

Probing Opaque Velocity Interfaces on Production Quadruped Platforms: A Field Study

Andrew Foster • Lena Vasquez • Rajiv Menon

Advanced Robotics Laboratory, Boston Dynamics, Waltham, MA, USA

PUBLISHED

August 18, 2026

REGISTERED

August 21, 2026

RECORD TYPE

Preprint

ABSTRACT

Production quadruped platforms expose velocity command interfaces whose internals (gait managers, safety layers, terrain modes) are undocumented, yet navigation planners routinely assume that commanded velocities are tracked instantly and uniformly. This paper reports a 14-day field characterization of three commercial quadruped platforms, two 70-class research units and one 45-class work unit, across laboratory, campus-path, and light outdoor terrain using GAUGE-style planner-conditioned active calibration. Each platform received 60 probe sessions (20 per terrain mode) logged at 100 Hz, with 0, 5, and 10 kg payloads rotated across sessions. We report command fidelity, 95th-percentile phase lag, saturation-envelope error, and mode-switch transients. On the primary unit with the full probe suite, the command fidelity RMS error is 9.7 percent with a p95 phase lag of 71 ms, compared with 11.1 percent and 79 ms on Platform B and 10.6 percent and 81 ms on Platform C. Restricting probes to the default mode raises the error to 10.5 percent, and removing disturbance rejection raises it to 10.7 percent. Payload is the dominant degradation factor: from 0 to 10 kg, the fidelity error rises from 8.9 to 12.6 percent and the phase lag from 64 to 88 ms. Probes also exposed three behaviors that break naive planning assumptions: an undocumented 3.2 m/s speed limiter in an outdoor mode, a 0.7 s gait-manager hysteresis that produced 0.4 m overshoot, and an unannounced careful mode under a 10 kg payload that halves the usable envelope. We derive interface assumptions that planners for such platforms should state explicitly.

INDEX TERMS legged robots, quadruped locomotion, velocity interfaces, active calibration, system identification, field robotics, motion planning

KAGG NOTE A velocity interface becomes a usable planning contract only when its envelope, latency, and mode behavior are measured under the commands, terrain modes, and payloads that the planner will actually use.

01 Introduction

Commercial quadrupeds have become routine field platforms for inspection, mapping, and autonomy research. Almost all of them expose the same narrow interface to higher-level software: a body-frame velocity command $u = (v_x, v_y, \omega_z)$ that is consumed by a proprietary locomotion stack. Behind that interface sit components that a planner never sees directly: a gait manager that selects and sequences stepping patterns, terrain modes that retune the controller for stairs, slippery ground, or outdoor surfaces, and safety layers that clip, delay, or reshape commands. Learned locomotion controllers deployed on such platforms [1, 2, 3] add further behavior that is shaped by training rather than by any published specification.

Navigation planners, in contrast, usually treat the velocity interface as transparent. Sampling-based and model-predictive local planners [4] roll out candidate commands through a unicycle or holonomic model with a box-shaped velocity limit copied from the datasheet, and learned navigation policies [5, 6] emit velocity or waypoint commands under the implicit assumption that the platform executes them faithfully. When the true interface adds latency, saturates earlier than documented, or changes its behavior after a mode switch, these assumptions fail silently: the planner’s predicted trajectory and the executed trajectory diverge exactly in the conditions (high speed, heavy payload, mode transitions) where margins matter most.

GAUGE [7] recently proposed treating the quadruped velocity controller as an opaque interface and calibrating it actively: it issues probe commands drawn from the region of command space that a given planner actually uses and fits a calibrated command-to-motion model online. GAUGE establishes the method; what is missing is evidence about what such calibration reveals on production platforms in the field, across different vendors, firmware versions, terrain modes, and payloads, and about which of the revealed behaviors actually invalidate the assumptions that planners make. This paper addresses that gap with a field study.

Over 14 days we characterized three commercial quadruped platforms (two 70-class research units and one 45-class work unit) on laboratory floor, campus paths, and light outdoor terrain, using GAUGE-style planner-conditioned probes, 60 sessions per platform, and payloads of 0, 5, and 10 kg. The contributions are as follows.

1. A field protocol for GAUGE-style planner-conditioned active calibration on production quadrupeds, extended with explicit terrain-mode probing, gait-disturbance rejection, residual-based detection of undocumented behaviors, and probe-safety rules for shared public spaces (Section 4).
2. A comparative report of command fidelity, p95 phase lag, saturation-envelope error, and mode-switch transients

across three platforms, together with an ablation of the probe suite and a payload sweep (Section 5).

