Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CacheMPC: Certified Cached Model Predictive Control for Quadruped Locomotion

Nimesh Khandelwal

nimesh20@iitk.ac.in

Indian Institute of Technology Kanpur

Mehul Anand

Indian Institute of Technology Roorkee

Shakti S. Gupta

Indian Institute of Technology Kanpur

Mangal Kothari

Indian Institute of Technology Kanpur

Research Article

Keywords: Legged robots, Model predictive control, Look-up table, Locality-sensitive hashing, Whole body control

Posted Date: August 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10472862/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CacheMPC: Certified Cached Model Predictive Control for Quadruped Locomotion

Nimesh Khandelwal[1*], Mehul Anand[2], Shakti S. Gupta[1], Mangal Kothari[3]

[1*]Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur, 208016, Uttar Pradesh, India.
[2]Department of Materials Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, 247667, Uttarakhand, India.
[3]Department of Aerospace Engineering, Indian Institute of Technology Kanpur, Kanpur, 208016, Uttar Pradesh, India.

*Corresponding author(s). E-mail(s): nimesh20@iitk.ac.in;
Contributing authors: mehul_a@mt.iitr.ac.in; ssgupta@iitk.ac.in;
mangalgnc@gmail.com;

Abstract

Model Predictive Control (MPC) is the standard predictive layer in hierarchical quadruped controllers, but the per-cycle QP solve limits the update rate achievable on embedded processors. Because legged gaits revisit a bounded region of state space, MPC solutions admit caching and reuse. This paper proposes *Certified CacheMPC*: a Locality-Sensitive-Hashed cache of horizon contact-force trajectories, partitioned by contact mode, retrieved at query time and accepted only when an a-posteriori per-query certificate confirms primal feasibility and a Lagrangian dual-gap upper bound on cost suboptimality. A bounded-budget controller schedule combines top-$\boldsymbol{K}$ certified retrieval, a deadline-bounded QP solve, and a shifted last-certified fallback. The framework is evaluated on a Unitree Go2 across **2,038** usable cold-controller MuJoCo trials, including a **600**-trial $\boldsymbol{n=50}$ campaign at three failure-boundary cells, and a first-deploy session on the on-robot NVIDIA Orin NX. The un-gated cache delivers a **25**× median solve-time speedup in simulation and an **18.7**× median speedup on hardware. At $\boldsymbol{n=50}$ no statistically significant difference in closed-loop stable rate is detected between the cache variants and the no-cache baseline at any tested cell. The certificate's contribution to closed-loop safety is not resolvable at the present sample size.

[1]Present address: ADASI, EDGE Group, Abu Dhabi, United Arab Emirates. This work was performed while M. Kothari was with the Indian Institute of Technology Kanpur.

Keywords: Legged robots, Model predictive control, Look-up table, Locality-sensitive hashing, Whole body control

1 Introduction

Hierarchical control architectures combining a low-rate predictive controller with a high-rate reactive controller have emerged as the state of the art for model-based legged locomotion [1–3, 16, 17, 29], complementing learning-based controllers that have demonstrated robust locomotion at scale [18, 19]. The predictive controller, typically a convex MPC on a simplified dynamics model, optimises contact forces over a finite horizon; the reactive controller, typically a Whole-Body Controller (WBC) [10, 28], uses the full rigid-body dynamics to produce physically consistent joint torques at high frequency. The quality of the MPC solution directly affects tracking and robustness, but the per-cycle solve cost limits the achievable MPC rate on embedded processors such as the NVIDIA Orin NX shipped with the Unitree Go2: a state-dependent contact horizon, larger problem size, or sequential preprocessing all push the per-tick MPC update into the millisecond range, leaving little headroom inside a typical 5 ms control budget. The MPC is therefore the limiting layer for the closed-loop update rate.

Several lines of work address this gap. Real-time iteration NMPC [20, 21] amortises a fresh nonlinear solve over multiple ticks but still requires solver progress every cycle. TinyMPC [4] pre-computes ADMM solver factorisations but requires linearisable dynamics. Neural function approximation [5] distils MPC solutions into networks but requires separate training and lacks runtime correctness guarantees; recent work on safe approximate MPC [23, 24] adds verification but operates offline. Explicit MPC [6, 22] pre-computes the solution as a piecewise-affine function of the state, but the polyhedral region count scales poorly with problem dimension. Sampling-based predictive controllers such as MPPI [31] sidestep the QP entirely at the cost of high parallel compute. The idea of caching and reusing MPC solutions appears in parametric settings [11], but without per-query safety guarantees.

This paper proposes Certified CacheMPC. A Locality-Sensitive-Hashed (LSH) cache, partitioned by contact mode so that the QP structural pattern is shared within a partition, returns previously computed contact-force trajectories at query time. An a-posteriori per-query certificate, evaluated against the current QP's own KKT data, produces a Lagrangian dual-gap quantity $\Gamma(\bar{\mathbf{U}})$ that upper-bounds the true suboptimality $\Delta J = J(\bar{\mathbf{U}}) - J^\star$ of any accepted retrieval, independent of where the proposal originated. A bounded-budget controller schedule combines top-K certified retrieval, a deadline-bounded QP solve, and a shifted last-certified fallback.

The contributions of this paper are:

1. A deterministic per-query QP certificate (Proposition 1) whose Lagrangian dual-gap quantity $\Gamma(\bar{\mathbf{U}})$ upper-bounds the true suboptimality of any feasible cached proposal independent of its provenance.

2. A bounded-budget controller schedule (Section 4.4) combining top-K certified retrieval, a deadline-bounded solver, and a shifted last-certified fallback, exercised empirically under deliberately tight deadlines in Section 5.5.
3. A simulation campaign of 2,038 usable cold-controller trials and a first-deploy hardware session on the Unitree Go2 with the on-robot NVIDIA Orin NX (Sections 5–6), characterising the speed, tracking, and failure-boundary safety of four cache configurations against the no-cache baseline.

The remainder of the paper is organised as follows. Section 2 reviews related work; Section 3 states the convex MPC and WBC formulations; Section 4 describes the certified framework and its propositions; Section 5 reports the simulation evaluation; Section 6 reports the hardware session; Section 7 discusses threats to validity; Section 8 concludes; Section 9 lists future-work directions.

2 Related Work

2.1 MPC for Quadruped Locomotion

Convex MPC formulations based on the single rigid body dynamics (SRBD) model have been widely adopted for quadruped control [1, 2, 16]. These formulations linearise the rotational dynamics and approximate friction cones as pyramids to cast the problem as a QP, enabling real-time solutions using solvers such as qpOASES [8] and HPIPM [9]. Torque-controlled feedback MPC variants [30] and whole-body trajectory optimisation frameworks [3, 29] extend the formulation to richer dynamics at the cost of a larger online problem.

2.2 Approaches to Accelerate MPC

Explicit MPC [6] pre-computes the optimal control law as a piecewise affine function of the state by solving a multi-parametric QP offline. The online evaluation reduces to a point-location problem in a polyhedral partition of the state space. While effective for low-dimensional systems, the number of regions grows exponentially with problem dimension, making the full enumeration impractical for the 13-state MPC problems common in quadruped control. Our framework can be viewed as an online, sample-based variant of explicit MPC: each cache entry is a local affine piece of the explicit law, populated lazily from the same QPs that the controller would have solved anyway, and accepted at query time only when an a-posteriori certificate vouches for it.

Function approximation methods learn a mapping from state to MPC solution using supervised learning. MPCNET [5] trains a neural network on MPC demonstrations and uses it for online inference. Similarly, iterative learning control (ILC) approaches [12] maintain a library of control actions indexed by trajectory phase, achieving faster execution through lookup rather than optimization. These methods require offline training or separate optimization stages and are difficult to adapt online to changing environments.

TinyMPC [4] achieves speedup by pre-computing and caching matrix factorizations within an ADMM solver, avoiding expensive operations at runtime. While effective

for convex problems with fixed structure, it does not address the broader question of reusing full MPC solutions across similar states.

