

Predictive Contact-Aware Path Following for Agile Quadrupeds in Narrow Passages

Felix Wagner • Ananya Iyer • Meilin Jiang • Carlos Ramirez

Legged Robotics Lab, Stanford University, Stanford, CA, USA

PUBLISHED

July 30, 2026

REGISTERED

August 4, 2026

RECORD TYPE

Preprint

ABSTRACT

In narrow passages, the contact sequence of a quadruped's legs and the lag between commanded and realized body velocity decide whether the robot clears the gap, yet most path followers treat the body as a point with ideal velocity response. This paper presents a predictive contact-aware path follower that couples a calibrated model of the opaque velocity interface, obtained with GAUGE, with the configuration-based motion prediction of AgniNav. For each candidate command sequence, the follower predicts the realized body trajectory, the gait phase, and the swept leg envelope over a 0.5-1.0 s horizon; it schedules body poses and contact phases so that the legs nearest the passage boundary enter in a favorable phase, and it rejects passage entries whose predicted trajectory cannot clear the gap within the clearance and time budgets. The approach is evaluated in an Isaac Lab narrow-gate suite with doorway-scale gaps of 55-90 cm, rope barriers, and pipe runs, and in Unitree Go2 trials through a 72 cm doorway-scale gap and two rope barriers at setpoints of 1.0-3.0 m/s. Compared with point-body path following under an ideal velocity assumption, the method raises first-attempt success from 75.5 to 84.6 percent and shortens time to pass from 3.2 to 2.8 s, at the cost of a smaller mean clearance margin of 13.2 cm versus 15.1 cm. Removing contact-phase scheduling, future-range rejection, or the calibrated interface model reduces success to between 72.4 and 74.4 percent.

INDEX TERMS quadruped locomotion, narrow passages, path following, velocity interface calibration, contact-phase scheduling, model predictive control, Unitree Go2

KAGG NOTE A quadruped clears a narrow passage when its realized motion and leg contacts, not its commanded velocity, fit the gap; a path follower should therefore predict both and refuse entries it cannot clear.

01 Introduction

Quadrupedal robots are increasingly deployed in buildings, plants, and field sites where the route is interrupted by doorways, gaps between equipment, rope cordons, and pipe runs. In such places the free width of the passage is often only a few tens of centimeters larger than the body of the robot. Whether the robot passes on the first attempt is then decided by details that a conventional navigation stack abstracts away: the realized body velocity lags the command issued to the locomotion controller, and the legs sweep laterally outside the trunk footprint during swing. A foot in mid-swing next to a door frame, or a body that is still yawing because the velocity controller has not yet settled, can turn a geometrically feasible path into a collision.

Most path followers used on legged robots inherit the assumptions of wheeled navigation. The robot is modeled as a point or a rigid footprint, and the commanded twist is assumed to be realized instantaneously [1]. Learned locomotion controllers [2, 3, 4, 5] make the velocity interface highly capable, but on commercial platforms they are typically opaque: the navigation layer sends a twist and observes the outcome, without access to the internal gait schedule or the transient response. Recent work has begun to close this gap from two directions. GAUGE [6] actively calibrates the command-to-motion response of an opaque quadruped velocity interface online, and AgniNav [7] plans locally over an abstract robot con-

figuration that encodes body geometry, sensors, and velocity limits, so that a single planning stack predicts motion for different embodiments. Neither, however, reasons about the contact sequence of the legs or decides explicitly whether a narrow passage should be entered at all.

This paper combines the two into a predictive path follower for narrow passages. The calibrated interface model predicts the realized body twist from the commanded twist; the configuration-based predictor converts that twist into body poses and a swept body-and-leg envelope; a gait-phase model predicts which legs are in stance or swing when the body crosses the narrowest section. The follower optimizes command sequences over a 0.5–1.0 s horizon to schedule body poses and contact phases jointly, and it rejects passage entries whose predicted trajectory cannot clear the passage within a clearance and time budget.

The contributions are as follows.

1. A formulation of narrow-passage path following for quadrupeds with opaque velocity interfaces in which clearance is evaluated on the predicted realized trajectory and on a phase-dependent leg envelope rather than on a point or rigid footprint.
2. A predictive follower that composes a GAUGE-calibrated interface model with AgniNav-style configuration-based motion prediction and schedules contact phases at passage entry by modulating the approach speed.