3. A documented set of three interface behaviors that break naive planning assumptions: an undocumented speed limiter, a gait-manager switching hysteresis, and an unannounced payload-triggered careful mode (Section 6).
4. A list of interface assumptions that any planner for these platforms should make explicit, each tied to the measurement that motivates it, plus a reproducibility note with probe parameters and a data availability statement (Sections 7 and 4.6).

02 Related Work

2.1 Active calibration of opaque velocity interfaces

GAUGE [7] is the method on which this study is built. It models the vendor locomotion controller as a black box between the planner’s velocity command and the measured body twist, selects probe commands conditioned on the distribution of commands that a specific planner issues, and fits a calibrated response model online so that the planner can roll out trajectories through the calibrated model instead of an idealized one. The present work does not propose a new calibration algorithm. It applies a GAUGE-style procedure to three production platforms under conditions GAUGE does not focus on (vendor terrain modes, rotated payloads, and public field sites) and reports what the calibrated models and their residuals reveal. Two procedural differences matter for interpreting the results: we probe every vendor terrain mode explicitly rather than only the default mode, and we add a gait-disturbance rejection stage before fitting. Both are ablated in Section 5.3.

GAUGE in turn sits within a long line of work on identifying dynamics online. Universal policies with online system identification [8] estimate physical parameters from recent history and condition a controller on them; RMA [9] learns an adaptation module that infers a latent extrinsics vector from proprioception, and meta-reinforcement learning [10] adapts a learned dynamics model from a few recent transitions. These methods identify the plant from the controller’s own perspective and assume access to joint-level signals. Our setting is the reverse: the plant includes the vendor controller, only the command and body twist are accessible, and the identified model is consumed by a planner. Probabilistic ensemble models [11] motivate our use of residual statistics, rather than point predictions alone, to decide when a calibrated model is no longer valid.

Active and safe exploration methods choose excitation to reduce model uncertainty while respecting constraints. Safe active dynamics learning [12] alternates exploration and exploitation with high-probability constraint satisfaction, and learning-based MPC for safe exploration [13] propagates model uncertainty to certify that exploratory inputs keep the system in a safe set. Our probe selection adopts the same structure (information gain traded against a risk cost under hard constraints), but because probes run near pedestrians we additionally impose procedural safety rules that are not expressed in the optimization (Section 4.5).

2.2 Simulation-to-reality calibration

Sim-to-real research addresses a closely related mismatch. Domain randomization [14] and dynamics randomization [15] train policies robust to parameter spread, and early quadruped

transfer work [16] combined system identification with randomization and actuator modeling. Xie et al. [17] showed for quadrupeds that careful identification can matter more than broad randomization. SimOpt [18] and BayesSim [19] close the loop by inferring simulator parameter distributions from real rollouts. These methods calibrate a simulator so that a controller transfers; we calibrate a reduced command-response model of a fixed, closed controller so that a planner’s predictions transfer. The two are complementary: an interface model of the kind fitted here could serve as the “plant” in a simulator used to train navigation policies.

2.3 Learned locomotion controllers behind the interface

The behaviors we measure are produced by locomotion controllers whose design increasingly relies on learning. Hwangbo et al. [1] trained velocity-tracking policies with a learned actuator model, and subsequent work produced blind [2] and perceptive [3] controllers robust to challenging terrain. Massively parallel training [20] made such controllers cheap to retrain, which is one reason vendor firmware can change interface behavior between releases. Rapid locomotion [21] pushed velocity-tracking policies to high speed, and Walk These Ways [22] exposes gait parameters as additional command inputs, illustrating that the mapping from command to motion depends on hidden or semi-hidden gait choices. Learned gait transitions [23] show that gait selection can be speed- and energy-dependent, which is precisely the kind of internal switching that appears externally as mode-switch transients. Vision-based controllers [24, 25] and agile navigation systems [26] demonstrate how much capability resides below the velocity interface. None of these works characterize the interface as a planner sees it on a closed commercial stack; that is the perspective adopted here.