Warm-starting is a widely used technique where the solution from the previous time step initializes the solver for the current problem. While effective for reducing iteration counts, warm-starting provides diminishing returns when the state changes significantly between MPC updates, and still requires running the solver to completion.

2.3 Cache-Based and Lookup Approaches

The idea of caching and reusing optimization solutions has been explored in various control contexts. In [11], a parameter-adaptive framework stores MPC solutions indexed by system parameters for fast retrieval. Our approach differs in that we cache solutions indexed by the full operational state using LSH, dynamically update the cache during operation, and exploit the contact-schedule structure specific to legged locomotion. To the best of our knowledge, CacheMPC is the first framework to combine LSH-based caching with contact-schedule partitioning and a per-query optimality certificate for quadruped MPC.

3 Background

3.1 Convex MPC for Quadruped Locomotion

We use a convex MPC formulation based on the SRBD model, following [1]. The robot is modeled as a single rigid body with mass m and rotational inertia $\mathbf{I}_b \in \mathbb{R}^{3\times3}$, actuated by contact forces at the feet. The state vector is defined as

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\Theta}^T & \mathbf{p}^T & \boldsymbol{\omega}^T & \dot{\mathbf{p}}^T & g \end{bmatrix}^T \in \mathbb{R}^{13}, \tag{1}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^3$ is the ZYX Euler angle vector representing the body orientation, $\mathbf{p} \in \mathbb{R}^3$ is the position of the center of mass (CoM), $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity in the world frame, $\dot{\mathbf{p}} \in \mathbb{R}^3$ is the CoM linear velocity, and g is the gravitational constant (appended for notational convenience). The control input is the stacked vector of contact forces:

$$\mathbf{u} = \begin{bmatrix} \mathbf{f}_1^T & \mathbf{f}_2^T & \mathbf{f}_3^T & \mathbf{f}_4^T \end{bmatrix}^T \in \mathbb{R}^{12}, \tag{2}$$

where $\mathbf{f}_i \in \mathbb{R}^3$ is the ground reaction force at foot i.

The continuous-time dynamics are linearized about the current state to obtain $\dot{\mathbf{x}} = \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{u}$, where

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{g}} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 \end{bmatrix}, \tag{3}$$

with $\mathbf{R}_z(\psi)$ being the yaw rotation matrix and $\hat{\mathbf{g}} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$. The input matrix $\mathbf{B}_c$ encodes the mapping from contact forces to CoM accelerations and angular accelerations through the moment arms of each foot relative to the CoM. The dynamics are discretized with timestep Δt using a zero-order hold to obtain $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$.

The MPC problem is formulated over a horizon of N steps as:

$$\begin{aligned} \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} \quad & \sum_{k=0}^{N-1} \left[(\mathbf{x}_k - \mathbf{x}_k^{\text{ref}})^T \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_k^{\text{ref}}) + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right] \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \\ & |f_{i,x}| \le \mu f_{i,z}, \quad |f_{i,y}| \le \mu f_{i,z}, \\ & f_{z,\min} \le f_{i,z} \le f_{z,\max}, \\ & f_{i,z} = 0 \quad \text{if foot } i \notin \mathcal{C}_k, \end{aligned} \tag{4}$$

where $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are the state and control weight matrices, μ is the friction coefficient, and $\mathcal{C}_k$ is the set of feet in contact at step k, determined by the gait schedule. The friction cone is linearized as a pyramid with four faces. The resulting problem is the strictly convex QP (8) and is solved with HPIPM [9] in production; the formulation is solver-agnostic and the certificate of Section 4 consumes only the assembled condensed QP data.

Fixed-contact horizon.

The per-stage contact set $\mathcal{C}_k$ is held at the current measured contact set $\mathcal{C}_0$ for every $k \in [0, N-1]$. The condensed QP (8) therefore reuses one $(\mathbf{B}_k, \underline{\mathbf{d}}, \overline{\mathbf{d}})$ block at every stage, partitioning by the four-bit contact mask $\boldsymbol{\sigma}$ is exact for the QP structure, and contact-switch anticipation is delegated to the gait scheduler upstream of the MPC. A horizon-anticipative variant of [1] that consumes a gait-table-driven contact sequence would re-key the cache on the full $(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{N-1})$ sequence rather than $\boldsymbol{\sigma} = \mathcal{C}_0$; the certificate of Section 4 extends unchanged to that variant because it operates on the assembled condensed QP and does not depend on the source of $\boldsymbol{\sigma}$.

3.2 KinoDynamic Whole Body Control

The reactive layer uses a KinoDynamic WBC that operates at the proprioceptive estimator rate (250 Hz on our build) and therefore at a higher frequency than the MPC. It consists of two cascaded stages: a Kinematic WBC (KinWBC) followed by a Weighted Whole Body Controller (W-WBC), forming the complete MPC + KinWBC + W-WBC control stack.

The KinWBC modifies the reference joint positions to better satisfy task-space objectives. Given the current robot configuration and the desired operational-space motions (base pose, foot positions), it computes updated joint position references using prioritized inverse kinematics, following [2]. This kinematic adjustment ensures that the joint references are consistent with the desired base and foot motions before being passed to the dynamic layer.

The W-WBC uses the full rigid-body dynamics (RBD) of the quadruped to compute joint torques. We adopt the weighted WBC formulation [10] that minimizes a weighted sum of task errors subject to the equation of motion:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T\boldsymbol{\tau} + \mathbf{J}_c^T\mathbf{F}_c, \tag{5}$$

where $\mathbf{q} \in \mathbb{R}^{18}$ is the generalized coordinate vector, $\mathbf{M}$ is the inertia matrix, $\boldsymbol{\eta}$ represents Coriolis and gravitational terms, $\mathbf{S}$ is the selection matrix for actuated joints, $\boldsymbol{\tau} \in \mathbb{R}^{12}$ are joint torques, and $\mathbf{F}_c$ are the contact forces.

The WBC optimizes joint accelerations $\ddot{\mathbf{q}}$, joint torques $\boldsymbol{\tau}$, and contact forces $\mathbf{F}_c$ simultaneously:

$$\begin{aligned} \min_{\ddot{\mathbf{q}},\boldsymbol{\tau},\mathbf{F}_c} \quad & \sum_i w_i \|\mathbf{J}_i\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \ddot{\mathbf{x}}_i^{\mathrm{cmd}}\|^2 \\ & + w_\tau\|\boldsymbol{\tau}\|^2 + w_f\|\mathbf{F}_c - \mathbf{F}_c^{\mathrm{ref}}\|^2 \\ \text{s.t.} \quad & \text{Eq. (5)}, \quad \text{contact constraints}, \end{aligned} \tag{6}$$

where $\mathbf{J}_i$ and $\ddot{\mathbf{x}}_i^{\mathrm{cmd}}$ are the Jacobian and commanded acceleration for task i (computed using the KinWBC-adjusted references), and $\mathbf{F}_c^{\mathrm{ref}}$ is the reference contact force from the MPC. The tracking term $\|\mathbf{F}_c - \mathbf{F}_c^{\mathrm{ref}}\|^2$ is the interface through which the MPC solution—cached or freshly solved—enters the W-WBC. The W-WBC enforces dynamic consistency with the full rigid-body dynamics and the friction and unilaterality constraints that the SRBD model abstracts away, while the KinWBC reconciles joint references with the realised task-space motion. The WBC is the final feasibility filter for the full-body dynamics; the per-query certificate of Section 4 bounds the SRBD-objective gap of any accepted reference by $\beta(\bar{\mathbf{U}})$, leaving model mismatch and full-body feasibility to the WBC.

4 Certified Memory-Augmented MPC

A retrieved control is accepted only when an a-posteriori certificate, evaluated against the current predictive problem's own optimality conditions, confirms both primal feasibility and a budgeted suboptimality bound. The certificate elevates a cache entry from a heuristic neighbour in feature space to a verified local piece of the explicit-MPC law. Because the optimal-control map is piecewise affine with discontinuities at active-set boundaries, a near neighbour in feature space can lie in a different critical region; the per-query certificate is what rules this out. Fig. 1 shows the resulting architecture.

