REAL-TIME INTELLIGENT CONTROL OF MULTIVARIABLE MOBILE ROBOT SYSTEMS: CONSTRAINED MODEL PREDICTIVE CONTROL WITH AN ADAPTIVE MODEL-CORRECTION LAYER

Sherzod A. Yarashev
Tashkent University
of Information Technologies named after
Muhammad al-Khwarizmi, Tashkent, Uzbekistan
ORCID: 0009-0009-5053-942X
Jurabek M. Abdiev
Arifov Institute of Ion-Plasma and Laser
Technologies, Academy of Sciences of the
Republic of Uzbekistan, Tashkent, Uzbekistan
E-mail: fiztexabdiev@gmail.com
ORCID: 0000-0003-4524-3439
https://doi.org/10.5281/zenodo.23906605

Abstract. Background: A mobile robot must compute constrained control actions within a fixed sampling interval, yet predictive control, model correction and solver timing are rarely studied together. **Objective:** To study constrained model predictive control (MPC) of a multivariable differential-drive robot as a real-time problem. **Methods:** A five-state, two-input torque model is analysed with the relative gain array. A linear time-varying MPC with input, rate and softened state constraints is solved as a sparse quadratic program with OSQP, with bounded recursive-least-squares gain-and-bias correction, a safeguard and a deadline fallback. Twelve simulated conditions (30 seeds each; gain 0.7 to 1.2, bias, disturbances, delay, 0 to 10 % packet loss) compared the MPC with a decoupled cascaded PID. **Results:** In the disturbed conditions the MPC lowered the position RMS error by 2.0 cm (95 % CI 2.0 to 2.1) and the residual after 5 s from 2.5 to 1.1 cm, but used 1.4 times the torque effort. The adaptive layer removed a further 0.21 cm of the residual (19 %) without detectable harm at ten times the sensor noise. In Python on a personal computer, a horizon of 20 samples needed 1.1 ms per sample (mean; 99th percentile 2.9 ms); at a 20 ms period the deadline was missed in at most 0.04 % of the samples for horizons up to 40. **Conclusions:** With a terminal weight, short-horizon constrained MPC fits these sampling rates and outperforms the PID at the price of rougher actuation. All results are simulations; hardware and embedded timing remain untested.

Keywords: multivariable mobile robot; model predictive control; real-time feasibility; adaptive model correction; OSQP; relative gain array; simulation study

1. Introduction

Autonomous mobile robots are deployed in logistics, manufacturing, inspection and agriculture, as one component of the wider move towards automated and controlled industrial processes. The controller must turn a state estimate into a command before a hard sampling deadline, under noise, delay, actuator limits and changing payload. A differential-drive robot is multivariable even in its simplest form: speed and yaw rate come from the same two wheel torques, so the forward and lateral channels are coupled through the manoeuvre, with a strength that changes with the turning rate. The usual response is a cascade of single-loop controllers tuned separately [1, 2], which is cheap but cannot anticipate saturation or use the reference to look ahead.

Model predictive control (MPC) solves at each sample a finite-horizon problem containing the model, reference and constraints, applies the first input and repeats [3–5]. Its industrial use is long established [6, 7], and predictive controllers on a model linearised along the reference have been reported for vehicles and mobile robots [8, 9], including a learning-based variant with robust constraints [10]. Kinematic tracking laws, chained-system control and dynamic windows need almost no computation [11–14]. Two difficulties limit MPC on small robots: model uncertainty, since the delivered torque differs from the commanded one (adaptive and learning-based MPC update the model online [15–18]); and real-time feasibility, since the optimisation must finish inside the sampling interval and its timing interacts with the loop [19, 20]. Fast solvers, code generators, explicit solutions and real-time iterations address the latter [21–27], but the dependence of solve time on horizon, constraints and solver settings, and the closed-loop cost of a missed deadline, are rarely reported together with accuracy.

This paper studies constrained predictive control of a multivariable differential-drive robot as a real-time problem. Its contributions are: (i) a torque-input state-space model with a coupling analysis by the relative gain array [28, 29]; (ii) a linear time-varying MPC (LTV-MPC) around the reference with input, rate and softened state constraints, written as a sparse quadratic program (QP), with a statement of what is and is not guaranteed in the light of Mayne et al. [30]; (iii) a bounded recursive-least-squares (RLS) layer estimating actuator gain and bias, with a model-selection safeguard; (iv) a real-time analysis with measured OSQP solve times and a deadline fallback; and (v) a pre-specified simulation study (30 seeds per condition) of a decoupled cascaded PID (R1), a nominal MPC (R2) and the MPC with the adaptive layer (R3), including where the layer does not help. All results are from simulation; no hardware experiment was performed, and timing is for Python on a personal computer, not an embedded target.

2. Materials and Methods

2.1 Robot and plant model

The robot has wheel radius r, track width L, mass m, yaw inertia I and viscous drag b_v, b_ω. With state $\mathbf{x} = [x, y, \theta, v, \omega]^\top$ (position, heading, speed, yaw rate), wheel torques $\mathbf{u} = [\tau_R, \tau_L]^\top$ and generalised torques $\eta_v = \tau_R + \tau_L$, $\eta_\omega = \tau_R - \tau_L$, the dynamics are

$$\dot{\mathbf{x}} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \\ -\alpha_v v + \beta_v \eta_v \\ -\alpha_\omega \omega + \beta_\omega \eta_\omega \end{bmatrix}, \qquad \begin{bmatrix} \eta_v \\ \eta_\omega \end{bmatrix} = \mathbf{T}\,\mathbf{u}, \quad \mathbf{T} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1}$$

with $\alpha_v = b_v/m$, $\alpha_\omega = b_\omega/I$, $\beta_v = 1/(mr)$, $\beta_\omega = L/(2rI)$; the first three rows are the standard unicycle kinematics [1, 5]. Measurements are $\mathbf{y}_k = \mathbf{x}(kT_s) + \mathbf{n}_k$ with Gaussian noise and sampling period T_s. The plant applies

$$\boldsymbol{\eta}^{act}(t) = g\,\boldsymbol{\eta} + \delta\,\mathbf{J}\,\boldsymbol{\eta} + \mathbf{c} + \mathbf{d}(t), \qquad \mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2}$$

where g is a common actuator gain (payload, slip), δ a small left-right asymmetry coupling the channels, $\mathbf{c} = [c_v, c_\omega]^\top$ a bias and $\mathbf{d}(t)$ step and sinusoidal disturbances. The controllers know only the nominal model ($g = 1$, $\delta = 0$, $\mathbf{c} = \mathbf{d} = \mathbf{0}$); the adaptive layer estimates g and $\mathbf{c}$ but not δ or $\mathbf{d}$, so a mismatch remains by design. A command computed at sample k is applied from $k + 1$ (one-sample delay); with probability p it is lost and the actuator holds the previous command, without notice to the controller.