2.4 Navigation planning over velocity interfaces

Local planners such as MPPI [4] depend on a forward model of how commands become motion; a biased model directly biases the selected trajectory. Learning-based navigation surveys [27] identify this model mismatch as a central source of failure. Inverse kinodynamics learning [28, 29] corrects commands for wheeled robots at high speed by learning the mapping from desired to required commands; it is the closest analog to interface calibration, but it targets a vehicle with documented dynamics and does not address discrete hidden modes. Self-supervised navigation systems [30] learn traversability and outcome models from experience, and legged navigation work couples vision with proprioception [31] or learns robot embeddings for legged navigation [32]. Traversability estimation [33] and GPU elevation mapping [34] supply the terrain representation used by such planners but leave the command interface implicit. Wheeled-legged systems [35] integrate navigation and locomotion training end to end, sidestepping the opaque interface entirely; production platforms do not offer that option. General navigation models [5, 6] and configuration-driven cross-embodiment planners such as AgniNav [36] make embodiment explicit through standardized actions or collision-envelope parameters; our interface assumptions can be read as the dynamic counterpart of such geometric descriptors. Finally, dynamics-aware metrics such as success weighted by completion time [37] reward planners that respect platform limits, which is only possible if those limits are measured.

2.5 Safety layers and safe operation

Control barrier functions [38] and Hamilton–Jacobi reachability [39] are the standard tools for certifying that commands keep a system within a safe set, and the survey by Brunke et al. [40] reviews how such certificates combine with learned models. Vendor safety layers likely implement related filters, but their thresholds are not published. Two of our recorded failures (an undocumented limiter and an unannounced careful mode) are consistent with such filters acting on the command stream, and they illustrate why a planner-side safety filter must be built on the *measured* envelope rather than the documented one.

03 Problem Formulation

Consider a platform $i \in \{A, B, C\}$ whose interface accepts a commanded body twist $u_k = (v_{x,k}, v_{y,k}, \omega_{z,k})$ at discrete times $t_k = k\Delta t$, where Δt is the command period. The measured body twist y_k is obtained from the platform’s state estimate and logged at $f_s = 100$ Hz. The interface has a hidden internal state comprising the effective mode $m_k \in \mathcal{M}_i$ (a finite set of gait and terrain modes, only partly documented), the internal state of the gait manager, and the payload mass p . The operator can request a mode $r_k \in \mathcal{M}_i$, but the effective mode m_k may lag or differ from r_k . We model the interface as

$$y_{k+1} = F_i(y_k, u_{k-D:k}, m_k, p) + w_k, \quad (1)$$

where F_i is unknown, $u_{k-D:k}$ is the command history over a window of D steps, and w_k collects terrain- and gait-induced disturbances. A *naive planner assumption* is that $y_k = u_k$ for every u_k inside the datasheet box $\mathcal{U}_{ds}$, independently of m_k and p . Calibration seeks a parametric surrogate $\hat{y}_k(\theta)$ of F_i that is accurate on the commands a given planner π issues. To make velocities and yaw rates commensurate we use the weighted norm $\|v\|_W^2 = v_x^2 + v_y^2 + \ell^2 \omega_z^2$, where ℓ is a characteristic body half-length.

We evaluate an interface and its calibrated model with four metrics computed on held-out validation segments $\mathcal{V}$ that are not used for fitting.

Command fidelity RMS error measures the fraction of commanded motion that the calibrated model fails to account for:

$$e_{\text{fid}} = 100 \sqrt{\frac{1}{|\mathcal{V}|} \sum_{k \in \mathcal{V}} \frac{\|y_k - \hat{y}_k\|_W^2}{\max(\|u_k\|_W^2, \nu^2)}} [\%], \quad (2)$$

where ν is a speed floor that prevents division by near-zero commands.

Phase lag (p95). For each validation window j containing persistent excitation, the lag ϕ_j is the delay that maximizes the normalized cross-correlation between the commanded and measured forward velocity; we report $L_{95} = Q_{0.95}(\{\phi_j\})$, the 95th percentile over windows, in milliseconds. A tail statistic is used because planners must budget for the slow cases, not the median.

Saturation-envelope error. Let α index directions in the normalized command space and let $\rho(\alpha)$ be the measured maximum attainable speed in direction α (from staircase probes), and $\hat{\rho}(\alpha)$ the envelope of the calibrated model. The error is $\varepsilon_{\text{sat}} = \frac{1}{N_\alpha} \sum_\alpha |\hat{\rho}(\alpha) - \rho(\alpha)|$ in m/s.

Mode-switch transient. After a requested mode change at time t_s , the transient T_{ms} is the time until $\|y_k - \bar{y}_{\text{new}}\|_W$ stays

within a band δ_{ms} of the new steady-state response $\bar{y}_{\text{new}}$, in seconds. This interval includes any switching hysteresis.

