

A Benchmark Protocol for Evaluating Opaque Velocity Interfaces in Legged Robots

Grace Lin • Wei Xu • Arjun Mehta • Ethan Cole

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

PUBLISHED

July 19, 2026

REGISTERED

July 21, 2026

RECORD TYPE

Preprint

ABSTRACT

Velocity commands are the de facto interface between navigation software and the locomotion controllers of legged robots, yet these controllers behave as opaque, stateful, and platform-specific systems whose command-to-motion response is rarely documented and never characterized in a common way. This paper proposes a benchmark protocol for evaluating such opaque velocity interfaces. The protocol consists of a fixed suite of 42 probe command sequences drawn from five families (step, sweep, spiral, saturation, and disturbance rejection), five standardized metrics (command fidelity, phase lag, steady-state bias, saturation envelope, and recovery time), and a reporting template that records firmware, gait mode, payload, and terrain alongside the results. GAUGE-style active calibration serves as the reference characterization method. The protocol is demonstrated on two ANYmal B units with identical firmware and different service histories and on one Unitree Go2, on a lab floor and a stair course, with three repetitions per sequence and platform. The characterization produced by the full suite attains a command fidelity error of 8.4 percent RMS and a 95th-percentile phase lag of 62 ms, compared with 9.0 percent and 66 ms for the reference-unit characterization. A probe-budget sweep shows that fidelity error falls from 17.8 to 8.4 percent as the suite grows from 6 to 42 sequences, with diminishing returns beyond 27. The recorded anomalies, including an unmodeled 0.4 s gait-mode transient and payload-dependent early saturation, illustrate behavior that single-unit characterizations miss. The probe suite and exact probe parameters are released with the paper.

INDEX TERMS legged robots, velocity interface, benchmarking, system identification, active calibration, quadruped locomotion

KAGG NOTE A legged robot's velocity interface should be reported like a sensor: with a fixed excitation protocol, standardized error metrics, and the configuration under which they were measured.

01 Introduction

Nearly every navigation stack deployed on a legged robot ends in the same place: a planar velocity command, typically a forward speed, a lateral speed, and a yaw rate, handed to a locomotion controller at a fixed rate. Model-based controllers built on convex model-predictive control and whole-body optimization [1, 2, 3] and learned policies trained in simulation [4, 5, 6, 7] expose this interface in nearly identical form. Above it, planners for rough terrain [8, 9], learned navigation models [10, 11], and sampling-based controllers [12] treat the command as if it were executed faithfully, or at best with a fixed delay and a box constraint.

Practitioners repeatedly discover that this assumption fails. The velocity controller is *opaque*: its internal structure, gains, and learned weights are not available to the planner. It is *stateful*: its response depends on gait mode, contact history, and internal filters, so the same command produces different motion depending on what preceded it. It is *platform-specific*: two robots of the same model with identical firmware can respond differently because of actuator wear, calibration drift, or service history. Recent work has begun to treat this interface explicitly; GAUGE [13] calibrates an opaque quadruped velocity interface by issuing planner-conditioned probe commands and fitting a command-to-motion model online. What is missing is a common way to *characterize and compare* such interfaces: a

fixed set of excitations, a fixed set of metrics, and a fixed set of reporting fields, so that a result obtained on one unit, platform, or firmware release can be compared with another.

This paper addresses that gap with a benchmark protocol. It does not propose a new locomotion controller, and it does not claim that any particular interface is better than another. Its object is the measurement procedure. The contributions are:

1. A formalization of the opaque velocity interface as a stateful, configuration-dependent operator, together with five standardized metrics (command fidelity, phase lag, steady-state bias, saturation envelope, and recovery time) with precise definitions that can be computed from logged commands and estimated body twist.
2. A fixed suite of 42 probe command sequences in five families (step, sweep, spiral, saturation, and disturbance rejection), specified to the level of exact amplitudes, durations, and ramp rates, and released for reuse.
3. A reporting template that ties every number to the unit, firmware, gait mode, payload, terrain, and estimator under which it was obtained.
4. A demonstration on two ANYmal B units and one Unitree Go2 on a lab floor and a stair course, using GAUGE-style active calibration [13] as the reference characterization

method, including an ablation over probe families, a sweep over probe budget, and a record of the anomalies that the protocol exposed.

Section 2 reviews related work. Section 3 formalizes the problem. Section 4 develops the protocol. Section 5 reports the demonstration, Section 6 discusses limitations, and Section 7 concludes.

