

Closed-Loop Field Calibration of Quadraped Velocity Interfaces

Wei Xu • Arjun Mehta • Ethan Cole

Autonomous Robotics Center, Carnegie Mellon University, Pittsburgh, PA, USA

PUBLISHED

September 5, 2026

REGISTERED

September 9, 2026

RECORD TYPE

Preprint

ABSTRACT

Quadraped navigation stacks usually command a locomotion controller through a velocity interface whose command-to-motion response must be calibrated before the planner's predictions can be trusted. Open-loop active calibration, such as planner-conditioned probing with GAUGE, yields an accurate model at calibration time, but the response drifts as payload, battery state, temperature, and wear change during deployment. This paper closes the calibration loop. The calibrated interface model is embedded in the local planner, and the prediction errors it incurs during normal operation are treated as passive observations that update a Gaussian posterior over the interface parameters. Active probes are issued only when the planner-relevant predictive uncertainty crosses a threshold, and a residual-decomposition damping rule separates genuine parameter change from structural model mismatch so that the estimator does not chatter when the operating point leaves the model class. The analysis gives conditions under which the posterior covariance and the parameter error remain bounded under bounded mismatch. The estimator was evaluated on an ANYmal B during a three-day field campaign with 18 h of operation, payloads of 0, 6, and 12 kg, battery levels from 100 to 30 percent, and 2.1 M observations logged at 50 Hz. At the same probe frequency of 0.4 probes per hour, navigation success was 90.1 percent, compared with 84.1 percent for recalibration every 4 h and 80.5 percent for a static day-0 model, while the improvement in interface drift was small (2.9 versus 3.0 and 3.4 percent per day).

INDEX TERMS quadraped robots, velocity interface calibration, closed-loop system identification, Bayesian parameter estimation, active probing, model mismatch, field robotics

KAGG NOTE Routine navigation is itself a calibration experiment: when its prediction errors are gated, damped against structural mismatch, and supplemented by probes only where the planner is uncertain, the velocity-interface model stays useful across payload and battery changes without a fixed recalibration schedule.

01 Introduction

Navigation stacks for quadrapedal robots such as ANYmal [2] are almost always layered. A perception and mapping layer builds a local terrain representation, a planner selects a body-frame velocity command, and a locomotion controller, increasingly a learned policy [3, 4, 5, 6], converts that command into joint torques. The *velocity interface* between planner and controller is where the planner's internal model of the robot meets the physical robot. Realized and commanded body twists differ in systematic ways: steady-state gains below unity, coupling between lateral and yaw channels, offsets, a response lag of several control periods, and saturation at the platform's speed caps. Any planner that rolls out candidate commands, whether a sampling-based model predictive controller [29] or a dynamic-window method [30], implicitly assumes a model of this response. When the model is wrong, predicted and executed trajectories separate, and the clearance margins that the planner believes it is keeping are no longer the margins the robot actually keeps.

GAUGE [1] addresses the identification of this response directly. It treats the velocity controller as an opaque interface, issues probe commands conditioned on the distribution of commands that the planner will actually use, and fits a calibrated command-to-motion model online during the probing session. The resulting model is accurate at the time it is fitted. In field de-

ployments, however, the interface is not stationary. A payload change shifts the mass and inertia that the controller must regulate; battery depletion reduces available actuator power and can cause the platform firmware to lower its speed cap; actuator temperature and mechanical wear alter tracking behavior over hours and days. The calibration produced by a probing session is therefore a snapshot, and the usual remedy, recalibrating on a fixed schedule, spends probing time when nothing has changed and reacts late when a change occurs between sessions.

The navigation stack already produces a continuous stream of potential calibration data: every command the planner issues and every twist that the state estimator subsequently measures. The difficulty is that these *passive* data are poor experiments. They are weakly exciting, because a robot on a mission mostly walks forward near cruise speed; they are corrupted by slips, contact events, and estimator transients; and some of the prediction errors they reveal are not caused by changed parameters at all but by structural mismatch, for example a shift of the speed cap that moves the operating point outside the region where the model class is valid. Feeding such data directly into a recursive estimator produces either slow adaptation, when the estimator is tuned conservatively, or parameter chattering, when it is tuned aggressively. The gap addressed in this paper is how to use passive operational data safely for interface calibration, and how to decide when they must be supplemented by active probes.

The proposed approach closes the calibration loop. The GAUGE-style interface model is embedded in the local planner, and its one-step prediction errors during normal operation are treated as gated observations that update a Gaussian posterior over the interface parameters. Active probes are issued only when the predictive uncertainty evaluated on the planner’s own upcoming command set crosses a threshold. A damping rule decomposes recent innovations into a part that a parameter update can explain and an unexplained remainder; the former triggers faster adaptation, while the latter inflates the effective measurement noise so that structural mismatch does not drive the parameters back and forth. The contributions are as follows.

1. A closed-loop calibration architecture in which a linear-in-parameters velocity-interface model is shared by a sampling-based local planner and a random-walk Kalman posterior, with gated passive observations drawn from normal navigation (Sections 3 and 4).
2. An uncertainty-triggered probing rule that measures relative predictive uncertainty on the planner’s upcoming command set and selects probes from a GAUGE-style library by planner-weighted information gain, with a covariance back-stop and a rate limit.
3. A model-mismatch damping rule based on a windowed decomposition of normalized innovations into explained and unexplained components.
4. An analysis that gives sufficient conditions for bounded posterior covariance and ultimately bounded parameter error under bounded mismatch, and shows that damping scales down mismatch-driven parameter excursions.
5. A three-day field validation on an ANYmal B with rotating payloads and battery depletion, compared against open-loop GAUGE probing, recalibration every 4 h, and a static day-0 model, together with an ablation study and an analysis of recorded failure cases (Section 5).