3. A future-range entry rejection test that refuses passage entries whose predicted trajectory violates a clearance margin or a time-to-pass budget, before the robot commits to the passage.
4. An evaluation in an Isaac Lab narrow-gate suite and on a Unitree Go2 that reports first-attempt success, clearance margin, time to pass, and correct entry rejections against three path-following baselines, together with an ablation and an analysis of recorded failure cases.

02 Related Work

2.1 Learned and model-based quadruped locomotion

Reinforcement learning has produced robust velocity-tracking controllers for quadrupeds. Hwangbo et al. trained policies in simulation with a learned actuator network and transferred them to ANYmal [2]; Lee et al. used a teacher-student scheme to traverse challenging terrain from proprioception alone [3]; RMA added an adaptation module that infers environment extrinsics online [8]; and massively parallel training in GPU simulation reduced training to minutes [4]. Miki et al. fused exteroception and proprioception for robust perceptive locomotion in the wild [5]. Walk These Ways exposed a family of gait parameters to the operator [9], and Rapid Locomotion pushed velocity tracking toward the speed limits of small quadrupeds [10]. Early sim-to-real work on the Minitaur established the role of accurate actuator and latency modeling [11]. These controllers define the velocity interface that our path follower commands. We do not modify them; instead, we model their closed-loop response, which on commercial platforms such as the Go2 is the only accessible abstraction.

Model-based controllers make the gait schedule explicit. Farshidian et al. formulated real-time motion planning for legged robots as sequential linear-quadratic model predictive control [12]; Kim et al. combined whole-body impulse control with convex MPC for highly dynamic gaits on the MIT Mini Cheetah [13]; and Grandia et al. added perceptive foothold constraints to nonlinear MPC [14]. DTC combines a learned tracking policy with optimized footholds [15]. These methods schedule contacts precisely, but they assume control at the joint or whole-body level. Our setting is the opposite: the contact schedule is internal to an opaque controller, and the navigation layer can influence it only indirectly, through the timing and magnitude of commanded twists. We borrow from this literature the Raibert-style foothold heuristic and the notion of a phase-dependent swing envelope, and use them for prediction rather than for actuation.

2.2 Perceptive and agile navigation for legged robots

A second line of work learns navigation and locomotion jointly. Rudin et al. trained end-to-end policies that combine local navigation with locomotion skills such as climbing and jumping [16], and ANYmal Parkour composed perception, a high-level navigation module, and a skill library to traverse obstacle courses at speed [17]. Robot Parkour Learning [18] and Extreme Parkour [19] learned agile skills from depth images, including tilting the body to pass narrow gaps; Agarwal et al. learned locomotion from egocentric vision [20]. These results show that legged robots can physically squeeze through tight openings, but the behavior is embedded in a learned policy trained for a specific robot and does not expose an explicit, inspectable decision about whether an opening should

be attempted. Fu et al. coupled vision and proprioception for navigation and used proprioceptive feedback to correct the planner’s model of the walking policy [21]; Truong et al. learned robot embeddings that let a navigation policy adapt to the dynamics of a legged base [22]. Both recognize that the navigation layer must model the locomotion layer, which is the premise of our work; we make that model explicit and calibrated, and use it for clearance prediction in a time-critical setting.

Traversability and mapping methods supply the geometry on which such decisions are made. Frey et al. learned traversability from locomotion-policy experience on volumetric representations [23], and GPU elevation mapping provides dense, real-time terrain maps for legged navigation [24]. Our follower consumes a local occupancy representation of the passage built from range data; any of these mapping back ends could provide it.

2.3 Local planning and embodiment-aware navigation

Classical local planners, such as those in the Nav2 stack [1], plan over a footprint and dynamic window. Sampling-based model predictive path integral control (MPPI) [25] optimizes over stochastic rollouts of a dynamics model and is a natural fit for non-smooth costs such as clearance to a gate edge; we use it as the optimizer of our follower. Learned navigation models such as BADGR [26], GNM [27], ViNT [28], and No-MaD [29] produce waypoints or actions from images across several robots, but delegate precise collision clearance to a downstream controller.

AgniNav [7] is the closest work on the planning side. It is a configuration-driven local planner: rather than hard-coding a robot model, it operates on an abstract configuration that specifies body geometry, sensor placement, and velocity and acceleration limits, so that the same planning stack can run on wheeled, legged, and other embodiments by swapping the configuration. Candidate motions are predicted from commanded velocities under that configuration and checked against the sensed environment. We adopt this configuration-based prediction as the backbone of our motion predictor and as the source of body geometry. We differ from AgniNav in three respects: we replace its nominal velocity-limit model with a calibrated, lag-aware interface model; we extend the configuration with a phase-dependent leg envelope, which matters for quadrupeds in passages narrower than about three body widths; and we add an explicit entry-rejection decision with a time budget, whereas AgniNav selects the best feasible local motion at each cycle.

