

A Unified Single-Rigid-Body NMPC/NMHE Framework for ANYmal-C Trotting:

Model Derivation, Output-Feedback Architecture, and Closed-Loop MuJoCo Validation

Sreeram Anil

Technical report, September 2026

Simulation study built around a real-time-iteration NMPC solver and its moving-horizon-estimation variant, developed as an extension of the author’s master’s thesis on SLQP-MPC. The solver algorithms are unpublished ongoing research and are not described here; this report covers the modelling, the architecture and the results.

Abstract

This report derives and validates a unified output-feedback control architecture for the ANYmal-C quadruped in which a nonlinear model predictive controller (NMPC) and a nonlinear moving-horizon estimator (NMHE) share one single-rigid-body-dynamics (SRBD) model, one symbolic source, and one algorithmic design language. The controller and the estimator are instances of a real-time-iteration NMPC solver and its moving-horizon-estimation variant, developed as an extension of the author’s master’s thesis on SLQP-MPC; both are used as unmodified libraries behind a small C interface, and their algorithms—ongoing unpublished work—are outside the scope of this report. The two solver-facing interfaces of the SRBD model—time-varying foot positions, contact schedule and disturbance parameters for the controller, and known inputs, measurement gates and an augmented random-walk disturbance state for the estimator—are realised entirely in the generated model header and in a thin driver. We derive the SRBD model in ZYX Euler coordinates with body-frame angular velocity, the gait-scheduled contact parametrisation, the friction-cone and attitude constraints, the leg-odometry and IMU measurement model with settled-stance gating, the augmented-state disturbance observer, and the joint closed loop with disturbance feed-forward. We explain why a disturbance modelled as a penalised estimator input is recovered too slowly for feed-forward and why carrying it as a random-walk state removes the lag. On a MuJoCo full-body simulation of ANYmal-C (18 DoF, 45 kg, 57 % of the mass in the legs) the loop performs a trot at 0.4 m/s, absorbs a 40 N lateral push, and executes a 0.3 rad/s turn with a 50-stage, 100 Hz NMPC costing 0.56 ms preparation and 56 μ s feedback per cycle, and a 20-node, 100 Hz NMHE costing 0.24 ms preparation, 1.1 μ s instantaneous output and 37 μ s feedback. Velocity is estimated to 1 cm/s, attitude to 1.4–2.6 mrad, a 60 N push to 61.8 N; force feed-forward of the estimated wrench reduces the push-induced drift from 0.74 m to 0.15 m. Five noise seeds succeed. The modelling, the simulation framework, the data and the animation are released; the solver libraries are not.

I. INTRODUCTION

Legged locomotion is the natural system-level test of a matched control–estimation pair. The controller must select ground reaction forces (GRFs) inside friction cones over a horizon that spans several contact switches, and the estimator must reconstruct the base state from proprioception alone—an IMU and the leg kinematics of whichever feet are on the ground—while the contact set changes every few hundred milliseconds. Both problems are constrained, nonlinear, and must be solved at hundreds of hertz. The reduced-order single-rigid-body model (SRBD) is the standard compromise between fidelity and speed for the controller [4–6]; for the estimator the same rigid-body model is the natural companion, because it makes the estimated quantities exactly the ones the controller consumes.

The solvers used here are a matched pair developed as an extension of the author’s master’s thesis on SLQP-MPC [2]: a real-time-iteration NMPC solver with state constraints, and a moving-horizon estimator with state and input constraints, both in plain C with compile-time dimensions and microsecond-scale

cycles [1]. That work is ongoing and unpublished, so this report treats both as libraries behind the small interface released with the code [3], and says nothing about how they work inside. What it does report is the system-level question they were built for: whether one reduced-order model, instantiated once as a controller and once as an estimator, closes an output-feedback walking loop on a full-body robot, and what the modelling has to get right for that to happen.

A. Contributions

1. A complete derivation of the ANYmal-C SRBD model, gait parametrisation, constraints, costs and proprioceptive measurement model, generated from one SymPy source and instantiated once for the controller (Sec. IV) and once for the estimator (Sec. V).
2. Two integration mechanisms that keep both solver libraries unmodified: an out-of-band stage-parameter table for the NMPC, and a known-input encoding of the contact schedule, foot anchors and measurement gates for the NMHE (Sec. IV-B, V-B).
3. An account of why a disturbance modelled as a penalised estimator input cannot be fed forward in a single-iteration setting, and its removal by an augmented random-walk disturbance state (Sec. V-A).
4. A leg-odometry measurement model with contact detection, settled-stance gating and contact-point Jacobians whose necessity is demonstrated by ablation during development (Sec. III-F).
5. A closed-loop validation on the full-body MuJoCo model with sensor noise, a lateral push and a turn, with timing, accuracy, ablation and noise-sensitivity data, an animation, and a released simulation framework (Sec. VII–VIII).

B. Scope and honesty statement

The SRBD is a deliberately coarse model of a robot whose legs carry 57 % of the mass. The report documents what this costs (touchdown residuals, a 5–10 % forward-speed deficit, torque-channel feed-forward that must stay open-loop) as carefully as what it buys. No third-party optimality oracle (IPOPT) was available in the build environment, so the quality of the solutions is established by the solver’s own fixed-parameter convergence to a 10^{-10} KKT residual and by the agreement of its two linearisation modes, not by an external solve. The solver algorithms are deliberately absent: what is reported about them is only what a user of the interface can measure, namely per-cycle wall times, memory footprint and horizon scaling.

C. Notation

Vectors are bold; $[\mathbf{a}]_{\times}$ is the cross-product matrix; $\mathbf{e}_j$ the j -th unit vector; $c_{\alpha}, s_{\alpha}, t_{\alpha}$ are cos, sin, tan; $\|\mathbf{a}\|_Q^2 = \mathbf{a}^T \mathbf{Q} \mathbf{a}$; $\succeq$ denotes positive semidefiniteness. Subscript k is a stage/node index, t a sampling instant, i a leg. Table 1 collects the symbols.

II. RELATED WORK

SRBD MPC for quadrupeds. Di Carlo et al. [4] introduced convex MPC on a linearised SRBD with gait-scheduled contacts and Raibert-style footholds for the MIT Cheetah 3; Bledt et al. [5] and Kim et al. [6] combined it with whole-body control. Nonlinear SRBD MPC solved by SQP/RTI appears in [7, 8] on ANYmal with the OCS2 toolbox, and the SLQ/DDP family [7] remains the workhorse for kino-dynamic models. Our controller is a nonlinear SRBD MPC in the sense of [8], with hard friction cones and attitude bounds handled inside the solver rather than by a separate QP.

Legged state estimation. Bloesch et al. [9] established the leg-odometry EKF that fuses IMU with kinematic foot constraints, treating foot positions as states; Camurri et al. [10] and Wisth et al. [11] extended this to factor graphs; moving-horizon formulations for legged systems have been explored more

TABLE 1. Symbols.

$\mathbf{p}, \mathbf{v}$	system-COM position/velocity, world frame
$\boldsymbol{\theta} = (\phi, \theta, \psi), \mathbf{R}, \mathbf{T}$	ZYX Euler angles, rotation $\mathcal{B} \rightarrow \mathcal{W}$, Euler-rate map
$\boldsymbol{\omega}$	angular velocity, body frame
$\mathbf{f}_i, \mathbf{r}_i, s_i$	GRF, contact point, contact flag of leg i
$\mathbf{d} = (\mathbf{d}_f, \mathbf{d}_\tau)$	disturbance wrench (world force, body torque)
$\mathbf{w}$	estimated input of the NMHE ($= \dot{\mathbf{d}}$)
$\boldsymbol{\pi}_k = (\mathbf{r}_k, \mathbf{s}_k, \mathbf{d})$	NMPC stage parameters
$\mathbf{u}_k$	NMHE known input (26)
s_i^m, g_v, δ_i	measurement gates, velocity gate, contact detection
$m, \mathbf{I}, \mu, f_{\max}$	mass, inertia, friction coefficient, force bound
N, M, h	NMPC horizon, NMHE window, sampling period
$Q, R, Q_N; V, W, \Lambda_a$	NMPC weights; NMHE weights and arrival information
$T_g, \beta, \phi_i, \sigma_i$	gait period, duty factor, leg phase, swing phase

recently as solver speed has made them affordable. Our estimator follows the leg-odometry principle but replaces the EKF by a constrained moving-horizon estimator on the same SRBD the controller uses, with a random-walk wrench state in the spirit of disturbance observers for offset-free MPC [12].

Solvers. The real-time-iteration idea—one Newton-type iteration per sampling instant, split into a preparation phase and a feedback phase—goes back to Diehl et al. [13] and is the operating mode assumed throughout this report; *acados* [15] is the reference open-source implementation and is a drop-in substitute behind the interface used here. The particular solvers used in this study are the author’s own [1, 2]; their formulation and numerics are unpublished and are not part of this report.

III. SYSTEM MODEL

A. Frames and attitude parametrisation

Let $\mathcal{W}$ be the inertial (world) frame with z up and $\mathcal{B}$ the base frame of ANYmal-C with x forward, y left, z up. The SRBD state is

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\theta} \\ \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^{12}, \quad \begin{array}{ll} \mathbf{p} \in \mathbb{R}^3 & \text{COM position in } \mathcal{W}, \\ \boldsymbol{\theta} = (\phi, \theta, \psi) & \text{roll, pitch, yaw (ZYX),} \\ \mathbf{v} \in \mathbb{R}^3 & \text{COM velocity in } \mathcal{W}, \\ \boldsymbol{\omega} \in \mathbb{R}^3 & \text{angular velocity in } \mathcal{B}. \end{array} \quad (1)$$

The rotation from $\mathcal{B}$ to $\mathcal{W}$ is

$$\mathbf{R}(\boldsymbol{\theta}) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi), \quad (2)$$

with the elementary rotations $\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$, $\mathbf{R}_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}$, $\mathbf{R}_z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Positive pitch is nose-down. The body angular velocity relates to the Euler rates through

$$\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta}) \boldsymbol{\omega}, \quad \mathbf{T}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix}, \quad (3)$$

where $t_\theta = \tan \theta$; $\mathbf{T}$ is singular only at $\theta = \pm\pi/2$, far outside the attitude bounds enforced by both the controller and the estimator (Sec. IV-A, V-B). The world-frame angular velocity is $\mathbf{R}\boldsymbol{\omega}$; we keep $\boldsymbol{\omega}$ in $\mathcal{B}$ because that is what a body-mounted gyroscope measures and because the inertia tensor is then constant.

1) Derivation of the Euler-rate map

Differentiating (2) and using $\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_{\times}$ (definition of the body angular velocity) gives $[\boldsymbol{\omega}]_{\times} = \mathbf{R}^T \dot{\mathbf{R}}$. Each factor contributes an elementary rotation about its own axis, $\dot{\mathbf{R}}_x(\phi) = \mathbf{R}_x[\dot{\phi} \mathbf{e}_1]_{\times}$ etc., so that

$$[\boldsymbol{\omega}]_{\times} = \mathbf{R}_x^T \mathbf{R}_y^T (\psi[\mathbf{e}_3]_{\times}) \mathbf{R}_y \mathbf{R}_x + \mathbf{R}_x^T (\dot{\theta}[\mathbf{e}_2]_{\times}) \mathbf{R}_x + \dot{\phi}[\mathbf{e}_1]_{\times},$$

and with $\mathbf{R}^T[\mathbf{a}]_{\times} \mathbf{R} = [\mathbf{R}^T \mathbf{a}]_{\times}$,

$$\boldsymbol{\omega} = \dot{\phi} \mathbf{e}_1 + \dot{\theta} \mathbf{R}_x^T \mathbf{e}_2 + \psi \mathbf{R}_x^T \mathbf{R}_y^T \mathbf{e}_3 = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & s_{\phi} c_{\theta} \\ 0 & -s_{\phi} & c_{\phi} c_{\theta} \end{bmatrix} \dot{\boldsymbol{\theta}} =: \mathbf{E}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}. \quad (4)$$

$\mathbf{E}$ has determinant c_{θ} ; its inverse is $\mathbf{T} = \mathbf{E}^{-1}$ in (3), singular at $\theta = \pm\pi/2$. The derivatives $\partial(\mathbf{T}\boldsymbol{\omega})/\partial\phi$ and $\partial(\mathbf{T}\boldsymbol{\omega})/\partial\theta$ that enter A_k (Appendix A) follow from $\partial\mathbf{T} = -\mathbf{T}(\partial\mathbf{E})\mathbf{T}$.

B. Single-rigid-body dynamics

Let m be the total robot mass and $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ the composite inertia about the COM expressed in $\mathcal{B}$, evaluated once in the nominal stance (Sec. III-G). Leg $i \in \{\text{LF, RF, LH, RH}\}$ ($i = 1, \dots, 4$) applies a world-frame ground reaction force $\mathbf{f}_i \in \mathbb{R}^3$ at the foot contact point $\mathbf{r}_i \in \mathbb{R}^3$ (world frame) when its contact flag $s_i \in \{0, 1\}$ is one. A world-frame disturbance force $\mathbf{d}_f$ and a body-frame disturbance torque $\mathbf{d}_{\tau}$ collect everything the SRBD does not model (leg inertia, external pushes, model error). The Newton–Euler equations of the lumped body are

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (5a)$$

$$\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta}) \boldsymbol{\omega}, \quad (5b)$$

$$m \dot{\mathbf{v}} = \sum_{i=1}^4 s_i \mathbf{f}_i + m \mathbf{g} + \mathbf{d}_f, \quad (5c)$$

$$\mathbf{I} \dot{\boldsymbol{\omega}} = \mathbf{R}(\boldsymbol{\theta})^T \sum_{i=1}^4 s_i [\mathbf{r}_i - \mathbf{p}]_{\times} \mathbf{f}_i - [\boldsymbol{\omega}]_{\times} \mathbf{I} \boldsymbol{\omega} + \mathbf{d}_{\tau}, \quad (5d)$$

with $\mathbf{g} = (0, 0, -9.81)$ and $[\mathbf{a}]_{\times}$ the cross-product matrix. Equation (5) is written once and reused by both solvers, with $\mathbf{d} = (\mathbf{d}_f, \mathbf{d}_{\tau})$ playing the role of a known stage parameter in the controller (Sec. IV) and of an estimated augmented state in the estimator (Sec. V). We write (5) compactly as

$$\dot{\mathbf{x}} = f_c(\mathbf{x}, \mathbf{f}; \mathbf{r}, \mathbf{s}, \mathbf{d}), \quad \mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_4) \in \mathbb{R}^{12}. \quad (6)$$

Two conventions matter for what follows. (i) The forces are multiplied by the flags s_i *inside the dynamics* rather than being removed from the decision vector: the controller’s input dimension stays fixed at $n_u = 12$ across the gait, swing-leg forces are simply decoupled from the dynamics and driven to zero by the cost and the cone $f_{i,z} \geq 0$, and the constraint set never changes dimension—which is what a solver with compile-time fixed shapes requires. (ii) The moment arm $\mathbf{r}_i - \mathbf{p}$ uses the COM, not the base origin; the COM offset of the leg configuration is part of the kinematic measurement model (Sec. III-F).