02 Related Work

2.1 Locomotion controllers and the velocity interface

ANYmal [2] established a torque-controllable quadruped with series-elastic actuators whose locomotion controller accepts body-frame velocity commands, and this interface has become the standard contract between navigation and locomotion on the platform. Hwangbo et al. [3] trained a velocity-tracking policy in simulation using a learned actuator network and transferred it to ANYmal; Lee et al. [4] used a teacher-student scheme to obtain a proprioceptive policy that follows velocity commands over rough natural terrain; and Miki et al. [5] added exteroception through a belief encoder that fuses height samples with proprioception. Rudin et al. [6] showed that massively parallel simulation trains such command-tracking policies within minutes. Margolis et al. pushed the commanded speed envelope on a smaller quadruped [7] and exposed additional behavior parameters, such as gait frequency and body height, alongside the velocity command [8]. Hoeller et al. [9] combined a navigation module with a set of learned skills, again communicating through commanded motions. A recent survey [10] documents how widespread this command interface has become. All of these controllers are trained with a command-tracking reward, so their response is close to identity inside the training distribu-

tion. The present work does not modify the controller. It starts from the observation that deployment conditions such as payload, battery state, and wear lie partly outside that distribution, and that the residual response is what the planner experiences.

2.2 Calibration of command-to-motion models

GAUGE [1] is the direct predecessor of this work. It models the velocity controller as an opaque system and identifies a compact command-to-motion model, including per-axis gains, cross-coupling, offsets, lag, and saturation, from short probe maneuvers. Its key idea is planner conditioning: probes are drawn from, and weighted by, the distribution of commands that the downstream planner is expected to issue, so that identification effort is spent where prediction accuracy matters for planning rather than uniformly over the command space. Calibration proceeds online during a probing session, and the fitted model is then handed to the planner. The present work adopts the GAUGE model class and its probe library but changes what happens after the session. Instead of freezing the model, it keeps the posterior alive during navigation, assimilates passive prediction errors, and reuses planner conditioning in a different role, namely as the statistic that decides *when* a further probe is needed. In this sense the method is a closed-loop extension of GAUGE rather than a replacement for it.

Outside legged robotics, Seegmiller et al. [11] identified wheeled-vehicle models by minimizing integrated prediction error over trajectories logged during normal driving, which is the closest precedent for passive calibration from operational data. That work assumes a wheeled vehicle with a physically structured slip model and does not address when to excite the system actively or how to react to structural mismatch. Ostafew et al. [12] learned residual path-tracking dynamics with Gaussian processes during repeated traversals and used them in a robust nonlinear MPC; the learning benefits from repetition along a taught path, whereas a calibration posterior over a small parametric model generalizes across paths. The broader system-identification literature [13] and, specifically, the identification-for-control viewpoint surveyed by Gevers [14] argue that experiments should be designed to be informative in the directions that matter for the eventual controller. The planner-weighted probe selection used here is a direct instance of that principle.

2.3 Online adaptation in legged locomotion

A large body of work adapts the locomotion controller itself. Dynamics randomization [16] and careful actuator modeling [17] make policies robust to parameter variation. RMA [15] trains an adaptation module that infers a latent encoding of environment and body parameters from proprioceptive history and conditions the base policy on it. Nagabandi et al. [18] meta-learn a dynamics model that adapts online from recent transitions. Model-based approaches such as adaptive force-based control [19] compensate unknown payloads within the whole-body controller. These methods reduce, but do not remove, the change in the closed-loop command-to-motion response that a planner sees: an adaptive controller with a heavier payload may still accelerate more slowly or saturate earlier. The method proposed here is complementary. It sits above the controller, treats it as opaque, and is therefore applicable whether or not the controller adapts internally.

2.4 Recursive estimation, consistency, and robustness

The posterior update is a Kalman filter [20] on a random-walk parameter model. Mehra [21] showed how innovation statistics can be used to estimate noise covariances online; the damping rule proposed here is an innovation-based noise adaptation in this tradition, but restricted to the part of the innovation that no parameter update can explain. Normalized innovation squared tests, as described by Bar-Shalom et al. [23], provide the consistency reference against which damping is triggered. Huber's M-estimation [24] supplies the per-sample down-weighting used to suppress slips. Kulhavý and Zarrop [25] analyzed forgetting as a mechanism for tracking time-varying parameters and its failure mode, covariance windup in unexcited directions, which the probe backstop addresses. Reif et al. [22] established stochastic boundedness of the extended Kalman filter under bounded covariance and observability conditions, and Ioannou and Sun [26] gave the robust adaptive control argument that parameter estimators can drift under unmodeled dynamics unless the adaptation is modified. The analysis in Section 4 follows both lines.

2.5 Triggered and active information acquisition

Trimpe and D'Andrea [27] proposed event-based state estimation in which transmissions are triggered when the estimation variance exceeds a threshold, which makes the trigger predictable and analyzable. The probe trigger used here has the same structure, but the triggered action is an informative experiment rather than a communication, and the variance is evaluated in the planner's command space rather than in the state space. Dual control, surveyed by Mesbah [28], optimizes excitation and performance jointly; the present method instead separates them, running the navigation task undisturbed until the trigger fires and then executing a short probe, which is simpler to certify and to schedule around mission constraints.

2.6 Field navigation for legged robots

Field-deployed legged navigation systems such as ArtPlanner [33], the CERBERUS system [36], and NeBula [37] combine GPU elevation mapping [34], legged state estimation [38, 39], and LiDAR-inertial odometry [40] with sampling- or graph-based planners. These systems typically assume an idealized or fixed-gain model of command following and compensate for its errors with conservative margins and replanning. Wild Visual Navigation [35] adapts a traversability estimator online from the robot's own experience, which is the perception-side analogue of the closed-loop calibration proposed here. Cautious MPC [31] propagates learned model uncertainty into constraint tightening, and safe model-based reinforcement learning [32] restricts exploration to regions with certified stability; both motivate the use of the calibration posterior to inflate collision margins in the local planner.