04 Method

4.1 Interface model structure

The surrogate is a mode-conditioned Hammerstein model with dead time: a static gain and saturation followed by first-order linear dynamics, one set of parameters per effective mode. Per axis this structure captures the three effects planners most often ignore (scale bias, delay, and clipping) with few parameters, which keeps identification well conditioned from a single session. The prediction is

$$\hat{y}_{k+1} = A_m \hat{y}_k + (I - A_m) \Pi_{\mathcal{E}_m(p)} (G_m u_{k-d_m}), \quad (3)$$

with $A_m = \text{diag}(e^{-\Delta t/\tau_{m,1}}, e^{-\Delta t/\tau_{m,2}}, e^{-\Delta t/\tau_{m,3}})$ holding the per-axis time constants $\tau_{m,j}$, G_m a diagonal gain matrix capturing scale bias, d_m an integer dead time in command periods, and $\Pi_{\mathcal{E}}$ the radial projection onto the attainable set $\mathcal{E}_m(p)$. The attainable set is star-shaped and described by its radial function,

$$\begin{aligned} \mathcal{E}_m(p) &= \{v : \|v\|_W \leq \rho_m(\alpha(v); p)\}, \\ \rho_m(\alpha; p) &= \rho_m^0(\alpha) (1 - \kappa_m p), \end{aligned} \quad (4)$$

where $\alpha(v)$ is the direction of v , ρ_m^0 is the unloaded envelope sampled at N_α directions and interpolated, and κ_m is a per-mode payload shrink coefficient. The linear payload term is the simplest form consistent with an envelope that contracts monotonically with load; it is a modeling choice, and Section 6 shows a case it does not capture.

Mode switches are modeled with a hysteresis h_m and a blending time constant τ_{sw} . If a switch from mode m to m' is requested at t_s , the prediction follows mode m until $t_s + h_m$ and then blends,

$$\begin{aligned} \hat{y}_k &= \beta_k \hat{y}_k^{(m')} + (1 - \beta_k) \hat{y}_k^{(m)}, \\ \beta_k &= 1 - e^{-(t_k - t_s - h_m)^+ / \tau_{\text{sw}}}, \end{aligned} \quad (5)$$

with $(x)^+ = \max(x, 0)$. The full parameter vector is $\theta = \{G_m, \tau_{m,j}, d_m, \rho_m^0, \kappa_m, h_m, \tau_{\text{sw}}\}_{m \in \mathcal{M}_i}$.

4.2 Planner-conditioned probe design

Following GAUGE [7], probes are chosen so that information about θ is concentrated where the target planner operates. Let $q_\pi(u, m)$ be the empirical occupancy of the planner over commands and modes, estimated from logs of the target MPPI-style local planner [4] on the same sites. A probe $\xi = (u_1, \dots, u_H)$ is drawn from parametric families (steps, staircases, chirps, turn-in-place sweeps, diagonal sweeps, and mode-toggle sequences) and scored by

$$\begin{aligned} J(\xi) &= \log \det \left(\Lambda_0 + \sum_{k=1}^H q_\pi(u_k, m_k) \Phi_k^\top \Sigma_w^{-1} \Phi_k \right) \\ &\quad - \lambda_r c_{\text{risk}}(\xi), \end{aligned} \quad (6)$$

where Λ_0 is the current information matrix of θ , $\Phi_k = \partial \hat{y}_k / \partial \theta$ is the sensitivity of the prediction, Σ_w is the disturbance covariance, and c_{risk} is a risk cost (predicted stopping distance relative to available clearance) weighted by

λ_r . The maximizer is selected subject to hard constraints: $\|u_k\|_W \leq \bar{u}_{\text{probe}}$, predicted stopping distance below free-space clearance, and the probe trajectory remaining inside the site geofence. The occupancy weighting is the GAUGE design choice; the log det criterion (D-optimality) is used because it is invariant to reparameterization of θ , which matters when parameters with different units are identified jointly. Unlike a default-mode-only campaign, the probe library includes every requested terrain mode and mode-toggle sequences, so that h_m , τ_{sw} , and per-mode envelopes are identifiable.

4.3 Online fitting with gait-disturbance rejection

Stepping produces periodic body-velocity oscillation at the stride frequency f_g and its harmonics, and foot slip produces sparse large residuals. Both bias a squared-error fit. We therefore filter the measured twist with a notch filter N_{f_g} tracking the stride frequency (estimated from the dominant spectral peak of vertical body velocity) and fit with a Huber loss ψ_δ under exponential forgetting:

$$\hat{\theta}_K = \arg \min_{\theta} \sum_{k \leq K} \gamma^{K-k} \psi_\delta(N_{f_g}(y_k) - \hat{y}_k(\theta)) + \|\theta - \theta_0\|_{P_0^{-1}}^2. \quad (7)$$

Here $\gamma \in (0, 1)$ is the forgetting factor, θ_0 and P_0 are the prior mean and covariance carried over from the previous session of the same platform and mode, and ψ_δ is quadratic for residuals below δ and linear above. Given d_m , the model is linear in the gain and in the filter coefficients inside the unsaturated region, so the minimization is carried out by iteratively reweighted recursive least squares; d_m is selected by enumerating integer delays, and the envelope samples ρ_m^0 are updated only from staircase segments that reach a plateau. The ablation variant without disturbance rejection replaces $N_{f_g}(y_k)$ with y_k and ψ_δ with the squared loss.