2.4 Modeling and calibrating velocity interfaces

The mismatch between commanded and realized velocity is usually handled implicitly, through conservative inflation of the footprint or through robust tracking. GAUGE [6] addresses it directly. It treats the quadruped velocity controller as an opaque interface, issues planner-conditioned probe commands, that is, commands drawn from the distribution of twists that the downstream planner actually uses, and fits a calibrated model of the command-to-motion response online, including lag, gain, saturation, and cross-axis coupling. The planner conditioning matters because opaque locomotion controllers are strongly nonlinear, and a model fit on generic excitation can be inaccurate in exactly the regime the planner exercises. We use GAUGE as the source of our interface model and extend its use in two ways: we propagate the model’s residual uncertainty

along the horizon to inflate clearance, and we use the calibration data to estimate the dependence of gait frequency on commanded speed, which the contact-phase predictor requires. GAUGE itself does not address passage clearance or contact timing.

2.5 Safety filtering and sim-to-real evaluation

Control barrier functions provide a principled way to enforce set invariance on top of a nominal controller [30]. Our clearance constraint plays a similar role but is evaluated on predicted trajectories of a black-box closed loop, where the barrier-function conditions cannot be certified; we therefore enforce it as a hard constraint on sampled rollouts and combine it with an entry-rejection rule. Sim-to-real transfer of locomotion relies on domain and dynamics randomization [31, 32] in GPU-parallel simulators such as Isaac Gym [33] and Orbit, now Isaac Lab [34]. Kadian et al. showed that simulation rankings do not necessarily predict real-world performance [35], and Yokoyama et al. argued for time-aware navigation metrics [36]. Accordingly, we report time to pass alongside success and evaluate on hardware as well as in simulation.

03 Problem Formulation

3.1 Robot, interface, and passage

Let $\mathbf{x}_k = (p_k^x, p_k^y, \psi_k) \in SE(2)$ denote the planar body pose at discrete time $t_k = k \Delta t$ in a gravity-aligned local odometry frame, and let $\boldsymbol{\nu}_k = (v_k^x, v_k^y, \omega_k)$ be the realized body twist expressed in the body frame. The navigation layer issues a commanded twist $\mathbf{u}_k = (u_k^x, u_k^y, u_k^\omega)$ to the onboard locomotion controller at period $\Delta t = 20$ ms. The controller is opaque: the map from $\mathbf{u}_{0:k}$ to $\boldsymbol{\nu}_k$ and to the internal gait schedule is unknown, but body odometry and binary foot-contact states $\mathbf{s}_k \in \{0, 1\}^4$ are observable.

The robot is described by a configuration $\mathcal{C}$ in the sense of AgniNav [7]: trunk length L , width W , and height H ; hip positions $\mathbf{h}_j$ for legs $j \in \{\text{FL}, \text{FR}, \text{RL}, \text{RR}\}$; foot radius r_f ; and velocity limits $\bar{\boldsymbol{\nu}}$. A passage $\mathcal{P}$ is a set of obstacle surfaces (door jambs, rope lines, pipes) together with an entry plane Π_{in} and an exit plane Π_{out} . A reference path $\gamma : [0, S] \rightarrow \mathbb{R}^2$ passes through both planes.

3.2 Objective

Given a speed setpoint v_{ref} , the task is to choose $\mathbf{u}_k$ so that the robot passes from Π_{in} to Π_{out} on the first attempt without contact between the robot and $\mathcal{P}$, within a time budget T_b , or to decline the entry before crossing Π_{in} when such a passage is not achievable. Let $\mathcal{E}(\mathbf{x}, \phi; \mathcal{C}) \subset \mathbb{R}^3$ denote the volume occupied by the trunk and legs at pose $\mathbf{x}$ and gait phase ϕ . The clearance margin at time k is

$$m_k = \min_{\mathbf{q} \in \mathcal{P}} \text{sd}(\mathbf{q}, \mathcal{E}(\mathbf{x}_k, \phi_k; \mathcal{C})), \quad (1)$$

where sd is the signed distance, positive outside the envelope. A passage is *cleared* if $m_k > 0$ for all k between entry and exit, and the time to pass T_{pass} is measured from the trigger point in front of Π_{in} to the instant the rear of the trunk crosses Π_{out} . An entry rejection is *correct* if the passage could not have been cleared within T_b at the commanded setpoint.