03 Problem Formulation

3.1 Velocity interface

Time is discretized with the planner period $\Delta t = 20$ ms, matching the 50 Hz rate at which observations were logged. At step t the planner issues a body-frame command $u_t = [v_{x,t}^c, v_{y,t}^c, \omega_t^c]^\top \in \mathcal{U} \subset \mathbb{R}^3$, comprising forward, lateral, and yaw-rate components. The state estimator returns the realized body twist $y_t \in \mathbb{R}^3$ in the same frame. The interface is the unknown map

$$y_{t+1} = F(y_t, u_{t-d}; \xi_t) + v_t, \quad (1)$$

where d is an integer command latency in planner periods, ξ_t collects unobserved conditions such as payload, battery state, actuator temperature, terrain, and wear, and v_t is measurement and gait-induced noise. The interface is *opaque*: the internals of F , including the controller's own state, are not accessible.

3.2 Model class

Following GAUGE [1], the interface is approximated by a first-order linear model with input saturation,

$$y_{t+1} = A y_t + B s(u_{t-d}) + c, \quad (2)$$

where $A = \text{diag}(a_1, a_2, a_3)$ holds per-axis lag coefficients, $B \in \mathbb{R}^{3 \times 3}$ holds gains and cross-coupling terms, $c \in \mathbb{R}^3$ is an offset, and $s(\cdot)$ clips each command component to the platform's speed cap $\bar{u}$. The steady-state gain is $G = (I - A)^{-1}B$. Writing row i as $y_{i,t+1} = \phi_t^{(i)\top} \theta_i$ with $\phi_t^{(i)} = [y_{i,t}, s(u_{t-d})^\top, 1]^\top \in \mathbb{R}^5$ and $\theta_i = [a_i, b_i^\top, c_i]^\top$, the model becomes linear in the stacked parameter vector $\theta = [\theta_1^\top, \theta_2^\top, \theta_3^\top]^\top \in \mathbb{R}^{15}$,

$$y_{t+1} = \Phi_t \theta + v_t, \quad \Phi_t = \text{blkdiag}(\phi_t^{(1)\top}, \phi_t^{(2)\top}, \phi_t^{(3)\top}) \in \mathbb{R}^{3 \times 15}. \quad (3)$$

The linear-in-parameters form is chosen deliberately. It admits an exact Gaussian posterior update at negligible cost, it is the model class whose probes GAUGE already provides, and its parameters have a direct physical reading that operators can inspect. Effects outside this class, such as speed-dependent gains or a changed speed cap, are not modeled and appear as mismatch.

3.3 Drift and mismatch

The true interface is described relative to the model class by a time-varying best-fit parameter θ_t^* and a mismatch term δ_t ,

$$y_{t+1} = \Phi_t \theta_t^* + \delta_t + v_t, \quad \theta_{t+1}^* = \theta_t^* + w_t, \quad (4)$$

where w_t models slow drift with $\mathbb{E}[w_t w_t^\top] \preceq Q$ and δ_t is the part of the response that no parameter in the class can reproduce. Abrupt events such as a payload swap appear as a large w_t at one step; operating-point changes such as a lowered speed cap appear as a nonzero δ_t .

3.4 Objectives and metrics

The calibration system maintains a posterior $p(\theta_t^* \mid \mathcal{D}_t) \approx \mathcal{N}(\hat{\theta}_t, P_t)$ given the data $\mathcal{D}_t$ received so far. Its goals are (i) to keep the planner-relevant predictive uncertainty, defined in Section 4, low; (ii) to do so with as few active probes as possible; and (iii) to remain stable, meaning bounded covariance, bounded parameter error, and bounded parameter excursions, when $\delta_t \neq 0$. These goals are evaluated through four campaign-level metrics: the *interface drift*, defined as the average daily growth of the relative steady-state prediction error of the deployed model, evaluated on probe segments before they are assimilated and expressed in percent per day; the *probe frequency* in probes per hour of operation; the *navigation success* rate, the percentage of navigation missions that reach the goal without collision, fall, or operator intervention; and the 95th percentile over the campaign of the relative posterior uncertainty σ_{rel} defined in (12).

04 Method

4.1 Architecture

The navigation stack has five components. LiDAR-inertial and leg odometry are fused to provide body pose and twist; a GPU elevation map [34] provides the local terrain; a global graph planner provides waypoints; a sampling-based local planner [29] runs at 50 Hz and issues velocity commands; and the platform's locomotion controller, treated as opaque, tracks them. The calibration estimator runs at the same 50 Hz rate in parallel with the local planner and shares the interface model with it. A probe supervisor monitors the trigger statistic and, when a probe is warranted and safe, briefly pauses mission progress to execute it. The per-step computation is summarized in Algorithm 1 at the end of this section.

4.2 Posterior update

The random-walk model (4) with $\delta_t = 0$ yields the standard Kalman recursion. The prediction step is $\hat{\theta}_{t|t-1} = \hat{\theta}_{t-1}$ and $P_{t|t-1} = P_{t-1} + Q$. Given an accepted observation, the innovation and its covariance are

$$\begin{aligned} r_t &= y_{t+1} - \Phi_t \hat{\theta}_{t|t-1}, \\ S_t &= \Phi_t P_{t|t-1} \Phi_t^\top + R_{\text{eff},t}, \end{aligned} \quad (5)$$

and the update is

$$\begin{aligned} K_t &= P_{t|t-1} \Phi_t^\top S_t^{-1}, \quad \hat{\theta}_t = \hat{\theta}_{t|t-1} + K_t r_t, \\ P_t &= (I - K_t \Phi_t) P_{t|t-1}, \end{aligned} \quad (6)$$

implemented in Joseph form for numerical symmetry. The effective measurement covariance

$$R_{\text{eff},t} = \frac{\lambda_t}{\rho_t} R \quad (7)$$

combines a nominal covariance R , identified from stationary walking during the initial GAUGE session, with a per-sample robust weight $\rho_t \in (0, 1]$ and the mismatch damping factor $\lambda_t \geq 1$ described below. The diagonal process covariance Q sets the assumed drift rate per step; it is the main tuning parameter trading tracking speed against noise.