1) Origin of the lumped model and meaning of the disturbance wrench

For the full multibody system with bodies b of mass m_b , COM positions $\mathbf{p}_b$ and total mass $m = \sum_b m_b$, Newton’s law for the system COM $\mathbf{p} = \frac{1}{m} \sum_b m_b \mathbf{p}_b$ is exact,

$$m \ddot{\mathbf{p}} = \sum_i \mathbf{f}_i + m \mathbf{g} + \mathbf{F}_{\text{ext}}, \quad (7)$$

because joint forces are internal. Likewise the angular momentum about the COM, $\mathbf{L} = \sum_b (m_b [\mathbf{p}_b - \mathbf{p}]_{\times} \dot{\mathbf{p}}_b + \mathbf{R}_b \mathbf{I}_b \mathbf{R}_b^{\top} \boldsymbol{\omega}_b^{\mathcal{W}})$, obeys exactly

$$\dot{\mathbf{L}} = \sum_i [\mathbf{r}_i - \mathbf{p}]_{\times} \mathbf{f}_i + \boldsymbol{\tau}_{\text{ext}}. \quad (8)$$

The SRBD approximates $\mathbf{L} \approx \mathbf{R} \mathbf{I} \boldsymbol{\omega}$ with $\mathbf{I}$ the composite inertia of the nominal configuration, i.e. it neglects (a) the momentum of the legs relative to the trunk and (b) the configuration dependence of $\mathbf{I}$. Differentiating with $\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_{\times}$ gives $\dot{\mathbf{L}} \approx \mathbf{R}(\mathbf{I}\dot{\boldsymbol{\omega}} + [\boldsymbol{\omega}]_{\times} \mathbf{I}\boldsymbol{\omega})$, and (8) becomes the last row of (5) after left-multiplication by $\mathbf{R}^{\top}$. The quantities dropped in (a)–(b) are exactly what the disturbance wrench absorbs:

$$\mathbf{d}_f = \mathbf{F}_{\text{ext}}, \quad \mathbf{d}_{\tau} = \mathbf{R}^{\top} \boldsymbol{\tau}_{\text{ext}} + \mathbf{R}^{\top} (\mathbf{R}(\mathbf{I}\dot{\boldsymbol{\omega}} + [\boldsymbol{\omega}]_{\times} \mathbf{I}\boldsymbol{\omega}) - \dot{\mathbf{L}}), \quad (9)$$

which makes the estimated $\hat{\mathbf{d}}_{\tau}$ in Fig. 5 interpretable: its gait-frequency component is the reaction of the swinging legs and the configuration dependence of the inertia, its mean the external torque. For the force channel the approximation is exact ($\mathbf{d}_f$ is only the external force), which is why force feed-forward is safe and torque feed-forward is not (Sec. VI). The velocity $\mathbf{v}$ in (1) is the velocity of the *system* COM, consistent with (7); this is why the measurement model (17) carries a relative COM-velocity term.

C. Discrete-time model

Both solvers integrate (6) with their own explicit RK4 step over the sampling period h (10 ms) with one substep,

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{f}_k; \boldsymbol{\pi}_k) = \mathbf{x}_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (10)$$

where $\boldsymbol{\pi}_k = (\mathbf{r}_k, \mathbf{s}_k, \mathbf{d}_k)$ is the stage parameter and $k_1, \dots, k_4$ the usual RK4 slopes evaluated with $\boldsymbol{\pi}_k$ frozen over the step. The sensitivities $A_k = \partial F / \partial \mathbf{x}$ and $B_k = \partial F / \partial \mathbf{f}$ are assembled by the solvers' chain rule from the continuous Jacobians $\partial f_c / \partial \mathbf{x}$ and $\partial f_c / \partial \mathbf{f}$, which are generated symbolically with common-subexpression elimination (Appendix A gives their structure). Contact flags are held constant inside a step; a contact switch is therefore aligned to the 10 ms grid, which is the resolution of the gait scheduler anyway.

Writing $J_x = \partial f_c / \partial \mathbf{x}$ and $J_u = \partial f_c / \partial \mathbf{f}$ at the four RK4 evaluation points $\mathbf{x}^{(1)} = \mathbf{x}_k$, $\mathbf{x}^{(2)} = \mathbf{x}_k + \frac{h}{2}k_1$, $\mathbf{x}^{(3)} = \mathbf{x}_k + \frac{h}{2}k_2$, $\mathbf{x}^{(4)} = \mathbf{x}_k + hk_3$ (superscripts on J below), the exact discrete sensitivities are the forward-mode chain rule of (10),

$$\begin{aligned} K_1^x &= J_x^{(1)}, & K_2^x &= J_x^{(2)}(I + \frac{h}{2}K_1^x), \\ K_3^x &= J_x^{(3)}(I + \frac{h}{2}K_2^x), & K_4^x &= J_x^{(4)}(I + hK_3^x), \\ K_1^u &= J_u^{(1)}, & K_2^u &= J_u^{(2)} + \frac{h}{2}J_x^{(2)}K_1^u, \\ K_3^u &= J_u^{(3)} + \frac{h}{2}J_x^{(3)}K_2^u, & K_4^u &= J_u^{(4)} + hJ_x^{(4)}K_3^u, \\ A_k &= I + \frac{h}{6}(K_1^x + 2K_2^x + 2K_3^x + K_4^x), \\ B_k &= \frac{h}{6}(K_1^u + 2K_2^u + 2K_3^u + K_4^u), \end{aligned} \quad (11)$$

which is what both solver libraries compute from the generated model. Per stage the SRBD costs four evaluations of f_c , J_x (12×12) and J_u (12×12) plus six $12 \times 12 \times 12$ products. The verification of Sec. VII-A checks (11) against central differences of the discrete map.

D. Gait scheduling and contact parametrisation

A trot is a periodic schedule with period T_g and duty factor β (stance fraction). With per-leg phase offsets $o = (0, \frac{1}{2}, \frac{1}{2}, 0)$ for (LF, RF, LH, RH), leg i has phase and contact flag

$$\varphi_i(t) = \left(\frac{t - t_0}{T_g} + o_i \right) \bmod 1, \quad s_i(t) = \mathbb{1}[\varphi_i(t) < \beta], \quad (12)$$

so that the diagonal pairs (LF,RH) and (RF,LH) alternate. Before t_0 the robot stands ($s \equiv 1$). We use $T_g = 0.7$ s and $\beta = 0.5$ (0.35 s stance, 0.35 s swing); $T_g = 0.5$ – 0.6 s produced markedly larger roll

excursions on this robot (Sec. VIII-E). The within-swing phase is $\sigma_i = (\varphi_i - \beta)/(1 - \beta) \in [0, 1]$ and the next touchdown time of a swinging leg is $t_{td,i} = t + (1 - \varphi_i)T_g$.

1) Footholds (Raibert heuristic)

Let $\ell_i \in \mathbb{R}^2$ be the nominal horizontal offset of foot i from the COM in $\mathcal{B}$ (from the nominal stance, ± 0.45 m longitudinally and ± 0.30 m laterally). For a leg that touches down at t_{td} , the target is

$$\begin{aligned} \mathbf{r}_i^{td} &= \mathbf{p} + \mathbf{v}(t_{td} - t) + \mathbf{R}_z(\psi_{td}) \begin{bmatrix} \ell_i \\ 0 \end{bmatrix} + \frac{1}{2}T_{st}\mathbf{v} + k_r(\mathbf{v} - \mathbf{v}^{cmd}), \\ \psi_{td} &= \psi + \omega_z^{cmd}(t_{td} - t), \quad \mathbf{r}_{i,z}^{td} = z_{ground}, \end{aligned} \quad (13)$$

with $T_{st} = \beta T_g$ the stance duration and $k_r = 0.12$ s a capture-point-like gain (the linear inverted pendulum gives $\sqrt{z/g} \approx 0.21$ s; smaller values proved sufficient and calmer). The neutral point $\frac{1}{2}T_{st}\mathbf{v}$ uses the *actual* velocity: using the commanded velocity places the foot ahead of the sweep whenever the robot is slower than commanded and brakes it (Sec. VIII-E). $\mathbf{v}^{cmd} = \mathbf{R}_z(\psi_{td})\mathbf{v}_{\mathcal{B}}^{cmd}$ rotates the body-frame command into $\mathcal{W}$; z_{ground} is the mean height of the currently anchored stance feet (Sec. III-F). All quantities in (13) are evaluated with the *estimated* state.

2) Swing-foot trajectory

For $\sigma \in [0, 1]$ the desired contact-point position and velocity are

$$\begin{aligned} \mathbf{p}_i^{sw}(\sigma) &= \mathbf{p}_i^{lo} + (\mathbf{r}_i^{td} - \mathbf{p}_i^{lo})\zeta(\sigma) + \begin{bmatrix} 0 \\ 0 \\ h_{sw}\beta_4(\sigma) + z_{land}\zeta(\sigma) \end{bmatrix}, \\ \zeta(\sigma) &= 10\sigma^3 - 15\sigma^4 + 6\sigma^5, \quad \beta_4(\sigma) = 16\sigma^2(1 - \sigma)^2, \end{aligned} \quad (14)$$

where $\mathbf{p}_i^{lo}$ is the lift-off position, $h_{sw} = 0.06$ m the apex height and $z_{land} = -0.015$ m a deliberate overshoot below the ground so that contact is established slightly *before* the scheduled touchdown. Both ζ and β_4 have zero velocity at $\sigma = 0, 1$; a $\sin(\pi\sigma)$ height profile, whose end slope is $-\pi h_{sw}/T_{sw} \approx -0.5$ m/s, produced 540–590 N impacts against 290–310 N commands and was replaced (Sec. VIII-E). The velocity reference is the analytic derivative of (14); the target $\mathbf{r}_i^{td}$ is re-evaluated at every control step so the trajectory bends smoothly towards the latest foothold.

E. Joint-torque mapping

Let $J_i(q) \in \mathbb{R}^{3 \times 3}$ be the world-frame Jacobian of foot i with respect to its three joints and $\mathbf{b}_i(q, \dot{q}) \in \mathbb{R}^3$ the gravity/Coriolis bias torques of those joints (obtained from the plant's inverse dynamics, as a torque-controlled robot would compute from its own model). Stance and swing legs are commanded at 500 Hz by

$$\boldsymbol{\tau}_i = \begin{cases} -J_i^T \mathbf{f}_i + \mathbf{b}_i & s_i = 1, \\ J_i^T (K_p(\mathbf{p}_i^{sw} - \mathbf{p}_i^{ft}) + K_d(\dot{\mathbf{p}}_i^{sw} - \dot{\mathbf{p}}_i^{ft})) + \mathbf{b}_i & s_i = 0, \end{cases} \quad (15)$$

with $K_p = 900$ N/m, $K_d = 30$ Ns/m and torques clipped at ± 80 Nm. The bias term in the stance branch is not cosmetic: without it the realised GRF is $\mathbf{f}_i + J_i^T \mathbf{b}_i$, i.e. 12 N per foot too large when standing and 20–25 N per foot in trot, which lifts the body until the NMPC's proportional height feedback balances it (a 5–9 cm offset; Sec. VIII-E). With it, the realised and commanded stance forces agree to 24 N rms including the touchdown transients and to -1 N on average.

1) Derivation of the stance mapping

For leg i with joint coordinates $q_i \in \mathbb{R}^3$ and dynamics $M_i \ddot{q}_i + \mathbf{b}_i(q, \dot{q}) = \boldsymbol{\tau}_i + J_i^T \mathbf{f}_i^{ft}$, where $\mathbf{f}_i^{ft}$ is the force the ground exerts on the foot and $\mathbf{b}_i$ collects gravity and Coriolis terms of the leg links (the coupling

with the base motion enters through the plant's inverse dynamics), the realised force under a commanded torque is $\mathbf{f}_i^{\text{ft}} = J_i^{-\top}(\mathbf{b}_i - \boldsymbol{\tau}_i + M_i \ddot{\mathbf{q}}_i)$. Setting $\boldsymbol{\tau}_i = -J_i^\top \mathbf{f}_i + \mathbf{b}_i$ yields $\mathbf{f}_i^{\text{ft}} = \mathbf{f}_i + J_i^{-\top} M_i \ddot{\mathbf{q}}_i$: the realised force equals the commanded one up to the leg's own inertial term, which is what remains in the 24 N rms figure and vanishes when standing. Omitting $\mathbf{b}_i$ leaves $\mathbf{f}_i^{\text{ft}} = \mathbf{f}_i + J_i^{-\top} \mathbf{b}_i$; in the nominal stance $J_i^{-\top} \mathbf{b}_i \approx 12$ N per foot, and the equilibrium the NMPC finds is then the height at which its proportional feedback offsets 4×12 N, a 5–9 cm error (Table 8). The same virtual-work identity $\boldsymbol{\tau} = J^\top \mathbf{F}$ justifies the swing branch of (15).

