inertial estimation reveals the crucial role it plays. In the absence of accurate inertial parameters, the MPC struggled to track reference swing trajectories, resulting in a progressively decaying motion (Fig. 4-top). Instead, when the inertia was estimated, the MPC adeptly tracked the swing trajectory.

2) *Localization with hybrid dynamics:* We compared our multi-contact localization approach against PRONTO [35]—a widely adopted localization framework for legged robots based on extended Kalman filter (EKF) and leg kinematics. To assess the impact of incorporating hybrid dynamics, PRONTO and the optimal estimator were configured to integrate proprioceptive information, namely IMU and encoders, exclusively, as done above. To provide a holistic evaluation, we employed PRONTO to close the loop with our MPC. Fig. 6 shows localization errors for both estimators, measured through motion capture.

The inclusion of hybrid dynamics significantly improved localization accuracy, especially in the presence of a dynamic

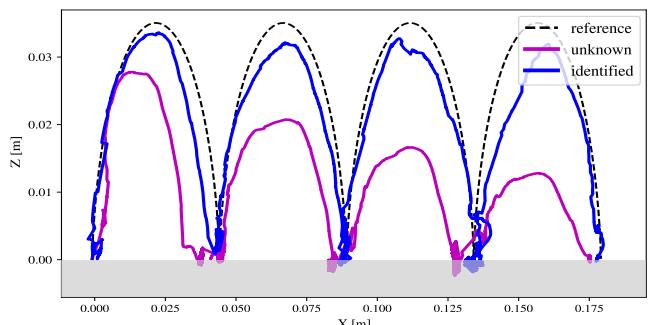


Fig. 5: Foot-swing tracking of the Go1 robot when carrying an unknown payload. Our MPC encountered difficulties in tracking the reference swing-foot trajectory (dashed line) when the payload was unknown (red line). Specifically, the Go1 robot failed to reach the desired step height, an error accumulated over time. In contrast, our MPC improved its foot-swing tracking performance when our estimator identified the payload (green line).

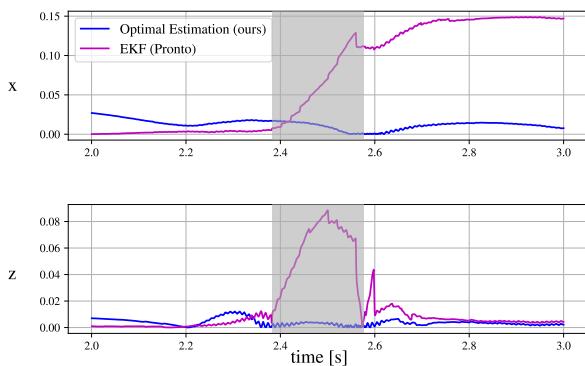


Fig. 6: Go1’s localization errors when performing a jump. Our optimal estimator (blue lines) exhibits a smaller estimation error compared to PRONTO.

maneuver like a jump, leading to our optimal estimator outperforming PRONTO’s estimation. This achievement is noteworthy because introducing the dynamics corrects almost all IMU drifts.

VII. CONCLUSION

We introduced (i) a multiple shooting optimal estimation algorithm tailored for multi-contact inertial estimation and localization and (ii) a novel smooth manifold with local submersion, named exponential eigenvalue, to ensure the full physical consistency of inertial parameters. Our exponential eigenvalue manifold was compared against the singularity-free log-Cholesky manifold, demonstrating improved convergence attributed to its reduced degree of nonlinearity. However, it presented singularities with symmetrical inertias. To address this, we proposed a nullspace approach, which handles these rank deficiencies. Additionally, our multiple-shooting rollout demonstrated superior numerical behavior compared to existing methods, resulting in better estimations. We demonstrated the effectiveness of our optimal estimation framework in complex scenarios such as humanoid brachiation and backflips. Practical benefits were further illustrated through experimental trials on the Go1 robot. Future work will involve an open-source C++ implementation and MHE development.