02 Related Work

2.1 Calibration of opaque velocity interfaces

GAUGE [13] is the work closest to ours. It treats the quadruped velocity controller as a black box and performs active calibration: rather than passively fitting a model to whatever commands a mission happens to produce, it issues probe commands conditioned on the planner that will consume the interface, so that excitation is concentrated in the region of command space that the planner actually uses, and it fits a calibrated model of command-to-motion response online. The calibrated model can then be used by the planner to anticipate lag, bias, and saturation. Our protocol differs in purpose rather than in modeling philosophy. GAUGE answers the question *how should a given planner calibrate a given interface*; the protocol answers *how should any interface be characterized so that results are comparable*. To make comparisons meaningful, the protocol fixes the excitation independently of any planner, which is precisely what planner-conditioned probing deliberately avoids. We therefore adopt GAUGE-style active calibration as the reference characterization method against which the fixed suite is assessed, and we adopt a response-model structure compatible with it (Section 4.4), so that a GAUGE calibration and a protocol characterization can be scored with the same metrics.

2.2 Locomotion controllers that expose velocity interfaces

The interfaces under test are produced by two broad classes of controller. Model-based controllers for ANYmal [14, 3] and the MIT Cheetah family [1, 2] compute contact forces and footholds by optimization and track a commanded base twist through a hierarchy of tasks; their response is in principle derivable from the model, but in practice depends on gait timing, swing-leg heuristics, and tuning that are not exposed to users. Learned controllers for ANYmal [4, 5, 6] and for low-cost quadrupeds [15, 16, 17, 18] are trained to maximize a velocity-tracking reward, often with dynamics randomization [19] and with massively parallel simulation [7]. Their tracking behavior is an emergent property of the reward and the training distribution, and is not documented in any parametric form. Several lines of work make the interface more explicitly stateful. Rapid Motor Adaptation [16] and concurrent estimator training [20] condition the policy on latent estimates of environment and body state, so the response to a command depends on an internal estimate that evolves over time. DreamWaQ [21] infers terrain implicitly from proprioception with a similar effect. Walk These Ways [22] adds gait parameters to the command itself, enlarging the interface beyond a planar twist. Hybrid schemes combine learned foothold selection with optimal control [23], and whole-body policies couple locomotion with manipulation [24]. Vision-conditioned and agile policies [25, 26, 27, 28, 29] further entangle the response with perception. None of these works is wrong to optimize tracking reward; the point is that the resulting command-to-motion map differs across controllers, terrains, and units in ways that the

reward value alone does not report. The protocol is designed to characterize any of them through the same probes without requiring access to their internals.

2.3 Consumers of the velocity interface

Navigation methods for legged robots rely on the interface in different ways. Elevation-mapping pipelines [8] and field-tested planners such as ArtPlanner [9] plan geometric paths and delegate tracking to a velocity controller with a nominal speed limit. Wild Visual Navigation [30] learns traversability online and likewise issues velocity commands to a fixed locomotion policy. End-to-end approaches merge local navigation and locomotion [31, 29] and thereby remove the explicit interface, at the cost of platform-specific training. Cross-embodiment navigation models [10, 11] standardize the action space across robots, while CeRLP [32] and AgniNav [33] make embodiment parameters explicit in local planning; all of these assume that the low-level controller realizes the standardized action with acceptable fidelity. Sampling-based predictive controllers such as MPPI [12] require a forward model of the command response and are directly sensitive to its accuracy. A shared characterization of the interface is useful to each of these consumers: it states, in a common vocabulary, how much fidelity, lag, bias, and headroom they can expect.

2.4 System identification and state estimation

The protocol borrows its excitation design from classical system identification. Step and swept-sine (chirp) inputs are standard tools for estimating gain, time constant, delay, and frequency response [34, 35]; persistency of excitation and input amplitude selection under saturation are treated there in depth. What classical identification does not supply is a convention for stateful, mode-switching, contact-rich systems whose output is itself an estimate. The realized body twist on a legged robot is produced by a state estimator that fuses leg kinematics and inertial measurements [36, 37], possibly with vision and LiDAR [38]. Estimator bias therefore enters every metric, and the reporting template records the estimator as part of the configuration.

2.5 Benchmarking protocols in robotics

Benchmark protocols elsewhere in robotics show the value of fixing excitations and metrics. The YCB object set and its protocols [39] made manipulation results comparable by standardizing objects and procedures. In embodied navigation, standardized task definitions and metrics such as success weighted by path length [40], and its dynamics-aware extension weighted by completion time [41], made navigation agents comparable, and Sim2Real predictivity analysis [42] showed why physical evaluation must be reported separately from simulation. To our knowledge no comparable protocol exists for the velocity interface itself, the layer on which these navigation metrics silently depend.