F. Proprioceptive measurement model

The estimator sees an attitude estimate from the IMU/AHRS, the gyroscope, and the leg kinematics. Let $\mathbf{p}_i^{\text{fk}}(q)$ be the foot contact point in $\mathcal{B}$ from forward kinematics, $\mathbf{c}(q)$ the COM offset from the base origin in $\mathcal{B}$ (a kinematic function of the joint angles only), and $\mathbf{p}_i = \mathbf{p}_i^{\text{fk}} - \mathbf{c}$ the COM-relative foot position. The measured outputs at time t are

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\theta}^{\text{ahrs}} \\ \boldsymbol{\omega}^{\text{gyro}} \\ g_v \mathbf{v}^{\text{kin}} \\ s_1^m \boldsymbol{\rho}_1 \\ \vdots \\ s_4^m \boldsymbol{\rho}_4 \end{bmatrix} \in \mathbb{R}^{21}, \quad h(\mathbf{x}; \mathbf{r}, \mathbf{s}^m, g_v) = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\omega} \\ g_v \mathbf{v} \\ s_1^m \mathbf{R}^\top (\mathbf{r}_1 - \mathbf{p}) \\ \vdots \\ s_4^m \mathbf{R}^\top (\mathbf{r}_4 - \mathbf{p}) \end{bmatrix}, \quad (16)$$

where $\mathbf{r}_i$ are the *anchored* foot positions (below), $s_i^m \in \{0, 1\}$ are measurement gates and $g_v \in \{0, 1\}$ a velocity gate. Only the last twelve rows are nonlinear (through $\mathbf{R}^\top$), and only through the attitude; they are what makes the estimation problem nonlinear and couples position to attitude.

1) Leg-odometry velocity

A foot that is stationary on the ground satisfies $\dot{\mathbf{p}}_i^{\text{ft}} = \mathbf{v}_{\text{base}} + J_i^{\text{rot}} \boldsymbol{\omega} + J_i \dot{\mathbf{q}}_i = 0$, where J_i^{rot} is the world-frame Jacobian of the contact point with respect to the base angular velocity. Hence each settled stance leg yields an estimate of the base velocity, and the COM velocity is

$$\mathbf{v}^{\text{kin}} = \frac{1}{|\mathcal{S}^m|} \sum_{i \in \mathcal{S}^m} \left(-J_i^{\text{rot}} \boldsymbol{\omega}^{\text{gyro}} - J_i \dot{\mathbf{q}}_i \right) + \dot{\mathbf{c}}_{\mathcal{W}}(q, \dot{q}), \quad (17)$$

where $\mathcal{S}^m = \{i : s_i^m = 1\}$ and $\dot{\mathbf{c}}_{\mathcal{W}}$ is the world-frame velocity of the COM relative to the base origin caused by joint motion (a kinematic quantity; with 57 % of the mass in the legs it reaches ± 0.1 m/s in trot and cannot be neglected). Two details were found to be essential: the Jacobians must be evaluated at the *contact point*, not at the foot-sphere centre, because the sphere rolls on the ground while the contact point stays put; and the gate $g_v = \mathbb{1}[\mathcal{S}^m \neq \emptyset]$ must be carried in the model (16), because the estimator's output weight is constant and a zero measurement with nonzero weight would otherwise be interpreted as “the robot is at rest”.

Derivation. The contact point of foot i is $\mathbf{p}_i^{\text{ft}} = \mathbf{p}_b + \mathbf{R}(\mathbf{p}_i^{\text{fk}}(q_i) + \mathbf{o}_i)$, with $\mathbf{p}_b$ the base origin and $\mathbf{o}_i$ the offset from the sphere centre to its lowest point expressed in $\mathcal{B}$ (for a sphere of radius r_f on flat ground, $\mathbf{o}_i = -r_f \mathbf{R}^\top \mathbf{e}_3$). Its world velocity is $\dot{\mathbf{p}}_i^{\text{ft}} = \mathbf{v}_b + \mathbf{R}[\boldsymbol{\omega}]_{\times} (\mathbf{p}_i^{\text{fk}} + \mathbf{o}_i) + \mathbf{R} \partial \mathbf{p}_i^{\text{fk}} / \partial q_i \dot{q}_i$, i.e. $J_i^{\text{rot}} = -\mathbf{R} [\mathbf{p}_i^{\text{fk}} + \mathbf{o}_i]_{\times}$ and $J_i = \mathbf{R} \partial \mathbf{p}_i^{\text{fk}} / \partial q_i$ (with the shank rotation included in the joint columns). Evaluating these at the sphere *centre* ($\mathbf{o}_i = 0$) omits the term $\mathbf{R} [\boldsymbol{\omega}_i^{\text{sh}}]_{\times} \mathbf{o}_i$ of the shank's rotation, which is what a rolling foot contributes: with $r_f = 3$ cm and shank rotation rates of 1–3 rad/s in trot this is a 3–9 cm/s bias, of the order of the velocities being estimated. The system-COM velocity follows from $\mathbf{p} = \mathbf{p}_b + \mathbf{R}\mathbf{c}(q)$: $\mathbf{v} = \mathbf{v}_b + \mathbf{R}[\boldsymbol{\omega}]_{\times} \mathbf{c} + \mathbf{R} \partial \mathbf{c} / \partial q \dot{q}$, the last two terms being $\dot{\mathbf{c}}_{\mathcal{W}}$ in (17).

2) Contact detection and settled-stance gating

Realised GRFs $\mathbf{f}_i^{\text{meas}}$ are available on torque-controlled robots from the joint-torque sensors, $\mathbf{f}_i^{\text{meas}} = -\mathbf{J}_i^{-\top}(\boldsymbol{\tau}_i^{\text{meas}} - \mathbf{b}_i)$; in the simulation they are emulated by the plant contact forces plus 2 N white noise. Contact is detected by $\delta_i = \mathbb{1}[\|\mathbf{f}_i^{\text{meas}}\| > 8 \text{ N}]$. The dynamics flag used by the estimator is $s_i^{\text{dyn}} = \max(s_i, \delta_i)$ (a foot that touches down early, which the overshoot z_{land} makes systematic, transmits force and must be modelled), and the measurement gate is

$$s_i^m(t) = \delta_i(t) \cdot \mathbb{1}[\text{in scheduled stance for } \geq t_{\text{settle}}], \quad t_{\text{settle}} = 40 \text{ ms}, \quad (18)$$

because the kinematic rows of a foot that is still sliding into its footprint are wrong by centimetres, and a velocity row taken from a foot that is still moving is wrong by metres per second (the impact-phase spikes reached 2.3 m/s in the unfiltered signal and drove the z estimate up by 1–2 cm at every touchdown before gating was introduced; Sec. VIII-E).

3) Foot anchoring

The anchors $\mathbf{r}_i$ in (16) are the estimated world positions of the feet at the moment they become stationary,

$$\mathbf{r}_i \leftarrow \hat{\mathbf{p}} + \mathbf{R}(\hat{\boldsymbol{\theta}}) \boldsymbol{\rho}_i \quad \text{at } s_i: 0 \rightarrow 1, \delta_i: 0 \rightarrow 1, \text{ and } s_i^m: 0 \rightarrow 1, \quad (19)$$

i.e. provisionally at the scheduled or detected touchdown (for the lever arm in the dynamics) and finally at the settled instant (for the output rows). Anchoring from the filtered estimate makes leg odometry a dead-reckoning process: absolute position is unobservable and $\hat{\mathbf{p}}$ drifts slowly (2 cm over 5 s here), which is inherent to proprioception and harmless for velocity control. It also introduces a subtle feedback: any systematic lag of $\hat{\mathbf{p}}$ at the anchoring instant is inherited by the anchor and pulls the subsequent velocity estimate. This is why the velocity row in (16) must be weighted strongly enough to be authoritative (Sec. V-B, VIII-E).

G. Physical constants

From the MuJoCo Menagerie ANYmal-C model in the nominal stance (HAA = 0, HFE = ± 0.5 rad, KFE = ∓ 1.0 rad, COM height 0.462 m): $m = 44.965 \text{ kg}$, $\mathbf{I} = \text{diag}(1.998, 3.881, 4.031) \text{ kg m}^2$ with off-diagonal terms below 5×10^{-3} , nominal COM-relative foot offsets ($\pm 0.451, \pm 0.301$) m, COM 5.3 cm below the base origin. The friction coefficient used by the controller is $\mu = 0.6$ against 0.8 in the plant, and the per-foot vertical bound is $f_{\text{max}} = 400 \text{ N}$ (0.9 mg).

IV. NONLINEAR MPC ON THE REDUCED-ORDER MODEL

A. The optimal control problem

At each control instant t with estimate $\hat{\mathbf{x}}_t$ and horizon $N = 50$ stages of $h = 10 \text{ ms}$ ($0.5 \text{ s}, \approx 0.7$ gait cycles), the controller solves the multiple-shooting problem

$$\min_{\mathbf{x}_{0:N}, \mathbf{f}_{0:N-1}} \sum_{k=0}^{N-1} \frac{1}{2} \left(\|\mathbf{x}_k - \mathbf{x}_k^r\|_Q^2 + \|\mathbf{f}_k - \mathbf{f}_k^r\|_R^2 \right) + \frac{1}{2} \|\mathbf{x}_N - \mathbf{x}_N^r\|_{Q_N}^2 \quad (20a)$$

$$\text{s.t. } \mathbf{x}_0 = \hat{\mathbf{x}}_t, \quad \mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{f}_k; \boldsymbol{\pi}_k), \quad (20b)$$

$$c(\mathbf{x}_k, \mathbf{f}_k; \boldsymbol{\pi}_k) \leq 0, \quad k = 0, \dots, N-1, \quad (20c)$$

with $n_x = 12$, $n_u = 12$, $n_c = 28$, $n_c^N = 0$. The stage parameters $\boldsymbol{\pi}_k = (\mathbf{r}_k, \mathbf{s}_k, \mathbf{d})$ are the predicted foot positions and contact flags of (12)–(13) evaluated at $t + kh$ —a leg’s $\mathbf{r}_{k,i}$ jumps to its planned foothold at each future touchdown inside the horizon, and stays at its anchor (19) while in stance—and $\mathbf{d}$ is the disturbance feed-forward of Sec. VI, held constant over the horizon.

1) References

With a body-frame velocity command $\mathbf{v}_{\mathcal{B}}^{\text{cmd}}$ and yaw-rate command ω_z^{cmd} , the references follow the commanded turning arc from the current estimate:

$$\begin{aligned} \psi_k^r &= \hat{\psi} + \omega_z^{\text{cmd}} k h, & \mathbf{v}_k^r &= \mathbf{R}_z(\psi_k^r) \mathbf{v}_{\mathcal{B}}^{\text{cmd}}, \\ \mathbf{p}_0^r &= \hat{\mathbf{p}}, & \mathbf{p}_{k+1}^r &= \mathbf{p}_k^r + h \mathbf{v}_k^r, & p_{k,z}^r &= z_{\text{ground}} + z^{\text{cmd}}, \\ \boldsymbol{\theta}_k^r &= (0, 0, \psi_k^r), & \boldsymbol{\omega}_k^r &= (0, 0, \omega_z^{\text{cmd}}), \end{aligned} \quad (21)$$

with $z^{\text{cmd}} = 0.45$ m. Because $\mathbf{p}^r$ is re-anchored at $\hat{\mathbf{p}}$ every sample, the horizontal position terms act as velocity tracking with a mild damping on drift within the horizon rather than as absolute position control (absolute position is unobservable from proprioception anyway). The force reference distributes the weight over the predicted stance feet,

$$\mathbf{f}_{k,i}^r = \frac{s_{k,i}}{\max(1, \sum_j s_{k,j})} m g \mathbf{e}_z, \quad (22)$$

which is the unique minimiser of $\sum_i \|\mathbf{f}_i\|^2$ under vertical force balance and has the practical role of a warm start for the swing-to-stance switches inside the horizon.

2) Weights

$Q = \text{diag}(5, 5, 300, 100, 100, 50, 20, 20, 20, 1, 1, 2)$ in SI units on $(\mathbf{p}, \boldsymbol{\theta}, \mathbf{v}, \boldsymbol{\omega})$, $R = 10^{-4} I_{12}$, $Q_N = 2Q$. These values are the result of the closed-loop tuning in Sec. VIII-E: more aggressive attitude/force weights ($Q_\theta = 300$, $R = 2 \times 10^{-5}$) were stable with ground-truth state feedback but excited a 25 Hz roll limit cycle through the estimator at the 20 ms control period first used, and remained marginal at 10 ms.

3) Constraints

For each foot i (six rows) and for the attitude (four rows), normalised to $O(1)$ as the augmented-Lagrangian machinery expects:

$$\mathbf{c} = \frac{1}{f_{\max}} \begin{bmatrix} f_{i,z} - f_{\max} \\ -f_{i,z} \\ f_{i,x} - \mu f_{i,z} \\ -f_{i,x} - \mu f_{i,z} \\ f_{i,y} - \mu f_{i,z} \\ -f_{i,y} - \mu f_{i,z} \end{bmatrix}_{i=1..4}, \quad \frac{1}{\vartheta_{\max}} \begin{bmatrix} \phi - \vartheta_{\max} \\ -\phi - \vartheta_{\max} \\ \theta - \vartheta_{\max} \\ -\theta - \vartheta_{\max} \end{bmatrix}, \quad (23)$$

with $\mu = 0.6$, $f_{\max} = 400$ N, $\vartheta_{\max} = 0.35$ rad. The pyramid (23) is the inner linearisation of the friction cone; the cone rows are imposed on *all* feet regardless of $s_{k,i}$ —for a swing foot the feasible set contains $\mathbf{f}_i = 0$, which is also the unconstrained minimiser of its R -term, so swing forces are driven to zero without a degenerate pair of opposing active rows. The constraint Jacobians are constant in the force block, $\partial c / \partial \mathbf{f}_i = f_{\max}^{-1} [\mathbf{e}_3 \quad -\mathbf{e}_3 \quad (\mathbf{e}_1 - \mu \mathbf{e}_3) \quad (-\mathbf{e}_1 - \mu \mathbf{e}_3) \quad (\mathbf{e}_2 - \mu \mathbf{e}_3) \quad (-\mathbf{e}_2 - \mu \mathbf{e}_3)]^T$, and $\pm \vartheta_{\max}^{-1}$ on ϕ, θ for the attitude rows; so the contribution of an active constraint row to the stage curvature is data-independent and sparse, whichever rows the solver selects.

B. Stage parameters

The controller's model callbacks receive a state and an input, and the library fixes all dimensions at compile time; the per-stage parameters π_k therefore travel out of band, in a table that the driver refills before every preparation phase and that the generated model reads for the stage whose linearisation is being formed. The requirement this places on the solver is that every stage is linearised with its own row of the table, and that the table is interpreted for the new stage indexing after the shift; the requirement

it places on the operating mode is that stage data are re-evaluated every sample, since a contact switch changes the input Jacobian of a foot from nonzero to exactly zero without moving the iterate. Both are stated in the interface documentation released with the code.

