IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 2, June 2026, pp. 307∼318
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i2.pp307-318

Design and implementation of NMPC for a two-DOF robotic arm using CasADi

Lahcen Boulbalah[1], Faiza Dib[1], Nabil Benaya[1], Khaddouj Ben Meziane[2]
[1]Department of Physics, Faculty of Sciences and Techniques, Abdelmalek Essaadi University, Tetouan, Morocco
[2]Laboratory of Innovation in Management and Engineering (LIMIE), Higher Institute of Engineering and Business (ISGA), Fez, Morocco

Article Info

Article history:

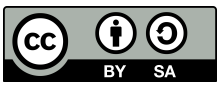
Received Nov 14, 2025
Revised Feb 1, 2026
Accepted Apr 19, 2026

Keywords:

CasADi
Dynamic model
Model predictive control
Nonlinear control
Two-link robot arm

ABSTRACT

Achieving accurate joint-space tracking in multi-link robotic arms is complicated by strong configuration-dependent nonlinearities and mandatory actuator limits that classical controllers are structurally unable to enforce. This paper presents a nonlinear model predictive control (NMPC) scheme for a two-degree-of-freedom (2-DOF) serial robotic arm, implemented within the CasADi symbolic computing environment to leverage automatic differentiation and sparse interior-point solving. The complete set of Lagrangian equations of motion-inertia, Coriolis, and gravity terms-is incorporated directly into the optimizer's prediction model through fourth-order Runge-Kutta (RK4) integration, eliminating the need for linearization. Torque, velocity, and angle bounds are imposed as native hard inequality constraints at every step of the finite-horizon optimization. Systematic simulations pit the proposed NMPC against a Ziegler-Nichols-tuned decentralized PID at two distinct sampling periods. The NMPC achieved a 95% reduction in peak tracking error relative to PID (0.0058 rad vs. 0.1347 rad for Joint 1), with mean error decreases of 64.65% and 57.58% for Joints 1 and 2 respectively, at an average solver time of 0.053 s-comfortably within the 0.1 s control cycle. The findings demonstrate that online NMPC with unabridged nonlinear dynamics is computationally practical for real-time joint control on standard computing hardware.



Corresponding Author:

Lahcen Boulbalah
Department of Physics, Faculty of Sciences and Techniques, Abdelmalek Essaadi University
Tetouan, Morocco
Email: boulbalah.lahcen@gmail.com

1. INTRODUCTION

High-fidelity joint-space tracking in robotic manipulators hinges on the ability to handle coupled, configuration-dependent nonlinearities while simultaneously respecting actuator saturation limits. Proportional-integral-derivative (PID) controllers are widely adopted in practice due to their simplicity, but the per-joint, decoupled structure means they cannot compensate Coriolis and inertia cross-coupling, nor can they enforce torque or velocity bounds in a principled manner [1], [2]. Model predictive control reframes the actuation problem as a constrained finite-horizon optimization executed at every sampling step. Linear model predictive control (MPC) variants are computationally lightweight but introduce linearization mismatch for plants with strong nonlinear behavior. Nonlinear model predictive control (NMPC) eliminates this mismatch by embedding the complete nonlinear plant model inside the optimizer; recent progress in sparse NLP solvers and symbolic computation tools has brought online NMPC within reach of both embedded processors and re-

search platforms [3], [4]. A wide range of alternative controllers has been studied for robotic manipulation. Neural-network path planners for mobile robots [5], fuzzy-logic regulators [6], [7], and adaptive controllers [8] each improve on basic PID performance, yet none enforces physical constraints as an integral design feature. MPC applied to robotic arms [9]-[11] substantiates the benefits of prediction-based actuation, though many reported implementations rely on linearized or reduced-order plant models. Recent contributions have pushed the boundary of intelligent hybrid control: a three-link manipulator controller combining adaptive fuzzy logic with sliding-mode disturbance rejection was developed in [12]; a chattering-free fuzzy super-twisting sliding-mode scheme for 2-DOF arms under parametric uncertainty was reported in [13]; and a neural-network-enhanced nonlinear PID with backstepping for wheeled mobile robots was presented in [14]. Notwithstanding these advances, hard constraint satisfaction is a structural property of MPC that is absent from sliding-mode and neural-network paradigms by construction. Laguerre-function MPC [15]-[17] and Particle swarm optimization (PSO)-optimized predictive controllers [18] demonstrate efficiency gains in power electronics and autonomous vehicle domains; however, full-order NMPC from the complete Lagrangian model remains the gold standard for arms with pronounced configuration-dependent nonlinearities [19], [20]. The specific contributions of this work are:

- A fully symbolic NMPC architecture for a 2-DOF serial arm implemented in CasADi, using RK4 integration of the complete Lagrangian model and executing at 10 Hz online
- Rigorous comparison against a Ziegler-Nichols-tuned PID baseline, yielding 95% peak-error reduction and mean improvements above 57% for both joints
- Structural guarantee of torque and velocity constraint satisfaction at every time step-a property not achievable with the PID baseline
- Demonstration that CasADi/IPOPT-based NMPC operates within the 0.1 s control period on off-the-shelf laptop hardware

Section 2 of this paper derives the full Lagrangian dynamic model of the two-link arm. Section 3 formulates the NMPC problem and describes the PID benchmark. Section 4 presents simulation results and a comparative analysis. Section 5 summarizes the work and identifies directions for future research.

2. PROBLEM FORMULATION AND METHODS

2.1. Dynamic equations

The plant under study is a planar two-link robotic arm. For each link $i \in \{1, 2\}$, θ_i denotes the joint angle, L_i the link length, and M_i the link mass; g is the gravitational acceleration. The physical configuration is illustrated in Figure 1.

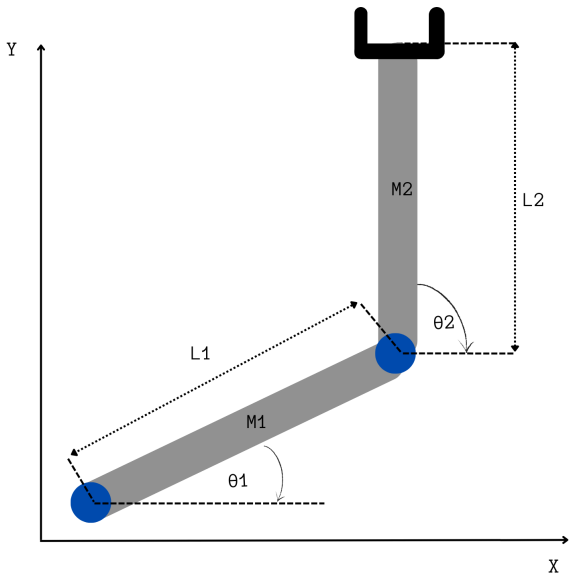


Figure 1. Two-link robot arm

The equations of motion are derived using the Lagrangian energy method [21]. This approach constructs the scalar Lagrangian $L = T - V$ from the total kinetic and potential energies, then applies the Euler-Lagrange operator to obtain the governing differential equations for each generalized coordinate.

2.1.1. Lagrangian formulation

With the joint angles θ_1 and θ_2 as generalized coordinates, the kinetic energy T and potential energy V are evaluated and combined to yield the Lagrangian L.

The total kinetic energy T is given by:

$$T = \frac{1}{2}M_1\dot{x}_1^2 + \frac{1}{2}M_2(\dot{x}_1^2 + \dot{x}_2^2 + 2\dot{x}_1\dot{x}_2\cos(\theta_2)), \tag{1}$$

where $\dot{x}_1$ and $\dot{x}_2$ are the velocities of the centers of mass of the links, and M_1 and M_2 are the masses of the first and second links, respectively.

Gravitational potential energy stored in both links summed over their respective center-of-mass heights gives:

$$V = (M_1 g L_1 \cos(\theta_1)) + (M_2 g(L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2))), \tag{2}$$

where g denotes gravitational acceleration. The Lagrangian is therefore:

$$L = T - V. \tag{3}$$

Substituting L into the Euler-Lagrange operator for each generalized coordinate yields the governing torque equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = \tau_i, \quad i = 1, 2, \tag{4}$$

with τ_1 and τ_2 being the joint torques that serve as control inputs.