The difficulty is that m_k depends on the realized poses $\mathbf{x}_k$ and phases ϕ_k , neither of which is commanded directly. The follower must therefore predict them from the commanded sequence.

04 Method

4.1 Overview

The follower runs a receding-horizon loop at 50 Hz. At each cycle it (i) updates the calibrated interface model and the gait-phase estimate, (ii) samples candidate command sequences over a horizon of N steps, (iii) predicts for each sequence the realized twist, body poses, gait phases, and swept envelope, (iv) evaluates path-tracking, contact-phase, and clearance terms, and (v) either executes the first command of the weighted optimum or, while the robot is still in front of the passage, rejects the entry.

4.2 Calibrated interface model

Following GAUGE [6], the response of the opaque controller is modeled per axis $i \in \{x, y, \omega\}$ as a delayed, saturated first-order system with cross-axis coupling:

$$\begin{aligned} \nu_{k+1}^i = & \nu_k^i + \frac{\Delta t}{\tau_i} \left(\sigma_i(g_i u_{k-d_i}^i \right. \\ & \left. + \sum_{l \neq i} c_{il} u_{k-d_l}^l) - \nu_k^i \right), \end{aligned} \quad (2)$$

where τ_i is the time constant, d_i the dead time in control periods, g_i the steady-state gain, c_{il} the coupling from axis l to axis i , and $\sigma_i(\cdot) = \text{clip}(\cdot, -\bar{\nu}_i, \bar{\nu}_i)$ the saturation. The parameter vector $\boldsymbol{\theta} = \{\tau_i, d_i, g_i, c_{il}, \bar{\nu}_i\}$ and the one-step residual covariance $\boldsymbol{\Sigma}_\nu$ are fitted by GAUGE from planner-conditioned probes and refined online by recursive least squares on executed commands. We use a first-order model because the closed-loop response of learned velocity controllers is dominated by a lag and a dead time, and a richer model would be poorly identified from the short probe sequences that GAUGE issues.

The body pose is propagated by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{R}(\psi_k) \boldsymbol{\nu}_k, \quad (3)$$

where $\mathbf{R}(\psi) = \text{diag}(\mathbf{R}_2(\psi), 1)$ and $\mathbf{R}_2(\psi)$ is the planar rotation by ψ . Uncertainty is propagated to first order: with $\mathbf{z}_k = (\mathbf{x}_k, \boldsymbol{\nu}_k)$ and $\mathbf{A}_k = \partial \mathbf{z}_{k+1} / \partial \mathbf{z}_k$,

$$\boldsymbol{\Sigma}_{k+1} = \mathbf{A}_k \boldsymbol{\Sigma}_k \mathbf{A}_k^\top + \mathbf{Q}, \quad \mathbf{Q} = \text{diag}(\mathbf{0}_3, \boldsymbol{\Sigma}_\nu), \quad (4)$$

and $\sigma_k = \sqrt{\mathbf{n}_k^\top \boldsymbol{\Sigma}_k \mathbf{n}_k}$ is the predicted position standard deviation along the unit normal $\mathbf{n}_k$ of the nearest passage surface, with $\boldsymbol{\Sigma}_k^p$ the position block of $\boldsymbol{\Sigma}_k$.

4.3 Gait-phase prediction

The onboard controller of the Go2 uses a trot in the speed range considered here. We represent its schedule by a scalar phase $\phi_k \in [0, 1)$ with leg offsets $o_{\text{FL}} = o_{\text{RR}} = 0$ and $o_{\text{FR}} = o_{\text{RL}} = 0.5$, so that leg j has phase $\phi_k^j = (\phi_k + o_j) \bmod 1$ and is in stance when $\phi_k^j < \beta$, with duty factor β . The phase advances as

$$\phi_{k+1} = (\phi_k + \Delta t f(\|\mathbf{v}_k\|)) \bmod 1, \quad f(v) = f_0 + f_1 v, \quad (5)$$

where $\mathbf{v}_k = (v_k^x, v_k^y)$. The coefficients f_0, f_1 and β are estimated from the foot-contact signals recorded during GAUGE calibration, and ϕ_k is re-anchored at every observed touchdown of the front-left leg. Because the stride frequency depends on the realized speed, the navigation layer can shift the phase at which the trunk reaches the passage by modulating the commanded speed; this is the only lever on the contact schedule that an opaque interface leaves to the planner.