4.4 Detection of undocumented behaviors

A calibrated model is only useful while its structure holds. To detect hidden limiters or unannounced modes we run a one-sided CUSUM test on the normalized residual $z_k = (\|y_k - \hat{y}_k\|_W - \mu_z)/\sigma_z$,

$$S_k = \max(0, S_{k-1} + z_k - c), \quad \text{alarm if } S_k > \eta, \quad (8)$$

where μ_z, σ_z are residual statistics from the validation set and c, η set the drift allowance and alarm threshold. An alarm in which the measured speed systematically falls short of the prediction only above some speed triggers a dedicated confirmation sequence (a fine-grained speed staircase) in the suspected mode; a sudden contraction of the envelope at constant command triggers a check of the platform's reported state. Alarms are logged as candidate interface behaviors and confirmed only after repetition.

4.5 Probe safety in public settings

Campus paths are shared with pedestrians, and probing deliberately drives the platform toward its limits. The protocol therefore separates *where* a probe may run from *what* it may command. High-speed staircases and saturation probes were executed only in the laboratory and in cordoned path segments with a dedicated spotter holding a hardware stop; on open paths the

Table 1: Field campaign configuration.

Item	Value
Platforms	A: research unit, v2.3 (primary); B: research unit, v2.1; C: work unit, v1.8
Platform classes	Two 70-class research units, one 45-class work unit
Duration and sites	14 days; laboratory, campus path, light outdoor terrain
Sessions	60 per platform (20 per terrain mode); 180 in total
Logging	100 Hz body twist and commands
Payloads	0, 5, 10 kg, rotated per session
Probing	GAUGE-style planner-conditioned probes (Section 4)

probe family was restricted to low-amplitude chirps and turn-in-place sweeps. Any person entering a fixed radius around the platform paused the probe, and the session resumed only after the area cleared, with the interrupted segment discarded from fitting. The risk term in Eq. (6) uses the most conservative envelope identified so far, so an undocumented limiter or careful mode cannot make the predicted stopping distance optimistic. Mode-toggle probes were issued only at standstill or low speed, because transients at speed are precisely the behavior under investigation. These rules reduce the coverage of high-speed probing outside the laboratory, which is a deliberate trade-off listed among the limitations.

4.6 Probe parameters and reproducibility

REPRODUCIBILITY NOTE

Commands are issued at 50 Hz ($\Delta t = 20$ ms); body twist is logged at 100 Hz. Probe families: steps and staircases in 0.2 m/s increments (S-series), linear chirps from 0.1 to 3 Hz with 0.3 m/s amplitude, turn-in-place sweeps, diagonal sweeps at $N_\alpha = 16$ directions, and mode-toggle sequences at standstill. Fitting uses forgetting factor $\gamma = 0.995$, Huber threshold $\delta = 0.05$ m/s, speed floor $\nu = 0.1$ m/s, characteristic length $\ell = 0.4$ m, delays enumerated up to $D = 10$ command periods, and settling band $\delta_{\text{ms}} = 0.05$ m/s for T_{ms} . Each platform received 60 sessions (20 per terrain mode) with payloads of 0, 5, and 10 kg rotated per session. **Data availability:** the session logs, probe definitions, and fitted interface parameters are available from the corresponding author on reasonable request; vendor-identifying firmware details are withheld under the terms of the platform agreements.

05 Experiments

5.1 Setup and platforms

The testbed consisted of three commercial quadruped platforms: two 70-class research units and one 45-class work unit. Platform A is the laboratory's primary research unit (firmware v2.3), Platform B is the second research unit (firmware v2.1), and Platform C is the work unit (firmware v1.8). Vendors are not named; the findings concern interface behavior of specific firmware releases, not product rankings. The campaign lasted 14 days across three site types: laboratory floor, campus paths, and light outdoor terrain. Each platform completed 60 GAUGE-style probe sessions, 20 in each terrain mode, for 180 sessions in total. Body twist was logged at 100 Hz, and payloads of 0, 5, and 10 kg were rotated per session. Table 1 summarizes the configuration.