2.2 Reference, coupling analysis and design of the controllers

Three 20 s references at $v_r = 0.6$ m/s are used: a straight line, an S-curve (opposite 90-degree arcs) and a figure-eight (two tangent circles of radius 0.8 m), with smooth-step yaw-rate pulses. The torques that reproduce the reference under the nominal model are

$$\eta_v^r = \frac{\alpha_v\, v_r}{\beta_v}, \qquad \eta_\omega^r = \frac{\dot{\omega}_r + \alpha_\omega\, \omega_r}{\beta_\omega}, \qquad \mathbf{u}_r = \frac{1}{2}\mathbf{T}\,\boldsymbol{\eta}^r \tag{3}$$

The S-curve and figure-eight need up to 0.40 and 0.37 N m per wheel against a limit of 0.5 N m; compensating a gain of 0.7 needs 0.57 and 0.53 N m, so the limit becomes active in the disturbed runs. At the level of wheel torques, the static gain matrix to (v, ω) is $\mathbf{G}(0)$ with entries $\pm 1/(rb_v)$ and $\pm L/(2rb_\omega)$, and the relative gain array (RGA) [28, 29] element is

$$\boldsymbol{\Lambda} = \mathbf{G} \circ (\mathbf{G}^{-1})^\top, \qquad \lambda_{11} = \left(1 - \frac{g_{12}\, g_{21}}{g_{11}\, g_{22}}\right)^{-1} = \frac{1}{2} \tag{4}$$

for any parameters, because the two torques enter the speed channel with equal sign and the yaw channel with opposite sign; pairing one wheel with one output gives maximal interaction, and a decoupled design must act on $\boldsymbol{\eta}$, where the static RGA is the identity. In a path-aligned frame (errors e_s, e_{ct}), linearisation at a frozen reference yaw rate gives terms $\pm\omega_r$ that rotate the errors into each other in a turn; the RGA of the transfer matrix from (η_v, η_ω) to (e_s, e_{ct}) is evaluated numerically at $s = 0.1 + j\omega$ (the real part avoids the poles $\pm j\omega_r$).

2.3 Linear time-varying MPC

Let $\mathbf{e} = [x - x_r,\ y - y_r,\ \theta - \theta_r,\ v - v_r,\ \omega - \omega_r]^\top$ (heading difference wrapped). Linearising Eq. (1) about the reference at sample j and discretising exactly with a zero-order hold (one matrix exponential per reference sample, precomputed) gives $\mathbf{A}_j, \mathbf{B}_j$, and with the estimates $(\hat{g}, \hat{\mathbf{c}})$ ($\hat{g} = 1$, $\hat{\mathbf{c}} = \mathbf{0}$ for R2) the prediction model is

$$\mathbf{e}_{i+1} = \mathbf{A}_{j+i}\,\mathbf{e}_i + \mathbf{B}_{j+i}(\hat{g}\,\mathbf{T}\,\mathbf{u}_i + \hat{\mathbf{c}} - \boldsymbol{\eta}_{j+i}^r), \qquad i = 0, \dots, N-1 \tag{5}$$

where $\mathbf{e}_0$ is the error predicted for sample $k+1$ from the state estimate and the command in transit. The cost, with weights $\mathbf{Q} \succeq 0$, $\mathbf{R} \succ 0$, $\mathbf{S}_\Delta \succeq 0$, terminal weight $\mathbf{P}_f$, feed-forward command $\mathbf{u}_i^{ff} = \frac{1}{2}\hat{g}^{-1}\mathbf{T}(\boldsymbol{\eta}_{j+i}^r - \hat{\mathbf{c}})$ and slacks $\mathbf{s}_i \geq 0$, is

$$\begin{aligned} J = {} & \frac{1}{2}\sum_{i=1}^{N-1} \mathbf{e}_i^\top \mathbf{Q}\,\mathbf{e}_i + \frac{1}{2}\,\mathbf{e}_N^\top \mathbf{P}_f\,\mathbf{e}_N \\ & + \frac{1}{2}\sum_{i=0}^{N-1} (\|\mathbf{u}_i - \mathbf{u}_i^{ff}\|_{\mathbf{R}}^2 + \|\mathbf{u}_i - \mathbf{u}_{i-1}\|_{\mathbf{S}_\Delta}^2) \\ & + \sum_{i=1}^{N} (w_s\,\mathbf{1}^\top \mathbf{s}_i + \frac{\rho_s}{2}\,\|\mathbf{s}_i\|^2) \end{aligned} \tag{6}$$

subject to input, increment and softened state constraints

$$\begin{aligned} -\tau_{max}\mathbf{1} \leq \mathbf{u}_i \leq \tau_{max}\mathbf{1}, \qquad -\bar{\Delta}\mathbf{1} \leq \mathbf{u}_i - \mathbf{u}_{i-1} \leq \bar{\Delta}\mathbf{1}, \qquad \underline{\mathbf{h}}_i - \mathbf{s}_i \\ \leq \mathbf{H}_i\,\mathbf{e}_i \leq \bar{\mathbf{h}}_i + \mathbf{s}_i \end{aligned} \tag{7}$$

where $\bar{\Delta} = \dot{\tau}_{max} T_s$ and the rows of $\mathbf{H}_i$ select the speed error, the yaw-rate error and the cross-track error $e_{ct} = -\sin\theta_r e_x + \cos\theta_r e_y$ (limits in Table 1). With $\mathbf{z} = \mathrm{col}(\mathbf{e}_{1..N}, \mathbf{u}_{0..N-1}, \mathbf{s}_{1..N}) \in \mathbb{R}^{10N}$ the problem is the sparse QP

$$\min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^\top \mathbf{P}\,\mathbf{z} + \mathbf{q}^\top \mathbf{z} \quad \text{s.t.} \quad \mathbf{l} \le \mathbf{A}_{qp}\,\mathbf{z} \le \mathbf{u}_{qp} \tag{8}$$

with $\mathbf{A}_{qp}$ of 18N rows: 5N dynamics equalities (Eq. (5), initial error on the right-hand side), 2N input bounds, 2N increment bounds, 6N two-sided state rows and 3N slack signs ($N = 20$: 200 variables, 360 rows). The sparsity pattern is fixed; only $\mathbf{A}_{j+i}, \mathbf{B}_{j+i}$, the cross-track coefficients, $\mathbf{P}_f$ and the vectors change, so the solver only receives a numerical update. The terminal weight $\mathbf{P}_f$ solves the Riccati equation of the frozen pair at the horizon end, with $\mathbf{B}_g = \mathbf{BT}$:

$$\mathbf{P}_f = \mathbf{Q} + \mathbf{A}^\top \mathbf{P}_f \mathbf{A} - \mathbf{A}^\top \mathbf{P}_f \mathbf{B}_g (\mathbf{R} + \mathbf{B}_g^\top \mathbf{P}_f \mathbf{B}_g)^{-1} \mathbf{B}_g^\top \mathbf{P}_f \mathbf{A} \tag{9}$$

What is guaranteed. Stability theory of constrained MPC requires a terminal cost that is a local control Lyapunov function and an invariant terminal set [3, 30]. Here only the following is proved. (a) The QP is feasible at every sample for every $\mathbf{e}_0$: the sequence $\mathbf{u}_i = \mathbf{u}_{-1}$ satisfies the input and increment bounds and the slacks absorb the state rows, so recursive feasibility holds trivially, at the price that state constraints are not hard. (b) With no active constraint and a frozen pair, the Riccati-weighted problem reproduces the infinite-horizon LQR, which stabilises the frozen error system. (c) Nothing is proved for the time-varying, constrained, adaptive, delayed closed loop (no terminal set, no bound on the linearisation error); its behaviour is established empirically.