03 Problem Formulation

3.1 The interface as an opaque operator

Let Δt denote the command period and let $t \in \{0, 1, \dots, T\}$ index command steps. At each step a client issues a planar command $\mathbf{u}_t = (u_t^x, u_t^y, u_t^\psi)^\top \in \mathcal{U} \subset \mathbb{R}^3$, consisting of forward speed, lateral speed (both in m/s, body frame), and yaw rate (rad/s). The robot's realized body twist, as reported by its state

estimator in the same frame, is $\mathbf{y}_t = (y_t^x, y_t^y, y_t^\psi)^\top$. We model the interface as an opaque, stateful operator

$$\mathbf{y}_t = \mathcal{H}_\theta(\mathbf{u}_{0:t}, \boldsymbol{\xi}_0; \mathbf{c}) + \epsilon_t, \quad (1)$$

where $\mathbf{u}_{0:t}$ is the command history, $\boldsymbol{\xi}_0$ is the unobserved internal state of the controller at the start of a sequence (gait mode, filter states, latent estimates), $\mathbf{c}$ is the configuration descriptor, θ collects unit-specific properties such as actuator condition, and ϵ_t is estimation and process noise. The configuration $\mathbf{c} = (\text{unit, firmware, gait mode, payload, terrain, estimator})$ is observable and must be reported; θ and $\boldsymbol{\xi}$ are not.

Two properties of $\mathcal{H}$ motivate the protocol. First, $\mathcal{H}$ depends on history, not only on $\mathbf{u}_t$, so any characterization must be defined relative to specific input sequences. Second, $\mathcal{H}$ depends on θ , so two units with equal $\mathbf{c}$ apart from the unit identifier may differ.

3.2 Characterization and its evaluation

A *characterization* is a model $\hat{\mathcal{H}}$ that predicts $\mathbf{y}$ from $\mathbf{u}$ under a stated configuration. A characterization is useful to a planner insofar as commands designed with it produce the intended motion. We therefore evaluate characterizations by *compensated replay*: given a reference twist trajectory $\mathbf{r}_{0:T}$, the characterization is inverted to produce a command sequence $\tilde{\mathbf{u}}_{0:T} = \hat{\mathcal{H}}^{-1}(\mathbf{r}_{0:T})$, the sequence is executed on the robot, and the metrics of Section 4.2 compare the realized $\mathbf{y}$ with $\mathbf{r}$. Compensated replay has two advantages over open-loop prediction error. It evaluates the characterization in the way a planner uses it, and it makes all five metrics, including recovery time after a disturbance, properties of measured motion rather than of a model.

3.3 Weighted error

Because linear and angular components have different units, errors are combined with the weighting $\mathbf{W} = \text{diag}(1, 1, \ell)$, where $\ell = 0.4$ m is a characteristic length that converts yaw rate to an equivalent tangential speed at the body perimeter. The weighted error at step t is

$$e_t = \|\mathbf{W}(\mathbf{y}_t - \mathbf{r}_t)\|_2. \quad (2)$$

The value $\ell = 0.4$ m is approximately half the body length of both platforms considered here; it is part of the protocol definition so that metrics are comparable across platforms.

04 Benchmark Protocol

The protocol has four parts: a probe suite (Section 4.1), metric definitions (Section 4.2), an evaluation procedure (Section 4.3), and a reference characterization model with its fitting procedure (Section 4.4). A reporting template (Section 4.5) completes it.

4.1 Probe suite

All probe amplitudes are expressed as fractions of the platform's nominal command envelope $\bar{\mathbf{u}} = (\bar{u}^x, \bar{u}^y, \bar{u}^\psi)$ for the active gait mode, taken from the vendor documentation and recorded in the template. Normalizing by $\bar{\mathbf{u}}$ makes the suite applicable to platforms whose maximum speeds differ by a factor of several. Commands are issued at 100 Hz ($\Delta t = 10$ ms), the same rate at which commands and estimated twist are logged. Every sequence begins and ends with 2 s of zero command so that the controller starts from and returns to a standing or stepping-in-place state, which limits the dependence on $\boldsymbol{\xi}_0$.

The suite contains 42 sequences, P-01 to P-42, summarized in Table 1.

Step probes isolate gain, time constant, dead time, and steady-state bias on one axis at a time. *Sweep probes* are logarithmic chirps

$$u^j(t) = A \bar{u}^j \sin\left(\frac{2\pi f_0 D}{\ln(f_1/f_0)} \left[\left(\frac{f_1}{f_0}\right)^{t/D} - 1\right]\right), \quad (3)$$

with $f_0 = 0.1$ Hz, $f_1 = 2.0$ Hz, duration $D = 40$ s, and amplitude fraction A ; they expose frequency-dependent attenuation and phase lag up to roughly the stride frequency. *Spiral probes* hold a constant forward command $u^x = c \bar{u}^x$ while ramping the yaw-rate command linearly from zero to $\pm 0.8 \bar{u}^\psi$ over 20 s, exciting the cross-axis coupling between forward speed and turning that per-axis probes miss. *Saturation probes* ramp a single linear axis from zero to $1.5 \bar{u}^j$ on the lab floor, or to $1.0 \bar{u}^j$ on the stair course, at a fixed rate ρ ; they locate the command at which realized speed stops following the command. *Disturbance-rejection probes* hold $u^x = 0.5 \bar{u}^x$ while a pendulum impactor of 5 kg mass released from a fixed drop height strikes the trunk longitudinally or laterally; they measure how quickly the interface returns to the commanded motion.