4.3 Passive observation channel

Every planner step produces a candidate observation (Φ_t, y_{t+1}) , but only a subset is informative about the interface. A sample is accepted, $g_t = 1$, only if the state estimator reports a healthy solution, the contact-consistency check of the leg odometry [38] reports no slip, the locomotion controller is in its walking mode rather than a stand or gait transition, and no probe is in progress. Rejected samples trigger only the prediction step. The latency d is taken from the initial GAUGE session and is not re-estimated online. Accepted samples are further weighted with a Huber function [24] of the normalized innovation,

$$\rho_t = \min \left(1, \frac{k_H}{\|S_t^{-1/2} r_t\|} \right), \quad (8)$$

so that isolated large residuals, for example from an undetected foot slip or a brief external contact, are down-weighted rather than rejected. The measured twist y_t also appears inside Φ_t , which makes the regression an errors-in-variables problem; at 50 Hz the resulting bias is small relative to gait-induced noise and is absorbed into R and the mismatch channel.

4.4 Model-mismatch damping

A large innovation can mean two different things. If the parameters have changed, a parameter update explains it and adaptation should be fast. If the response has left the model class, no parameter update explains it, and adapting would only move $\hat{\theta}$ to wherever the current operating point happens to pull it. The damping rule distinguishes the two cases on a sliding window $\mathcal{W}_t$ of the last W accepted samples. It first computes the window-optimal correction

$$\delta\theta_t^* = \arg \min_{\delta\theta} \sum_{s \in \mathcal{W}_t} \|r_s - \Phi_s \delta\theta\|_{S_s^{-1}}^2 + \|\delta\theta\|_{P_t^{-1}}^2, \quad (9)$$

a 15-dimensional regularized least-squares problem, and then the total and unexplained normalized innovation statistics

$$\begin{aligned} \varepsilon_t^{\text{tot}} &= \frac{1}{mW} \sum_{s \in \mathcal{W}_t} r_s^\top S_s^{-1} r_s, \\ \varepsilon_t^{\text{un}} &= \frac{1}{mW} \sum_{s \in \mathcal{W}_t} \|r_s - \Phi_s \delta\theta_t^*\|_{S_s^{-1}}^2, \end{aligned} \quad (10)$$

with $m = 3$. For a consistent filter both statistics are close to one [23]. The damping factor follows the unexplained part through a first-order low-pass filter,

$$\lambda_{t+1} = (1 - \beta) \lambda_t + \beta \min \left(\bar{\lambda}, \max \left(1, \frac{\varepsilon_t^{\text{un}}}{1 + \kappa} \right) \right), \quad (11)$$

where $\kappa > 0$ is a tolerance, $\beta \in (0, 1)$ a smoothing constant, and $\bar{\lambda}$ an upper bound that keeps the filter from ignoring data entirely. The explained part triggers a covariance boost: if $\varepsilon_t^{\text{tot}} - \varepsilon_t^{\text{un}} > \kappa_J$, then $P_t \leftarrow P_t + Q_J$ with $Q_J \succ Q$. The boost lets the filter follow an abrupt change such as a payload swap; it also raises the predictive uncertainty and therefore makes a probe more likely, which matters when the passive data do not excite the changed parameters. The design follows the innovation-based noise adaptation of Mehra [21], but applies it only to the component that the model class cannot absorb.

4.5 Uncertainty-triggered active probes

The trigger statistic is evaluated in the planner's command space. Let $\mathcal{U}_t$ be the set of commands in the local planner's current nominal sequence together with those of its highest-weight sampled rollouts over the planning horizon. For a command u , the predicted steady-state twist is $\bar{y}(u; \theta) = (I - A)^{-1} (B s(u) + c)$ and its linearized covariance is $\Sigma_{\bar{y}}(u) = J_u P_t J_u^\top$ with $J_u = \partial \bar{y} / \partial \theta$ evaluated at $\hat{\theta}_t$. The relative planner-relevant uncertainty is

$$\sigma_{\text{rel}}(t) = \max_{u \in \mathcal{U}_t} \frac{\sqrt{\text{tr} \Sigma_{\bar{y}}(u)}}{\max \left(\|\bar{y}(u; \hat{\theta}_t)\|, v_0 \right)}, \quad (12)$$

where v_0 prevents division by small speeds. A probe is requested when either $\sigma_{\text{rel}}(t) > \tau$ for a hold time T_h , or the backstop condition $\lambda_{\text{max}}(P_t) > \bar{p}$ holds, and it is granted only if at least T_{min} has passed since the previous probe and the elevation map shows sufficient free space around the robot. The hold time prevents single-step spikes from triggering probes; the backstop guarantees that directions irrelevant to the current command set cannot wind up without bound [25]; and the rate limit bounds probing overhead.