4.1 Feature Vector and Candidate Generation via LSH

Each cache entry is indexed by a feature vector $\boldsymbol{\phi} \in \mathbb{R}^{18}$ that compresses the operationally relevant state:

$$\boldsymbol{\phi} = \left[\dot{\mathbf{p}}^T \;\; \dot{\mathbf{p}}_{\mathrm{des}}^T \;\; ({}^{\mathcal{B}}\mathbf{r}_{p_1})^T \;\; \cdots \;\; ({}^{\mathcal{B}}\mathbf{r}_{p_4})^T\right]^T, \tag{7}$$

combining the measured and desired CoM velocities and the four body-frame foot positions. The cache is partitioned by contact mode $\boldsymbol{\sigma}$ so that the QP structure (constraint shape, active swing/stance bits) is shared within a partition; a trotting gait, for

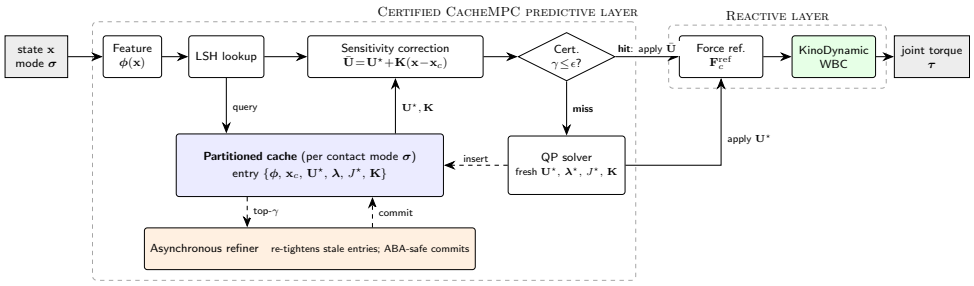


Fig. 1 Certified CacheMPC architecture. The predictive layer hashes $\boldsymbol{\phi}(\mathbf{x})$ into the contact-mode-partitioned cache and accepts a sensitivity-corrected candidate $\bar{\mathbf{U}}$ only when ρ_{feas} and the dual-gap $\Gamma(\bar{\mathbf{U}})$ satisfy the budget $\beta(\bar{\mathbf{U}})$; on rejection the QP is solved and the fresh tuple is inserted. An asynchronous refiner re-tightens stale entries, and the accepted $\mathbf{F}_c^{\text{ref}}$ is consumed by the KinoDynamic WBC.

example, populates only two partitions. Approximate nearest-neighbour search inside each partition uses Locality-Sensitive Hashing [7, 25, 26] with the Euclidean p-stable construction of [25]: L tables of k hash functions $h_{\mathbf{a},b}(\boldsymbol{\phi}) = \lfloor (\mathbf{a}\cdot\boldsymbol{\phi} + b)/w \rfloor$, $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $b \sim \mathcal{U}[0, w]$, giving sub-linear query time in 18-dimensional space. LSH serves strictly as a candidate generator; acceptance is decided by the certificate of Section 4.3.

4.2 Condensed Problem and Cache Entry Schema

After eliminating the predicted states via the linearised dynamics of Section 3.1, the convex MPC at state $\mathbf{x}$ and contact mode $\boldsymbol{\sigma}$ is the strictly convex QP

$$\min_{\mathbf{U}} \ \tfrac{1}{2}\mathbf{U}^\top \mathbf{H}\,\mathbf{U} + \mathbf{g}^\top \mathbf{U} \quad \text{s.t.} \quad \underline{\mathbf{d}} \le \mathbf{C}\,\mathbf{U} \le \overline{\mathbf{d}}, \tag{8}$$

where $\mathbf{U} \in \mathbb{R}^n$ stacks the horizon contact forces, $\mathbf{H} \succ 0$ (regularised by the controller), and $(\mathbf{C}, \underline{\mathbf{d}}, \overline{\mathbf{d}})$ collect the linearised friction pyramid, f_z bounds, and contact-activation constraints. Let $J(\mathbf{U}; \mathbf{x}) = \tfrac{1}{2}\mathbf{U}^\top \mathbf{H}\mathbf{U} + \mathbf{g}^\top \mathbf{U}$ with constrained optimum $J^\star$. Each cache entry stores $(\boldsymbol{\phi}, \mathbf{x}_c, \mathbf{U}^\star, \boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u, J^\star, \mathbf{K})$, where $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u) \ge \mathbf{0}$ are the dual multipliers of the two sides of (8), and $\mathbf{K} = \partial \mathbf{U}^\star / \partial \mathbf{x}$ is the active-set sensitivity from the entry's KKT system. For the SRBD formulation $(\mathbf{C}, \underline{\mathbf{d}}, \overline{\mathbf{d}})$ depend on the contact mode but not on the state, so a cached control feasible at insertion stays feasible for every query in the same partition; only suboptimality must be bounded. State-dependent constraint terms restore a non-trivial feasibility component in the certificate below.

4.3 Per-Query Certificate

Given a candidate control $\bar{\mathbf{U}}$ (retrieved, possibly sensitivity-corrected) the certificate returns two quantities.

Primal feasibility.
The exact violation is

$$\rho_{\text{feas}}(\bar{\mathbf{U}}) = \max_i \max\Big(\underline{d}_i - (\mathbf{C}\bar{\mathbf{U}})_i,\ (\mathbf{C}\bar{\mathbf{U}})_i - \overline{d}_i,\ 0\Big), \tag{9}$$

computed in one matrix–vector product.

Dual-gap bound on suboptimality.
The Lagrangian dual of (8) at any $\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u \geq \mathbf{0}$ is, with $\mathbf{s} = \mathbf{g} + \mathbf{C}^\top(\boldsymbol{\lambda}_u - \boldsymbol{\lambda}_\ell)$,

$$q(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u) = -\tfrac{1}{2}\mathbf{s}^\top \mathbf{H}^{-1}\mathbf{s} - \boldsymbol{\lambda}_u^\top \overline{\mathbf{d}} + \boldsymbol{\lambda}_\ell^\top \underline{\mathbf{d}}, \tag{10}$$

and by weak duality $q \leq J^\star$. Hence for any feasible $\bar{\mathbf{U}}$,

$$\begin{aligned} \Gamma(\bar{\mathbf{U}}) \ &= \ J(\bar{\mathbf{U}}; \mathbf{x}) - q(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u) \\ &\geq \ J(\bar{\mathbf{U}}; \mathbf{x}) - J^\star \ = \ \Delta J(\bar{\mathbf{U}}) \ \geq \ 0 \end{aligned} \tag{11}$$

is a rigorous upper bound on the true suboptimality ΔJ of $\bar{\mathbf{U}}$. Throughout the paper ΔJ denotes the true gap and Γ the certificate's upper-bound quantity. Evaluating (10) costs one Cholesky factorisation of $\mathbf{H}$ and a back-substitution. The reported implementation evaluates Γ at $\boldsymbol{\lambda} = \mathbf{0}$ (the unconstrained relaxation $q = -\frac{1}{2}\mathbf{g}^\top \mathbf{H}^{-1}\mathbf{g}$), which is the loosest valid weak-duality bound. A cached-dual variant of the bound would be tight whenever the query shared the entry's active set.

Acceptance rule.
Define the per-query relative budget

$$\beta(\bar{\mathbf{U}}) \ = \ \epsilon_{\text{abs}} + \epsilon_{\text{rel}}\,|J(\bar{\mathbf{U}})|. \tag{12}$$

The cached candidate is accepted iff

$$\rho_{\text{feas}}(\bar{\mathbf{U}}) \leq \epsilon_{\text{feas}} \ \text{ and } \ \Gamma(\bar{\mathbf{U}}) \leq \beta(\bar{\mathbf{U}}), \tag{13}$$

otherwise the QP is solved online and the new $(\mathbf{U}^\star, \boldsymbol{\lambda}^\star, J^\star, \mathbf{K})$ tuple is inserted. The budget β is state-dependent through $|J(\bar{\mathbf{U}})|$; a per-tick constant guarantee requires an operating tube with an upper bound $J_{\max}$ on $|J|$, giving $\beta \leq \beta_{\max} = \epsilon_{\text{abs}} + \epsilon_{\text{rel}} J_{\max}$.