4.4 Body and leg envelope

The trunk is a box $\mathcal{B}(\mathbf{x}; L, W, H)$ taken from the configuration $\mathcal{C}$. For each leg, the touchdown foothold is predicted with the Raibert heuristic

$$\mathbf{p}_j^{\text{td}} = \mathbf{h}_j(\mathbf{x}_{\text{td}}) + \frac{\beta}{2f} \mathbf{v}_{\text{td}} + k_r(\mathbf{v}_{\text{td}} - \mathbf{u}_{\text{td}}^{xy}), \quad (6)$$

where $\mathbf{h}_j(\mathbf{x})$ is the hip position in the world frame, $\mathbf{v}_{\text{td}}$ and $\mathbf{u}_{\text{td}}^{xy}$ are the predicted realized and commanded planar velocities at touchdown, and k_r is a feedback gain. The correction term captures the controller's tendency to step wider and longer when the realized velocity lags the command, which is exactly the situation in which lag causes lateral contact. During swing, with normalized swing progress $s = (\phi^j - \beta)/(1 - \beta)$, the foot moves from the lift-off to the touchdown foothold with an outward lateral excursion and apex height

$$\mathbf{p}_j(s) = (1 - s) \mathbf{p}_j^{\text{lo}} + s \mathbf{p}_j^{\text{td}} + \sin(\pi s)(\delta_\ell \mathbf{e}_j^\perp + h_{\text{sw}} \mathbf{e}_z), \quad (7)$$

where $\mathbf{e}_j^\perp$ is the outward lateral unit vector of leg j , δ_ℓ the lateral excursion, and h_{sw} the swing apex height. The envelope is

$$\mathcal{E}(\mathbf{x}_k, \phi_k; \mathcal{C}) = \mathcal{B}(\mathbf{x}_k) \cup \bigcup_j \mathbb{B}(\mathbf{p}_j(\phi_k^j), r_f) \cup \overline{\mathbf{h}_j \mathbf{p}_j}, \quad (8)$$

the union of the trunk, a sphere of radius r_f at each foot, and the segment from hip to foot, which conservatively covers the leg links. The predicted margin is then the risk-adjusted version of (1),

$$\hat{m}_k = \min_{\mathbf{q} \in \hat{\mathcal{P}}_k} \text{sd}(\mathbf{q}, \mathcal{E}(\hat{\mathbf{x}}_k, \hat{\phi}_k; \mathcal{C})) - \kappa \sigma_k, \quad (9)$$

where $\hat{\mathcal{P}}_k$ are the passage surfaces in the local range map and κ sets the confidence level. Evaluating (9) along the predicted trajectory is what we call the *future-range* test, as opposed to a *present-range* test that evaluates distances only at the current pose.

4.5 Predictive path following with contact scheduling

Let $e_k^\perp$ be the lateral deviation of the predicted trunk center from γ , ψ_k^γ the path tangent heading at the closest point, and $v_k^\parallel$ the realized speed along the path. For a command sequence $\mathbf{U} = (\mathbf{u}_0, \dots, \mathbf{u}_{N-1})$ the cost is

$$\begin{aligned} J(\mathbf{U}) = \sum_{k=0}^{N-1} & \left[w_e (e_k^\perp)^2 + w_\psi (\psi_k - \psi_k^\gamma)^2 \right. \\ & + w_v (v_k^\parallel - v_{\text{ref}})^2 \\ & + w_\Delta \|\mathbf{u}_k - \mathbf{u}_{k-1}\|^2 + w_m \text{softplus}\left(\frac{m_s - \hat{m}_k}{\epsilon}\right) \\ & \left. + w_\phi \ell_\phi(\mathbf{U}), \right] \end{aligned} \quad (10)$$

where m_s is a soft target margin and ϵ a smoothing width. The contact-phase term penalizes entering the narrowest cross-section Π_{min} of the passage while a leg adjacent to its boundary is near mid-swing, where its lateral excursion in (7) is largest:

$$\ell_\phi(\mathbf{U}) = \sum_{j \in \mathcal{J}_{\text{adj}}} \mathbb{I}[\hat{\phi}_{k_g}^j \geq \beta] \sin^2(\pi \hat{s}_j(k_g)), \quad (11)$$

where k_g is the predicted step at which the front of the hips $\mathbf{h}_{\text{FL}}, \mathbf{h}_{\text{FR}}$ crosses Π_{min} , and $\mathcal{J}_{\text{adj}}$ are the legs whose hips are within W of a passage surface at that step. Because k_g and $\hat{\phi}_{k_g}^j$ both depend on the commanded speed profile through (2)

Table 1: Design parameters of the follower.