The stair course is used only for the three forward saturation ramps P-25 to P-27, which are executed in the platform's stair gait mode; all other sequences are executed on the flat lab floor. This keeps the stair exposure short and repeatable while still capturing gait-mode-dependent saturation.

4.2 Metric definitions

Let $\mathcal{T}$ denote the set of logged steps in the evaluation sequences, excluding the zero-command padding. Each metric is defined so that lower values are better.

Command fidelity is the normalized root-mean-square weighted error, in percent:

$$F = 100 \cdot \frac{\sqrt{\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} e_t^2}}{\sqrt{\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\mathbf{W} \mathbf{r}_t\|_2^2}}. \quad (4)$$

Phase lag is computed per axis j over sliding windows w of 2 s length with 1 s overlap. In each window the lag is the shift that maximizes the cross-correlation between reference and response,

$$\lambda_{w,j} = \Delta t \cdot \arg \max_{k \in \{0, \dots, 30\}} \sum_{t \in w} r_t^j y_{t+k}^j, \quad (5)$$

searched over 0 to 300 ms. Windows in which the reference has a standard deviation below $0.05 \bar{u}^j$ are discarded because the lag is then ill-defined. The reported value Λ_{95} is the 95th percentile of $\lambda_{w,j}$ over all retained windows and axes. A tail percentile is used because planners fail on worst-case lag rather than average lag.

Steady-state bias is evaluated on the plateau segments $\mathcal{S}$ of step probes, defined as the final 1 s of each 4 s hold:

$$B = 100 \cdot \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{|\bar{y}_s^j - r_s^j|}{|r_s^j|}, \quad (6)$$

where $\bar{y}_s^j$ is the mean response on segment s and r_s^j the constant reference there.

Table 1: Probe suite: 42 command sequences in five families. Amplitudes are fractions of the gait-mode nominal envelope $\bar{\mathbf{u}}$.

Family	IDs	Count	Parameters	Purpose
Step	P-01–P-12	12	3 axes (x, y, ψ) $\times$ 4 amplitudes (0.25, 0.5, 0.75, 1.0); 4 s at $+\bar{A}$, 2 s zero, 4 s at $-\bar{A}$	Gain, time constant, dead time, steady-state bias
Sweep	P-13–P-21	9	3 axes $\times$ 3 amplitudes (0.2, 0.4, 0.6); log chirp 0.1–2.0 Hz over 40 s	Frequency-dependent attenuation and phase lag
Saturation	P-22–P-30	9	3 ramp rates (0.1, 0.2, 0.4 $\bar{u}^j/\text{s}$) $\times$ {forward, lab floor, to 1.5; forward, stair course, to 1.0; lateral, lab floor, to 1.5}	Saturation envelope per axis and terrain
Spiral	P-31–P-36	6	Forward $c \in \{0.3, 0.6, 0.9\} \times$ yaw ramp to $\pm 0.8 \bar{u}^\psi$ over 20 s	Cross-axis coupling between speed and turning
Disturbance	P-37–P-42	6	Forward hold 0.5; 5 kg pendulum, longitudinal and lateral strikes $\times$ drop heights 0.1, 0.2, 0.3 m	Recovery time after external perturbation

Saturation envelope. On a saturation ramp, the measured threshold u_{sat}^j is the smallest command at which the ratio y^j/u^j , low-pass filtered with a 0.5 s moving average, falls below 0.9 and remains below it for at least 1 s. The saturation envelope of a unit is the set of these thresholds over axes and terrains. A characterization predicts a threshold $\hat{u}_{\text{sat}}^j$; the reported metric is the mean absolute threshold error $E_{\text{sat}} = \text{mean} |\hat{u}_{\text{sat}}^j - u_{\text{sat}}^j|$ in m/s over the saturation probes.

Recovery time. For a disturbance at time t_d , the recovery time is

$$\tau_{\text{rec}} = \min \{ \tau \geq 0 : e_t \leq \eta \quad \forall t \in \mathcal{I}_\tau \}, \quad (7)$$

where $\mathcal{I}_\tau = [t_d + \tau, t_d + \tau + 0.5 \text{ s}]$ and the tolerance is $\eta = 0.1 \text{ m/s}$. The reported value R_{95} is the 95th percentile over disturbance trials.