2.4 Adaptive layer and state estimation

The layer estimates $\boldsymbol{\theta} = [g, c_v, c_\omega]^\top$. The velocity subsystems are linear, so with $a_v = e^{-\alpha_v T_s}$ and $\gamma_v = \beta_v(1 - a_v)/\alpha_v$ (the yaw channel is analogous) the one-step relation gives the regressions

$$z_{v,k} = \frac{y_{v,k} - a_v\, y_{v,k-1}}{\gamma_v} = g\,\eta_{v,k-1} + c_v + \varepsilon_{v,k}, \qquad z_{\omega,k} = g\,\eta_{\omega,k-1} + c_\omega + \varepsilon_{\omega,k} \tag{10}$$

where $\eta_{\cdot,k-1}$ is the command believed applied (that of sample $k-2$; a lost packet makes the belief wrong and appears as noise). RLS with forgetting $\lambda = e^{-T_s/\tau_f}$ is applied to both scalar equations, with scaled regressors $\boldsymbol{\varphi}$:

$$\mathbf{K}_k = \frac{\mathbf{P}_{k-1}\boldsymbol{\varphi}}{\lambda + \boldsymbol{\varphi}^\top \mathbf{P}_{k-1}\boldsymbol{\varphi}}, \quad \hat{\boldsymbol{\theta}}_k = \Pi(\hat{\boldsymbol{\theta}}_{k-1} + \mathbf{K}_k(z - \boldsymbol{\varphi}^\top \hat{\boldsymbol{\theta}}_{k-1})), \quad \mathbf{P}_k = \frac{1}{\lambda}(\mathbf{I} - \mathbf{K}_k \boldsymbol{\varphi}^\top)\mathbf{P}_{k-1} \tag{11}$$

Πprojects onto $g \in [0.5, 1.5]$, $|c_v| \le 0.25$ N m, $|c_\omega| \le 0.15$ N m, and the diagonal of $\mathbf{P}$ is capped so that the covariance cannot wind up when the regressor lacks excitation (a straight line at constant speed makes g and c_v inseparable); bounded-covariance RLS addresses the same issue [12]. A safeguard compares exponentially weighted (3 s) squared scaled prediction errors of the adapted model, J_a, and the nominal model, J_n, and passes the estimate to the MPC only if $J_a < 0.9 J_n$; otherwise $(1, \mathbf{0})$ is used. It is a model-selection rule, not a stability guarantee; R3u denotes R3 without it. The state estimate comes from a fixed-gain predictor-corrector observer, a steady-state simplification of the Kalman filter [6, 7]: the previous estimate is propagated through the corrected nonlinear model with the command believed applied, and the innovation is multiplied by $\mathbf{L}_o = \mathrm{diag}(0.6, 0.6, 0.6, 0.5, 0.5)$, set by hand without a consistency test.

2.5 Real-time cycle and analysis

The loop is: reference, then LTV-MPC (sparse QP solved by OSQP), then safety filter (clipping, rate limit, fallback), then delay and packet loss, then actuators and plant with unknown gain, bias and disturbance, then sensors and the fixed-gain observer back to the MPC; the adaptive layer estimates gain and bias from the measured velocities and applied commands and feeds them to the MPC model and the observer. Algorithm 1 gives the cycle.

Algorithm 1. Real-time control cycle at sample k **(period** T_s**, deadline** T_{dl}**, horizon** N**)**

1. Acquire $\mathbf{y}_k$; start timer. 2. Observer: $\hat{\mathbf{x}}_k$ from $\mathbf{y}_k$ and the command believed applied.
3. Adapt: $(\hat{\mathbf{g}}, \hat{\mathbf{c}})$ by Eq. (11) and the safeguard (nominal for R1, R2).
4. Delay compensation: $\mathbf{e}_0 \leftarrow$ wrapped error of the one-step prediction from $\hat{\mathbf{x}}_k$ with command $\mathbf{u}_{k-1}$ against $\mathbf{x}_{r,k+1}$.
5. Update the QP data in place (warm start from the previous solution). 6. Solve, with iteration cap and time limit T_{dl}.
7. **If** solved in time: $\mathbf{u}_k \leftarrow \mathbf{u}_0^{star}$, store the plan, age $a \leftarrow 0$. **Else:** if $a + 1 < N$ use the shifted old plan, $\mathbf{u}_k \leftarrow \mathbf{U}_{a+1}^{star}$; otherwise coast down, $\mathbf{u}_k \leftarrow \mathbf{u}_{k-1} - \mathrm{clip}(\mathbf{u}_{k-1}, \pm\bar{\Delta})$.
8. Safety filter: clip to $[-\tau_{max}, \tau_{max}]$ and $\mathbf{u}_{k-1} \pm \bar{\Delta}$; record the request violation. 9. Transmit; log timing; $k \leftarrow k + 1$.

In the closed-loop simulations the deadline is represented by the solver's iteration cap (4000) and by injected misses, not by a wall-clock timeout, so accuracy results do not depend on machine speed; the wall-clock behaviour is measured separately by running the loop as fast as possible and comparing each sample's computation time with the deadline. The computation time is the sum of observer, adaptation, QP assembly, solver update and solve, and the deadline-miss fraction is

$$\rho_{dl}(T_{dl}) = \frac{1}{K}\sum_{k=1}^{K} 1[t_k^{cmp} > T_{dl}], \qquad T_{dl} = T_s \text{ or } \frac{1}{2}T_s \tag{12}$$

OSQP is an operator-splitting (ADMM) method with one fixed quasi-definite KKT solve per iteration [22, 23]; with the sparse formulation the factorisation and iteration scale roughly linearly in N, whereas condensing yields a dense $2N \times 2N$ Hessian with cubic scaling [4, 21]. Timing was measured serially in one Python process on the machine of Table 1, with garbage collection disabled and raised process priority; the machine was shared, so unrelated processes were running and the figures include their scheduling jitter. Five seeds of the disturbed S-curve (2000 to 5000 samples per cell) were used. They are Python/OSQP timings on a personal computer, not embedded-target measurements.

2.6 Simulation design, metrics and statistics

Table 1. Simulation settings used for all reported results.

Item	Value
Robot, limits	$m = 12$ kg, $I = 0.6$ kg m^2, $r = 0.08$ m, $L = 0.40$ m, $b_v = 4$ N s/m, $b_\omega = 0.4$ N m s/rad; $\tau_{max} = 0.5$ N m, $\dot{\tau}_{max} = 4$ N m/s; $v \in [0,0.9]$ m/s, $\lvert\omega\rvert \le 1.2$ rad/s, $\lvert e_{ct}\rvert \le 0.35$ m
Mismatch (disturbed)	$g \sim U(0.7,1.2)$, $\delta \sim U(-0.05,0.05)$, $c_v \sim N(0,0.04^2)$, $c_\omega \sim N(0,0.02^2)$ N m; step in η_v at 10 s, $\pm U(0.04,0.08)$ N m; sinusoid in η_ω, 0.4 Hz, $U(0.03,0.06)$ N m. Benign: $g = 1$, no bias, no disturbance
Delay, loss, noise	one sample; $p \in \{0,5,10\}$ % (nested patterns); noise s.d. $(0.02,0.02,0.01,0.01,0.01)$ for x, y (m), θ (rad), v (m/s), ω (rad/s)

Item	Value
Initial error	along $U(-0.1,0.1)$ m; cross $\pm U(0.15,0.30)$ m; heading $U(-0.15,0.15)$ rad; velocities $U(-0.05,0.05)$
Sampling, MPC	$T_s \in \{20,50\}$ ms, $N \in \{5,\dots,40\}$ (main: 50 ms, $N = 20$); $\mathbf{Q} = \mathrm{diag}(q, q, q/4, 10, 1)$, $q = 200$; $\mathbf{R} = 0.5\mathbf{I}$; $\mathbf{S}_\Delta = 0.5\mathbf{I}$; $w_s = 10^3$, $\rho_s = 10^2$
PID, RLS	outer $k_{ps} = 0.85$, $k_{is} = 0.012$, $k_{ct} = 2.99$, $k_\theta = 3.28$; speed loop $k_{pv} = 4.68$, $k_{iv} = 0.88$, $k_{dv} = 0.05$; yaw loop $k_{p\omega} = 0.85$, $k_{i\omega} = 0.076$, $k_{d\omega} = 0.01$; RLS $\tau_f = 6$ s, prior s.d. (0.3,0.2,0.1)
OSQP, machine	v1.1.3, $\varepsilon_{abs} = \varepsilon_{rel} = 10^{-4}$, ≤ 4000 iterations, adaptive ρ, warm start; Intel Core i3-1115G4 (3.0 GHz), Windows 11, Python 3.14.4, NumPy 2.4.4, SciPy 1.17.1