C. What the controller is asked to do

The controller runs one real-time iteration per sample [13]: a preparation phase that re-linearises the problem around the current warm start and shifts it by one stage, and a feedback phase that maps the fresh state estimate to the input to apply, available $0.11 \mu\text{s}$ later. Three properties of the interface are what the walking problem needs, and they are the only ones this study relies on.

Constraints inside the solve. The rows (23) are enforced by the solver rather than by clipping the result. This matters at every contact switch: a foot leaving stance has its normal-force lower bound active at zero force while the entering foot’s pyramid becomes relevant, and the switch is resolved within the cycle instead of producing a force command that the leg controller would have to saturate.

Warm start with a one-stage shift. The stage table, the references and the internal iterate all shift together, so a stage that was “swing” at index $k + 1$ is “swing” at index k one sample later. The contact switch is then a non-event for the iteration: the solver needed a second inner pass on only 3 % of the samples of the main run, all of them at switches.

An initial convergence phase. At $t = 0$ the problem is solved to a KKT residual of 10^{-6} at the frozen initial state (three cycles from the force reference (22)); afterwards exactly one cycle per sample is executed. Sec. VII-D quantifies how far one cycle stays from the converged solution during walking.

1) Controllability of the diagonal support

A remark on why an SRBD horizon of 0.5 s suffices for a trot. With two point feet at $\mathbf{r}_1, \mathbf{r}_2$ the achievable moment about the COM is $\boldsymbol{\tau} = [\mathbf{r}_1 - \mathbf{p}]_{\times} \mathbf{f}_1 + [\mathbf{r}_2 - \mathbf{p}]_{\times} \mathbf{f}_2$; its component along the support line $\boldsymbol{\ell} = \mathbf{r}_1 - \mathbf{r}_2$ is $\boldsymbol{\ell}^T \boldsymbol{\tau} = \mathbf{f}_1^T (\boldsymbol{\ell} \times (\mathbf{r}_1 - \mathbf{p})) + \mathbf{f}_2^T (\boldsymbol{\ell} \times (\mathbf{r}_2 - \mathbf{p}))$, which is nonzero unless the COM lies on the line. In trot the COM is ≈ 3 cm off the diagonal (the feet are not symmetric about the COM because of the swing-leg mass distribution and the Raibert offsets), so the instantaneous rigid-body dynamics are controllable in all six directions, but with a small lever arm about the diagonal; the roll-about-the-diagonal mode is therefore governed mainly by the *next* foot placement, which is why the horizon must contain the next touchdown and why the foothold heuristic (13) and the NMPC are complementary rather than redundant: the NMPC balances within a stance, the footholds between stances. This is also the mechanism behind the $T_g = 0.5\text{--}0.6$ s failures of Sec. VIII-E: a shorter stance leaves less time to steer the roll mode with a small lever arm.

V. MOVING-HORIZON ESTIMATION ON THE AUGMENTED MODEL

A. Why the disturbance is a state and not an input

The estimator’s natural instantiation makes the disturbance an estimated *input*, $\mathbf{w} = \mathbf{d}$ in (5), penalised by $\frac{1}{2} \|\mathbf{w}\|_W^2$ with $n_x = 12$, $n_w = 6$. That version was implemented first and rejected.

The reason is observability of the newest value, not a property of any particular solver. A disturbance acting over one 10 ms interval moves the velocity by $h\mathbf{d}_f/m \approx 2 \times 10^{-4}$ m/s per newton and the position by $h^2\mathbf{d}_f/(2m) \approx 10^{-6}$ m per newton. Against a velocity measurement of 3 cm/s and a foot-position measurement of 1 cm standard deviation, a single sample therefore says almost nothing about the disturbance that acted during it, and with a white-noise prior there is nothing else in the problem that ties it to its neighbours. A single real-time iteration per sample then recovers each disturbance value only gradually while it remains in the window, and the value at the *head* of the window—the one the controller needs—is systematically biased towards zero. This is what was observed: under a 60 N standing push the window held $34 \rightarrow 30 \rightarrow 23 \rightarrow 17 \rightarrow 9 \rightarrow 3$ N towards the newest node, the head of the window reported 36–40 N of a 60 N push, and the velocity estimate lagged the truth by 40 %.

The remedy is the standard one for persistent disturbances [12]: carry the disturbance in the state as a random walk. The estimator's model is the augmented system

$$\mathbf{x}^e = \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \end{bmatrix} \in \mathbb{R}^{18}, \quad \dot{\mathbf{x}}^e = \begin{bmatrix} f_c(\mathbf{x}, \mathbf{f}; \mathbf{r}, \mathbf{s}, \mathbf{d}) \\ \mathbf{w} \end{bmatrix}, \quad \mathbf{w} = \dot{\mathbf{d}} \in \mathbb{R}^6, \quad (24)$$

in which the estimated input $\mathbf{w}$ is the *increment* of the wrench, so the prior constrains how fast the disturbance may drift (here ≈ 8 N and 2 Nm per sample at unit cost) instead of how large it may be, and the level $\mathbf{d}$ accumulates the information of the whole window. With this model the same 60 N push is estimated at 61.8 N with a rise time of ≈ 0.3 s and the velocity estimate is unbiased (Sec. VIII-C). The cost is six extra states.

B. The window problem

At sample t with window length $M = 20$ ($h = 10$ ms, 0.2 s), the estimator solves

$$\min_{\mathbf{x}_{0:M}^e, \mathbf{w}_{0:M-1}} \frac{1}{2} \|\mathbf{x}_0^e - \mathbf{x}_a\|_{\Lambda_a}^2 + \sum_{k=0}^M \frac{1}{2} \|\mathbf{y}_k - h(\mathbf{x}_k^e, \mathbf{u}_k)\|_V^2 + \sum_{k=0}^{M-1} \frac{1}{2} \|\mathbf{w}_k\|_W^2 \quad (25a)$$

$$\text{s.t. } \mathbf{x}_{k+1}^e = F^e(\mathbf{x}_k^e, \mathbf{u}_k, \mathbf{w}_k), \quad c(\mathbf{x}_k^e, \mathbf{w}_k) \leq 0, \quad c_M(\mathbf{x}_M^e) \leq 0, \quad (25b)$$

where the *known input* carries everything the estimator must be told about the contact situation,

$$\mathbf{u}_k = (\mathbf{f}_k^{\text{meas}}, \mathbf{r}_k, \mathbf{s}_k^{\text{dyn}}, \mathbf{s}_k^m, g_{v,k}) \in \mathbb{R}^{33}, \quad (26)$$

with the interval input $\mathbf{f}^{\text{meas}}$ being the mean realised GRF over $[t_k, t_{k+1}]$ and the node input carrying the anchors and gates at t_k . Using the *realised* rather than the commanded forces was the second decisive modelling choice: commanded forces make every touchdown impact (realised forces exceeding commands by 250 N for one sample) look like a disturbance that saturates the wrench bounds and lags the roll estimate by a factor of two; realised forces make the SRBD residual zero-mean with isolated ± 40 –170 N single-sample spikes at touchdown (Sec. VII-C).

1) Weights and prior

V is diagonal with standard deviations $\sigma_\theta = 0.01$ rad (attitude), $\sigma_\omega = 0.05$ rad/s, $\sigma_v = 0.03$ m/s, $\sigma_\rho = 0.01$ m; W has $\sigma_{d_f} = 800$ N/s and $\sigma_{d_\tau} = 200$ Nm/s, i.e. the wrench may drift by ≈ 8 N and 2 Nm per sample at unit cost. The initial arrival information is $\Lambda_{a,0} = \text{diag}(10^2 I_3, 10^3 I_3, 10 I_3, 10 I_3, 10^{-3} I_3, 10^{-2} I_3)$ on $(\mathbf{p}, \boldsymbol{\theta}, \mathbf{v}, \boldsymbol{\omega}, \mathbf{d}_f, \mathbf{d}_\tau)$ around the true initial state and zero wrench; thereafter the arrival cost is produced by the estimator itself as the window slides. The choice $\sigma_v = 0.03$ (rather than 0.1) and $\sigma_\rho = 0.01$ (rather than 0.005) removed a -2.3 cm/s forward-velocity bias caused by the anchoring feedback of Sec. III-F (Sec. VIII-E).

2) Constraints

Stage rows ($n_c = 28$): wrench bounds $|d_{f,j}| \leq 800$ N, $|d_{\tau,j}| \leq 300$ Nm (twelve rows, pure state rows), attitude $|\phi|, |\theta| \leq 0.6$ rad (four rows), and increment bounds $|w_j| \leq 2000$ (twelve rows); newest node ($n_c^M = 16$): the wrench and attitude rows. All rows are normalised by their bound. The wrench bounds are deliberately wide: they are a safety net against unbounded excursions of an unobservable direction (e.g. $d_{\tau,z}$ while no foot is settled), not a design tool.

C. What the estimator is asked to do

The estimator is used exactly as the controller is: one real-time iteration per sample, with the work split into a preparation phase that runs before the measurement arrives (residuals and linearisations of

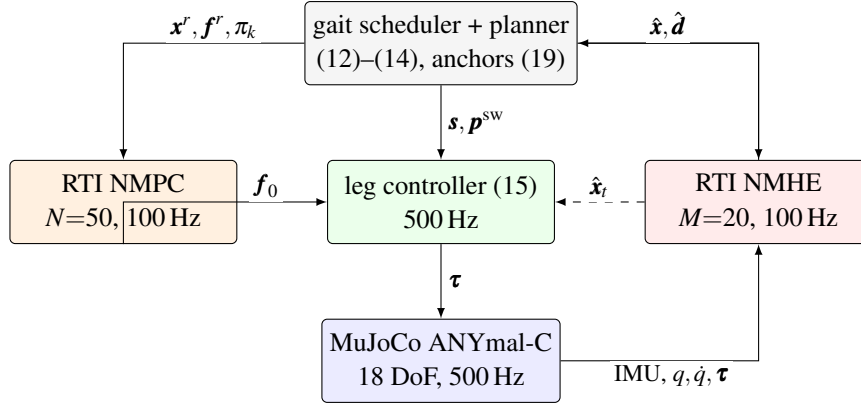


FIGURE 1. Unified loop. The planner is the only component that sees both solvers; it turns the NMHE state into the NMPC references, stage parameters and swing targets, and anchors feet from the estimate. The dashed arrow indicates that the estimate reaches the NMPC through the planner.

the window as it stands) and a feedback phase that maps the fresh measurement to an estimate. Three properties of the interface matter for the loop.

Time-varying known inputs. Contact flags, foot anchors and measurement gates are part of the known input (26) of an interval, and an interval’s known input never changes after the interval has elapsed. The structural change of the measurement model at a contact event is therefore confined to the entering interval; nothing earlier in the window has to be revisited.

Two outputs per sample. The estimator returns an estimate derived from the measurement innovation, available $1.1 \mu\text{s}$ after the measurement, and a smoothed newest-node estimate $37 \mu\text{s}$ later, once the inner iterations have run. On this problem the two are statistically indistinguishable (velocity rms 8–9 mm/s for both); the controller consumes the smoothed one, and the disturbance wrench $\hat{\mathbf{d}}$ is read from the same node.

A window that forgets slowly. When the window slides, the information of the leaving interval is folded into an arrival cost rather than discarded. For the augmented model this is where the wrench’s memory lives: $\hat{\mathbf{d}}$ at the head of the window is anchored by every interval that has ever left it, not only by the twenty currently visible, which turns a 0.2 s window into a disturbance observer whose forgetting horizon is set by W alone.

D. Bounds and the model domain

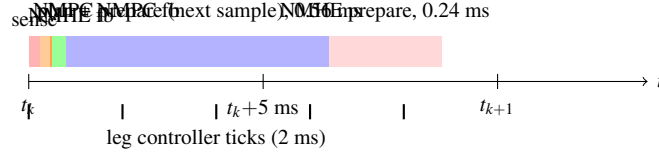
Beyond the rows above, the estimator is given a box that acts as a model domain rather than as a design constraint: $z \in [0.05, 1.5] \text{ m}$, $|\phi|, |\theta| \leq 1.2 \text{ rad}$, $\|\mathbf{v}\|_\infty \leq 5 \text{ m/s}$, $\|\boldsymbol{\omega}\|_\infty \leq 20 \text{ rad/s}$, and per-cycle step bounds of $(1, 1, 1) \text{ m}$, $(0.5, 0.5, 1) \text{ rad}$, $(2, 2, 2) \text{ m/s}$, $(3, 3, 3) \text{ rad/s}$ on the rigid-body block and $(400, 400, 400) \text{ N}$, $(150, 150, 150) \text{ Nm}$ on the wrench block. Their purpose is to keep a transient measurement outlier from moving the iterate into a region where the attitude parametrisation or the inertia inverse is meaningless. Neither the box nor the step bounds became active after engagement in any reported run.

VI. THE JOINT ARCHITECTURE

Figure 1 summarises the closed loop; Table 2 its rates. The two solvers share the generated header `anymal_models.h`, produced by one SymPy script that emits the controller’s function set (with the stage table) and the estimator’s function set (with the augmented state) from the same expressions (5), verified against each other at run time (Sec. VII-A).

TABLE 2. Rates and dimensions of the loop.

Component	rate	dims	notes
MuJoCo plant	500 Hz	$n_q = 19, n_v = 18$	torque actuators, ± 80 Nm
leg controller	500 Hz	12	(15)
RTI NMHE	100 Hz	18/6/21/33/28/16	$n_x/n_w/n_y/n_u/n_c/n_c^M, M = 20$
RTI NMPC	100 Hz	12/12/28/0	$n_x/n_u/n_c/n_c^N, N = 50$
planner	100 Hz	—	shares $\hat{\mathbf{x}}$ with both


 FIGURE 2. Schedule of one 10 ms sample (not to scale within the first millisecond). Sensing, the estimator feedback, planning and the controller feedback complete in ≈ 0.3 ms; both preparations for the next sample overlap the plant's motion.

1) Ordering within a sample

At a 10 ms boundary the driver (i) forms the measurement (16), detects contacts, updates the anchors and gates and runs the estimator cycle, (ii) builds references and stage parameters from the fresh estimate and runs the controller cycle, (iii) starts accumulating the realised GRF for the next interval input. The interval input must be the force applied *after* the NMPC update of the same instant; using the pre-update force (a one-sample staleness) produced spurious 40 Nm torque disturbances at every force change and was the first bug found in closed loop.