4.3 Evaluation procedure

Each sequence is executed three times per platform. Characterizations are fitted and scored with leave-one-repetition-out cross-validation: a characterization is fitted on two repetitions of the fitting sequences and scored by compensated replay against the references of the held-out repetition, and the three folds are pooled before percentiles are taken. For compensated replay the reference trajectories $\mathbf{r}$ are the probe sequences themselves, so the evaluation exercises the same five families that define the suite. Because a characterization that has seen a probe family is expected to compensate it better, the ablation and budget sweep in Section 5 vary only the sequences used for fitting; scoring always uses the held-out repetition of the full 42-sequence suite.

4.4 Reference characterization model and fitting

A characterization needs a parametric form that can be inverted for compensated replay. We use a per-axis first-order-plus-dead-time model with saturation, rate limiting, and bias, which is compatible with the model class used by GAUGE-style calibration [13] and standard in identification practice [34]. For axis j ,

$$\tau_j \dot{y}^j(t) + y^j(t) = k_j \sigma_j(u^j(t - d_j)) + b_j, \quad (8)$$

where k_j is the static gain, τ_j the time constant, d_j the dead time, b_j a constant bias, and σ_j the saturation-and-rate-limit nonlinearity, which clips its argument to $[-\hat{u}_{\text{sat}}^j, \hat{u}_{\text{sat}}^j]$ and limits its rate of change to $\pm a_j$ per second. Exact discretization at period Δt gives

$$\hat{y}_{t+1}^j = \alpha_j \hat{y}_t^j + (1 - \alpha_j) \left(k_j \sigma_j(u_{t-n_j}^j) + b_j \right), \quad (9)$$

where $\alpha_j = e^{-\Delta t/\tau_j}$ and the integer delay is $n_j = \text{round}(d_j/\Delta t)$. Cross-axis coupling, which the spiral probes excite, is represented by an additive term $\kappa u_t^x u_t^\psi$ on the lateral axis, capturing the lateral drift that many quadruped controllers exhibit when turning at speed. The parameter vector ϕ for one configuration therefore collects $k_j, \tau_j, d_j, b_j, \hat{u}_{\text{sat}}^j$, and a_j for each axis $j \in \{x, y, \psi\}$ together with κ , 19 parameters in total.

Fitting. The integer delays are found by grid search over $n_j \in \{0, \dots, 15\}$, that is 0 to 150 ms. For each delay candidate, the remaining parameters minimize the regularized simulation error

$$\hat{\phi} = \arg \min_{\phi} \sum_{t \in \mathcal{T}_{\text{fit}}} \|\mathbf{W}(\mathbf{y}_t - \hat{\mathbf{y}}_t(\phi))\|_2^2 + \gamma \|\phi - \phi_0\|_2^2, \quad (10)$$

solved with Levenberg–Marquardt, where $\hat{\mathbf{y}}_t(\phi)$ is the free-run simulation of the discrete model driven by the logged commands, ϕ_0 is a prior centered on unit gain, zero bias, a 0.15 s time constant, and the nominal envelope $\bar{\mathbf{u}}$, and $\gamma = 10^{-3}$ is a small ridge weight that keeps unexcited parameters at their prior. Simulation error rather than one-step prediction error is used because a planner rolls the model forward over a horizon; one-step error systematically underweights the time constant. The saturation thresholds are initialized from the saturation probes by the procedure of Section 4.2 before joint refinement.

Inversion for compensated replay. Because the reference is known in advance during replay, the model can be inverted with a preview of $n_j + 1$ steps. Away from saturation,

$$\hat{u}_t^j = \frac{1}{k_j} \left(\frac{r_{t+n_j+1}^j - \alpha_j r_{t+n_j}^j}{1 - \alpha_j} - b_j \right), \quad (11)$$

which is subsequently clipped to the predicted envelope and rate limit. The preview requirement is not restrictive in practice: planners that issue velocity commands typically hold a short horizon of intended motion, and MPPI-style controllers [12] hold one explicitly. When the inverse exceeds the predicted envelope, the clipped command makes the realized motion lag the reference; the saturation-threshold error therefore propagates directly into fidelity and recovery time, which is why E_{sat} is reported separately.

GAUGE-style reference calibration. As the reference characterization method we implement GAUGE-style active calibration [13] with the same model class: probe commands are generated around the command distribution of a nominal waypoint-following planner, and the model is updated online as data arrive. The reference characterization therefore differs from the

Table 2: Reporting template: fields that accompany every protocol result.