R1 is a decoupled cascade: a PI loop on the along-track error gives a speed set-point, a proportional law on cross-track and heading errors gives a yaw-rate set-point in the manner of [11], and two independent PID velocity loops with static drag feed-forward give η_v, η_ω, passed through the same safety filter; it has no reference-acceleration feed-forward. The main study has 12 conditions (three trajectories benign; three trajectories disturbed at 0, 5, 10 % loss), each with 30 seeds (0 to 29) whose random draws are shared by all controllers (paired). PID gains (60 random candidates), MPC weights (24-point grid) and τ_f (four values) were tuned on separate seeds (100 to 107) in three disturbed conditions; test seeds were never used. Further studies: a horizon sweep at both periods on the disturbed figure-eight (R3 vs R1; also without terminal weight); a noise stress test (noise × 1,3,6,10; straight line and figure-eight; benign and disturbed; R2, R3, R3u); and injected deadline misses (0 to 50 %, disturbed S-curve, R3).

Table 2. Evaluation metrics.

Metric	Definition
Position RMS (cm)	$\sqrt{\frac{1}{K}\sum_k((x_k - x_{r,k})^2 + (y_k - y_{r,k})^2)}$, whole run (primary) and for $t \geq 5$ s
Settling (s); overshoot (%)	last time the position error exceeds 0.10 m (20 s if never settled); largest opposite cross-track excursion after the first zero crossing, relative to the initial one
State / request violation (%)	largest excess of v, $\lvert\omega\rvert$, $\lvert e_{ct}\rvert$ over its limit (true plant state) / of command or increment before the safety filter, relative to the limit
Effort (N m); roughness (%)	$\sqrt{\frac{1}{K}\sum_k(\tau_{R,k}^2 + \tau_{L,k}^2)}$; mean $\lvert\Delta\mathbf{u}_k\rvert$ relative to $\bar{\Delta}$
Success (%)	no corridor violation, speed excess ≤ 5 %, mean error < 0.10 m in the last 2 s
Timing (ms)	mean, p95, p99, max of QP solve and total per sample; miss fraction Eq. (12)

The primary comparisons, fixed in advance, are paired differences of the whole-run position RMS between R2 and R1, R3 and R2, and R3 and R1, per condition and pooled over the disturbed conditions; secondary outcomes are the metrics above. For $n = 30$ pairs the mean difference has the interval

$$\bar{d} \pm t_{0.975,\,n-1}\,\frac{s_d}{\sqrt{n}} \qquad (13)$$

and success rates carry Wilson intervals. No multiplicity correction is applied; intervals are descriptive, and pooled intervals treat seeds of different conditions as independent pairs. Large language

models (Claude, Anthropic; accessed through Claude Code) were used under the authors' direction for language editing, structuring and for writing and running the simulation code reported in Sections 2 and 3.

3. Results

3.1 Coupling and verification

Figure 1 shows the references and the coupling. The static RGA has $\lambda_{11} = 0.5$ and the gain matrix a condition number of 2.0. In the path-aligned model the interaction is absent on a straight line ($\lambda_{11} = 1.00$) and, along the figure-eight at 0.5 rad/s, is 0.77 on the steady circles and peaks at 2.60 where $|\omega_r|$ passes the evaluation frequency (Fig. 1c): a 23 % deviation from the decoupled value in the steady turn and more than a factor of two at transitions, which motivates a model that contains the coupling.

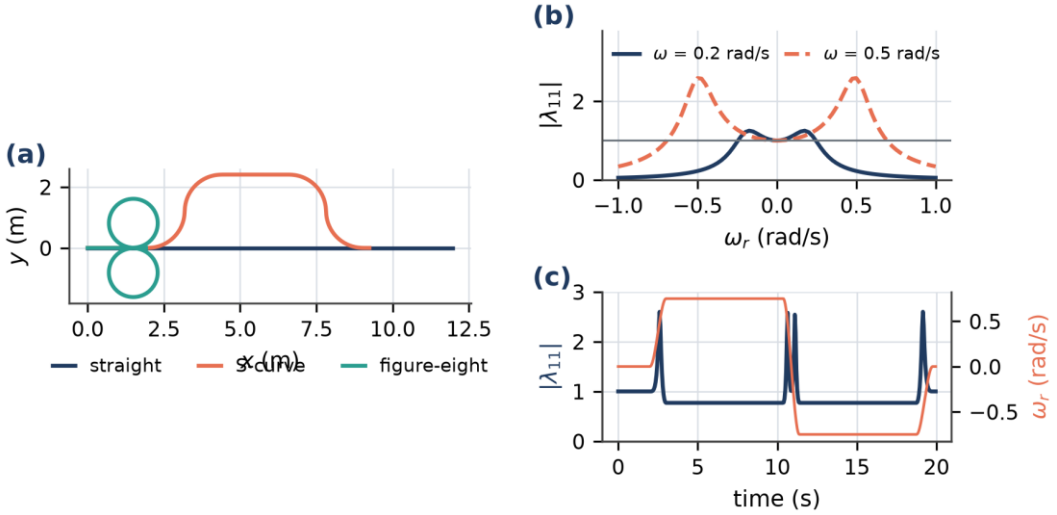


Fig. 1. (a) References. (b) $|\lambda_{11}|$ of the path-aligned error model against the frozen reference yaw rate ω_r at two frequencies ($s = 0.1 + j\omega$). (c) $|\lambda_{11}|$ at 0.5 rad/s along the figure-eight with the reference yaw rate. Simulation.

Verification. (i) The Jacobians agree with central finite differences of Eq. (1) at 50 random points to 3e-10. (ii) Driving the nominal plant open loop with the torques of Eq. (3) reproduces the reference to 1e-13 m (straight), 0.025 m (S-curve) and 0.032 m (figure-eight) at most (feed-forward held constant over each sample). (iii) The linearised one-step prediction differs from the nonlinear step by a median relative error of 2e-04 to 1e-03 (maximum 7e-03). (iv) The sparse QP was compared with an independent condensed formulation (states eliminated by recursion) solved by the trust-region method of SciPy on 36 problems with $N = 10$, scaled to activate constraints (slacks non-zero in 24). At the operating tolerance of 10^{-4} objectives agree to a relative 1e-04, but the primal can differ more: the median over problems of the largest element-wise difference of the input sequence is 2e-03 N m and the worst problem differs by 0.13 N m, because the objective is flat along weakly active constraints. With tolerance 10^{-9} and polishing the largest difference is 4e-04 N m (relative objective 5e-07); the SciPy reference needs 375 ms per problem.