2) Latency budget

Both solvers expose an instantaneous output: the controller's first input is available $0.11 \mu\text{s}$ after the estimate, and the estimator's innovation-based estimate $\hat{\mathbf{x}}_t^{\text{fast}}$ $1.1 \mu\text{s}$ after the measurement. The measurement-to-actuation latency of the joint loop is therefore dominated by the leg controller's 2 ms period and the NMHE's smoothing feedback ($37 \mu\text{s}$), not by the optimisation: the sample ordering of Algorithm 1 places the two instantaneous outputs back to back so that a fresh GRF command reaches the joints ≈ 0.3 ms after the sensors were read, with the remaining ≈ 0.5 ms of NMPC preparation for the *next* sample overlapping the plant's motion—the RTI prepare/feedback split at the system level. Figure 2 illustrates the schedule.

3) Disturbance feed-forward

The NMPC receives $\mathbf{d} = (\hat{\mathbf{d}}_f, \mathbf{0})$ for all stages: the estimated force is fed forward, the estimated torque is not. The torque channel is estimated with a within-trot standard deviation of 4–6 Nm (the reaction of the swinging legs on the trunk, which the SRBD does not model) and its feed-forward closed a positive-feedback loop through the roll dynamics that doubled the roll excursion and quadrupled the NMPC's inner-iteration load (Table 5); the force channel is smooth (2–3 N within trot) and its feed-forward reduces the drift under the lateral push by a factor of five. This is a bandwidth statement, not a solver limitation: a disturbance observer should only be closed over frequencies it estimates faithfully.

VII. VERIFICATION

A. Model and derivatives

The generated C model was checked at random points against (i) the SymPy expressions it was generated from (1.7×10^{-13} maximum deviation, 8.5×10^{-14} for the augmented set), (ii) central finite differences of all Jacobians, including the RK4 chain rule for the discrete A_k, B_k ($2.5\text{--}3.2 \times 10^{-9}$ relative), and (iii)

each other: the controller’s dynamics $f_c(\mathbf{x}, \mathbf{f}; \boldsymbol{\pi})$ and the first twelve rows of the estimator’s augmented dynamics agree bitwise for the same arguments, which is the operational meaning of “one model source”.

B. Model consistency on the plant

The SRBD residual $m\Delta\mathbf{v}/h - \sum_i \mathbf{f}_i^{\text{meas}} - m\mathbf{g}$ evaluated with the interval-mean realised forces of the closed-loop run is zero-mean (< 0.05 N in all axes over 6 s of trot) with standard deviation (22, 6, 29) N and isolated single-sample spikes of up to (132, 36, 169) N at touchdowns—the velocity jump of the landing leg’s mass, which the lumped model attributes to the trunk. The torque balance with the commanded forces and the true attitude closes to 0.6 Nm rms while standing. These numbers bound what the disturbance state has to absorb and confirm the sign and frame conventions of (5).

C. Fixed-parameter convergence of the NMPC

At a mid-swing instant of the walk ($t = 2.35$ s, LF/RH in stance, two contact switches inside the horizon) the iteration was run at frozen $\hat{\mathbf{x}}$ from a cold start ($\mathbf{f} \equiv 0$) and from the weight-distributing warm start of (22). The KKT residual decreases as $1.1 \times 10^1 \rightarrow 1.4 \times 10^{-1} \rightarrow 2.0 \times 10^{-3} \rightarrow 1.4 \times 10^{-5} \rightarrow 7.4 \times 10^{-8} \rightarrow 4.1 \times 10^{-10}$ (cold) and $6.0 \times 10^{-2} \rightarrow 1.2 \times 10^{-2} \rightarrow 2.2 \times 10^{-5} \rightarrow 7.3 \times 10^{-7} \rightarrow 1.5 \times 10^{-9} \rightarrow 4.0 \times 10^{-11}$ (warm), i.e. six cycles to 10^{-10} in both cases with a contraction factor of $\approx 10^{-2}$ per cycle (Fig. 10). The cold and warm starts converge to the same solution, including the same set of constraints active at the contact switches inside the horizon, which is the check that the warm start used in closed loop is not biasing the solution. In closed loop the one-cycle KKT residual stays at $7\text{--}8 \times 10^{-2}$ in the unnormalised units of (20) ($\approx 10^{-3}$ relative to the cold-start value) and the maximum constraint violation of the committed input is 3.8×10^{-2} in normalised units (8.6 N on a 400 N cone edge, one sample, at a touchdown).

D. Estimator mode consistency

The estimator can re-linearise every stage of the window at every sample, or re-use stage information that the window structure makes re-usable. Running both modes on an identical plant trajectory (ground-truth-fed control, so that both see the same data) gives estimates whose rms errors agree to four significant digits in every component, with a maximum pointwise difference of 1.3×10^{-4} (state, SI units) and 9.6×10^{-3} N (wrench) over 800 samples, while the preparation phase costs 215 μs instead of 1170 μs . The cheaper mode is used in all closed-loop runs; the agreement is the check that it is not trading accuracy for speed.

VIII. CLOSED-LOOP RESULTS

A. Setup

Plant: MuJoCo 3.13, ANYmal-C from the Menagerie with the position servos replaced by direct torque motors (± 80 Nm) and the foot contacts stiffened (`solimp 0.9 0.95 0.001, solref 0.005 1,  $\mu_{\text{plant}} = 0.8$` ; the Menagerie’s soft feet, `solimp 0.015 1 0.03`, creep laterally under sustained load and bias leg odometry by 25 %), 2 ms time step, joint damping 1 Nms/rad and dry friction 0.1 Nm as shipped. Sensor noise (white, per sample): attitude 5 mrad, gyro 0.02 rad/s, joint velocities 0.05 rad/s, forward kinematics 3 mm, realised GRF 2 N. Scenario: stand for 1 s; ramp a forward command to 0.4 m/s over 1 s and trot; lateral push of 40 N on the trunk for 0.8 s from $t = 3.5$ s; yaw-rate command 0.3 rad/s from $t = 5$ s; end at 8 s. Timing host: shared cloud vCPU (Intel Xeon, 2.8 GHz class), gcc 13 -O3 -march=native, single core, wall-clock per call via `clock_gettime`; the simulator, renderer and Python driver share the core, so the numbers are conservative relative to an isolated single-core measurement.

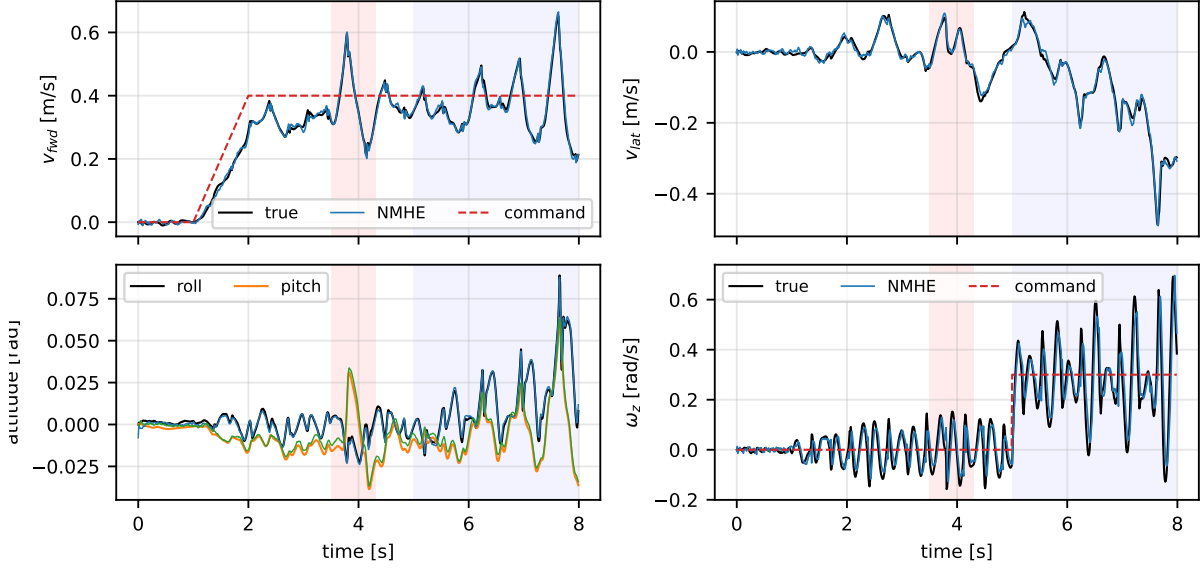


FIGURE 3. Main run: body-frame forward and lateral velocity, attitude and yaw rate; true (black), NMHE estimate (blue/green), commands (dashed). Shaded: push (red), turn (blue).

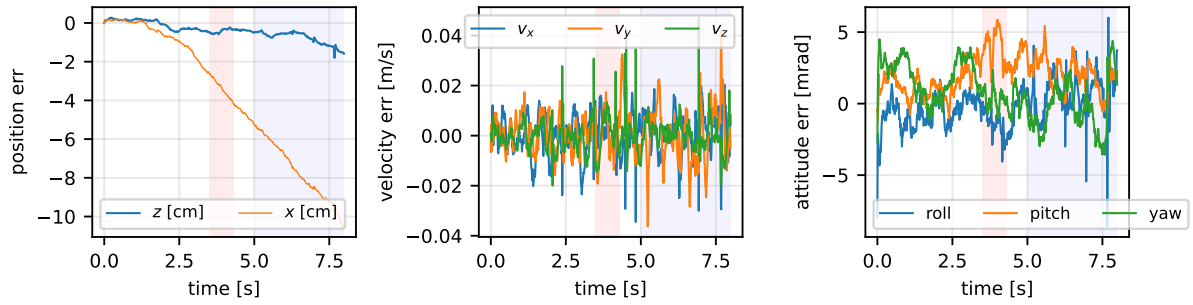


FIGURE 4. Main run: estimation errors of the smoothed NMHE output. Position drifts (leg odometry is dead reckoning); velocity and attitude do not.

B. Main run

Figure 3 shows the closed-loop tracking, Fig. 4 the estimation errors, Fig. 5 the estimated wrench, and Fig. 6 frames of the animation. In steady trot the forward speed is 0.32 ± 0.02 m/s during the ramp-out phase (2.5–3.5 s) and 0.38 m/s in the turn, against a 0.4 m/s command; the deficit is consistent with the -5 N per foot longitudinal drag of the plant’s joint damping and friction, which appears in the estimated $\hat{d}_{f,x}$ as $+1.9$ N (net of the feed-forward). Roll stays within ± 0.09 rad and pitch within ± 0.06 rad over the whole run including push and turn; height is 0.452 ± 0.002 m against 0.45 m commanded. The yaw rate settles at 0.28 rad/s for a 0.3 rad/s command; during the turn the body-frame lateral velocity drifts to -0.2 to -0.4 m/s, i.e. the heading is tracked but the velocity vector lags it and the robot crabs partly sideways along the arc—a limitation of the foothold heuristic (13), which places feet from the current velocity rather than from the commanded arc, not of the solvers. Estimation errors (rms over $t > 1.5$ s): position (1.0, 0.5, 0.4) cm (odometry drift, not noise), attitude (1.4, 2.6, 1.7) mrad, velocity (9, 10, 7) mm/s, angular velocity (0.19, 0.07, 0.07) rad/s—the roll-rate error being the trot’s 10–20 Hz roll vibration which the SRBD does not predict and the gyro row (0.05 rad/s) smooths. The 40 N push is estimated at 39.5 N (mean over its last 0.5 s) with the 0.3 s rise time expected from W , and the estimate returns to -1.4 N within 0.3 s after release. Within-trot wrench fluctuations are (2.9, 2.2, 2.1) N and (5.8, 3.7, 5.4) Nm.

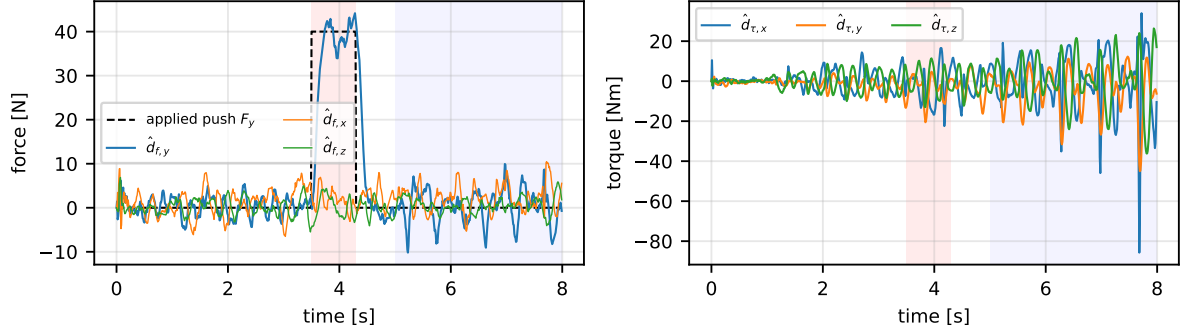


FIGURE 5. Main run: estimated wrench state $\hat{\mathbf{d}}$. The 40 N lateral push (dashed) is recovered by the augmented random-walk state; the torque channel carries the swing-leg reaction at the gait frequency.

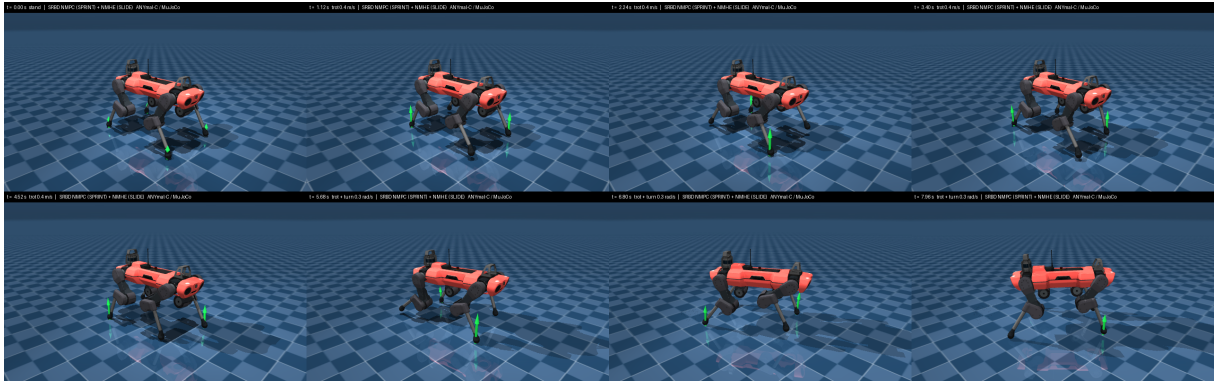


FIGURE 6. Frames from the released animation (media/anymal_walk_hd.mp4): stand, trot, push, turn. Green arrows are the commanded ground reaction forces $\mathbf{f}_i$ (scaled 1 m $\equiv$ 400 N), the red arrow the push.