Dual-feasibility validation.
The bound (10) requires $\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u \geq \mathbf{0}$ and finite. Because the reported implementation evaluates Γ at $\boldsymbol{\lambda} = \mathbf{0}$, dual-feasibility is satisfied by construction. Any cached-dual variant of the bound would need to guard the reused multipliers for finiteness and non-negativity and fall back to $\boldsymbol{\lambda} = \mathbf{0}$ on failure, so that the bound the controller acts on is always weak-duality-valid.

Proposition 1 (Deterministic per-query QP certificate) *Let* (8) *be the current QP with* $\mathbf{H} \succ \mathbf{0}$ *and optimum* $J^\star$*. Assume* $\bar{\mathbf{U}}$ *is feasible for* (8)*, i.e.* $\underline{\mathbf{d}} \leq \mathbf{C}\bar{\mathbf{U}} \leq \overline{\mathbf{d}}$*. Then for any finite, dual-feasible* $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u) \geq \mathbf{0}$ *the dual-gap quantity* $\Gamma(\bar{\mathbf{U}})$ *defined in* (11) *satisfies*

$$\Delta J(\bar{\mathbf{U}}) \;=\; J(\bar{\mathbf{U}}; \mathbf{x}) - J^\star \;\leq\; \Gamma(\bar{\mathbf{U}}).$$

If, in addition, the acceptance rule (13) *succeeds, then* $\Delta J(\bar{\mathbf{U}}) \leq \beta(\bar{\mathbf{U}})$.

Proof Feasibility of $\bar{\mathbf{U}}$ gives $J^\star \leq J(\bar{\mathbf{U}})$; weak duality at any dual-feasible $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u)$ gives $q \leq J^\star$. Subtracting yields $\Delta J = J(\bar{\mathbf{U}}) - J^\star \leq J(\bar{\mathbf{U}}) - q = \Gamma$, and the acceptance rule (13) enforces $\Gamma \leq \beta$. □

The implementation accepts $\rho_{\text{feas}} \leq \epsilon_{\text{feas}}$ with $\epsilon_{\text{feas}} = 10^{-4}$, three orders of magnitude below the smallest constraint scale; this is a numerical tolerance for round-off in ρ_{feas} evaluation rather than a relaxation of feasibility in Proposition 1. Proposition 1 also makes no reference to the source of $\bar{\mathbf{U}}$ or $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u)$: the bound is valid whether the proposal is obtained by approximate nearest-neighbour search, drawn from an offline dictionary, asynchronously refined, or constructed by sensitivity-corrected affine extrapolation.

Proposition 2 (Retrieval safety is independent of approximate nearest-neighbour search) *Suppose the controller applies a cached proposal* $\bar{\mathbf{U}}$ *only after the acceptance rule* (13) *succeeds with* $\rho_{\text{feas}}(\bar{\mathbf{U}}) = 0$ *(exact feasibility), and otherwise solves* (8) *exactly. Then for any retrieval policy — including one that returns an arbitrary proposal, returns no proposal, or returns one that violates the cached entry's active-set assumptions — every applied control* $\mathbf{U}_{\text{applied}}$ *satisfies* $\Delta J(\mathbf{U}_{\text{applied}}) \leq \beta(\mathbf{U}_{\text{applied}})$.

Proof If the acceptance rule (13) succeeds with $\rho_{\text{feas}}(\bar{\mathbf{U}}) = 0$, Proposition 1 gives $\Delta J(\bar{\mathbf{U}}) \leq \beta(\bar{\mathbf{U}})$; otherwise the controller applies the QP optimum $\mathbf{U}^\star$, for which $\Delta J = 0$. □

The numerical feasibility tolerance ϵ_{feas} inherits the same caveat as Proposition 1. Approximate-nearest-neighbour errors therefore reduce the cache hit rate but cannot compromise acceptance correctness; the same statement covers stale refiner entries, sensitivity-corrected proposals whose linearisation has crossed an active-set boundary, and discovered proposals whose anchor was sampled offline.

4.4 Bounded-Budget Controller Schedule

The controller enforces a deterministic schedule at every MPC tick. Let the tick budget T_{total} be partitioned as $T_{\text{total}} = T_{\text{cache}} + T_{\text{solve}} + T_{\text{fallback}}$:

1. Within T_{cache}, attempt certified retrieval over the top-K LSH candidates; if any candidate $\bar{\mathbf{U}}$ satisfies (13), apply it and return.
2. Within T_{solve}, solve the QP (8); if the solver returns within budget, apply $\mathbf{U}^\star$ and insert into the cache.
3. If the deadline is missed, apply a shifted last-certified feasible sequence.

The controller executes this schedule at every tick. Section 5.5 reports an empirical stress trial under deliberately tight deadlines. In nominal MPC the analogue of step (3) is absent: a solver that exceeds its deadline leaves the controller without a defined action.

4.5 Sensitivity-Augmented Retrieval and Active-Set Filtering

A cache entry computed at anchor state $\mathbf{x}_c$ can be upgraded from a constant proposal to a first-order affine proposal via the stored sensitivity:

$$\bar{\mathbf{U}}(\mathbf{x}) = \mathbf{U}^\star + \mathbf{K}\,(\mathbf{x} - \mathbf{x}_c). \tag{14}$$

Scope. (14) is exact only when the query parameter $\theta = (\mathbf{x}, \mathbf{x}_g, \mathbf{r}_p, \boldsymbol{\sigma}, \mu, \dots)$ varies along $\mathbf{x}$ alone, with reference $\mathbf{x}_g$, foot positions $\mathbf{r}_p$, contact mode $\boldsymbol{\sigma}$, and model parameters held at the anchor's values. In the present quadruped use-case the actual cache neighbourhood varies along all of these axes simultaneously, so (14) is a *first-order proposal heuristic*, not an exact piece of the explicit-MPC law. Proposition 1 is what makes its use safe: the certificate is evaluated on the corrected candidate, and the candidate is applied only if the certificate succeeds. Errors in the linearisation (missing $\partial/\partial\mathbf{x}_g$, $\partial/\partial\mathbf{r}_p$, $\partial/\partial\boldsymbol{\sigma}$ terms) reduce hit rate but cannot compromise the bound.

Active-set region filter.

The cached duals $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u)$ identify the entry's active set. As a hit-quality pre-screen, the candidate $\bar{\mathbf{U}}$ is rejected when its binding/non-binding pattern is inconsistent with the cached one within a cert-region band of 25 N. This filter is not required for Proposition 1 and does not represent exact membership in a critical region; it costs one matrix–vector product against $\mathbf{C}$ per query, and avoids the Cholesky in (10) for proposals that would fail (13) anyway. The region filter is enabled in the production configuration.

4.6 Certified Online Execution

Algorithm 1 summarises the certified retrieval. The condensed-QP data $(\mathbf{H}, \mathbf{g}, \mathbf{C}, \underline{\mathbf{d}}, \overline{\mathbf{d}})$ is assembled once per tick because the certificate consumes it; this is the only online cost shared between the hit and miss paths. The hit path additionally pays the certificate cost (one Cholesky of $\mathbf{H}$, one back-substitution, and an $\mathcal{O}(n)$ primal-cost evaluation) and skips the QP solve. An asynchronous refiner re-optimises entries with the largest Γ values under a monotone-improvement invariant and runs off the real-time hit/miss path; the per-step guarantee remains the certificate of (13). Cache entries are serialised to disk on shutdown so subsequent sessions start populated.