Symbol	Value	Meaning
Δt	20 ms	Control and prediction period
T_H	0.5–1.0 s	Prediction horizon, Eq. (13)
N_s	512	MPPI samples per cycle
κ	2	Uncertainty inflation in Eq. (9)
m_{min}	5 cm	Hard clearance margin
m_s	10 cm	Soft target margin
T_b	3.5 s	Time-to-pass budget
ρ_{min}, n_r	0.05, 5	Rejection threshold and persistence

and (5), minimizing (10) schedules contacts by small speed adjustments during the approach rather than by direct gait commands. The same mechanism applies to rope barriers and pipe runs, where h_{sw} and the foothold (6) determine whether a foot touches down too close to the obstacle line.

The hard constraints are

$$\hat{m}_k \geq m_{\text{min}} \quad \forall k, \quad \hat{T}_{\text{pass}}(\mathbf{U}) \leq T_b, \quad (12)$$

where $\hat{T}_{\text{pass}}$ is the predicted time to pass. If the exit plane is reached within the horizon, $\hat{T}_{\text{pass}}$ is read from the rollout; otherwise the remaining path length is divided by the calibrated steady-state speed $g_x u_{N-1}^x$ and added to the elapsed and horizon times.

The horizon adapts to speed so that it covers the passage crossing while staying short enough for the interface model to remain accurate:

$$T_H = N \Delta t = \text{clip}\left(\frac{d_{\text{min}} + L}{\max(v_0^\parallel, v_\epsilon)}, 0.5 \text{ s}, 1.0 \text{ s}\right), \quad (13)$$

with d_{min} the distance to Π_{min} . We optimize (10) with MPPI [25]: N_s perturbed sequences $\mathbf{U}^{(n)} = \bar{\mathbf{U}} + \mathcal{E}^{(n)}$ are rolled out through (2)–(9), sequences violating (12) are discarded, and the nominal sequence is updated with weights

$$\omega^{(n)} \propto \exp(-(J(\mathbf{U}^{(n)}) - J_{\text{min}})/\lambda). \quad (14)$$

MPPI is chosen over gradient-based MPC because ℓ_ϕ and the signed-distance terms are discontinuous in $\mathbf{U}$ when k_g jumps between steps.

4.6 Passage entry rejection

While the trunk is in front of Π_{in} , the follower computes the feasible fraction ρ of sampled sequences that satisfy (12). Entry is rejected when $\rho < \rho_{\text{min}}$ for n_r consecutive cycles before the trigger point; the robot then decelerates to a standoff pose on γ and reports the rejection to the mission layer, which may re-approach at a lower setpoint. Once Π_{in} is crossed, rejection is disabled and the follower continues with the best available sequence, because stopping inside the passage is usually worse than completing it. The time budget makes the rejection test sensitive to cases where clearance is achievable only by a slow realignment that the mission cannot afford, which a pure clearance test would accept.

4.7 Implementation

The configuration $\mathcal{C}$ uses the Go2 trunk dimensions and hip offsets from the manufacturer model. The local range map is accumulated from the onboard LiDAR in the odometry frame and cropped to the passage region. Rollouts are evaluated in parallel on the onboard computer. Table 1 lists the design parameters.

05 Experiments

5.1 Setup and platform

Simulation. The Isaac Lab [34] narrow-gate suite contains doorway-scale gaps of 55–90 cm, rope barriers, and pipe runs, arranged in 18 layouts. Each layout is run with 5 seeds and 25 trials per seed, i.e., 2,250 trials per method.

Hardware. Real-world trials use a Unitree Go2 passing through a 72 cm doorway-scale gap and two rope barriers, with 10 first-attempt runs per gap class. Speed setpoints range from 1.0 to 3.0 m/s in both domains.

Baselines. All methods share the reference path, the range map, and the onboard locomotion controller. (i) *Point-body path following (ideal velocity)* tracks γ assuming $\nu_k = \mathbf{u}_k$ and checks clearance of an inflated point. (ii) *Path following with measured lag model only* replaces the ideal assumption with a lag model measured offline but keeps the point-body clearance check. (iii) *Path following with measured lag + present range* adds an entry check based on the range measured at the current pose.