REFERENCES

- [1] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman and Hall, 2004.
- [2] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, “On-Manifold Preintegration for Real-Time Visual–Inertial Odometry,” *IEEE Trans. Rob. (T-RO)*, vol. 33, 2017.
- [3] D. Wisth, M. Camurri, and M. Fallon, “VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots,” *IEEE Rob. Autom. Lett. (RA-L)*, vol. 39, 2023.
- [4] P.-B. Wieber, “Holonomy and nonholonomy in the dynamics of articulated motion,” in *Fast Motions in Biomechanics and Robotics*, 2005.
- [5] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. Caldwell, and C. Semini, “Application of Wrench-Based Feasibility Analysis to the Online Trajectory Optimization of Legged Robots,” *IEEE Rob. Autom. Lett. (RA-L)*, vol. 3, 2018.
- [6] D. Gabay, “Minimizing a differentiable function over a differential manifold,” *J. Optim. Theory Appl.*, vol. 37, 1982.
- [7] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, “Crocodyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control,” in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2020.
- [8] J. Sola, J. Deray, and D. Atchutan, “A micro Lie theory for state estimation in robotics,” *arXiv preprint arXiv:1812.01537*, 2018.
- [9] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Cambridge University Press, 2009.
- [10] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Rev.*, vol. 12, 2005.
- [11] R. H. Byrd, J. Nocedal, and R. A. Waltz, “KNITRO: An integrated package for nonlinear optimization,” in *Lrg. Scal. Nonlin. Opt.*, 2006.
- [12] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Progr.*, vol. 106, 2006.
- [13] “Harwell Subroutine Library, AEA Technology, Harwell, Oxfordshire, England. A catalogue of subroutines,” <http://www.hsl.rl.ac.uk/>.
- [14] C. Mastalli, J. Marti-Saumell, W. Merkt, J. Sola, N. Mansard, and S. Vijayakumar, “A Feasibility-Driven Approach to Control-Limited DDP,” *Autom. Rob.*, vol. 46, 2022.
- [15] F. Dellaert and M. Kaess, *Factor Graphs for Robot Perception*. Now Foundations and Trends, 2017.
- [16] R. Hartley, J. Mangelson, L. Gan, M. Ghaffari Jadidi, J. M. Walls, R. M. Eustice, and J. W. Grizzle, “Legged Robot State-Estimation Through Combined Forward Kinematic and Preintegrated Contact Factors,” in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2018.
- [17] V. Agrawal, S. Bertrand, R. Griffin, and F. Dellaert, “Proprioceptive State Estimation of Legged Robots with Kinematic Chain Modeling,” in *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2022.
- [18] R. E. Bellman, *The Theory of Dynamic Programming*. RAND Corporation, 1954.
- [19] M. Kobilarov, D.-N. Ta, and F. Dellaert, “Differential dynamic programming for optimal estimation,” in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2015.
- [20] A. Oshin, M. D. Houghton, M. J. Acheson, I. M. Gregory, and E. A. Theodorou, “Parameterized Differential Dynamic Programming,” in *Rob.: Sci. Sys. (RSS)*, 2022.
- [21] C. Mastalli, S. P. Chhatoi, T. Corbères, S. Tonneau, and S. Vijayakumar, “Inverse-Dynamics MPC via Nullspace Resolution,” *IEEE Trans. Rob. (T-RO)*, vol. 39, 2023.
- [22] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, “Constrained Differential Dynamic Programming: A primal-dual augmented Lagrangian approach,” in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2022.
- [23] H. Li, W. Yu, T. Zhang, and P. M. Wensing, “A Unified Perspective on Multiple Shooting In Differential Dynamic Programming,” in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2023.
- [24] C. G. Atkeson, C. H. An, and J. M. Hollerbach, “Estimation of Inertial Parameters of Manipulator Loads and Links,” *The Int. J. of Rob. Res. (IJRR)*, vol. 5, 1986.
- [25] J. Carpentier and N. Mansard, “Analytical derivatives of rigid body dynamics algorithms,” in *Rob.: Sci. Sys. (RSS)*, 2018.
- [26] S. Singh, R. P. Russell, and P. M. Wensing, “Efficient Analytical Derivatives of Rigid-Body Dynamics Using Spatial Vector Algebra,” *IEEE Rob. Autom. Lett. (RA-L)*, vol. 7, 2022.
- [27] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [28] S. Traversaro, S. Brossette, A. Escande, and F. Nori, “Identification of fully physical consistent inertial parameters using optimization on manifolds,” in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2016.
- [29] C. Rucker and P. M. Wensing, “Smooth Parameterization of Rigid-Body Inertia,” *IEEE Rob. Autom. Lett. (RA-L)*, vol. 7, 2022.
- [30] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, “Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics,” in *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2018.
- [31] P. M. Wensing, S. Kim, and J.-J. E. Slotine, “Linear Matrix Inequalities for Physically Consistent Inertial Parameter Identification: A Statistical Perspective on the Mass Distribution,” *IEEE Rob. Autom. Lett. (RA-L)*, vol. 3, 2017.
- [32] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, “Agile maneuvers in legged robots: a predictive control approach,” *arXiv preprint arXiv:2203.07554*, 2022.
- [33] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. New York, USA: Springer, 2006.
- [34] R. H. Byrd, M. E. Hribar, and J. Nocedal, “An Interior Point Algorithm for Large-Scale Nonlinear Programming,” *SIAM J. Optim.*, vol. 9, 1999.
- [35] M. F. Fallón, M. Antone, N. Roy, and S. Teller, “Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing,” in *IEEE Int. Conf. Hum. Rob. (ICHR)*, 2014.