4.7 Stability Considerations

The classical suboptimal-MPC analysis [14, 15] establishes Lyapunov decrease towards an equilibrium under the standard nominal-MPC assumptions [33]; robust extensions under bounded disturbance [32] give input-to-state stability rather than asymptotic

Algorithm 1 Certified CacheMPC Retrieval

Require: state $\mathbf{x}$, contact mode $\boldsymbol{\sigma}$, cache $\mathcal{H}$, tolerances $(\epsilon_{\text{feas}}, \epsilon_{\text{abs}}, \epsilon_{\text{rel}})$
Ensure: force reference $\mathbf{F}_c^{\text{ref}}$

$\boldsymbol{\phi} \leftarrow \Phi(\mathbf{x})$; select partition $\mathcal{H}_{\boldsymbol{\sigma}}$
$\mathcal{N} \leftarrow$ LSH-TopK$(\mathcal{H}_{\boldsymbol{\sigma}}, \boldsymbol{\phi}, K)$ {up to K neighbour entries, sorted by $\|\boldsymbol{\phi} - \boldsymbol{\phi}_j\|$}
assemble $(\mathbf{H}, \mathbf{g}, \mathbf{C}, \underline{\mathbf{d}}, \overline{\mathbf{d}})$ {condense; no solve}
accept $\leftarrow$ FALSE
for all $(\mathbf{U}_j^\star, \boldsymbol{\lambda}_j, \mathbf{K}_j, \mathbf{x}_{c,j}) \in \mathcal{N}$ **in order, while in** T_{cache} **budget do**
  $\bar{\mathbf{U}} \leftarrow \mathbf{U}_j^\star + \mathbf{K}_j(\mathbf{x} - \mathbf{x}_{c,j})$ {sensitivity correction}
  if region-filter active set of $\boldsymbol{\lambda}_j$ inconsistent with $\mathbf{C}\bar{\mathbf{U}}$ vs. $\underline{\mathbf{d}}, \overline{\mathbf{d}}$ **then**
    continue {region-filter reject; try next candidate}
  end if
  $(\rho_{\text{feas}}, \Gamma) \leftarrow$ Certificate$(\mathbf{H}, \mathbf{g}, \mathbf{C}, \underline{\mathbf{d}}, \overline{\mathbf{d}}, \bar{\mathbf{U}}, \boldsymbol{\lambda} = \mathbf{0})$ {production uses $\boldsymbol{\lambda} = \mathbf{0}$; cached duals stored for future tighter bound}
  $\beta \leftarrow \epsilon_{\text{abs}} + \epsilon_{\text{rel}}|J(\bar{\mathbf{U}})|$
  if $\rho_{\text{feas}} \le \epsilon_{\text{feas}}$ **and** $\Gamma \le \beta$ **then**
    accept $\leftarrow$ TRUE; **break**
  end if
end for
if accept **then**
  $\mathbf{F}_c^{\text{ref}} \leftarrow \bar{\mathbf{U}}$ {certified hit}
else
  $(\mathbf{U}^\star, \boldsymbol{\lambda}^\star, J^\star, \mathbf{K}^\star) \leftarrow$ SolveMPC$(\mathbf{x}, \boldsymbol{\sigma})$
  insert into $\mathcal{H}_{\boldsymbol{\sigma}}$; $\mathbf{F}_c^{\text{ref}} \leftarrow \mathbf{U}^\star$
end if
forward $\mathbf{F}_c^{\text{ref}}$ to the KinoDynamic WBC

convergence. Periodic locomotion violates two of those assumptions: the closed-loop attractor is a phase-parameterised orbit rather than a point, and the applied torque passes through a WBC projection layer that the MPC does not model. A formal practical-orbital stability theorem under the per-query budget $\beta(\bar{\mathbf{U}})$ of (12) would require a phase-conditioned value function, recursive feasibility under contact-mode switches, a strong-convexity conversion from cost gap to control deviation (scaling as $\sqrt{\beta/\sigma_{\min}(\mathbf{H})}$ rather than linearly in β), and a disturbance model that subsumes WBC projection error, SRBD–plant mismatch, and state-estimator error.

Within the present paper the certificate plays an operational role: it bounds the per-query cost gap by $\beta(\bar{\mathbf{U}})$ exactly when (13) succeeds, and the proposal mechanism (cache, refiner, sensitivity correction) maximises the rate at which the controller stays inside the certified regime. The KinoDynamic WBC is the final feasibility filter for the full rigid-body dynamics and absorbs part of the residual model mismatch.

5 Experiments and Results

5.1 Experimental Setup

We evaluate CacheMPC on the Unitree Go2 quadruped robot in MuJoCo [13]. The convex predictive backend uses an SRBD model with horizon $N = 5$ at $\Delta t = 50$ ms ($T = 0.25$ s); the WBC runs at the proprioceptive estimator rate (250 Hz). Friction coefficient $\mu = 0.3$ and per-foot normal-force bounds $0 \le f_{z,i} \le 150$ N (the lower bound is set to 0 in the production configuration so that swing feet with $f_{i,z} = 0$ remain feasible). The LSH index uses $L = 35$ tables of $k = 5$ hash functions each, bucket width $w \propto r$ for an acceptance radius $r = 0.20$. The feature vector $\boldsymbol{\phi} \in \mathbb{R}^{18}$ matches (7): $\boldsymbol{\phi} = [\,\dot{\mathbf{p}}_B^\top \mid \dot{\mathbf{p}}_{\mathrm{des},B}^\top \mid ({}^{\mathcal{B}}\mathbf{r}_{p_1})^\top \mid \cdots \mid ({}^{\mathcal{B}}\mathbf{r}_{p_4})^\top\,]^\top$, with both velocities and foot positions expressed in the body frame; the desired CoM linear velocity is rotated into the body frame and concatenated with the desired yaw rate. Body-frame bucketing makes the index SE(2)-invariant.

Compute platforms.

Timing in Section 5 is reported on the development workstation (P-W). The on-robot platform (P-O), used in Section 6, is the NVIDIA Orin NX 16 GB module shipped on the Go2; the ARM64 Orin NX is approximately 3–5× slower than the P-W workstation on the inner QP solve and on Eigen-heavy assembly code. Per-stage breakdowns are wall-clock measurements taken inside the controller covering the entire MPC update call.

We evaluate four cache configurations on each of two downstream architectures. Each configuration is fully specified by three caching flags in the production controller's YAML configuration; Table 1 lists the runtime values of `use_certificate`, `use_sensitivity`, and `use_region_filter`. The region-filter flag defaults to `true` in the C++ controller, so Cache-Cert retains the region filter even when its YAML override is absent. Convex disables caching entirely.

Table 1 Cache configurations and their three runtime flags.

Configuration	Certificate	Sensitivity	Region filter
Convex (no cache)	—	—	—
Cache-NoCert	off	off	off
Cache-Cert	on	off	on
Cache-Full	on	on	on

The downstream architectures are (a) **WBC**, the KinoDynamic Whole-Body Controller of Section 3.2 that consumes the MPC's contact-force reference, and (b) **MRT**, a model-reference-tracking controller that maps the MPC's contact forces to joint torques directly via $\boldsymbol{\tau} = -\mathbf{J}_{aa}^\top \mathbf{F}_c$ with a swing-leg foot impedance term, removing the WBC's full-dynamics projection so that the MPC reference reaches joint torque without an absorption layer.

Trials are run on a cold-controller IID protocol: one fresh controller spawn per trial, paired pseudo-random seeds, and a cache file that is wiped at trial start. The simulation evaluation comprises four exploratory campaigns covering the eight (configuration, architecture) cells across velocity-sweep trot, a push ladder from 10 to 100 N, and three terrain scenarios (two stair heights and a 15° slope). Inspection of the exploratory $n = 15$ data identified three failure-boundary cells with the largest apparent cache-vs-no-cache separation; these cells were prospectively specified and then re-powered to $n = 50$ with fresh seeds in a 600-trial campaign. Combined with the exploratory campaigns, the full evaluation comprises 2,038 usable cold-controller trials (1,438 exploratory after two high-push convex trials were lost to a controller startup failure unrelated to caching, plus 600 re-power). A per-cell anomaly scanner taints trials emitting controller anomalies, NaN/Inf streams, or RMSE values above physically reasonable caps; all cells reported pass these checks. Per-campaign tabulations are provided in the supplementary material.

Hit-rate metrics.