5.2 Main comparison

Table 2 reports the four interface metrics. The row labeled "Ours" corresponds to the primary unit (Platform A) character-

ized with the full probe suite; it is the same configuration as the first row of the ablation in Table 3. The rows for Platforms B and C report the same metrics for the other two units. Because each row is a different platform and firmware, differences between rows reflect both the interfaces themselves and how well the surrogate in Eq. (3) captures them; the table is a characterization, not a ranking of methods.

Three observations follow. First, residual command-fidelity error lies between 9.7% and 11.1% on all three units, so even after calibration roughly a tenth of commanded motion is unexplained by the surrogate, and an uncalibrated planner would face a larger discrepancy. The spread across platforms is 1.4 percentage points. Second, p95 phase lag ranges from 71 to 81 ms. Platform C’s lag exceeds the primary unit’s by 10 ms and Platform B’s by 8 ms; at a 20 ms command period these tails correspond to roughly four command periods, which a planner that assumes immediate response does not represent. Third, the saturation-envelope error is between 0.18 and 0.21 m/s, and mode-switch transients are 0.9 to 1.0 s on all units. A one-second transient is long relative to typical replanning horizons and is the quantity most directly tied to the overshoot failure in Section 6.

5.3 Ablation of the probe suite

Table 3 isolates the two procedural extensions on the primary unit.

Restricting probes to the default mode raises the error from 9.7% to 10.5% (0.8 percentage points, about 8% relative). This is expected under Eq. (3): when non-default modes are never excited, their gains, delays, and envelopes are extrapolated from the default mode, and validation segments recorded in other terrain modes are then mispredicted. Removing disturbance rejection raises the error to 10.7% (1.0 percentage point, about 10% relative), the larger of the two effects. Without the notch filter and robust loss, stride-frequency oscillation and slip residuals are absorbed into the gain and time-constant estimates. Both extensions therefore contribute, and neither alone accounts for the full-suite result.

5.4 Payload sweep

Table 4 reports fidelity error and p95 phase lag as a function of payload.

Payload is the strongest single factor in the study. From 0 to 10 kg the fidelity error rises by 3.7 percentage points (from 8.9% to 12.6%, about 42% relative) and the p95 lag by 24 ms (from 64 to 88 ms, 37.5% relative). The degradation is not linear: the second 5 kg adds 2.4 percentage points and 14 ms, versus 1.3 percentage points and 10 ms for the first 5 kg. A planner that calibrates once without payload and then carries 10 kg would therefore underestimate the p95 lag by 24 ms, and the accelerating growth suggests that extrapolating from light-payload calibration is unsafe. The 88 ms p95 lag at 10 kg is 4.4 command periods at 50 Hz.

06 Failure Cases

The probes surfaced three interface behaviors that violate naive planning assumptions.

6.1 Undocumented speed limiter (Platform A)

On the primary unit (firmware v2.3), an undocumented speed limiter engages at 3.2 m/s in the vendor’s “outdoor” mode. The limit is absent from the datasheet and was caught by probe se-

quence S-11, a forward speed staircase in that mode of the kind triggered by the residual monitor in Eq. (8). A planner that uses the datasheet box $\mathcal{U}_{ds}$ in outdoor mode would plan trajectories above 3.2 m/s that the platform silently clips, producing systematic along-track lag that grows with segment length. The limiter illustrates that the envelope is mode-dependent, so ρ_m in Eq. (4) must be identified per mode rather than once per platform.

6.2 Gait-manager hysteresis on mode switch (Platform B)

Platform B’s gait manager adds a 0.7 s hysteresis on mode switch. Naive planners that assume instant switching overshoot by 0.4 m, because they schedule the speed and turning profile of the new mode at the moment the switch is requested while the platform continues to execute the old mode for 0.7 s. The recorded 1.0 s mode-switch transient for Platform B in Table 2 is consistent with this hysteresis plus a shorter blending phase, which is the structure encoded by h_m and τ_{sw} in Eq. (5).

6.3 Payload-triggered careful mode (Platform C)

With a 10 kg payload on Platform C, the saturation envelope shrinks by 21%. In addition, two sessions triggered an unannounced “careful mode” that halves the usable envelope. The 21% shrink is the kind of smooth, payload-dependent contraction that the κ_m term in Eq. (4) is designed to capture; the careful mode is not, because it is a discrete hidden mode entered without any change in the requested mode r_k . It is the clearest example of an effective mode $m_k \neq r_k$. Because it occurred in only two sessions, its trigger conditions could not be isolated, and a planner must treat it as an event that can occur at any time under heavy load.