When a probe is granted, the supervisor selects from a GAUGE-style library $\mathcal{P}$ of short step, ramp, and turn-in-place sequences [1] the one that maximizes planner-weighted information gain,

$$\pi^* = \arg \max_{\pi \in \mathcal{P}} \text{tr} (W_t (P_t - P_\pi^+)), \quad (13)$$

$$(P_\pi^+)^{-1} = P_t^{-1} + \sum_{s \in \pi} \Phi_s^\top (\lambda_t R)^{-1} \Phi_s,$$

where the regressors Φ_s along π are predicted with the current mean model and $W_t = \sum_{u \in \mathcal{U}_t} J_u^\top J_u / \max(\|\tilde{y}(u; \hat{\theta}_t)\|, v_0)^2$ weights parameter directions by their effect on the planner's predictions. This is an A-optimal design criterion in the identification-for-control sense [14]. Probe magnitudes are scaled to the range of $\mathcal{U}_t$, so the probe excites the operating region the planner is about to use. Probe samples pass through the same filter as passive samples.

4.6 Integration with the local planner

The local planner is a model predictive path integral controller [29] that samples N_s command sequences over a horizon of H steps and propagates each through the posterior-mean interface model,

$$\hat{y}_{k+1} = \hat{A} \hat{y}_k + \hat{B} s(u_{k-d}) + \hat{c}, \quad x_{k+1} = x_k \oplus \hat{y}_k \Delta t, \quad (14)$$

where $x_k \in SE(2)$ is the planar body pose and $\oplus$ denotes pose composition. Because the optimization acts through the calibrated model, command pre-compensation for gains, coupling, and lag is implicit. The rollout cost combines progress toward the next global waypoint, an elevation-map traversability term, and a collision term evaluated with a footprint inflated by

$$m_k = m_0 + \kappa_m \sigma_{\text{pos},k}, \quad (15)$$

where $\sigma_{\text{pos},k}$ is the standard deviation of the predicted position at step k , obtained by propagating P_t through the sensitivities of (14) along the nominal sequence. This uncertainty-dependent tightening follows cautious MPC [31]: an uncertain interface model makes the planner keep more clearance rather than trusting an optimistic prediction. It also means that calibration quality affects navigation success through two paths, the accuracy of the predicted trajectory and the width of the margins.

4.7 Stability under model mismatch

The analysis concerns the parameter estimator in closed loop with the trigger rule. The following assumptions are made.

- **A1 (bounded regressors).** For accepted samples, $\|\Phi_t\| \leq \bar{\phi}$. This holds because commands are clipped by $s(\cdot)$ and twists are bounded by the platform's speed limits.
- **A2 (bounded mismatch and noise).** $\|\delta_t\| \leq \bar{\delta}$, and v_t is zero-mean with $\mathbb{E}[v_t v_t^\top] \preceq R$, $R \succeq r_{\min} I$.
- **A3 (bounded drift).** $\mathbb{E}[w_t w_t^\top] \preceq Q$ with $Q \succ 0$.
- **A4 (probe excitation).** Every library probe satisfies $\sum_{s \in \pi} \Phi_s^\top R^{-1} \Phi_s \succeq \alpha I$ for some $\alpha > 0$; that is, each probe alone renders all 15 parameters observable.

Lemma 1 (covariance bounds). Under A1, A3, and A4, with $1 \leq \lambda_t \leq \bar{\lambda}$, $\rho_t \in [\rho_{\min}, 1]$, and the trigger rule with

backstop $\bar{p}$ and rate limit $T_{\min}$, there exist $0 < p_{\min} \leq p_{\max} < \infty$ such that $p_{\min} I \preceq P_t \preceq p_{\max} I$ for all t .

Proof sketch. The lower bound follows from the information form: each accepted sample adds at most $\bar{\phi}^2/(r_{\min} \rho_{\min})$ to the information matrix, while every prediction step adds $Q \succ 0$ to the covariance, so P_t cannot collapse. For the upper bound, between probes $\lambda_{\max}(P_t)$ can exceed $\bar{p}$ only for the time needed for a probe to be granted, at most $T_{\min}$ plus the probe duration T_π , during which it grows by at most $\|Q\|(T_{\min} + T_\pi)/\Delta t$ plus the bounded covariance boosts. Each probe, by A4, yields $P^+ \preceq (\bar{\lambda}/\alpha) I$. Combining the two cases gives $p_{\max} = \max(\bar{p}, \bar{\lambda}/\alpha) + \|Q\|(T_{\min} + T_\pi)/\Delta t + n_J \|Q_J\|$, where n_J bounds the number of boosts between probes; n_J is finite because each boost raises $\lambda_{\max}(P_t)$ toward the backstop. $\square$

Proposition 1 (bounded error and bounded excursions).

Let $\hat{\theta}_t = \theta_t^* - \hat{\theta}_t$. Under A1 to A4 and the conclusions of Lemma 1, (a) the mean squared error is ultimately bounded,

$$\limsup_{t \rightarrow \infty} \mathbb{E} \|\hat{\theta}_t\|^2 \leq c_1 \text{tr} Q + c_2 \text{tr} R + c_3 \bar{\delta}^2, \quad (16)$$

with constants c_1, c_2, c_3 that depend only on $p_{\min}, p_{\max}, \bar{\phi}$, and $\bar{\lambda}$; and (b) the per-step parameter excursion driven by mismatch satisfies

$$\|K_t \delta_t\| \leq \frac{p_{\max} \bar{\phi} \bar{\delta}}{\lambda_t r_{\min}}. \quad (17)$$

Proof sketch. The error obeys $\tilde{\theta}_{t+1} = (I - K_t \Phi_t) \tilde{\theta}_t - K_t (\delta_t + v_t) + w_t$. With P_t bounded above and below, $V_t = \tilde{\theta}_t^\top P_t^{-1} \tilde{\theta}_t$ is a stochastic Lyapunov function for the homogeneous part, decreasing in expectation by a factor that depends on $p_{\min}/p_{\max}$, in the manner of Reif et al. [22]. The inputs w_t, v_t , and δ_t enter additively with bounded gains, so a bounded-input bounded-state argument, as in robust adaptive control [26], gives (a). For (b), $\|K_t\| \leq \|P_{t|t-1}\| \|\Phi_t\| \|S_t^{-1}\| \leq p_{\max} \bar{\phi} / (\lambda_t r_{\min})$ because $S_t \succeq R_{\text{eff},t} \succeq \lambda_t r_{\min} I$. $\square$