The paper reports three hit-rate quantities, each labelled in context. The *raw LSH-neighbour hit rate* is the fraction of MPC ticks for which the LSH index returns at least one candidate within bucket width w. The *certified-acceptance rate* is the fraction for which a returned candidate additionally satisfies the certificate budget (13). The *controller hit-path rate* is the fraction for which the controller takes the cache-hit code path (no QP solve); it equals the certified-acceptance rate when the certificate is on, and the raw LSH-neighbour rate for Cache-NoCert.

Production tolerances.

The condensed Hessian is regularised with $w_{qp} = 10^{-2}$, preserving weak duality. The certificate is evaluated with $\boldsymbol{\lambda} = \mathbf{0}$ throughout. Certificate tolerances are $\epsilon_{\text{abs}} = 5.0$, $\epsilon_{\text{rel}} = 0.5$, $\epsilon_{\text{feas}} = 10^{-4}$, with a cert-region band of 25 N. Base proportional gains for (x, y, ψ) are zero because their references are velocity-integrated. Per-trial logged $|J(\bar{\mathbf{U}})|$ values lie in $[10, 250]$, giving $\beta \in [10, 130]$ across the trot scenario.

5.2 Solution Retrieval Time

Per-tick MPC update times on the trot scenario, summarised across $n = 15$ cold-controller trials per cell on the WBC architecture, are reported in Table 2. Speedups are per-trial p50 ratios against the matched-seed Convex baseline.

The un-gated cache attains the largest median speedup because its hit path skips both the certificate evaluation and the QP solve, reducing to an LSH lookup followed by the apply step. The cert-gated variants pay the certificate cost (Cholesky on $\mathbf{H}$ plus the dual-gap evaluation) on every tick and plateau at a moderate median speedup; their mean and tail are inflated by certificate-rejected misses that pay both the certificate and the QP solve. The per-trial p50 distributions for the cache variants are non-overlapping with the Convex baseline across the campaign. Hit-rate curves are shown in Fig. 2.

Table 2 Per-tick MPC update time on the trot scenario, $n = 15$ trials per cell. Entries are the across-trial median in microseconds.

Configuration	Mean	p50	p95	p99	Speedup
Convex	759	711	1086	1758	—
Cache-NoCert	184	**28**	1522	2281	**25.4**×
Cache-Cert	842	198	2213	3048	3.6×
Cache-Full	865	207	2105	2956	3.4×

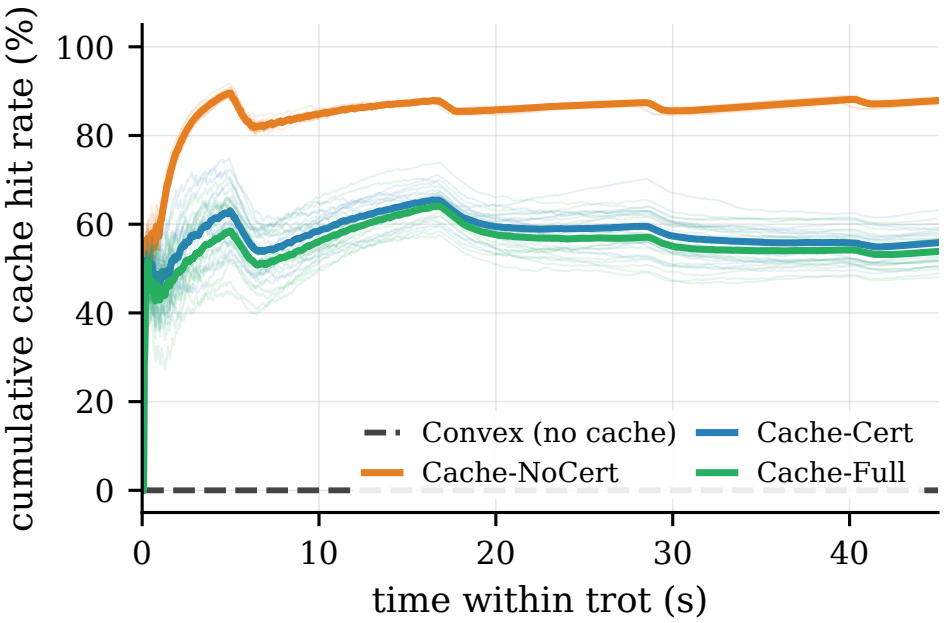


Fig. 2 Cumulative cache hit rate over time within trot, $n = 15$ trials. Faint traces are per-trial curves; bold lines are per-variant means. The Cache-NoCert plateau is the raw LSH-neighbour rate; the Cache-Cert and Cache-Full plateaus are the certified-acceptance rate.

5.3 Locomotion Performance

Tracking performance during the trot velocity sweep (0–0.6 m/s) is summarised in Table 3. Per-trial v_x, v_y, and combined velocity RMSE were aggregated across $n = 15$ matched-seed trials per configuration.

All four cache configurations track within ±14% of the matched-seed Convex baseline on velocity RMSE, the trial-to-trial spread overlaps across cells, and no trial produced a fall. Position RMSE varies widely across cells (bootstrap CI 0.6–1.2 m)

Table 3 Tracking RMSE on the 0–0.6 m/s trot velocity sweep, $n = 15$ trials per cell. All RMSE values in m/s.

Configuration	v_x RMSE	v_y RMSE	Vel RMSE	Stable
Convex (no cache)	0.116	0.067	0.098	15/15
Cache-NoCert	0.131	0.080	0.112	15/15
Cache-Cert	0.113	0.076	0.098	15/15
Cache-Full	0.133	0.067	0.110	15/15

because the base-frame proportional gains for (x, y, ψ) are zero in the production configuration and position drifts under velocity-integrated references; velocity RMSE is therefore the discriminating tracking metric.

5.4 Failure-Boundary Safety at $n = 50$

Closed-loop safety is characterised by whether the cache changes the stable rate at conditions that push the no-cache baseline to fail. The exploratory campaigns bracket two failure boundaries (WBC architecture between 80 and 100 N of lateral push; MRT architecture between 50 and 60 N, with an additional terrain failure at the 10 cm step height). The three cells with the largest apparent cache-vs-no-cache separation in the exploratory data were prospectively specified and re-powered to $n = 50$ with fresh seeds; Table 4 reports the result.

Table 4 Failure-boundary stable rate per configuration, $n = 50$ trials per cell. Fisher p-values are against the no-cache baseline of the same cell and are uncorrected; the Holm–Bonferroni threshold over the 12 comparisons is $\alpha/12 \approx 0.0042$, so the bold entry is not corrected-significant.

Cell	Configuration	Stable	Rate	Fisher p
WBC, 80 N lateral push				
	Convex (no cache)	19/50	38%	—
	Cache-NoCert	28/50	56%	0.109
	Cache-Cert	24/50	48%	0.419
	Cache-Full	31/50	**62%**	**0.027**
MRT, 50 N lateral push				
	Convex	10/50	20%	—
	Cache-NoCert	9/50	18%	1.000
	Cache-Cert	7/50	14%	0.595
	Cache-Full	6/50	12%	0.414
MRT, 10 cm step traversal				
	Convex	43/50	86%	—
	Cache-NoCert	44/50	88%	1.000
	Cache-Cert	43/50	86%	1.000
	Cache-Full	46/50	92%	0.525