C. Force commands and the predicted horizon

Figure 7 compares commanded and realised GRFs over one gait cycle: the diagonal pair carries ≈ 210 –230 N per foot vertically with $|f_x| \leq 20$ N, the commands stay inside the pyramid with margin, and the realised forces follow to within the touchdown transient (the first 20–40 ms of each stance) and the leg-inertia term of Sec. III-E. Figure 8 shows one NMPC solution: the horizon spans two contact switches, the predicted height rises 1 mm over 0.5 s towards its reference (the NMPC’s height loop is proportional; offset removal is left to the disturbance feed-forward), and the force plan hands the load over between the diagonal pairs at the switches without transient—the warm start (22) and the shifted iterate make a contact switch a non-event for the solver, which is why the inner loop runs a second pass on only 3 % of the samples.

D. Timing

Table 4 and Fig. 9 give the per-sample distributions. The NMPC preparation (0.56 ms median, 0.71 ms p99, 1.07 ms max) is dominated by the linearisation of 50 stages; its feedback (56 μ s median) includes an occasional second inner pass (p99 194 μ s, max 412 μ s at contact switches). The applied-input latency is 0.11 μ s. The NMHE preparation (0.24 ms median) covers the linearisation of the window; its instantaneous output costs 1.1 μ s and the smoothed feedback 37 μ s. One complete estimate-then-control cycle therefore costs ≈ 0.9 ms median and ≤ 2.1 ms worst case against the 10 ms period, on a shared core, in plain C without SIMD—roughly 9 % utilisation with a $5\times$ worst-case margin.

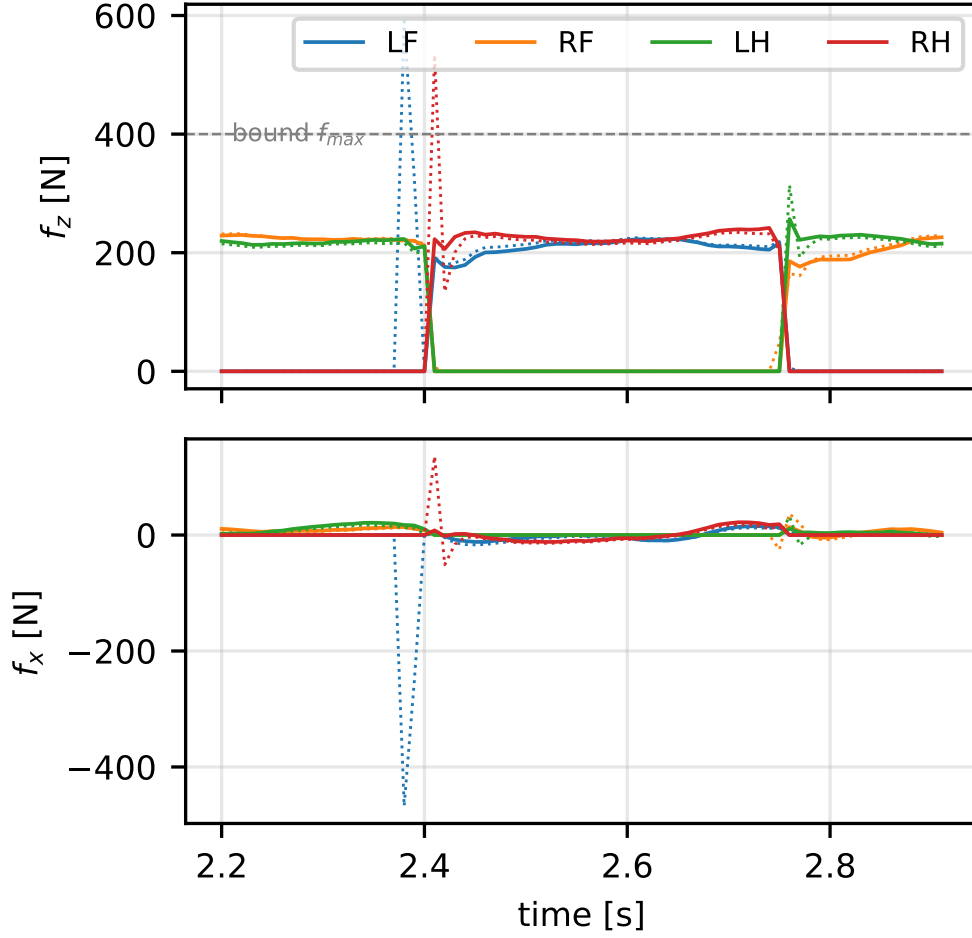


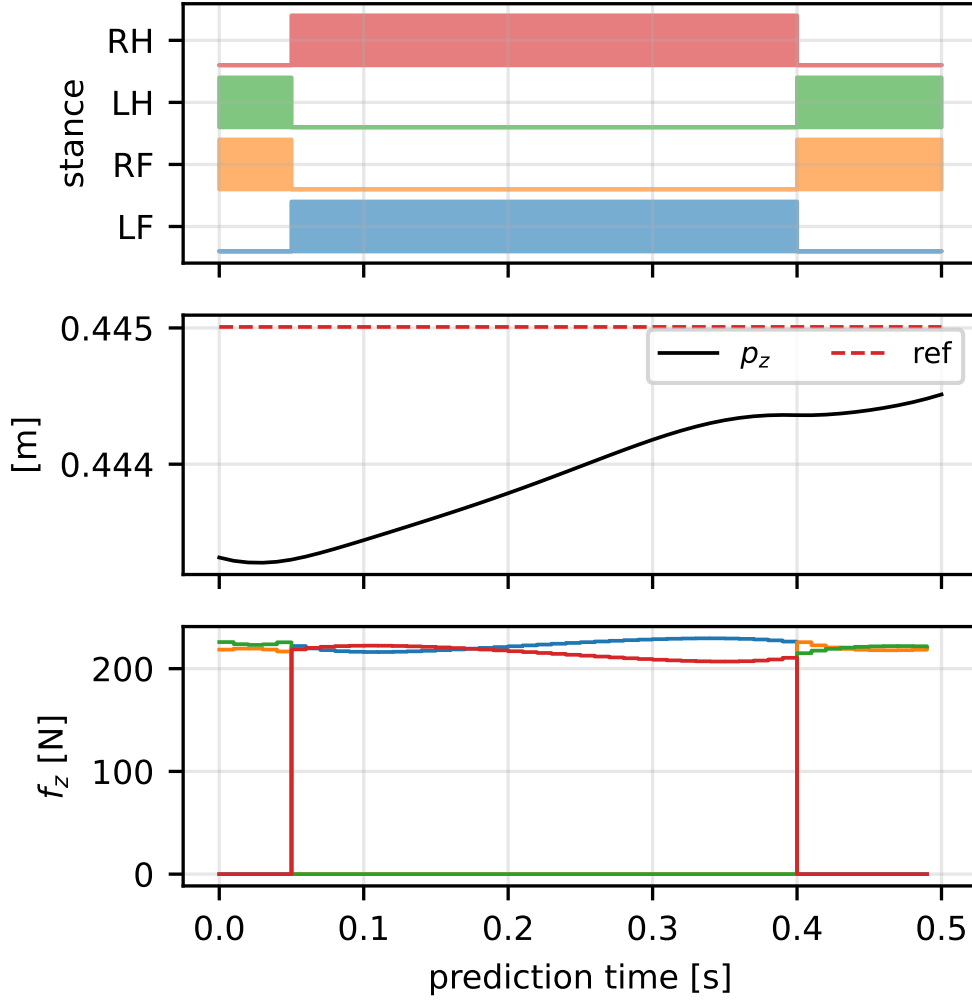
FIGURE 7. Commanded (solid) and realised (dotted) vertical and longitudinal GRFs over one gait cycle of the main run.

1) Horizon scaling

Rebuilding the NMPC for $N \in \{10, 20, 30, 50, 80, 120\}$ (same h , same weights) and timing it on the walking scenario gives Table 3: preparation and feedback are affine in N with slopes of 10.9 and 1.15 μs per stage and the applied-input latency is flat at 0.1 μs , as expected for a stage-wise structured solver; memory grows by 16.0 kB per stage. All horizons from 0.1 s upward trot stably at 0.3 m/s on this scenario (no push); $N = 120$ (1.2 s) walks slower because a reference arc re-anchored at the current position is a poor description of the gait that far ahead. The choice $N = 50$ is a comfortable 0.7 gait cycles at 5 % of the period.

E. Ablations

Table 5 and Fig. 11 compare, on the same seed and scenario: (a) ground-truth state feedback (no estimator), (b) the NMHE without feed-forward, (c) force feed-forward (the proposed configuration), (d) full wrench feed-forward, and (e) the NMHE in exact mode. Two observations. First, the estimator costs essentially nothing in nominal control performance: roll and pitch excursions with the NMHE (0.09/0.06 rad) are within a factor two of the truth-fed run (0.04/0.05 rad) and the forward speed is higher (0.32 vs 0.29 m/s), because the estimated drag is fed forward. Second, feed-forward of the estimated force is what turns the push from a 0.74 m lateral excursion (uncompensated, 0.26 m with truth feedback) into 0.15 m, while feed-forward of the torque as well degrades everything (roll 0.18 rad, speed overshoot to 0.64 m/s in the turn, NMPC feedback p99 of 1.1 ms from inner-iteration churn). The exact-mode NMHE reproduces the


 FIGURE 8. One NMPC solution ($t = 2.35$ s): predicted contact schedule, height, and vertical force plan over the 0.5 s horizon.

reuse-mode run to the resolution of the table at five times the estimator cost.

F. Sensitivity to sensor noise

Scaling all sensor noises jointly by a factor $\alpha \in \{0, \frac{1}{2}, 1, 2, 4\}$ (nominal: attitude 5 mrad, gyro 0.02 rad/s, joint velocities 0.05 rad/s, forward kinematics 3 mm, GRF 2 N) gives Table 6. Estimation errors grow essentially linearly with α up to $\alpha = 2$ (velocity 4 mm/s noise-free to 16 mm/s at $\alpha = 2$; attitude 0.8 to 2.6 mrad), the push estimate stays within 4 % of 40 N, and the closed loop remains stable with roll excursions growing from 0.06 to 0.13 rad. At $\alpha = 4$ (20 mrad attitude, 0.08 rad/s gyro, 0.2 rad/s joint velocities, 12 mm kinematics, 8 N forces) the loop falls during the turn: the roll-rate estimate error reaches 1 rad/s and the NMPC, which sees the noisy roll rate through Q_ω , over-reacts. The noise-free run is not better than the nominal one in control performance (roll 0.06 vs 0.09 rad), confirming that the nominal sensor noise is not what limits the loop; the SRBD residual is.

G. Robustness over noise seeds and what it took

Five independent noise realisations complete the full scenario without a fall: roll maxima 0.03–0.09 rad, pitch 0.05–0.06 rad, forward speed 0.28–0.32 m/s during the ramp-out, yaw rate 0.27–0.29 rad/s, push excursion 0.03–0.13 m, velocity rms ≤ 12 mm/s (Table 7). Table 8 lists, in the order they were found,

TABLE 3. NMPC horizon scaling on the walking scenario (medians over 300 samples; times in μs , memory in kB).

N	horizon [s]	prep	prep p99	feedback	u_0 latency	RAM
10	0.1	113	156	11.7	0.11	168
20	0.2	220	263	22.1	0.12	328
30	0.3	311	359	31.7	0.10	488
50	0.5	531	656	54.5	0.11	809
80	0.8	889	1184	90.4	0.12	1290
120	1.2	1304	1759	136.8	0.12	1931

 TABLE 4. Per-sample wall times [μs] in the main run (800 NMPC and 800 NMHE cycles) and static memory.

solver	phase	med	p99	max	
NMPC	preparation	564	708	1065	809 kB
	feedback	56	194	412	
	u_0 latency	0.11			
NMHE	preparation	241	560	1019	579 kB
	prep. (full relin.)	1190	1468	1840	
	instant output	1.1			
	feedback	37	68	423	

the modelling decisions that separated a loop that falls within two strides from one that walks; each was isolated by comparing ground-truth-fed and estimator-fed closed loops and estimator-only runs on identical trajectories. None of them required a change to either kernel; all of them are recorded because a reader porting the framework to another robot will meet them again.

IX. DISCUSSION AND LIMITATIONS

What the SRBD costs on this robot. With the legs carrying 57 % of the mass, the lumped model is coarse: the touchdown residuals of Sec. VII-C, the gait-frequency torque channel of Fig. 5, the 0.19 rad/s roll-rate estimation error and the 5–10 % speed deficit are all leg-mass effects. They are absorbed by three mechanisms that this report makes explicit—realised forces as known inputs, the augmented wrench state, and the choice of what to feed forward—rather than hidden in tuning. A centroidal or kino-dynamic model would remove most of them at the price of a $n_x \geq 24$ NMPC; nothing in the solver interfaces prevents that extension, and the stage-parameter mechanism carries over unchanged.

Contact softness. Leg odometry assumes stationary feet; compliant contacts violate it. The Menagerie’s soft feet gave a 25 % velocity bias that no estimator can correct without modelling the compliance; we stiffened the plant, which is the honest choice for a study of the estimator rather than of the contact model. Real ANYmal feet are stiff; real ground is not always.

Odometry drift and yaw. Absolute position and, strictly, yaw are unobservable from proprioception; the AHRS row supplies yaw here. Position drifted 2 cm over 5 s in the main run—an artefact of the anchoring rule (19), not of the solver—and is irrelevant for velocity-commanded locomotion.

No external optimality oracle. IPOPT was not available in the build environment. Optimality is therefore established by the fixed-parameter convergence of Sec. VII-D (to 10^{-10} KKT residual from a cold start) and by the agreement of the two estimator linearisation modes in Sec. VII-E, not by a third-party solve. The release contains the CasADi model builders that the companion papers use for their IPOPT oracles, so the check can be run where IPOPT is installed.