Two remarks qualify the result. First, damping does not remove the asymptotic bias caused by a persistent mismatch; the term $c_3 \bar{\delta}^2$ remains, because no estimator in the model class can reproduce a response outside it. What damping changes is the *bandwidth* with which $\hat{\theta}$ responds to mismatch: by (17), a mismatch that switches sign as the operating point moves in and out of saturation produces excursions reduced by λ_t , which is the chattering the rule is designed to prevent. Second, damping slows genuine adaptation by the same factor, which is why λ_t tracks only the unexplained innovation and why explained innovations instead trigger the covariance boost. Assumption A4 is the strongest assumption; a probe that excites only a subset of parameters gives the corresponding bounds only on that subspace, with the backstop eventually requesting a complementary probe.

4.8 Algorithm summary and cost

ALGORITHM 1: CLOSED-LOOP CALIBRATION CYCLE (ONE PLANNER PERIOD)

1. Predict: $\hat{\theta} \leftarrow \hat{\theta}, P \leftarrow P + Q$.
2. If the sample passes the gate g_t : compute r_t, S_t, ρ_t from (5) and (8); update with $R_{\text{eff},t}$ from (7); append the sample to $\mathcal{W}_t$.

3. Every W accepted samples: solve (9), compute (10), update λ by (11), and apply the covariance boost if the explained part exceeds κ_J .
4. Evaluate $\sigma_{\text{rel}}(t)$ by (12) on the planner's current command set; if the trigger or backstop holds and the probe is permitted, select π^* by (13) and execute it.
5. Pass $(\hat{\theta}, P)$ to the local planner for rollouts (14) and margins (15).

The filter state is 15-dimensional with a 3-dimensional innovation, so each update costs a few microseconds on an embedded CPU, and the windowed problem (9) is a single 15-by-15 linear solve. The computational load is negligible compared with the planner's rollouts, which reuse the same model evaluations.

05 Experiments

5.1 Platform and field campaign

All experiments used an ANYmal B quadruped carrying a rotating multi-beam LiDAR with a nominal range of about 100 m, an inertial measurement unit, and onboard computers for perception, planning, and calibration. The locomotion controller supplied with the platform was used without modification and treated as opaque. The campaign lasted three days and comprised 18 h of operation in three environments: an indoor laboratory, outdoor campus paths, and a stair course. Payloads of 0, 6, and 12 kg were rotated daily, and the battery level ranged from 100% down to 30%. In total, 2.1 M command-response observations were logged at 50 Hz, which corresponds to roughly 11.7 h of logged locomotion within the 18 h of operation. Each configuration was initialized with the same day-0 GAUGE calibration.

REPRODUCIBILITY NOTE

All configurations share the interface model class (2), the probe library, the local planner, and the day-0 calibration; they differ only in how and when the posterior is updated. All reported metrics are campaign-level aggregates from the run-server logs.

5.2 Compared configurations

Four configurations were compared. *Open-loop GAUGE probes only* executes GAUGE probes during operation and refits the model from probe data, but ignores passive prediction errors. *Recalibrate every 4 h* runs a GAUGE calibration session on a fixed schedule, the conventional field practice. *Static model* keeps the day-0 calibration throughout; probes are still executed and logged so that its drift can be evaluated on the same probe segments, but they are not used to update the model. *Ours* is the full method of Section 4. As Table 1 shows, all four configurations ended the campaign with the same probe frequency of 0.4 probes per hour, roughly seven probes over the 18 h, so the comparison is at an equal probing budget. Three ablations remove one component each: passive observation updates (only triggered probes update the posterior), uncertainty-triggered probes (only passive data update the posterior), and model-mismatch damping ($\lambda_t \equiv 1$).

5.3 Main results

Table 1 summarizes the comparison. The clearest effect is in navigation success. The closed-loop estimator reached 90.1%, which is 6.0 percentage points above recalibration every 4 h (84.1%), 8.5 points above open-loop GAUGE probing (81.6%), and 9.6 points above the static day-0 model (80.5%). Because the probe frequency is identical across configurations, this difference cannot be attributed to a larger probing budget; it reflects how the same number of probes is combined with the information available between them.

The differences in interface drift and in posterior uncertainty are much smaller. Drift was 2.9%/day for the closed-loop estimator, 3.0%/day for both GAUGE-based baselines, and 3.4%/day for the static model; the 95th-percentile relative uncertainty was 0.22, 0.23, and 0.24, respectively. The gap between any updating configuration and the static model (0.4 to 0.5%/day) is larger than the gap between the closed-loop estimator and the scheduled baselines (0.1%/day). The latter is small enough that, without per-segment variability estimates, it should not be read as a meaningful accuracy improvement on its own. The more defensible interpretation is that the closed-loop estimator does not achieve a substantially lower average error than scheduled recalibration, but reaches comparable accuracy without waiting for a scheduled session and, as the failure cases below show, recovers much faster after abrupt changes. Campaign averages of drift and uncertainty dilute such transient episodes, whereas navigation success is sensitive to them, because missions attempted during a poorly calibrated period are the ones that fail.

5.4 Ablation study

Table 2 isolates each component. Removing passive observation updates has the largest effect: navigation success drops by 15.4 points to 74.7%, and drift rises to 3.4%/day, the same value as the static model. Passive data are therefore the main source of calibration information during operation; the triggered probes alone do not keep the model current. That this variant falls below even the static model in navigation success (74.7% versus 80.5%) was not anticipated. One plausible explanation, which the logged aggregates do not isolate, is that without passive updates the posterior covariance grows between probes under the random-walk model, which widens the uncertainty-dependent margins (15) and makes narrow passages and stair approaches harder to plan through. The static configuration does not feed an updated covariance to the planner in this way.