07 Interface Assumptions for Planners

The measurements motivate the following assumptions, which any planner for these platforms should make explicit rather than inherit implicitly from a datasheet. Table 5 links each to its evidence.

1. **Envelope is a function of mode and payload.** The attainable set must be indexed by the effective mode and payload, $\mathcal{E}_m(p)$, and identified per firmware release; datasheet limits are an upper bound at best (limiter at 3.2 m/s; 21% shrink at 10 kg).
2. **Response is delayed by a load-dependent tail.** Planner rollouts should include a dead time and lag budgeted at the p95 level for the current payload (64 to 88 ms across the sweep), not a nominal or median value.
3. **Mode switches take time.** A requested mode change should be modeled as a scheduled transition with hysteresis and settling (0.9 to 1.0 s transients; 0.7 s hysteresis on Platform B), and planners should avoid committing to new-mode maneuvers inside that window.
4. **Effective mode is not the requested mode.** Hidden modes can be entered without announcement (careful mode under 10 kg), so the planner should monitor tracking residuals online and fall back to the most conservative identified envelope when they drift.
5. **Residual tracking error is not negligible.** Even calibrated, roughly 10% of commanded motion is unexplained (9.7% to 12.6% across conditions); collision margins should be sized from this residual rather than from localization error alone.

Table 2: Main comparison of interface metrics. Lower is better for all columns.

Method	Command fidelity RMS (%)	Phase lag p95 (ms)	Saturation envelope error (m/s)	Mode-switch transient (s)
Platform B (research, v2.1 firmware)	11.1	79	0.19	1.0
Platform C (work unit, v1.8 firmware)	10.6	81	0.21	1.0
Ours	9.7	71	0.18	0.9

Table 3: Ablation of the probe suite (command fidelity RMS error; lower is better).

Variant	Command fidelity RMS (%)
All probe modes (terrain modes on)	9.7
Default mode only	10.5
Probe suite, no disturbance rejection	10.7

Table 4: Payload sweep (lower is better).

Payload (kg)	Command fidelity RMS (%)	Phase lag p95 (ms)
0	8.9	64
5	10.2	74
10	12.6	88

6. Calibration is tied to conditions. Calibrations obtained in the default mode or without disturbance rejection are measurably worse (10.5% and 10.7% versus 9.7%) and should not be reused across modes, payloads, or firmware versions without revalidation.

08 Limitations

The study has several limitations. It covers three platforms and specific firmware releases, so the measured behaviors may change with vendor updates; the limiter and careful mode in particular are properties of v2.3 and v1.8 firmware, respectively, as observed. The campaign lasted 14 days and was limited to laboratory, campus-path, and light outdoor terrain; stairs, deformable ground, and adverse weather were not probed. The main comparison mixes platform and firmware differences with surrogate-model adequacy, and only the primary unit was ablated, so the ablation does not establish that the same gains hold on Platforms B and C. The payload sweep reports aggregate values and does not separate the platforms, so platform-specific payload sensitivities beyond the Platform C envelope shrink are not available. The careful mode appeared in only two sessions, which is too few to characterize its trigger. The surrogate in Eq. (3) is low-order by design and cannot represent discrete hidden modes, which are handled only by detection. All measurements rely on the platforms’ own state estimates, so slow estimator bias would be indistinguishable from interface bias. Finally, the public-setting safety rules restricted high-speed probing to the laboratory and cordoned segments, so high-speed behavior on open paths is extrapolated rather than measured.

09 Conclusion

This field study applied GAUGE-style planner-conditioned active calibration to three production quadruped platforms and found that their velocity interfaces deviate from the transparent model that planners commonly assume in consistent, measurable ways. Calibrated residual error remained near 10%, p95 lag reached 71 to 81 ms across units and 88 ms under a 10 kg

payload, and mode switches took about a second. Probing all terrain modes and rejecting gait disturbances each improved calibration, and the probes exposed three behaviors (an undocumented limiter, switching hysteresis, and an unannounced careful mode) that a datasheet-driven planner would not anticipate. The practical conclusion is that the interface assumptions listed in Section 7 should be stated explicitly by any planner deployed on such platforms and re-measured whenever mode, payload, or firmware changes.