Metrics. First-attempt success is the percentage of attempts that clear the passage without contact on the first try. Clearance margin is the minimum distance between the robot and the passage during the crossing, in centimeters. Time to pass is defined in Section 3. Correct passage-entry rejections counts entries declined in cases where the passage could not be cleared at the commanded setpoint, as logged by the run server.

5.2 Main comparison

Table 2 reports the main comparison. The proposed follower reaches 84.6% first-attempt success, 9.1 percentage points above the strongest baseline, point-body path following with ideal velocity (75.5%), and 11.2 and 10.7 points above the two lag-model baselines. It also passes fastest, with a time to pass of 2.8 s against 3.0–3.2 s for the baselines, i.e., 0.4 s or 12.5% shorter than the ideal-velocity baseline. Correct entry rejections are slightly higher (7.9 versus 7.6, 7.1, and 6.8).

The clearance margin moves in the opposite direction: 13.2 cm for our follower against 15.1 cm for the ideal-velocity baseline and 13.9 and 14.5 cm for the lag-model baselines. We attribute this, tentatively, to explicit scheduling: the follower knows its predicted envelope and steers toward the soft margin target m_s rather than toward the passage center line, and it completes crossings that the baselines fail, which are likely to be tighter ones. The data do not separate these effects. The reduction of 1.9 cm in mean margin is accompanied by higher success, but it also means that the follower relies on the accuracy of its predictions; Section 5.4 discusses a marginal case.

A notable result concerns the lag-model baselines. Adding a measured lag model to point-body path following lowers success from 75.5% to 73.4%, and adding a present-range entry check recovers only part of the loss (73.9%). Modeling the lag without modeling the leg envelope or the time budget changes when the follower slows down and turns, but its clearance test still uses the wrong geometry, so the lag compensation alone does not translate into more passes.

5.3 Ablation

Table 3 removes one component at a time. Removing contact-phase scheduling ($w_\phi = 0$) reduces success to 72.4% (−12.2 points), removing the future-range rejection test, i.e., replacing

(9) along the horizon by a present-range check, reduces it to 73.3% (−11.3 points), and replacing the calibrated interface model by the nominal configuration limits reduces it to 74.4% (−10.2 points). Each component therefore contributes more than 10 points, and every ablated variant falls below the ideal-velocity baseline at 75.5%. The components are complementary rather than additive: phase scheduling relies on the calibrated model to predict when the trunk reaches $\Pi_{\min}$, and the rejection test relies on both to predict whether the crossing fits the budget. A partial predictive system commits to entries on the basis of an incomplete prediction, and in our evaluation that was worse than the simpler point-body follower.

5.4 Failure cases

Rejection at high speed. In the 62 cm gap at a 3.0 m/s setpoint, the follower correctly rejected 8 of 10 attempts. The predicted trajectory required 4.1 s to pass, exceeding the 3.5 s budget by 0.6 s: at this setpoint the interface lag forces a long deceleration and lateral realignment before the narrow section, and the predicted crossing is slow even though the setpoint is high. With the setpoint lowered to 2.4 m/s, the robot passed. This case illustrates the intended behavior of the time budget, and also its cost: a mission planner that does not lower the setpoint after a rejection will not reach the goal.

Marginal contact at a rope barrier. In one rope-barrier trial at 2.8 m/s, the right foreleg made a contact with a 6 cm margin, inside the corridor, and the trial was logged as a marginal pass. The margin is above $m_{\min}$ but below m_s , which indicates that the predicted foothold (6) of the right foreleg was close to the rope line at that speed. At high speed, the Raibert correction term and the phase prediction both accumulate error over the horizon, and a single trial is insufficient to establish whether this is a systematic bias.

Sim-to-real gap of the ideal-velocity baseline. The ideal-velocity baseline reached 91% first-attempt success in simulation but 63% on the real Go2, a drop of 28 points, mostly induced by lag. In simulation the locomotion controller tracks commands closely enough that the ideal-velocity assumption is nearly harmless; on hardware, the velocity lag shifts the realized trajectory relative to the planned one exactly where clearance is smallest. This observation motivated the calibrated interface model and is consistent with the finding that simulation rankings need not transfer to hardware [35].