Removing the uncertainty trigger reduces success by 6.3 points to 83.8%, close to the recalibrate-every-4-h baseline, and raises drift to 3.1%/day. Passive data alone thus track slow drift but not the changes they fail to excite, as the payload-swap case below illustrates. Removing damping costs 2.5 points of success (87.6%) and raises drift to 3.2%/day, consistent with the analysis in Section 4: without damping, structural mismatch moves the parameters away from values that are correct within the region where the model class is valid.

5.5 Failure cases

Payload swap. On day 2 the payload was changed to 12 kg. The estimator without uncertainty-triggered probes needed 2.1 h, or 126 min, to adapt, whereas the full method issued 3 probes and re-converged in 11 min, roughly an order of magnitude faster. The mechanism matches the design in Section 4. A

Table 1: Main comparison over the three-day campaign. Drift is the daily growth of the relative steady-state prediction error; uncertainty is the 95th percentile of σ_{rel} .

Method	Interface drift over campaign (%/day)	Probe frequency (per hour)	Navigation success (%)	Posterior uncertainty p95 (relative)
Open-loop GAUGE probes only	3.0	0.4	81.6	0.23
Recalibrate every 4 h	3.0	0.4	84.1	0.23
Static model (day-0 calibration)	3.4	0.4	80.5	0.24
Ours	2.9	0.4	90.1	0.22

Table 2: Ablation of the closed-loop estimator.

Variant	Interface drift over campaign (%/day)	Navigation success (%)
Ours (closed-loop + triggered probes)	2.9	90.1
w/o passive observation updates	3.4	74.7
w/o uncertainty-triggered probes	3.1	83.8
w/o model-mismatch damping	3.2	87.6

heavier payload changes gains and lag on all axes, but normal forward walking excites mainly the forward channel, so the passive data leave lateral and yaw parameters poorly determined. In the full method, the explained part of the innovation triggers the covariance boost, σ_{rel} crosses the threshold, and the probe selection (13) targets the directions the planner needs. These 3 probes account for a substantial share of the roughly seven probes of the whole campaign, which illustrates that triggered probing concentrates effort around changes rather than spreading it uniformly in time.

Low battery and speed cap. At a battery level of 32%, the platform’s speed cap changed. Commands near the previous cap were then clipped by the controller but not by the model’s saturation $s(\cdot)$, a mismatch of exactly the kind (4) attributes to δ_t . Because the robot moved in and out of the saturated regime as the planner varied its commands, the sign of the residual alternated. With damping, the unexplained innovation raised λ_t and the estimator remained calm, with a p95 relative uncertainty of 0.22; without damping it chattered, and the p95 uncertainty reached 0.39, about 1.8 times higher. This episode is the clearest empirical counterpart of Proposition 1(b). It also exposes a limitation: damping prevents harm but does not model the new cap, which would require extending the model class or estimating $\bar{u}$ explicitly.

Scheduled recalibration overhead. In the recalibrate-every-4-h baseline, 14% of the calibration time was spent probing under static conditions in which the model was already converged. This is the expected inefficiency of a schedule that is blind to the posterior. Because the probe frequency in Table 1 is the same for all configurations, this result does not imply fewer probes for the closed-loop estimator over the campaign; rather, it shows that a fixed schedule spends part of its budget where it is not needed, while the triggered rule spends it where the uncertainty is.

5.6 Discussion

Taken together, the results support a specific and limited claim. At an equal probing budget, the closed-loop estimator produced the highest navigation success of all configurations, and the ablation attributes most of that gain to passive observation updates, with smaller contributions from triggered probes and mismatch damping. The failure cases provide mechanistic evidence for the two components whose campaign-level effect

is modest: triggered probes shortened recovery after a payload swap from 2.1 h to 11 min, and damping kept the uncertainty at 0.22 rather than 0.39 when the operating point moved outside the model class. The results do not show a large reduction in average interface drift: the 0.1%/day difference relative to the scheduled baselines is small. The interface itself drifts with payload, battery, and wear regardless of how it is estimated, and an estimator can only track that drift, not prevent it.

06 Limitations

The study has several limitations. First, the evidence comes from a single robot, a single locomotion controller, and one three-day campaign; the results table reports campaign aggregates without per-mission counts or confidence intervals, so small differences, in particular the 0.1%/day drift difference and the 0.01 uncertainty differences, should not be overinterpreted. Second, the model class (2) is first-order, linear, and has a fixed saturation level; the low-battery episode shows that a changed speed cap is handled only by damping, not by modeling, and speed-dependent gains are likewise left to the mismatch channel. Third, the stability analysis relies on Assumption A4, that each probe excites all parameters, and on bounded mismatch; it provides sufficient conditions and qualitative scaling, not tight constants, and it does not cover the coupling between estimator and planner through the margins (15). Fourth, the ablation without passive updates performed worse than the static model, and the explanation offered in Section 5 remains a hypothesis that the logged aggregates do not isolate. Fifth, the trigger threshold, hold time, rate limit, and damping constants were set by the authors for this campaign; their sensitivity was not studied, and the coincidence of equal probe frequencies across configurations means that the trade-off between probing budget and navigation success was not mapped. Finally, probes pause mission progress; in time-critical missions, the cost of that pause would need to be weighed explicitly.