3.2 Tracking accuracy

Table 3 gives the position RMS error in the 12 conditions ($T_s = 50$ ms, $N = 20$, 30 seeds), with the paired differences and their 95 % confidence intervals. The whole-run RMS contains the transient from the initial cross-track error of 0.15 to 0.30 m, which every controller must remove, so the RMS after 5 s is also given. The MPC (R2) has a lower whole-run RMS than the PID in all 12 conditions. Pooled over the 270 disturbed pairs, R2 − R1 is -2.04 [-2.09, -1.99] cm (29 % of the PID value of 6.9 cm), and -1.95 [-2.01, -1.88] cm in the benign plant; after 5 s it is -1.44 [-1.52, -1.36] cm (residual 2.5 cm for the PID

against 1.1 cm for the MPC). The adaptive layer changes the whole-run RMS by -0.035 [-0.042, -0.027] cm (0.7 %) and the RMS after 5 s by -0.21 [-0.24, -0.17] cm (19 % of the 1.07 cm residual of R2). In the benign plant the safeguard keeps the nominal model in 100 % of samples, so R3 equals R2 there. The success rate is 100 % for all controllers in all conditions, so the success criterion (which allows a 5 % speed excess) does not discriminate; the differences lie in accuracy, transients and constraint handling.

Table 3. Position RMS error (cm, mean ± s.d., 30 seeds; simulation) over the whole run, and paired differences (cm, mean [95 % CI]) over the whole run and, for R3 − R2, after 5 s ($T_s = 50$ ms, $N = 20$). Negative values favour the first-named controller. Success rate was 100 % for all controllers in all conditions.

Condition	RMS R1	RMS R2	RMS R3	R2 − R1	R3 − R2	R3 − R2, >5 s
Straight, benign	6.8 ± 1.2	5.0 ± 0.9	5.0 ± 0.9	-1.84 [-1.95, -1.73]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
S-curve, benign	7.1 ± 1.2	5.0 ± 1.0	5.0 ± 1.0	-2.07 [-2.17, -1.97]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
Figure-eight, benign	6.7 ± 1.2	4.8 ± 1.0	4.8 ± 1.0	-1.94 [-2.06, -1.81]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
Straight, dist., 0 % loss	6.6 ± 1.4	4.8 ± 1.0	4.7 ± 1.0	-1.86 [-2.01, -1.70]	-0.04 [-0.06, -0.01]	-0.19 [-0.30, -0.09]
Straight, dist., 5 % loss	6.7 ± 1.5	4.8 ± 1.1	4.8 ± 1.1	-1.91 [-2.08, -1.73]	-0.03 [-0.06, -0.01]	-0.18 [-0.28, -0.08]
Straight, dist., 10 % loss	7.0 ± 1.3	5.1 ± 1.0	5.0 ± 1.0	-1.91 [-2.08, -1.75]	-0.03 [-0.06, -0.01]	-0.20 [-0.30, -0.10]
S-curve, dist., 0 % loss	7.2 ± 1.5	5.0 ± 1.2	4.9 ± 1.2	-2.24 [-2.38, -2.10]	-0.04 [-0.07, -0.02]	-0.23 [-0.35, -0.12]
S-curve, dist., 5 % loss	7.2 ± 1.3	5.0 ± 1.1	5.0 ± 1.2	-2.19 [-2.31, -2.06]	-0.03 [-0.04, -0.01]	-0.17 [-0.23, -0.11]
S-curve, dist., 10 % loss	7.0 ± 1.3	4.9 ± 1.0	4.9 ± 1.0	-2.10 [-2.24, -1.96]	-0.03 [-0.05, -0.01]	-0.16 [-0.26, -0.06]
Figure-eight, dist., 0 % loss	6.9 ± 1.4	4.8 ± 1.2	4.8 ± 1.2	-2.07 [-2.24, -1.90]	-0.04 [-0.06, -0.02]	-0.28 [-0.38, -0.18]
Figure-eight, dist., 5 % loss	6.8 ± 1.3	4.8 ± 1.0	4.7 ± 1.0	-2.02 [-2.19, -1.86]	-0.04 [-0.07, -0.02]	-0.25 [-0.36, -0.13]
Figure-eight, dist., 10 % loss	7.1 ± 1.2	5.0 ± 0.9	5.0 ± 1.0	-2.10 [-2.24, -1.96]	-0.03 [-0.04, -0.01]	-0.20 [-0.29, -0.11]

3.3 Transients, constraints and actuation

Table 4 pools the 270 disturbed runs per controller. The MPC settles faster and with less overshoot. The violations observed in the plant were exclusively exceedances of the yaw-rate limit; the speed limit and corridor were never violated. The PID exceeded the yaw-rate limit in 30 % of runs (mean 1.94 %, worst 18.4 %), the MPC in 2.2 % (mean 0.08 %, worst 8.4 %), although its state constraints are soft. MPC requests respect the torque limit (mean excess 0.02 %) and increment limit (0.07 %, solver tolerance); the largest PID request exceeds the increment limit by 74 % on average and is clipped. Under packet loss a held command is followed by one that has moved two samples, so the applied increment can exceed the limit although every command respects it: by 76 % on average (largest per run) for the MPC and 5 % for

the PID, because the MPC commands move at the rate limit. **The price of the better transient is actuation activity:** the MPC uses 1.41 times the RMS torque of the PID (0.29 against 0.21 N m) and its commands change by 55 % of the increment limit per sample against 18 % (Table 4). The weights were tuned for accuracy, not smoothness, and the observer gain not to the noise, so part of this is noise passing through the position feedback; this is the main respect in which the PID is preferable here.

Table 4. Secondary metrics pooled over the nine disturbed conditions (270 runs per controller; simulation; mean ± s.d.) and paired differences with 95 % CIs. † Roughness (mean $|\Delta \mathbf{u}|$ as % of the increment limit) was recorded only in the three loss-free disturbed conditions (90 runs).

Metric	R1	R2	R3	R2 − R1	R3 − R2
Settling time (s)	3.59 ± 3.07	1.13 ± 0.24	1.13 ± 0.24	-2.47 [-2.83, -2.10]	0.006 [0.003, 0.008]
Overshoot (%)	18.4 ± 14.2	4.8 ± 2.5	4.3 ± 1.8	-13.6 [-15.2, -12.1]	-0.48 [-0.68, -0.29]
State violation (%)	1.94 ± 3.86	0.08 ± 0.67	0.06 ± 0.61	-1.87 [-2.34, -1.39]	-0.018 [-0.052, 0.017]
Control effort (N m)	0.207 ± 0.042	0.291 ± 0.024	0.297 ± 0.028	0.085 [0.082, 0.088]	0.0058 [0.0047, 0.0069]
Roughness (%)†	17.8 ± 0.8	55.4 ± 1.9	56.0 ± 1.5	37.6 [37.2, 38.0]	0.66 [0.45, 0.88]

3.4 Behaviour of the adaptive layer

In the disturbed plant the end-of-run gain estimate follows the true gain (Fig. 2a) with mean error -0.037 and s.d. 0.033, i.e. biased low by about 3.7 % (unmodelled asymmetry, disturbances absorbed as bias, regressor noise). In the benign plant the error has mean -0.0005 and s.d. 0.017: no drift under nominal noise. Packet loss degrades the estimate because the controller assumes its command was applied (Fig. 2c): on the straight line the mean error grows from -0.032 at 0 % loss to -0.054 at 10 %, and it is larger on the straight line than on the S-curve (0.032 against 0.023 at 0 %), as expected from weaker excitation. The closed-loop benefit nevertheless persisted: R3 − R2 after 5 s is -0.23 cm at 0 % and -0.19 cm at 10 % loss. In the noise stress test (noise multiplied by 1, 3, 6, 10; Fig. 2b) the benefit of the unguarded layer in the disturbed plant (R3u − R2 after 5 s, straight line and figure-eight) is -0.38 and -0.41 cm at nominal noise and -0.28 and -0.28 cm at 3 ×, and is indistinguishable from zero at 6 × (-0.17 and -0.15 cm) and 10 × (0.07 and -0.31 cm). The safeguarded R3 reverts to the nominal model in 50 %, 85 %, 100 % and 100 % of samples at the four noise scales (disturbed figure-eight), so at 6 × R3 equals R2. In the benign plant the unguarded layer leaves the RMS unchanged within the intervals in seven of eight cases (exception: straight line at 3 ×, 0.03 cm, 0.01 to 0.06); no condition had an upper confidence limit above 0.71 cm. No noise-driven degradation was found in the tested range; what disappears at high noise is the benefit.