Field	Content
Unit	Platform model, unit identifier, service history
Firmware	Locomotion software and version
Gait mode	Active mode per sequence and nominal envelope $\bar{u}$
Payload	Mass and mounting location
Terrain	Lab floor or stair course, per sequence
Estimator	Source of realized twist y
Suite	Sequences used for fitting and for scoring; repetitions
Metrics	F , Λ_{95} , B , E_{sat} , R_{95}
Anomalies	Flagged events with probe identifier; re-runs kept

protocol characterization in its excitation (planner-conditioned rather than fixed) and in where it was obtained, not in model form. This isolates the effect of the excitation and of per-unit fitting, which is what the protocol is meant to measure.

4.5 Reporting template

A metric value is only interpretable together with the configuration c under which it was measured. The template (Table 2) lists the fields that must accompany every result. Fields that are unknown must be reported as unknown rather than omitted.

REPRODUCIBILITY NOTE

Probe parameters: commands and estimated twist logged at 100 Hz ($\Delta t = 10$ ms); 2 s zero padding before and after every sequence; amplitudes as fractions of the gait-mode nominal envelope $\bar{u}$. Step: axes x, y, ψ ; amplitudes 0.25, 0.5, 0.75, 1.0; profile 4 s hold at $+A$, 2 s zero, 4 s hold at $-A$. Sweep: log chirp 0.1 to 2.0 Hz over 40 s; amplitudes 0.2, 0.4, 0.6; axes x, y, ψ . Saturation: ramp rates $\rho = 0.1, 0.2, 0.4 \bar{u}^j/\text{s}$; forward axis on lab floor to $1.5 \bar{u}^x$, forward axis on stair course to $1.0 \bar{u}^x$, lateral axis on lab floor to $1.5 \bar{u}^y$. Spiral: $c = 0.3, 0.6, 0.9$; yaw ramp to $\pm 0.8 \bar{u}^\psi$ over 20 s. Disturbance: $u^x = 0.5 \bar{u}^x$; 5 kg pendulum impactor; longitudinal and lateral strikes; drop heights 0.1, 0.2, 0.3 m. Metric constants: $\ell = 0.4$ m; lag windows 2 s, 1 s overlap, search 0 to 300 ms; plateau = final 1 s of each hold; saturation ratio 0.9 for 1 s; recovery tolerance $\eta = 0.1$ m/s held for 0.5 s. Three repetitions per sequence per platform, leave-one-repetition-out scoring. The probe suite is released with sequence identifiers P-01 to P-42 as listed in Table 1.

05 Experiments

5.1 Platforms and setup

The testbed consists of two ANYmal B units, designated AMB-1 and AMB-2, and one Unitree Go2. The two ANYmal B units run identical firmware but have different service histories; AMB-2 is designated the reference unit. The Go2 runs its vendor locomotion software, recorded as GoW v3.4. Experiments were conducted on the lab floor and on a stair course. The full probe suite of 42 command sequences (Table 1) was replayed on each platform with logging at 100 Hz and three repetitions per sequence per platform, giving $42 \times 3 = 126$ scheduled sequence executions per platform and 378 across the three platforms; one additional re-run on AMB-2 is described in Section 5.5. The realized body twist was taken from each platform’s onboard state estimator, as recorded in the reporting template.

5.2 Main comparison

Table 3 reports the five protocol metrics for three characterizations, scored with the held-out compensated replay of Section

4.3. The first two rows are reference characterizations obtained with GAUGE-style active calibration and labeled, as in the run-server record, by the unit from which they were derived: the reference unit AMB-2, and the Go2 running GoW v3.4. They represent the common practice of characterizing one representative unit and reusing the result. The third row, labeled Ours, is the per-unit characterization fitted on the full 42-sequence suite.

The full-suite characterization attains the lowest value on all five metrics. Command fidelity error is 8.4% RMS against 9.0% for the AMB-2 reference characterization and 9.6% for the Go2 (GoW v3.4) reference, reductions of 0.6 and 1.2 percentage points. The 95th-percentile phase lag is 62 ms against 66 and 65 ms, steady-state bias is 3.1% against 3.6% and 3.3%, saturation threshold error is 0.14 m/s against 0.16 and 0.15 m/s, and 95th-percentile recovery time is 1.9 s against 2.1 s for both references. These margins are consistent in direction but modest in size, and because the record reports pooled values without per-fold dispersion, they should not be read as statistically established differences. The more robust conclusion is qualitative: a fixed, broad excitation fitted per unit is at least as informative as a planner-conditioned calibration transferred from a representative unit, and the protocol is sensitive enough to resolve differences of this size between characterizations.

The phase-lag values also give a sense of scale for planners. A 95th-percentile lag of 62 to 66 ms corresponds to six to seven command periods at 100 Hz, so a planner that assumes instantaneous execution is wrong by several steps at the tail even after compensation.