Constraint semantics. The stage inequalities are enforced by the solver as penalised constraints and can be transiently violated during fast transients; the largest violation of a committed input was 8.6 N on a 400 N cone edge, for one sample, at a touchdown. Applications requiring strict inviolability should tighten μ or add a projection at the leg-controller level.

Timing. The reported times include Python/MuJoCo contention on a shared cloud core and are conservative. The per-sample cost of the NMPC grows linearly in the horizon (Table 3) and that of the

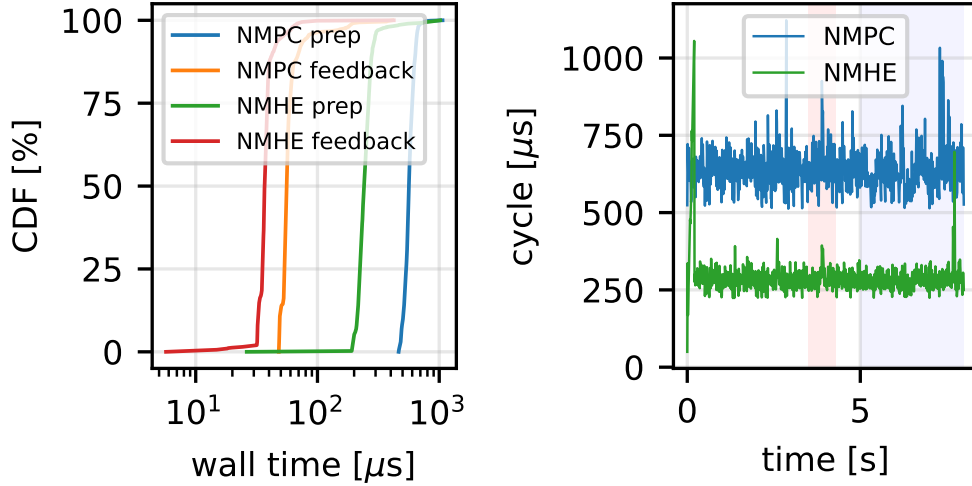


FIGURE 9. Left: empirical CDFs of the per-sample phases. Right: complete cycle times along the run; the NMPC’s bursts are contact-switch samples with a second inner pass.

TABLE 5. Ablations on the main scenario (seed 0). Extremes over $t > 1.5$ s; p_y at the end of the push ($t = 4.3$ s) and at $t = 8$ s; estimation rms over $t > 1.5$ s; NMPC feedback p99 [μs].

configuration	roll max	pitch max	$z_{\min}$	$z_{\max}$	v_{fwd} walk/turn	ψ turn	p_y push/end	$\hat{v}$ rms [mm/s]	$\hat{\theta}$ rms [mrad]	fb p99
truth-fed control (no NMHE)	0.044	0.047	0.445	0.457	0.29/0.29	0.29	0.26/0.70	—	—	189
NMHE, no feed-forward	0.073	0.051	0.448	0.462	0.32/0.41	0.28	0.24/0.74	(9,10,8)	(1.5,2.6,1.7)	180
NMHE, force feed-forward (proposed)	0.089	0.063	0.448	0.464	0.32/0.38	0.28	0.03/0.15	(9,10,7)	(1.4,2.6,1.7)	194
NMHE, full wrench feed-forward	0.176	0.211	0.444	0.503	0.38/0.64	0.19	0.05/0.33	(22,32,18)	(1.7,2.5,1.7)	1110
NMHE exact mode, force feed-forward	0.094	0.066	0.448	0.464	0.32/0.38	0.28	0.03/0.15	(9,10,7)	(1.4,2.6,1.7)	216

NMHE is constant in the window length, which is what makes the 10 ms budget comfortable. Both would benefit from blocked linear-algebra kernels and SIMD, and the NMPC from a shorter horizon at a coarser stage length (25 stages of 20 ms cover the same 0.5 s at half the cost; the 20 ms period was abandoned here only because the first tuning was too aggressive for it).

Towards the real robot. The framework’s interface to a physical ANYmal is small: joint angles, velocities and torques at 500 Hz, an AHRS attitude and gyro, and a contact-detection threshold. The realised-GRF reconstruction $-J_i^{-T}(\tau_i^{\text{meas}} - \mathbf{b}_i)$ uses the same leg model as the controller and is standard on torque-controlled quadrupeds; the AHRS yaw can be replaced by integrating the gyro if no magnetometer is available (yaw then drifts like position, which the velocity-commanded loop tolerates). The two kernels’ static memory (809 kB and 579 kB) and worst-case cycle times (≤ 2.1 ms together on a shared core) fit any onboard computer; on a Cortex-A class embedded target the N -scaling of Table 3 suggests $N = 30$ (0.3 s) as a safe choice, which walked as well as $N = 50$ in the horizon sweep.

X. CONCLUSION

A single symbolic SRBD model, instantiated once as an RTI NMPC with gait-scheduled stage parameters and once as an RTI moving-horizon estimator with an augmented wrench state and a known-input encoding of the contact situation, closes an output-feedback trot on a full-body ANYmal-C simulation with millisecond-scale solver cycles, centimetre-per-second velocity estimates, and a disturbance observer accurate enough to feed forward. Both solver libraries were used unmodified, behind an interface small enough that another RTI package can be substituted without touching the framework. One iteration per sample sufficed throughout: a second inner pass was needed on 3 % of the samples, all at contact switches. The engineering that made the loop work—realised forces as known inputs, settled-stance gating, anchoring at detection, bias-compensated torque mapping, force-only feed-forward—is documented as

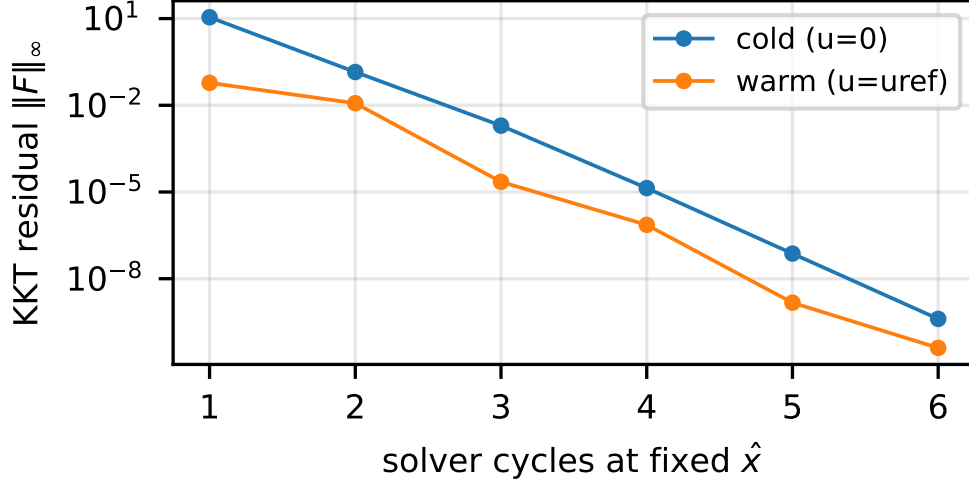

 FIGURE 10. Fixed-parameter convergence on the walking OCP at $t = 2.35$ s from a cold and a warm start.

 TABLE 6. Joint sensor-noise scaling α , proposed configuration, seed 0. Errors are rms over $t > 1.5$ s; push estimate is the mean $\hat{d}_{f,y}$ over the last 0.5 s of the 40 N push.

α	fall	roll	$\hat{\mathbf{v}}$ [mm/s]	$\hat{\boldsymbol{\theta}}$ [mrad]	$\hat{\omega}_x$	$\hat{z}$ [mm]	push [N]
0	no	0.059	(4,7,4)	(0.8,1.4,1.3)	0.16	1.1	40.7
0.5	no	0.071	(5,7,5)	(1.0,2.1,1.2)	0.17	3.7	40.0
1	no	0.089	(9,10,7)	(1.4,2.6,1.7)	0.19	6.4	39.5
2	no	0.126	(16,17,13)	(2.6,4.0,3.4)	0.24	9.7	38.5
4	yes ($t \approx 7$ s)	—	—	—	1.07	—	35.7

ablations so that it can be reused rather than rediscovered.

A. JACOBIAN STRUCTURE OF THE SRBD MODEL

Writing $\mathbf{m}_i = \mathbf{r}_i - \mathbf{p}$ and $\boldsymbol{\tau} = \sum_i s_i [\mathbf{m}_i]_{\times} \mathbf{f}_i$ (world frame), the nonzero blocks of $\partial f_c / \partial \mathbf{x}$ are

$$\begin{aligned}
 \frac{\partial \mathbf{p}}{\partial \mathbf{v}} &= \mathbf{I}, & \frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{\omega}} &= \mathbf{T}(\boldsymbol{\theta}), & \frac{\partial \dot{\boldsymbol{\theta}}}{\partial(\phi, \theta)} &= \begin{bmatrix} \frac{\partial \mathbf{T}}{\partial \phi} \boldsymbol{\omega} & \frac{\partial \mathbf{T}}{\partial \theta} \boldsymbol{\omega} \end{bmatrix}, \\
 \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \mathbf{p}} &= \mathbf{I}^{-1} \mathbf{R}^{\top} \sum_i s_i [\mathbf{f}_i]_{\times}, & \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\theta}} &= \mathbf{I}^{-1} \begin{bmatrix} \frac{\partial \mathbf{R}^{\top}}{\partial \phi} \boldsymbol{\tau} & \frac{\partial \mathbf{R}^{\top}}{\partial \theta} \boldsymbol{\tau} & \frac{\partial \mathbf{R}^{\top}}{\partial \psi} \boldsymbol{\tau} \end{bmatrix}, \\
 \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\omega}} &= \mathbf{I}^{-1} ([\mathbf{I} \boldsymbol{\omega}]_{\times} - [\boldsymbol{\omega}]_{\times} \mathbf{I}),
 \end{aligned}$$

and of $\partial f_c / \partial \mathbf{f}$

$$\frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{f}_i} = \frac{s_i}{m} \mathbf{I}, \quad \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \mathbf{f}_i} = s_i \mathbf{I}^{-1} \mathbf{R}^{\top} [\mathbf{m}_i]_{\times},$$

which vanish for swing feet. The augmented estimator model adds $\partial \dot{\mathbf{v}} / \partial \mathbf{d}_f = \mathbf{I} / m$, $\partial \dot{\boldsymbol{\omega}} / \partial \mathbf{d}_{\tau} = \mathbf{I}^{-1}$ and $\partial \dot{\mathbf{d}} / \partial \mathbf{w} = \mathbf{I}$. The output Jacobian of (16) has the nonzero blocks $\partial(s_i^m \mathbf{R}^{\top}(\mathbf{r}_i - \mathbf{p})) / \partial \mathbf{p} = -s_i^m \mathbf{R}^{\top}$ and $\partial(\cdot) / \partial \boldsymbol{\theta} = s_i^m [\partial \mathbf{R}^{\top} / \partial \phi \quad \partial \mathbf{R}^{\top} / \partial \theta \quad \partial \mathbf{R}^{\top} / \partial \psi](\mathbf{r}_i - \mathbf{p})$. All expressions are generated symbolically with common-subexpression elimination; the generated header is 2.7 kLOC and evaluates the controller's full function set in $\approx 1 \mu\text{s}$ per stage.

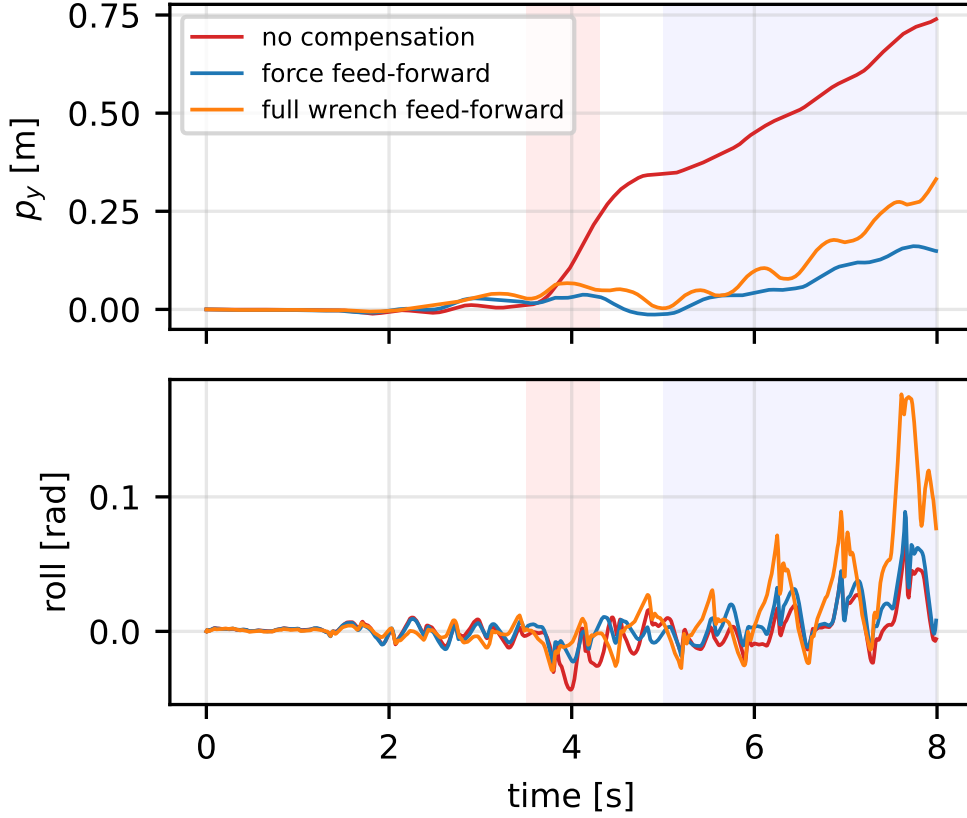


FIGURE 11. Push recovery: lateral position and roll for no / force / full wrench feed-forward.