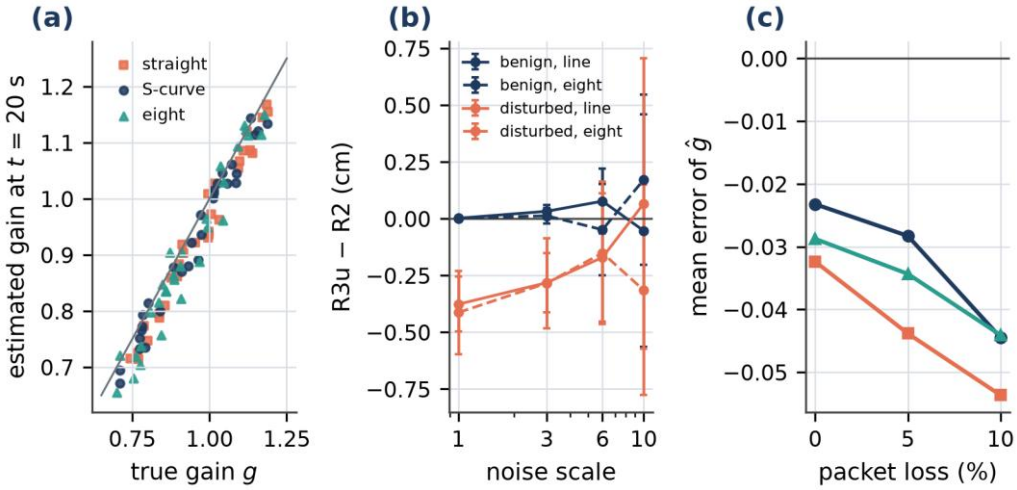


Fig. 2. Adaptive layer (simulation). (a) End-of-run gain estimate of R3u against the true gain (disturbed, no loss; line: identity). (b) R3u − R2 in RMS after 5 s against sensor-noise scale, 95 % CIs; values above zero mean the layer hurts. (c) Mean error of the gain estimate against packet loss.

3.5 Horizon, solver and real-time feasibility

Accuracy versus horizon (Fig. 3a,b). With the Riccati terminal weight, the whole-run RMS of R3 does not depend on N from 5 to 40: it varies by 0.001 cm at 50 ms (4.966 to 4.968 cm) and 0.007 cm at 20 ms, against PID values of 7.1 and 6.9 cm. The weight makes the unconstrained solution of every horizon equal to the infinite-horizon regulator, so a longer horizon helps only through constraints, rarely active here. Without the weight the RMS at 50 ms is 9.44, 8.61, 6.66, 5.28 and 4.99 cm for $N = 3,5,10,20,40$, against 4.97 cm with the weight at $N = 20$.

Computation time (Table 5, Fig. 3d–f). At $N = 20$ and 50 ms the mean computation time per sample is 1.12 ms: solve 0.47 ms, solver update 0.27 ms, QP assembly in Python 0.16 ms, the rest (observer, adaptation, delay compensation) 0.21 ms; the 99th percentile is 2.86 ms and the maximum 23.6 ms. Assembly and update together (0.44 ms) cost about as much as the solve, a property of the Python implementation. The mean solve time grows by about 0.020 ms per horizon step (intercept 0.12 ms). The mean iteration count is 26 ($N = 5$) and 26 ($N = 40$) at 50 ms but 27 and 53 at 20 ms (cause not investigated). The one-period deadline is missed in at most 0.05 % of samples at 50 ms and in 0.02 % at 20 ms with $N = 40$; the half-period deadline in at most 0.25 % (50 ms) and 2.26 % (20 ms, $N = 40$). Dropping the state rows lowers the mean solve time at $N = 20$ from 0.49 to 0.42 ms (box only: 0.38 ms); a cold start needs 114 iterations against 26 (warm) and 24 (shifted); a tolerance of 10^{-5} gives 0.60 ms against 0.39 ms at 10^{-3}.

Table 5. Measured computation time per sample of R3 (Python, OSQP, Intel Core i3-1115G4, shared machine; five seeds of the disturbed S-curve; not an embedded-target measurement). Means of the solve phase; deadline-miss fractions from Eq. (12); rows for other N omitted.

T_s (ms)	N	QP size	Solve (ms)	Total mean (ms)	Total p99 (ms)	Total max (ms)	Iter.	Miss T_s (%)	Miss $T_s/2$ (%)
20	5	50 / 90	0.19	0.70	2.09	7.8	27	0.00	0.00
20	20	200 / 360	0.92	1.66	6.17	19.7	52	0.00	0.18
20	40	400 / 720	2.04	3.20	11.98	20.6	53	0.02	2.26
50	5	50 / 90	0.22	0.82	2.70	17.1	26	0.00	0.00

T_s (ms)	N	QP size	Solve (ms)	Total mean (ms)	Total p99 (ms)	Total max (ms)	Iter.	Miss T_s (%)	Miss $T_s/2$ (%)
50	20	200 / 360	0.47	1.12	2.86	23.6	26	0.00	0.00
50	40	400 / 720	0.91	1.83	4.49	9.7	26	0.00	0.00

Deadline misses and fallback (Fig. 3c). The QP result was discarded with probability 5, 20 and 50 % per sample (this injects the consequence of a miss; it does not measure one). Relative to no misses the whole-run RMS changes by 0.005, 0.029 and 0.153 cm (95 % CI for 50 %: 0.100 to 0.206); the coast-down mode is reached in 0.7 samples of 400 on average at 50 %, and the success criterion is still met in 100 % of runs. Coast-down requires $N - 1$ consecutive misses; bursts were not tested, and the coast-down command is a placeholder for a platform-specific safe stop.