5.3 Ablation over probe families

Table 4 varies the sequences used for fitting while scoring on the full held-out suite. Fitting on step probes only, the 12 sequences P-01 to P-12, raises fidelity error to 11.3%, 2.9 percentage points above the full suite. Steps identify gain, delay, and bias well but leave the frequency response, cross-axis coupling, and envelope weakly constrained, so the compensated commands err on chirps, spirals, and ramps. The reduced suite of 18 sequences, a fixed quick-check subset containing six steps, three sweeps, three saturation ramps, three spirals, and three disturbance probes, recovers most of the gap at 9.7%, still 1.3 percentage points above the full suite.

5.4 Probe budget sweep

Table 5 sweeps the number of fitting sequences. Subsets are nested and grow in a fixed round-robin order across the five families, so every budget includes every family and each larger subset contains the smaller ones; the 18-sequence subset of the sweep is therefore distinct from the hand-selected reduced suite of Table 4, which explains the small difference between their fidelity values (9.6% and 9.7%).

Both metrics improve monotonically with budget. Fidelity error falls from 17.8% at 6 sequences to 8.4% at 42, a reduction of 9.4 percentage points, and the 95th-percentile phase lag falls from 118 to 62 ms. The gains are concentrated at small budgets: moving from 6 to 18 sequences removes 8.2 of the 9.4 points, while moving from 27 to 42 removes 0.5 points and 3 ms of lag. For routine regression checks, for example after a firmware update, a budget of 18 to 27 sequences is therefore a reasonable compromise; for a characterization that will be published or used to compare platforms, the protocol specifies the full suite.

Table 3: Main comparison. Lower is better for all metrics. Ours denotes the per-unit characterization fitted on the full 42-sequence suite.

Method	Command fidelity RMS (%)	Phase lag p95 (ms)	Steady-state bias (%)	Saturation threshold error (m/s)	Recovery time p95 (s)
AMB-2 (reference unit)	9.0	66	3.6	0.16	2.1
Unitree Go2 (GoW v3.4)	9.6	65	3.3	0.15	2.1
Ours	8.4	62	3.1	0.14	1.9

Table 4: Ablation over the probe families used for fitting.

Variant	Command fidelity RMS (%)
Full probe suite (42 sequences)	8.4
Reduced suite (18 sequences)	9.7
Step-only probes	11.3

Table 5: Probe budget sweep (nested subsets, all families represented).

Probe budget (sequences)	Command fidelity RMS (%)	Phase lag p95 (ms)
6	17.8	118
12	12.4	91
18	9.6	70
27	8.9	65
42	8.4	62

5.5 Recorded failure cases

The protocol is also intended to surface behavior that a parametric characterization does not capture. Three events were recorded.

Gait-mode transient on AMB-1. When entering the stairs gait, AMB-1 exhibits a 0.4 s mode-switch transient that neither reference model captures; it is logged as the P-27 anomaly. P-27 is the fastest forward saturation ramp on the stair course. The transient is a direct instance of the stateful behavior formalized in Eq. (1): the response depends on a discrete internal mode that changes during the sequence. A 0.4 s transient corresponds to 40 command periods, far beyond the 62 to 66 ms tail lag of the continuous model. Because AMB-1 and AMB-2 run identical firmware, the event also illustrates why the protocol characterizes units individually and records service history.

Payload-dependent saturation on the Go2. Under an 8 kg payload, Go2 saturation appears 0.3 m/s earlier than its datasheet suggests. This is twice the 0.15 m/s saturation threshold error of the Go2 reference characterization in Table 3 and illustrates why the protocol measures the envelope under the reported payload rather than reading it from documentation; the template’s payload field exists for exactly this reason.

Sensor dropout on AMB-2. One spiral probe on AMB-2 was affected by an encoder dropout, recorded in the run-server log as a wheel-encoder dropout. The trial was re-run, and both runs are kept in the record. Keeping both runs, flagged, rather than silently replacing the faulty one, is the protocol’s prescribed handling: a dropout is a property of the system under test as deployed, and discarding it would bias the record toward ideal conditions.

06 Limitations

The demonstration covers three units of two platform models, two terrains, and one stair course; it is sufficient to exercise every part of the protocol but not to establish population-level statements about either platform. The record reports pooled metric values without per-fold dispersion, so the differences in Table 3 are reported as observed and not as statistically tested. Realized twist is taken from each platform’s onboard estimator,

so estimator bias is confounded with interface error; an external reference such as motion capture would separate the two and is recommended where available. The reference model class is per-axis first order with a single bilinear coupling term, which by construction cannot represent discrete mode switches such as the AMB-1 transient; the protocol detects such events but the reference model does not explain them. The GAUGE-style reference is our implementation within a shared model class, not the original authors’ code, and its numbers should be read as characterizing that implementation. Finally, the nominal envelope used to scale probe amplitudes is taken from vendor documentation, which the Go2 payload observation shows can be optimistic; a conservative first pass is advisable on unfamiliar platforms.