07 Conclusion

This paper presented a closed-loop extension of planner-conditioned active calibration for quadraped velocity interfaces. The calibrated interface model is shared between the local planner and a Kalman posterior; prediction errors during normal navigation serve as gated passive observations; probes are triggered only when the planner-relevant predictive uncertainty is high; and a residual-decomposition damping rule keeps the estimator from chattering under structural model mismatch. The analysis gives conditions for bounded covariance and ultimately bounded parameter error, and shows that damping reduces mismatch-driven parameter excursions in proportion to the damping factor. On an ANYmal B over a three-day campaign with payload and battery changes, the method achieved 90.1% navigation success compared with 84.1% for recalibration every 4 h, at the same probe frequency

Table 3: Recorded failure cases and the corresponding behavior of the compared estimators.

Event	Observed behavior	Interpretation
Day 2, 12 kg payload swap	Passive-only estimator took 2.1 h to adapt; the triggered-probe version issued 3 probes and re-converged in 11 min.	Passive data are weakly exciting; the covariance boost and trigger convert an abrupt change into a short targeted experiment.
Low battery (32%)	The platform's speed cap changed, moving the operating point; the mismatch damped the estimator instead of chattering (p95 uncertainty 0.22 vs 0.39 undamped).	The change lies outside the model class; damping limits mismatch-driven excursions as predicted by (17).
Recalibrate-every-4-h baseline	14% of the calibration time was spent probing in static conditions where the model was already converged.	Fixed schedules spend probing effort independently of need.

of 0.4 per hour, with only a small change in average interface drift. The main open directions are richer model classes that absorb speed-cap and speed-dependent effects, multi-robot and multi-controller validation with per-mission statistics, and a systematic study of the trade-off between probing budget and navigation performance.

References

- [1] T. Zang, H. Huang, S. Cheng, and W. Zhang. GAUGE: Planner-Conditioned Active Calibration of Opaque Quadruped Velocity Interfaces. arXiv:2607.02841, 2026.
- [2] M. Hutter et al.. ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- [3] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. Science Robotics, 4(26), 2019.
- [4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. Science Robotics, 5(47), 2020.
- [5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. Science Robotics, 7(62), 2022.
- [6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. Conference on Robot Learning (CoRL), 2021.
- [7] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid Locomotion via Reinforcement Learning. Robotics: Science and Systems (RSS), 2022.
- [8] G. B. Margolis and P. Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. Conference on Robot Learning (CoRL), 2022.
- [9] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. Science Robotics, 9(88), 2024.
- [10] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv. Learning-Based Legged Locomotion: State of the Art and Future Perspectives. The International Journal of Robotics Research, 2025.
- [11] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly. Vehicle Model Identification by Integrated Prediction Error Minimization. The International Journal of Robotics Research, 32(8), 2013.
- [12] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot. Robust Constrained Learning-Based NMPC Enabling Reliable Mobile Robot Path Tracking. The International Journal of Robotics Research, 35(13), 2016.
- [13] L. Ljung. System Identification: Theory for the User, 2nd ed.. Prentice Hall, 1999.
- [14] M. Gevers. Identification for Control: From the Early Achievements to the Revival of Experiment Design. European Journal of Control, 11(4-5), 2005.
- [15] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. Robotics: Science and Systems (RSS), 2021.
- [16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. IEEE International Conference on Robotics and Automation (ICRA), 2018.
- [17] J. Tan et al.. Sim-to-Real: Learning Agile Locomotion for Quadruped Robots. Robotics: Science and Systems (RSS), 2018.
- [18] A. Nagabandi et al.. Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. International Conference on Learning Representations (ICLR), 2019.
- [19] M. Sombolstan, Y. Chen, and Q. Nguyen. Adaptive Force-Based Control for Legged Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.
- [20] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME, Journal of Basic Engineering, 82(1), 1960.
- [21] R. K. Mehra. On the Identification of Variances and Adaptive Kalman Filtering. IEEE Transactions on Automatic Control, 15(2), 1970.
- [22] K. Reif, S. Günther, E. Yaz, and R. Unbehauen. Stochastic Stability of the Discrete-Time Extended Kalman Filter. IEEE Transactions on Automatic Control, 44(4), 1999.
- [23] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation. Wiley, 2001.
- [24] P. J. Huber. Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics, 35(1), 1964.
- [25] R. Kulhavý and M. B. Zarrop. On a General Concept of Forgetting. International Journal of Control, 58(4), 1993.
- [26] P. A. Ioannou and J. Sun. Robust Adaptive Control. Prentice Hall, 1996.
- [27] S. Trimpe and R. D'Andrea. Event-Based State Estimation with Variance-Based Triggering. IEEE Transactions on Automatic Control, 59(12), 2014.
- [28] A. Mesbah. Stochastic Model Predictive Control with Active Uncertainty Learning: A Survey on Dual Control. Annual Reviews in Control, 45, 2018.
- [29] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving. IEEE Transactions on Robotics, 34(6), 2018.
- [30] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. IEEE Robotics and Automation Magazine, 4(1), 1997.
- [31] L. Hewing, J. Kabzan, and M. N. Zeilinger. Cautious Model Predictive Control Using Gaussian Process Regression. IEEE Transactions on Control Systems Technology, 28(6), 2020.
- [32] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. Safe Model-Based Reinforcement Learning with Stability Guarantees. Advances in Neural Information Processing Systems (NeurIPS), 2017.

- [33] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. *Field Robotics*, 3, 2023.
- [34] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation Mapping for Locomotion and Navigation Using GPU. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [35] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. *Robotics: Science and Systems (RSS)*, 2023.
- [36] M. Tranzatto et al.. CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge. *Field Robotics*, 2, 2022.
- [37] A. Agha et al.. NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge. *arXiv:2103.11470*, 2021.
- [38] 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. *Robotics: Science and Systems (RSS)*, 2012.
- [39] D. Wisth, M. Camurri, and M. Fallon. VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots. *IEEE Transactions on Robotics*, 39(1), 2023.
- [40] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics*, 38(4), 2022.