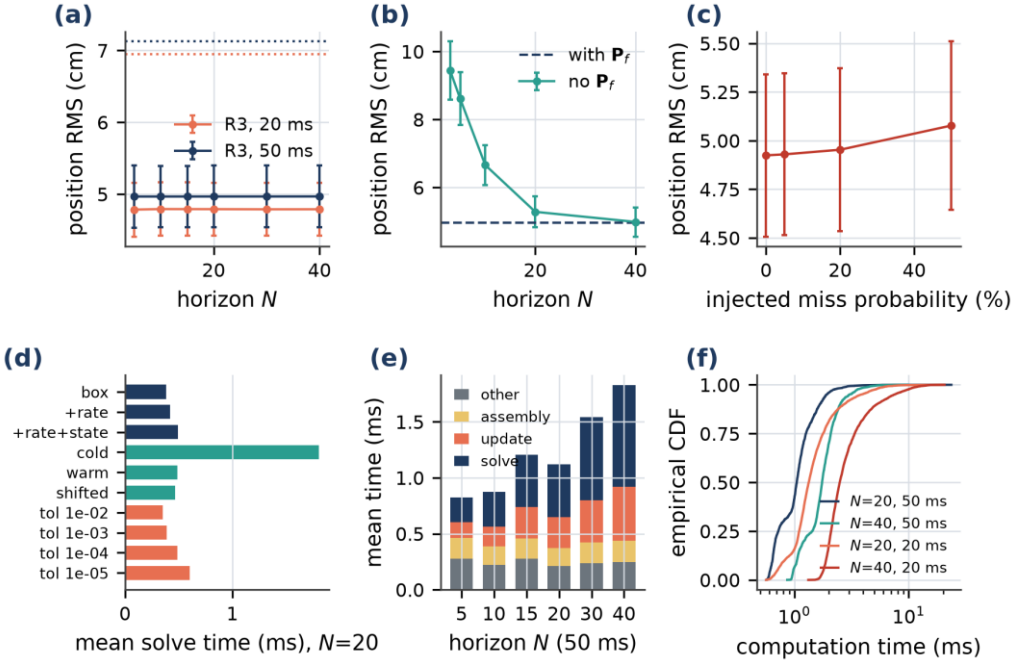


Fig. 3. Horizon and real-time results (simulation and measurement on a shared personal computer). (a) Whole-run RMS of R3 against horizon at 20 and 50 ms; dotted: PID. (b) The same at 50 ms without terminal weight. (c) RMS against injected deadline-miss probability (R3, disturbed S-curve). (d) Mean QP solve time at $N = 20$ for constraint sets (navy), warm starts (teal) and tolerances (orange). (e) Mean computation time split by phase at 50 ms. (f) Empirical distribution of total computation time; N and T_s as labelled.

4. Discussion

A constrained predictive controller built on the reference-linearised model outperformed a decoupled cascaded PID in every condition, with a large advantage in transients (settling 2.5 s shorter, overshoot 14 percentage points lower) and a moderate one in whole-run error (2.0 cm), which is dominated by the initial offset all controllers remove. The comparison must be read for what it is: the PID receives reference velocities and static drag compensation but not reference accelerations or coupling terms, whereas the MPC receives the full inverse-dynamics feed-forward and the coupling model; the study compares a model-based constrained controller with a decoupled one and does not isolate the optimisation itself. Acceleration feed-forward would narrow the gap by an unmeasured amount. The adaptive layer gave a small but detectable benefit (0.21 cm, about 19 % of the residual of the nominal MPC) because receding-horizon feedback already compensates most of the gain error; the gain estimate was biased low by about 3.7 %.

The layer did not hurt in the tested range, the benefit vanished at 6 × the nominal noise where the safeguard returned the nominal model, and loss biased the estimate (-0.032 to -0.054 between 0 and 10 % loss on the straight line) without removing the benefit. Bursty loss and unmodelled dynamics remain unexplored.

The advantage over a decoupled design agrees with reports on predictive path tracking [8–10]. The role of excitation in adaptive MPC [15–17] appears as the dependence of the gain estimate on the trajectory, and the horizon behaviour with and without a terminal weight reflects the terminal-cost conditions of the stability theory [3, 30]. The timing agrees in kind with the premise of embedded predictive-control solvers [21–24]: optimisation time is small against periods of tens of milliseconds, and warm starting and tolerance matter more than formulation at small horizons; but jitter, not the mean, determines loop performance [19], and here it is seen only through percentiles. Practically, a short horizon suffices ($N = 5$ to 10; solve time grows by about 0.020 ms per step), in-place updates or generated code would save more than a faster solver, and the fallback is cheap insurance (0.15 cm extra RMS at 50 % misses, if misses are not consecutive). Whether a platform meets its deadline needs the worst-case execution time on the target.

Limitations and threats to validity. (i) Everything is simulated and the plant structure equals the design structure except for gain, bias, asymmetry and disturbances; slip, motor dynamics, ground contact and encoder effects are absent, so absolute errors are optimistic and the ranking under unmodelled dynamics is untested. (ii) The PID is a limited representative of decoupled control (60 random candidates on eight seeds, no acceleration feed-forward, designed at 50 ms and reused at 20 ms). (iii) The observer is hand-tuned with no consistency analysis, and the MPC command chatter is consistent with noise passing through the position feedback. (iv) State constraints are soft and violations are measured on the true state; stability of the closed loop is not proved. (v) Timing is for Python and OSQP 1.1.3 on a shared personal computer; it is not a worst-case execution time and does not transfer to an embedded processor. (vi) At the solver tolerance 10^{-4} optimal inputs can differ from the exact solution by up to 0.13 N m; the closed-loop effect of a tighter tolerance was not studied. (vii) The success criterion does not discriminate, the initial offset dominates the whole-run RMS, and intervals are uncorrected for multiplicity and over seeds, not robots. (viii) Loss was modelled as independent drops with hold.

Future work comprises validation in a robot simulator and on a physical robot with logged timing; embedded solver code with worst-case execution time on the target; a baseline with acceleration feed-forward and a tuned Kalman filter; obstacle avoidance on top of the tracking layer, for example with a dynamic-window planner [13, 14]; and robust or tube formulations with set-membership updates [17] or learning-based correction [18].

5. Conclusions

A constrained LTV-MPC of a multivariable differential-drive robot, written as a sparse QP, tracked three trajectories more accurately than a decoupled cascaded PID in all 12 simulated conditions (2.0 cm lower whole-run error; residual after 5 s 1.1 against 2.5 cm) at the price of 1.4 times the torque effort and rougher commands. An adaptive gain-and-bias correction with a model-selection safeguard removed a further 19 % of the residual, did not harm the loop up to ten times the sensor noise, and lost its benefit at high noise. With a Riccati terminal weight accuracy was independent of the horizon from 5 to 40 samples, and the Python computation time was 1.1 ms on average at $N = 20$, so a 50 ms period was met with a large margin and a 20 ms period with a miss fraction of at most 0.04 % for $N \leq 40$. The deadline fallback kept the error within 0.15 cm of the miss-free value at 50 % injected misses. These conclusions are limited to the simulated plant and to a Python implementation; validation on a physical robot and an embedded target remains to be done.

Declarations

Author contributions (CRediT). Both authors contributed equally to conceptualization, methodology, software, validation, formal analysis, visualization, and writing (original draft, review and editing). **Funding.** This research received no external funding. **Competing interests.** The authors declare no competing interests. **Data and code availability.** Simulation scripts and generated data are available from the corresponding author on request. **Use of AI.** Large language models (Claude, Anthropic; models Claude Fable 5.1, Claude Sonnet 5.5 and Claude Opus 5.5, accessed through Claude Code) were used under the authors' direction for language editing, structuring of the manuscript, drafting of figures, and for writing and running the simulation code reported in Sections 2 and 3. The authors reviewed and verified the content and take full responsibility for it.