07 Conclusion

Velocity commands are the narrow waist of legged-robot autonomy, yet the controllers behind them are opaque, stateful, and unit-specific. This paper proposed a benchmark protocol for characterizing such interfaces: a fixed 42-sequence probe suite in five families, five metrics with precise definitions, a compensated-replay evaluation procedure, and a reporting template. On two ANYmal B units and a Unitree Go2, the full-suite characterization reached 8.4% command fidelity error, 62 ms tail phase lag, 3.1% steady-state bias, 0.14 m/s saturation threshold error, and 1.9 s tail recovery time, slightly better on every metric than GAUGE-style reference characterizations taken from a single representative unit. The probe-budget sweep shows that most of the benefit is obtained with 18 to 27 sequences, and the recorded anomalies show the protocol surfacing gait-mode transients, payload-dependent saturation, and sensor faults that a single characterization misses. The probe suite and its exact parameters are released so that other groups can report velocity interfaces in the same terms. Future work will extend the reference model to hybrid, mode-switching dynamics and add an external motion-capture reference.

References

- [1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.
- [2] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim. Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control. arXiv:1909.06586, 2019.
- [3] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter. Dynamic Locomotion Through Online Nonlinear Motion Optimization for Quadrupedal Robots. IEEE Robotics and Automation Letters, 3(3), 2018.
- [4] 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.
- [5] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.

- [6] 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.
- [7] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [8] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization. IEEE Robotics and Automation Letters, 3(4), 2018.
- [9] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. arXiv:2303.01420, 2023.
- [10] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [11] 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.
- [12] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information Theoretic MPC for Model-Based Reinforcement Learning. IEEE International Conference on Robotics and Automation (ICRA), 2017.
- [13] T. Zang, H. Huang, S. Cheng, and W. Zhang. GAUGE: Planner-Conditioned Active Calibration of Opaque Quadruped Velocity Interfaces. arXiv:2607.02841, 2026.
- [14] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger. ANYmal – A Highly Mobile and Dynamic Quadrupedal Robot. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- [15] 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.
- [16] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [17] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid Locomotion via Reinforcement Learning. arXiv:2205.02824, 2022.
- [18] L. Smith, I. Kostrikov, and S. Levine. A Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning. arXiv:2208.07860, 2022.
- [19] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [20] G. Ji, J. Mun, H. Kim, and J. Hwangbo. Concurrent Training of a Control Policy and a State Estimator for Dynamic and Robust Legged Locomotion. arXiv:2202.05481, 2022.
- [21] I. M. A. Nahrendra, B. Yu, and H. Myung. DreamWaQ: Learning Robust Quadrupedal Locomotion With Implicit Terrain Imagination via Deep Reinforcement Learning. arXiv:2301.10602, 2023.
- [22] G. B. Margolis and P. Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. arXiv:2212.03238, 2022.
- [23] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. RLOC: Terrain-Aware Legged Locomotion using Reinforcement Learning and Optimal Control. arXiv:2012.03094, 2020.
- [24] Z. Fu, X. Cheng, and D. Pathak. Deep Whole-Body Control: Learning a Unified Policy for Manipulation and Locomotion. arXiv:2210.10044, 2022.
- [25] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning Vision-Guided Quadrupedal Locomotion End-to-End with Cross-Modal Transformers. arXiv:2107.03996, 2021.
- [26] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.
- [27] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [28] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. arXiv:2309.05665, 2023.
- [29] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [30] M. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. arXiv:2305.08510, 2023.
- [31] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [32] H. Xi, M. Tan, X. Zhang, S. Cheng, S. Wang, Y. Gu, X. Shen, and W. Zhang. CeRLP: A Cross-embodiment Robot Local Planning Framework for Visual Navigation. arXiv:2603.19602, 2026.
- [33] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. Agni-Nav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [34] L. Ljung. System Identification: Theory for the User, 2nd ed.. Prentice Hall, 1999.
- [35] R. Pintelon and J. Schoukens. System Identification: A Frequency Domain Approach, 2nd ed.. Wiley-IEEE Press, 2012.
- [36] 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.
- [37] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle. Contact-Aided Invariant Extended Kalman Filtering for Legged Robot State Estimation. arXiv:1805.10410, 2018.
- [38] D. Wisth, M. Camurri, and M. Fallon. VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots. arXiv:2107.07243, 2021.
- [39] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols. arXiv:1502.03143, 2015.
- [40] P. Anderson et al.. On Evaluation of Embodied Navigation Agents. arXiv:1807.06757, 2018.
- [41] N. Yokoyama, S. Ha, and D. Batra. Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation. arXiv:2103.08022, 2021.
- [42] A. Kadian et al.. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? arXiv:1912.06321, 2019.