TABLE 7. Five noise seeds, proposed configuration.

seed	fall	roll max	pitch max	v_{fwd}	ψ	p_y push	$\hat{v}$ rms [mm/s]
0	no	0.089	0.063	0.32	0.28	0.03	(9,10,7)
1	no	0.031	0.048	0.28	0.29	0.13	(8,11,6)
2	no	0.038	0.047	0.30	0.28	0.12	(10,12,7)
3	no	0.080	0.054	0.31	0.27	0.04	(10,11,7)
4	no	0.071	0.055	0.30	0.29	0.06	(10,10,6)

B. DERIVATIONS AND PROOFS

A. Raibert foothold from the linear inverted pendulum

Consider the horizontal COM dynamics of a body of height z_0 supported at a fixed foot position r : $\ddot{p} = \frac{g}{z_0}(p - r)$. Over a stance of duration T_{st} the body sweeps symmetrically about the foot if $p(0) - r = -\frac{1}{2}T_{st}v$, in which case the net horizontal impulse vanishes and the velocity is preserved, $v(T_{st}) = v(0)$; the “neutral point” of (13) is this symmetric placement. To change the velocity by $\Delta v = v^{cmd} - v$ the foot is displaced by δ , producing $\Delta v \approx -\frac{g}{z_0}T_{st}\delta$ to first order, i.e. $\delta = -\frac{z_0}{gT_{st}}\Delta v$. Raibert’s gain k_r in $\delta = k_r(v - v^{cmd})$ is therefore of the order $z_0/(gT_{st}) = 0.45/(9.81 \cdot 0.35) = 0.13$ s for this robot, and the capture-point value $\sqrt{z_0/g} = 0.21$ s is the gain that brings the body to rest in one step. The value $k_r = 0.12$ s used here is the LIP value; both larger (0.2) and much smaller (0.03–0.05) gains walked, with more lateral wander at the small end.

TABLE 8. Development ablations (chronological): symptom, cause, fix, and effect. “Truth-fed” = NMPC on ground-truth state; “est-only” = estimator run on a truth-fed closed loop.

Symptom	Cause	Fix	Effect
Standing loop rolls at 25 Hz with NMHE, stable truth-fed	NMHE interval input was the force <i>before</i> the NMPC update of the same instant (one-sample stale)	accumulate realised force after the update	spurious ± 40 Nm torque estimate removed
Body rises 5–9 cm in trot	$\tau = -J^T \mathbf{f}$ ignores leg gravity: realised GRF = $\mathbf{f} + J^{-T} \mathbf{b}$	add bias torques in stance (15)	realised–commanded -1 N mean
540–590 N impacts at touchdown	$\sin(\pi\sigma)$ swing height has -0.5 m/s end slope; targets in sphere-centre frame vs contact-point feedback (3 cm late)	bump profile, $z_{\text{land}} = -1.5$ cm, consistent frame	impacts ≤ 170 N single-sample
Speed 0.17–0.25 for 0.4 m/s command; $T_g = 0.5$ – 0.6 s unstable	foot placed ahead of the sweep (neutral point from commanded velocity)	neutral point from actual velocity, $T_g = 0.7$ s, k_r 0.05–0.12	0.32–0.38 m/s
$\hat{\mathbf{z}}$ jumps $+1$ cm per touchdown, $\hat{\mathbf{v}}$ spikes 2 m/s	kinematic velocity from a foot still moving; anchors from a foot still sliding	settled-stance gates (18), contact detection, contact-point Jacobians	$\hat{\mathbf{z}}$ max error 1.7 cm
$\hat{\mathbf{z}}$ drifts -2.5 cm per stride in closed loop only	scheduled-stance force gating dropped early-contact forces (1.5 cm overshoot lands early)	$s^{\text{dyn}} = \max(s, \delta)$, anchor at detection	drift 2 cm over 5 s
Push estimated at 36–40 of 60 N; velocity lags 40 %	disturbance modelled as a penalised estimator input is weakly observable over one interval (Sec. V-A)	random-walk wrench state (24)	61.8 N, no lag
-2.3 cm/s forward-velocity bias	anchoring from the filtered estimate inherits estimator lag; $\sigma_v = 0.1$ too weak	$\sigma_v = 0.03$, $\sigma_p = 0.01$	bias < 1 mm/s
Falls after push with full wrench feed-forward	torque channel carries gait-frequency swing reaction; feed-forward closes a roll loop	feed forward force only	roll 0.09 vs 0.18 rad
Speed runaway to 0.9 m/s while turning	world-frame velocity command integrated while the body yaws	body-frame command along the arc (21)	0.38 m/s, ψ 0.28

B. Swing-foot polynomials

The horizontal blend $\zeta(\sigma) = 10\sigma^3 - 15\sigma^4 + 6\sigma^5$ is the unique quintic with $\zeta(0) = 0$, $\zeta(1) = 1$ and zero first and second derivatives at both ends, so that the foot leaves and arrives with zero velocity and acceleration relative to the trajectory endpoints; its peak velocity is $\frac{15}{8}(\mathbf{r}^{\text{td}} - \mathbf{p}^{\text{lo}})/T_{\text{sw}}$, i.e. ≈ 0.5 m/s for a 10 cm step in 0.35 s. The height bump $\beta_4(\sigma) = 16\sigma^2(1 - \sigma)^2$ is the lowest-degree polynomial with $\beta_4(0) = \beta_4(1) = 0$, $\beta_4(\frac{1}{2}) = 1$ and zero end slopes; its end curvature is $32h_{\text{sw}}/T_{\text{sw}}^2 \approx 16$ m/s², which is what limits the residual touchdown velocity when tracking is imperfect. The landing offset $z_{\text{land}}\zeta(\sigma)$ adds an end slope of zero and a terminal target 1.5 cm below the surface, which the swing PD converts into a $K_p z_{\text{land}} = 13.5$ N downward pre-load at the scheduled touchdown; the contact detector then hands the leg to the force branch of (15) within one 2 ms step.

C. Gated measurement rows

A row of (16) multiplied by $s_i^m = 0$ has zero residual *and* zero Jacobian, so it contributes nothing to the estimation problem while the gate is closed; this is how a fixed output weight accommodates a measurement set that changes at every contact event, without changing the dimension of the estimator. The attitude and gyro rows are never gated and, with the arrival cost, keep the window problem well posed during the ≤ 40 ms intervals in which no foot is settled.

C. ALGORITHMS

Algorithm 1 One sample of the unified loop (every 10 ms)

-
- 1: **sense:** IMU attitude and gyro, joint angles/velocities, joint torques $\rightarrow$ realised GRF $\mathbf{f}^{\text{meas}}$; contact detection δ_i
 - 2: **gates/anchors:** s_i from (12); s_i^m from (18); update $\mathbf{r}_i$ by (19); form $\mathbf{y}$ by (16), node input $\mathbf{u}_t$ by (26)
 - 3: **NMHE:** $\text{nmhe_step}(\mathbf{u}_{t-1 \rightarrow t}, \mathbf{u}_t, \mathbf{y}) \rightarrow \hat{\mathbf{x}}^{\text{fast}}$ (1.1 μs), then $\hat{\mathbf{x}}, \hat{\mathbf{d}}$ (37 μs)
 - 4: **plan:** references (21), force references (22), stage table $\pi_{0:N}$ from (12), (13) and $\mathbf{d} = (\hat{\mathbf{d}}_f, \mathbf{0})$; swing targets (14)
 - 5: **NMPC:** $\text{nmpe_set_par}(\pi_{0:N})$, $\text{nmpe_set_ref}(\mathbf{x}^r, \mathbf{f}^r)$; $\text{nmpe_step}(\hat{\mathbf{x}}, \text{shift}) \rightarrow \mathbf{f}_0$ (applied 0.11 μs after $\hat{\mathbf{x}}$)
 - 6: **leg control (5 $\times$ 2 ms):** (15) with $\mathbf{f}_0, s_i, \mathbf{p}_i^{\text{sw}}$; accumulate the interval-mean realised GRF for $\mathbf{u}_{t \rightarrow t+1}$
-

Algorithm 2 Stage-parameter table for the NMPC (per sample)

-
- 1: **for** $k = 0, \dots, N$ **do**
 - 2: $t_k \leftarrow t + kh$; $\mathbf{s}_k \leftarrow$ contact flags (12) at t_k
 - 3: **for each leg** i **do**
 - 4: **if** $s_{k,i} = 1$ and $s_{k-1,i} = 0$ (touchdown inside the horizon) **then**
 - 5: $\mathbf{r}_{k,i} \leftarrow$ foothold (13) for $t_{\text{td}} = t_k$
 - 6: **else**
 - 7: $\mathbf{r}_{k,i} \leftarrow \mathbf{r}_{k-1,i}$ (anchor (19) for $k = 0$)
 - 8: **end if**
 - 9: **end for**
 - 10: $\pi_k \leftarrow (\mathbf{r}_k, \mathbf{s}_k, \hat{\mathbf{d}}_f, \mathbf{0})$; $\mathbf{f}_k^r \leftarrow$ (22); $\mathbf{x}_k^r \leftarrow$ (21)
 - 11: **end for**
-

D. PARAMETERS

TABLE 9. All parameters of the framework.

mass, inertia	$m = 44.965 \text{ kg}$, $\mathbf{I} = \text{diag}(1.998, 3.881, 4.031) \text{ kg m}^2$
friction (NMPC / plant)	$\mu = 0.6 / 0.8$; $f_{\text{max}} = 400 \text{ N}$
gait	$T_g = 0.7 \text{ s}$, $\beta = 0.5$, offsets $(0, \frac{1}{2}, \frac{1}{2}, 0)$
swing	$h_{\text{sw}} = 0.06 \text{ m}$, $z_{\text{land}} = -0.015 \text{ m}$, $K_p = 900 \text{ N/m}$, $K_d = 30 \text{ Ns/m}$
footholds	$k_r = 0.12 \text{ s}$, $\ell_i = (\pm 0.451, \pm 0.301) \text{ m}$, $z^{\text{cmd}} = 0.45 \text{ m}$
torque limits	$\pm 80 \text{ Nm}$
NMPC	$N = 50$, $h = 10 \text{ ms}$, $\mathbf{Q} = (5, 5, 300, 100, 100, 50, 20, 20, 20, 1, 1, 2)$, $\mathbf{R} = 10^{-4} \mathbf{I}$, $\mathbf{Q}_N = 2\mathbf{Q}$, $\Delta \mathbf{x}_{\text{max}} = 3$, $\Delta \mathbf{u}_{\text{max}} = 600 \text{ N}$, $\vartheta_{\text{max}} = 0.35 \text{ rad}$
NMHE	$M = 20$, $h = 10 \text{ ms}$, $\sigma_\theta = 0.01 \text{ rad}$, $\sigma_\omega = 0.05 \text{ rad/s}$, $\sigma_v = 0.03 \text{ m/s}$, $\sigma_p = 0.01 \text{ m}$, $\sigma_{d_f} = 800 \text{ N/s}$, $\sigma_{d_\tau} = 200 \text{ Nm/s}$
NMHE bounds	$ d_f \leq 800 \text{ N}$, $ d_\tau \leq 300 \text{ Nm}$, $ \phi , \theta \leq 0.6 \text{ rad}$, $ d \leq 2000$
gating	$t_{\text{settle}} = 40 \text{ ms}$, contact threshold 8 N (node) / 4 N (interval)
sensor noise	5 mrad, 0.02 rad/s, 0.05 rad/s (joints), 3 mm (FK), 2 N (GRF)
plant	MuJoCo 3.13, $\Delta t = 2 \text{ ms}$, foot solimp 0.9 0.95 0.001, solref 0.005 1

E. REPRODUCIBILITY

The released repository [3] contains `model/anymal_srbd.py` (the symbolic source, which emits the C model header), `sim/robot.py` (plant, sensors, leg controller), `sim/gait.py` (schedule, footholds, swing trajectories, references), `sim/framework.py` (the loop), `sim/solvers.py` (the ctypes bindings and the derivative checks), `sim/run_walk.py` (all experiments), `sim/make_figs.py` (the figures), `sim/render_hd.py` (the animation), `interface/solver_abi.h` (the documented C interface the framework binds to, with notes on substituting another RTI package), `results/` with every logged trajectory and summary used in this report, and `media/` with the animation. The ANYmal-C assets are the MuJoCo Menagerie’s [17] (BSD-3); `sim/anymal_c_torque.xml` is the modified model. The two solver libraries are not redistributed: they are ongoing unpublished research, and the repository documents only the interface they present.

REFERENCES

- [1] S. Anil, “A constrained real-time-iteration NMPC solver and its moving-horizon-estimation variant,” manuscript in preparation, 2026. (Algorithms and source confidential pending publication.)
- [2] S. Anil, “SLQP-MPC for a linear inverted pendulum: exploiting the horizon structure of the linear-programming subproblem,” master’s thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2026.
- [3] S. Anil, “Unified SRBD NMPC/NMHE framework for ANYmal-C,” source repository, 2026.
- [4] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control,” in *Proc. IEEE/RSJ IROS*, 2018, pp. 1–9.
- [5] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot,” in *Proc. IEEE/RSJ IROS*, 2018.
- [6] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” arXiv:1909.06586, 2019.
- [7] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An efficient optimal planning and control framework for quadrupedal locomotion,” in *Proc. IEEE ICRA*, 2017, pp. 93–100.
- [8] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, “Feedback MPC for torque-controlled legged robots,” in *Proc. IEEE/RSJ IROS*, 2019, pp. 4730–4737.
- [9] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, “State estimation for legged robots—consistent fusion of leg kinematics and IMU,” in *Robotics: Science and Systems VIII*, 2013.
- [10] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, “Pronto: A multi-sensor state estimator for legged robots in real-world scenarios,” *Front. Robot. AI*, vol. 7, art. 68, 2020.
- [11] D. Wisth, M. Camurri, and M. Fallon, “Robust legged robot state estimation using factor graph optimization,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4507–4514, 2019.
- [12] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model-predictive control,” *AIChE J.*, vol. 49, no. 2, pp. 426–437, 2003.
- [13] M. Diehl, H. G. Bock, and J. P. Schlöder, “A real-time iteration scheme for nonlinear optimization in optimal feedback control,” *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [14] M. H. Raibert, *Legged Robots That Balance*. MIT Press, 1986.
- [15] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, “acados—a modular open-source framework for fast embedded optimal control,” *Math. Program. Comput.*, vol. 14, pp. 147–183, 2022.
- [16] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *Proc. IEEE/RSJ IROS*, 2012, pp. 5026–5033.
- [17] K. Zakka *et al.*, “MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo,” Google DeepMind, 2022.