References

- [1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. arXiv:1901.08652, 2019.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [3] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. arXiv:2201.08117, 2022.
- [4] G. Williams, A. Aldrich, and E. Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. arXiv:1509.01149, 2015.
- [5] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [6] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. arXiv:2306.14846, 2023.
- [7] T. Zang, H. Huang, S. Cheng, and W. Zhang. GAUGE: Planner-Conditioned Active Calibration of Opaque Quadruped Velocity Interfaces. arXiv:2607.02841, 2026.
- [8] W. Yu, J. Tan, C. K. Liu, and G. Turk. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. arXiv:1702.02453, 2017.
- [9] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [10] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to Adapt in Dynamic, Real-World Environments Through Meta-Reinforcement Learning. arXiv:1803.11347, 2018.
- [11] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. arXiv:1805.12114, 2018.
- [12] T. Lew, A. Sharma, J. Harrison, A. Bylard, and M. Pavone. Safe Active Dynamics Learning and Control: A Sequential Exploration-Exploitation Framework. arXiv:2008.11700, 2020.
- [13] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-Based Model Predictive Control for Safe Exploration. arXiv:1803.08287, 2018.
- [14] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. arXiv:1703.06907, 2017.

Table 5: Planner-facing interface assumptions and the evidence that motivates them.

Assumption	Naive default	Evidence in this study	Planner action
Mode- and payload-indexed envelope	Fixed datasheet box	3.2 m/s limiter in outdoor mode (A); 21% shrink at 10 kg (C)	Identify $\mathcal{E}_m(p)$ per mode, payload, firmware
Load-dependent latency tail	Instant tracking	p95 lag 64 to 88 ms over 0 to 10 kg	Roll out with p95 dead time and lag
Non-instant mode switch	Instant switching	0.7 s hysteresis, 0.4 m overshoot (B); 0.9 to 1.0 s transients	Schedule switch; freeze new-mode maneuvers
Hidden effective mode	$m_k = r_k$	Careful mode in two 10 kg sessions (C)	Residual monitor; conservative fallback
Residual tracking error	Zero after calibration	9.7% to 12.6% fidelity error	Inflate margins from residual
Condition-specific calibration	Calibrate once	10.5% (default only), 10.7% (no rejection) vs 9.7%	Recalibrate per mode, payload, firmware

- [15] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [16] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. arXiv:1804.10332, 2018.
- [17] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg. Dynamics Randomization Revisited: A Case Study for Quadrupedal Locomotion. arXiv:2011.02404, 2020.
- [18] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Isaac, N. Ratliff, and D. Fox. Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. arXiv:1810.05687, 2018.
- [19] F. Ramos, R. C. Possas, and D. Fox. BayesSim: Adaptive Domain Randomization via Probabilistic Inference for Robotics Simulators. arXiv:1906.01728, 2019.
- [20] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [21] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid Locomotion via Reinforcement Learning. arXiv:2205.02824, 2022.
- [22] G. B. Margolis and P. Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. arXiv:2212.03238, 2022.
- [23] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots. Fast and Efficient Locomotion via Learned Gait Transitions. arXiv:2104.04644, 2021.
- [24] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.
- [25] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [26] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [27] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: a Survey. arXiv:2011.13112, 2020.
- [28] X. Xiao, J. Biswas, and P. Stone. Learning Inverse Kinodynamics for Accurate High-Speed Off-Road Navigation on Unstructured Terrain. arXiv:2102.12667, 2021.
- [29] H. Karnan, K. S. Sikand, P. Atreya, S. Rabiee, X. Xiao, G. Warnell, P. Stone, and J. Biswas. VI-IKD: High-Speed Accurate Off-Road Navigation using Learned Visual-Inertial Inverse Kinodynamics. arXiv:2203.15983, 2022.
- [30] G. Kahn, P. Abbeel, and S. Levine. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. arXiv:2002.05700, 2020.
- [31] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling Vision and Proprioception for Navigation of Legged Robots. arXiv:2112.02094, 2021.
- [32] J. Truong, D. Yarats, T. Li, F. Meier, S. Chernova, D. Batra, and A. Rai. Learning Navigation Skills for Legged Robots with Learned Robot Embeddings. arXiv:2011.12255, 2020.
- [33] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. arXiv:2305.08510, 2023.
- [34] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation Mapping for Locomotion and Navigation using GPU. arXiv:2204.12876, 2022.
- [35] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter. Learning Robust Autonomous Navigation and Locomotion for Wheeled-Legged Robots. arXiv:2405.01792, 2024.
- [36] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. AgniNav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [37] N. Yokoyama, S. Ha, and D. Batra. Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation. arXiv:2103.08022, 2021.
- [38] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control Barrier Functions: Theory and Applications. arXiv:1903.11199, 2019.
- [39] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. arXiv:1709.07523, 2017.
- [40] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. arXiv:2108.06266, 2021.