06 Limitations

The study has several limitations. First, the hardware evaluation is small, with 10 first-attempt runs per gap class and a single doorway width, so the hardware numbers carry substantial sampling uncertainty, and the per-domain breakdown recorded for the ideal-velocity baseline was not compiled for the other methods. Second, the follower influences the contact schedule only through speed modulation and assumes a trot whose frequency depends affinely on speed; controllers that switch gaits or use non-periodic stepping would require a different phase model. Third, the leg envelope in (8) is a geometric approximation; the marginal rope-barrier contact suggests that foothold prediction at speeds near 3 m/s is the weakest part of the prediction. Fourth, the mean clearance margin of the follower is smaller than that of every baseline, so its advantage depends on the calibrated model remaining valid; payload changes, worn feet, or slippery floors that alter the interface response would require recalibration before the margin can be trusted. Finally, the

Table 2: Main comparison on the narrow-passage evaluation. Higher is better for success and correct rejections; lower is better for time to pass.

Method	First-attempt success (%)	Clearance margin (cm)	Time to pass (s)	Passage entry rejections (correct)
Point-body path following (ideal velocity)	75.5	15.1	3.2	7.6
Path following with measured lag model only	73.4	13.9	3.0	7.1
Path following with measured lag + present range	73.9	14.5	3.1	6.8
Ours	84.6	13.2	2.8	7.9

Table 3: Ablation of the proposed follower. Δ is the difference to the full method in percentage points.

Variant	Success (%)	Δ
Ours (predictive contact scheduling)	84.6	–
w/o contact-phase scheduling	72.4	–12.2
w/o future-range rejection	73.3	–11.3
w/o calibrated interface model	74.4	–10.2

time budget is a fixed parameter, and the high-speed rejection case shows that the follower declines passages that are feasible at lower setpoints unless the mission layer adapts the setpoint.

07 Conclusion

This paper presented a predictive contact-aware path follower for quadrupeds with opaque velocity interfaces. By predicting the realized body motion with a GAUGE-calibrated interface model, converting it into a phase-dependent body-and-leg envelope with AgniNav-style configuration-based prediction, scheduling contact phases at passage entry through speed modulation, and rejecting entries whose predicted trajectory violates a clearance or time budget, the follower raised first-attempt success from 75.5% to 84.6% and reduced time to pass from 3.2 s to 2.8 s relative to point-body path following, with a smaller mean clearance margin of 13.2 cm. Ablations show that contact-phase scheduling, future-range rejection, and the calibrated interface model each contribute between 10.2 and 12.2 points. Future work will extend the phase model to gait-switching controllers, couple the rejection test with setpoint adaptation, and enlarge the hardware evaluation.

References

- [1] S. Macenski, F. Martín, R. White, and J. Ginés Clavero. The Marathon 2: A Navigation System. arXiv:2003.00368, 2020.
- [2] 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.
- [3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [4] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [5] 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.
- [6] T. Zang, H. Huang, S. Cheng, and W. Zhang. GAUGE: Planner-Conditioned Active Calibration of Opaque Quadruped Velocity Interfaces. arXiv:2607.02841, 2026.
- [7] 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.
- [8] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [9] G. B. Margolis and P. Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. arXiv:2212.03238, 2022.
- [10] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid Locomotion via Reinforcement Learning. arXiv:2205.02824, 2022.
- [11] 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.
- [12] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifftthaler, and J. Buchli. Real-Time Motion Planning of Legged Robots: A Model Predictive Control Approach. arXiv:1710.04029, 2017.
- [13] 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.
- [14] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter. Perceptive Locomotion through Nonlinear Model Predictive Control. arXiv:2208.08373, 2022.
- [15] F. Jenelten, J. He, F. Farshidian, and M. Hutter. DTC: Deep Tracking Control. arXiv:2309.15462, 2023.
- [16] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [17] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [18] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. arXiv:2309.05665, 2023.
- [19] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [20] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.
- [21] 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.
- [22] 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.
- [23] J. Frey, D. Hoeller, S. Khattak, and M. Hutter. Locomotion Policy Guided Traversability Learning using Volumetric Representations of Complex Environments. arXiv:2203.15854, 2022.
- [24] 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.
- [25] G. Williams, A. Aldrich, and E. Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. arXiv:1509.01149, 2015.
- [26] G. Kahn, P. Abbeel, and S. Levine. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. arXiv:2002.05700, 2020.

- [27] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [28] 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.
- [29] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. arXiv:2310.07896, 2023.
- [30] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control Barrier Functions: Theory and Applications. arXiv:1903.11199, 2019.
- [31] 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.
- [32] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [33] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. arXiv:2108.10470, 2021.
- [34] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.
- [35] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance?. arXiv:1912.06321, 2019.
- [36] N. Yokoyama, S. Ha, and D. Batra. Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation. arXiv:2103.08022, 2021.