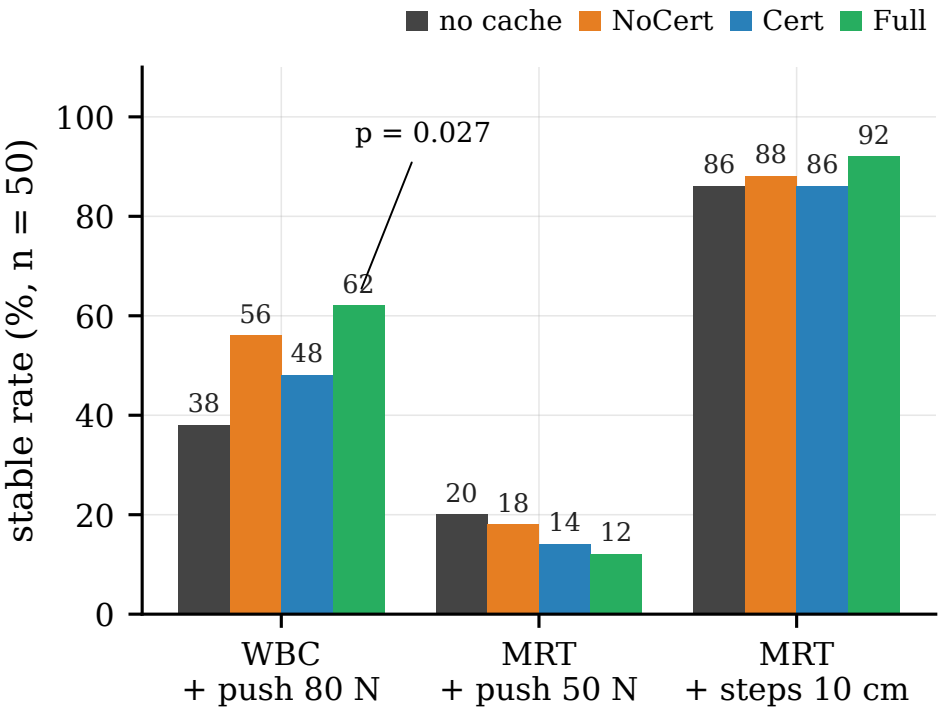


Fig. 3 Failure-boundary stable rate, $n = 50$ trials per cell, at three prospectively specified cells. The annotated $p = 0.027$ (Cache-Full vs. Convex on the WBC 80 N cell) is the only within-cell comparison below 0.05 and is uncorrected; it does not survive Holm–Bonferroni over the 12-comparison family.

No comparison between any cache variant and the no-cache baseline survives Holm–Bonferroni correction over the 12 cell–variant comparisons at $n = 50$. The largest within-cell separation in the table is descriptive only.

The operational statement supported by the data is that caching without a per-query gate does not measurably harm closed-loop stability at the failure boundaries tested. The certificate's runtime safety contribution is masked at this sample size by the downstream architecture — the WBC's full-dynamics projection and joint-level proportional-derivative loop on one side, the foot-impedance term on the MRT side. At the matched 10 N exploratory cell, all four configurations survive every trial and the per-trial v_y RMSE distributions overlap.

5.5 Bounded-Budget Stress Test

The schedule of Section 4.4 was exercised under deliberately tight deadlines on a single 45 s trot trial. Table 5 reports the outcome.

Table 5 Stress trial under tight deadlines on 45 s of trot. Schedule parameters: $K = 3$, $T_{\text{cache}} = 200\,\mu$s, $T_{\text{solve}} = 2000\,\mu$s. The closed-loop velocity RMSE is reported against a matched-seed unstressed trial.

Quantity	Value
Controller ticks (TROT)	2239
Cache-budget exhausts (fall to QP solve)	469
Solver-deadline overruns (fall to shifted fallback)	56
Total deadline-fallback ticks	525 (23.4%)
Vel RMSE (this trial)	0.079 m/s
Vel RMSE (matched unstressed trial)	0.078 m/s
Fall?	no

The schedule meets its wall-clock budget on every tick of the trial without a fall. This is descriptive single-trial evidence of behaviour under timing pressure, not a statistical comparison.

6 First Hardware Deployment

A first-deploy session was conducted on the Unitree Go2 with the on-robot NVIDIA Orin NX 16 GB module (P-O); the workstation used for the simulation campaign is denoted P-W. The cross-compiled controller binary follows the same code path as the simulation controller, with the hardware build enabling Unitree DDS communication and the on-robot proprioceptive estimator. The cache file is seeded from the simulation-trained Cache-Full cache obtained at the end of the WBC trot campaign. The session is an informal free-walk: the operator drives the robot through approximately 90 s of trot per configuration via the gamepad and applies 20–30 lateral pushes by hand. Push events are detected from $|v_y| > 0.3$ m/s during trot.

The scope of this session is limited: timing characterisation, qualitative push response, and confirmation that the simulation-trained cache transfers. The pushes are operator-applied and not controlled for force, direction, or timing across configurations, so the section reports qualitative free-walk evidence rather than a controlled comparative safety result.

Table 6 reports the headline numbers; Fig. 4 plots the per-tick update-time CDF.

The un-gated cache reaches the predicted on-robot p50 timing and shows no fall across the push events. The simulation-trained cache transfers: certified acceptance is non-zero on the first trot phase. Cache-Full is the inverted case relative to simulation, slower than Convex on the Orin NX, and its mechanism is summarised in Table 7.

The hardware inversion is consistent with the certificate of Proposition 1 acting as designed: Γ inflates above $\beta(\bar{\mathbf{U}})$ for the majority of LSH-neighbour candidates under the on-robot QP data, so most candidates are rejected and fall through to the full QP solve. Certificate-rejected ticks pay both the certificate evaluation and the QP solve, putting Cache-Full systematically above Convex.

Table 6 Hardware results on the Orin NX. The hit-rate row reports the raw LSH-neighbour rate for Cache-NoCert and the certified-acceptance rate for Cache-Full. Timing entries are in μs; push events use $|v_y| > 0.3$ m/s during trot.

Metric	Convex	Cache-NoCert	Cache-Full
Hit rate	—	55.8%	16.0%
Mean	1966	1324	2910
p50	1883	**101**	3342
p95	2478	3753	4267
p99	3856	5139	5741
Push events	21	32	20
Peak $\lvert v_y \rvert$ (m/s)	0.84	0.83	0.68
Min p_z (m)	0.301	0.296	0.282
Falls	0	0	0

Table 7 Cache-Full on the Orin NX: the certificate's high rejection rate explains the inversion of the simulation-measured speedup. The raw LSH-neighbour rate is taken from the matched Cache-NoCert run as a cross-run proxy; the derived rejection rate assumes equal raw-LSH rates across runs.

Quantity	Value
Raw LSH-neighbour rate (proxy, Cache-NoCert)	55.8%
Certified-acceptance rate (Cache-Full)	16.0%
Derived certificate rejection rate	$\approx$71%
Cache-Full p50 vs Convex p50 (sim)	**3.4**× faster
Cache-Full p50 vs Convex p50 (hardware)	1.8× slower

7 Threats to Validity

Single platform and dynamics setting.

The simulation campaign uses friction $\mu = 0.3$, the default MuJoCo contact solver, and flat or scripted-terrain ground. Within these conditions it covers two architectures (WBC, MRT), four cache configurations, a push ladder from 10 to 100 N, and two terrain types. Generalisation across payload, contact-model variation, friction coefficient, and quadruped platform is not characterised. The hardware session covers a single platform (Unitree Go2 + Orin NX) under operator-driven free-walk on flat ground.

Loose certificate bound.

The certificate is evaluated in production with $\boldsymbol{\lambda} = \mathbf{0}$, the unconstrained relaxation; this gives the loosest valid weak-duality bound on ΔJ. A cached-dual variant of the bound, which would be tight whenever the query shares the entry's active set, is not used in the present evaluation.