References

1. Siciliano, B.; Khatib, O., Eds. Springer Handbook of Robotics, 2nd ed.; Springer: Cham, Switzerland, 2016. doi:10.1007/978-3-319-32552-1.
2. U. Mayinova and J. M. Abdiev, “Enhancing energy efficiency in industrial enterprises through advanced automation and control systems,” JizPI Xabarnomasi, vol. 1, no. 1, pp. 989–992, 2024.
3. Abdiev, J. (2026a). Beyond faradaic efficiency: A closure audit of charge, carbon utilization, and energy reporting in electrochemical CO2 reduction. Journal of Electroanalytical Chemistry, 1022, Article 120675. https://doi.org/10.1016/j.jelechem.2026.120675.
4. Abdiev, J. (2026b). Data for: Reassessing isospin-resolved decay of the ^{31}P giant dipole resonance: Covariance-aware channel competition and cascade closure from archival γ-X-γ′ data [Data set]. Mendeley Data. https://doi.org/10.17632/538h8zxwxn.1.
5. Abdiev, J. M. (2022). Physical terms problems and their solutions [Working paper]. SSRN. https://doi.org/10.2139/ssrn.4024965.
6. Abdiev, J., Abdieva, N. M., & Khasanova, D. Y. (2022). Physical terms, problems and their solutions [Working paper]. SSRN. https://doi.org/10.2139/ssrn.4066756.
7. Abdiev, J., & Malikov, R. (2022). The role of teacher and student cooperation in the modern educational process of Google Docs. Web of Scientist: International Scientific Research Journal, 3(1), 579–589.
8. Abdiev, J., & Safarov, O. (2022). Basalt fiber-basic (primary) concepts. PhilPapers. https://philpapers.org/rec/ABDBF.
9. Abdiev, J., Safarov, O., & Julanov, H. (2022). Study of the properties of polymer composites: Reinforcement based on glass and basalt fibers. Eurasian Scientific Herald, 7, 77–88.
10. Abdiev, J. M., Leyderman, A. Yu., Saidov, A. S., & Usmonov, Sh. N. (2021, April 21). Perenos toka v poluprovodnikovykh p-n-strukturakh s bazovoy n-oblasti iz varizonnogo poluprovodnika [Current transport in semiconductor p-n structures with a base n-region made of a graded-gap semiconductor]. In Yosh olimlar va fizik talabalarning I respublika ilmiy anjumani (YoOFTRIA-I) materiallari [Proceedings of the 1st republican scientific conference of young scientists and physics students (YoOFTRIA-I)] (Vol. 1, pp. 114–117). Tashkent. (in Russian).
11. Abdiev, J. M., & Ziyadullaev, Z. R. (2022, April 22). Si taglikka o‘stirilgan (Si2)1-x(GaP)x qattiq qorishma epitaksial qatlamining kimyoviy tarkibi va morfologiyasini o‘rganish [Study of the chemical composition and morphology of the epitaxial layer of the (Si2)1-x(GaP)x solid solution grown on a Si substrate]. In Fizika fanining rivojida iste‘dodli yoshlarning o‘rni (RIAK-XV): Respublika ilmiy-amaliy konferensiyasi ishlari to‘plami [The role of talented youth in the development of physics

(RIAK-XV): Proceedings of the republican scientific-practical conference] (Vol. 2, No. 15, pp. 184–186). (in Uzbek).

12. Abdiyev, J. M., Leyderman, A. Yu., Saidov, A. S., & Usmonov, Sh. N. (2021). Injection diffusion processes in the weak linear graded-band semiconductor p-n-structures. Euroasian Journal of Semiconductors Science and Engineering, 3(1), 20–26. https://doi.org/10.37681/2181-1652-019-x-2021-1-3.
13. Kenjayev, S., & Abdiev, J. (2024). Expanding the role of mathematical modeling in quantum computers: The application of theoretical physics in next-generation computing technologies. Journal of Integrated Education and Research, 3(5), 149–162. https://ojs.rmasav.com/index.php/ojs/article/view/1675.
14. Leyderman, A. Yu., Saidov, A. S., Abdiyev, J. M., Usmonov, Sh. N., & Suyarov, Q. T. (2021, August 15). Influence of weak graded gap on injection diffusion regimes of current transport in semiconductor p-n-structures. In USBIK 2021 Online International Congress on Natural Sciences (Vol. 8, pp. 40–44).
15. Mayinova, U., & Abdiev, J. M. (2024). Enhancing energy efficiency in industrial enterprises through advanced automation and control systems. JizPI Xabarnomasi, 1(1), 989–992.
16. Mustafayev, T., Akhmedzhanov, F., Abdiev, J., & Khasanov, S. (2025a). Correction: An exploration of anisotropic acoustic wave attenuation in quartz crystals. Brazilian Journal of Physics, 55(1), Article 18. https://doi.org/10.1007/s13538-024-01657-x.
17. Mustafayev, T., Akhmedzhanov, F., Abdiev, J., & Khasanov, S. (2025b). An exploration of anisotropic acoustic wave attenuation in quartz crystals. Brazilian Journal of Physics, 55(1), Article 1. https://doi.org/10.1007/s13538-024-01638-0.
18. Payzullaev, A. N., Allaev, B. A., Mirzaev, S. Z., Abdiev, J. M., Urinov, J., & Parkash, A. (2023). The impact of silicon dioxide nanoparticle size on the viscosity and stability of nanofluids: A comprehensive study. ECS Advances, 2(3), Article 031001. https://doi.org/10.1149/2754-2734/ace121.
19. Saidov, A. S., Saparov, D. V., Usmonov, Sh. N., Kutlimratov, A., Abdiev, J. M., Kalanov, M., Razzakov, A. Sh., & Akhmedov, A. M. (2021). Investigation of the crystallographic perfection and photoluminescence spectrum of the epitaxial films of (Si2)1-x(GaP)x ($0 \le x \le 1$) solid solution, grown on Si and GaP substrates with the crystallographic orientation (111). Advances in Condensed Matter Physics, 2021, Article 3472487. https://doi.org/10.1155/2021/3472487.
20. Saidov, A. S., Usmonov, Sh. N., Karshiev, A. B., & Abdiev, J. M. (2022). Influence of the varizonal Si1-xGex solid solution composition on the thermovoltaic effect in n-Si-p-(Si)1-xGex structure. IOP Conference Series: Earth and Environmental Science, 1112(1), Article 012040. https://doi.org/10.1088/1755-1315/1112/1/012040.
21. Urinov, Kh. O., Jumanov, Kh. A., Khidirov, A. M., Urinov, S. Kh., Abdiyev, J. M., Jumaboyev, T. A., & Eshmirzayev, M. R. (2020). Magnetocaloric effect in polycrystalline cobalt. Journal of Physics: Conference Series, 1515(2), Article 022079. https://doi.org/10.1088/1742-6596/1515/2/022079.
22. Taylanov, N. A., Urinov, S. Kh., & Abdiev, J. M. (2022). A fourth-order Runge–Kutta method for numerical solution of the Kuramoto–Sivashinsky equation.
23. Urinov, J. O., Abdiev, J. M., & Malikov, R. R. (n.d.). FexV3−xO4: Magnetic and electrical properties of spinel.
24. Komilovich, J. L. H., Ugli, A. A. M., Safarov, O., Ugli, A. J. M., & Ugli, Y. A. A. (2022). Understanding the magneto caloric effect in super spin glass cobalt-based nanoparticles.
25. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E. N. The explicit linear quadratic regulator for constrained systems. Automatica **2002**, 38(1), 3–20. doi:10.1016/S0005-1098(01)00174-1.

26. Zeilinger, M. N.; Jones, C. N.; Morari, M. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. IEEE Trans. Autom. Control **2011**, 56(7), 1524–1534. doi:10.1109/TAC.2011.2108450.
27. Diehl, M.; Bock, H. G.; Schlöder, J. P. A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM J. Control Optim. **2005**, 43(5), 1714–1736. doi:10.1137/S0363012902400713.
28. Bristol, E. On a new measure of interaction for multivariable process control. IEEE Trans. Autom. Control **1966**, 11(1), 133–134. doi:10.1109/TAC.1966.1098266.
29. Skogestad, S.; Postlethwaite, I. Multivariable Feedback Control: Analysis and Design, 2nd ed.; Wiley: Chichester, UK, 2005. ISBN 978-0-470-01168-3.
30. Mayne, D. Q.; Rawlings, J. B.; Rao, C. V.; Scokaert, P. O. M. Constrained model predictive control: Stability and optimality. Automatica **2000**, 36(6), 789–814. doi:10.1016/S0005-1098(99)00214-9.