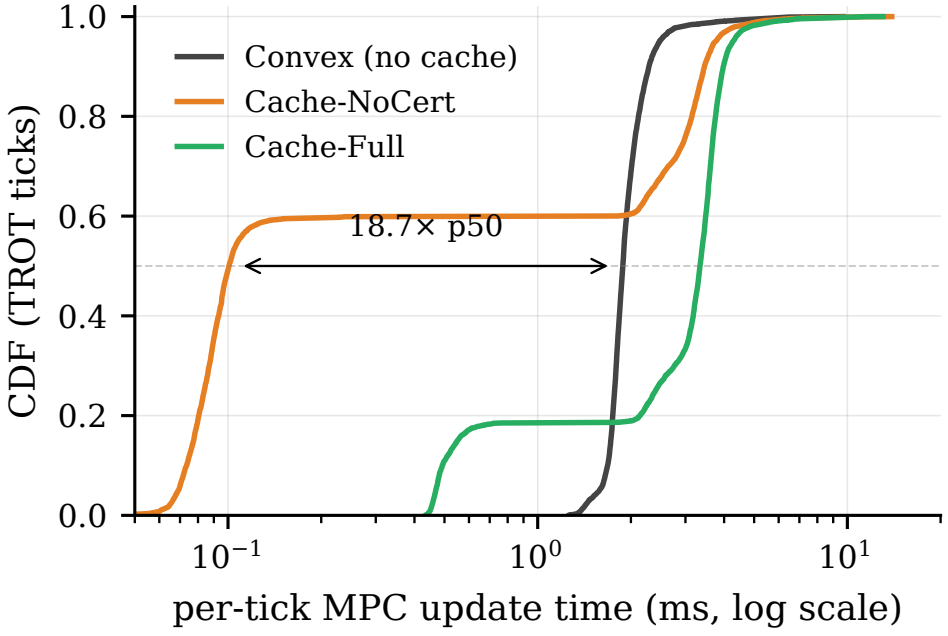


Fig. 4 Per-tick MPC update-time CDF on the Orin NX, trot ticks only. The 18.7× p50 Cache-NoCert speedup over Convex is visible at the 0.5 line; Cache-Full is bimodal with a certified-acceptance plateau near 0.5 ms and a climb at ∼3.5 ms from certificate-rejected ticks.

Single push direction.

The push tests use a single lateral direction, both at the simulation failure boundary and on hardware. A $(\pm x, \pm y)$ sweep at $n = 50$ would characterise the geometry of the boundary more fully.

Sample-size-bounded negative result.

The $n = 50$ Fisher p-values reported in Table 4 are uncorrected and do not establish equivalence: "no significant difference detected" is a statement about test power, not about the magnitude of the underlying effect. The certificate may carry measurable closed-loop value in regimes that the present evaluation cannot resolve.

Unit of analysis.

The timing distributions in Table 2 are aggregated within trial and summarised across trials; per-tick hypothesis testing is avoided because per-tick samples within a trial are autocorrelated. The per-trial p50 ratios are descriptive across $n = 15$ independent cold-controller spawns per cell.

Baseline coverage.

The cache variants are compared only against the no-cache Convex baseline of the same MPC implementation. Warm-started convex MPC, kd-tree nearest-neighbour retrieval [27], fixed offline dictionaries, and learned approximate-MPC policies [5, 23] are not included in the present comparison.

Reproducibility protocol.

Every cold-controller trial records the binary SHA-256, the YAML configuration that ran, the per-trial cache file, the per-tick controller log, and the scenario time-series log. Per-cell trial logs are gated by an anomaly scanner with pre-specified rejection criteria (controller anomalies, NaN/Inf streams, RMSE values above physically reasonable caps); all $n = 50$ re-power cells and all reported exploratory cells pass these checks. The same controller binary path is used in simulation and on hardware; only the communication and estimator backends differ. Aggregate per-cell tabulations, per-knob sweep tables, and the full hardware-session breakdown are available from the authors on reasonable request.

8 Conclusion

This paper presented Certified CacheMPC, a memory-augmented MPC framework in which an LSH-indexed cache returns previously computed contact-force trajectories and an a-posteriori per-query certificate produces a Lagrangian dual-gap quantity $\Gamma(\bar{\mathbf{U}})$ that upper-bounds the true suboptimality ΔJ of any feasible accepted retrieval. The certificate is deterministic and source-independent. A bounded-budget controller schedule combines top-K certified retrieval, a deadline-bounded QP solve, and a shifted last-certified fallback.

The framework was evaluated through a cold-controller simulation campaign and a first-deploy session on the on-robot NVIDIA Orin NX; the headline numbers are in Tables 2–7. The un-gated cache produces a large median solve-time speedup in simulation that transfers to hardware. No statistically significant difference between the cache variants and the no-cache baseline is detected at $n = 50$ at any tested failure-boundary cell. The certificate functions as a verifiable upper bound on the cost gap of any applied control, rather than as a runtime safety mechanism whose effect the present evaluation can resolve.

9 Future Work

The directions opened by this work are grouped into four families. *Tighter certified bounds.* Reuse of cached multipliers $(\boldsymbol{\lambda}_\ell, \boldsymbol{\lambda}_u)$ would tighten Γ towards ΔJ whenever the query shares the entry's active set, given a guarded staleness check; a Hoffman-bound perturbation analysis would extend Proposition 1 to $\rho_{\text{feas}} > 0$. *Formal closed-loop theory.* A practical-orbital stability theorem on top of the per-query budget $\beta(\bar{\mathbf{U}})$ and a worst-case real-time theorem on top of the bounded-budget schedule of Section 4.4 are natural next steps, requiring a phase-conditioned value function, recursive feasibility under contact-mode switches, and a preemption-safe inner solver. *Empirical*

breadth. A constant-velocity hardware tracking matrix, a powered non-inferiority hardware safety matrix, $(\pm x, \pm y)$ push sweeps at $n = 50$, and richer terrain (steps, slopes, granular ground) are natural next campaigns. *Baselines and cache lifecycle.* Warm-started convex MPC, kd-tree neighbour search, fixed offline dictionaries, and learned approximate-MPC policies provide stronger reference points; on-robot fine-tuning and online refinement of the cache are expected to reduce the certificate rejection rate observed on hardware.

Acknowledgements. We thank the MRL Lab at IIT Kanpur for providing the resources to conduct the research and experiments for this paper.

Declarations

- **Funding**: Not applicable.
- **Competing interests**: The authors declare no competing interests.
- **Ethics approval and consent to participate**: Not applicable.
- **Consent for publication**: Not applicable.
- **Data availability**: Aggregate per-cell tabulations, per-knob sweep tables, and the full hardware-session breakdown are available from the authors on reasonable request.
- **Materials availability**: Not applicable.
- **Code availability**: Available from the authors on reasonable request.
- **Author contributions**: N. Khandelwal conceived the main idea of the Certified CacheMPC framework, implemented the predictive and reactive controllers and their integration, developed the per-query certificate, designed and conducted the simulation and hardware experiments, performed the data collection and processing, and wrote the primary manuscript draft. M. Anand implemented the LSH-based caching and retrieval module. S. S. Gupta and M. Kothari provided technical supervision, editorial oversight, and final manuscript revisions.

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," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 1–9.

[2] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A platform for pushing the limits of dynamic quadruped control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6295–6301.

[3] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4688–4695, 2021.

[4] K. N. Nguyen *et al.*, "TinyMPC: Model-predictive control on resource-constrained microcontrollers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[5] B. Carius, R. Ranftl, V. Koltun, and M. Hutter, "MPC-Net: A first principles guided policy search," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2897–2904, 2020.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[7] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. 30th Annu. ACM Symp. Theory Comput. (STOC)*, 1998, pp. 604–613.

[8] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.

[9] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," in *Proc. IFAC World Congr.*, 2020.

[10] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2016, pp. 558–564.

[11] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Robot.*, vol. 24, no. 5–6, pp. 719–737, 2010.

[12] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, 2006.

[13] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 5026–5033.

[14] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Trans. Autom. Control*, vol. 44, no. 3, pp. 648–654, 1999.

[15] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Conditions under which suboptimal nonlinear MPC is inherently robust," *Syst. Control Lett.*, vol. 60, no. 9, pp. 747–755, 2011.

[16] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 2245–2252.

[17] 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," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 38–44.

[18] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, p. eaau5872, 2019.

[19] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Proc. Conf. Robot Learn. (CoRL)*, 2022, pp. 22–31.

[20] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[21] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, 2022.

[22] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[23] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 543–548, 2018.

[24] J. A. Paulson and A. Mesbah, "Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction," *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 719–724, 2020.

[25] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geom. (SoCG)*, 2004, pp. 253–262.

[26] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[27] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[28] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *Int. J. Humanoid Robot.*, vol. 2, no. 4, pp. 505–518, 2005.

[29] 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 preprint arXiv:1909.06586, 2019.

[30] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for torque-controlled legged robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 4730–4737.

[31] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 344–357, 2017.

[32] D. Limón, T. Alamo, F. Salas, and E. F. Camacho, "Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties," *Automatica*, vol. 42, no. 5, pp. 797–803, 2006.

[33] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.