2.1.2. Inertia matrix

Differentiating L twice with respect to the generalized velocities produces the configuration-dependent inertia matrix $\mathbf{M}(\theta_2)$:

$$\mathbf{M}(\theta_2) = \begin{bmatrix} D_1 & D_2 \\ D_2 & D_4 \end{bmatrix}, \tag{5}$$

where:

$$D_1 = (M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2\cos(\theta_2), \tag{6}$$

$$D_2 = M_2L_2^2 + M_2L_1L_2\cos(\theta_2), \tag{7}$$

and

$$D_4 = M_2L_2^2. \tag{8}$$

2.1.3. Coriolis matrix

Inter-joint velocity interactions generate Coriolis forces that are collected in the matrix $\mathbf{C}(\dot{\theta}_1, \dot{\theta}_2, \theta_2)$:

$$\mathbf{C}(\dot{\theta}_1, \dot{\theta}_2, \theta_2) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \tag{9}$$

where:

$$C_1 = -M_2L_1L_2\left(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right)\sin(\theta_2), \tag{10}$$

and

$$C_2 = -M_2L_1L_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_2). \tag{11}$$

2.1.4. Gravity vector

The reaction torques at each joint due to gravity are assembled into the vector:

$$\mathbf{G}(\theta_1, \theta_2) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \tag{12}$$

where:

$$G_1 = -(M_1 + M_2)gL_1 \sin(\theta_1) - M_2 g L_2 \sin(\theta_1 + \theta_2), \tag{13}$$

and

$$G_2 = -M_2 g L_2 \sin(\theta_1 + \theta_2). \tag{14}$$

2.1.5. State equations

Joint accelerations $\ddot{\theta}$ are recovered by inverting the inertia matrix:

$$\ddot{\theta} = \mathbf{M}^{-1}(\tau - \mathbf{C} - \mathbf{G}), \tag{15}$$

where $\tau = [\tau_1, \tau_2]^\top$ collects the joint torques acting as control inputs, and $\mathbf{C}$, $\mathbf{G}$ are the velocity-coupling and gravitational terms derived above.

2.1.6. Forward Kinematics

The Cartesian coordinates (x, y) of the end-effector are obtained from the joint angles through the standard planar chain geometry:

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2), \tag{16}$$

and

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2). \tag{17}$$

These geometric expressions permit joint-space tracking metrics to be directly interpreted as end-effector positioning accuracy in the workspace.

3. CONTROL STRATEGY

At each control step, the proposed NMPC formulates and solves a constrained nonlinear program (NLP) that selects joint torques so as to jointly minimize trajectory tracking deviation and actuator energy consumption over a finite prediction horizon. Future states within that horizon are propagated using a fourth-order Runge-Kutta (RK4) integration scheme [22], which preserves the arm's configuration-dependent inertial and Coriolis characteristics without approximation. Bounds on joint position, velocity, and torque are imposed directly inside the NLP as hard inequality constraints, so the optimizer can only produce solutions that are physically realizable.

3.1. Cost function

The NLP objective accumulates, over the N-step prediction window, a quadratic penalty on state deviation from the reference trajectory and a quadratic penalty on the applied torques, together with a terminal stage cost:

$$J = \sum_{k=0}^{N-1} \left(\mathbf{x}_k - \mathbf{x}_{\text{ref}}\right)^\top \mathbf{Q} \left(\mathbf{x}_k - \mathbf{x}_{\text{ref}}\right) + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k + \mathbf{x}_N^\top \mathbf{Q}_N \mathbf{x}_N \tag{18}$$

Here $\mathbf{x}_k \in \mathbb{R}^4$ collects the joint angles and angular velocities at prediction step k, $\mathbf{x}_{\text{ref}}$ is the reference trajectory to be tracked, and $\mathbf{u}_k \in \mathbb{R}^2$ holds the two joint torques. The positive semi-definite weight $\mathbf{Q}$ balances angle and velocity tracking accuracy, $\mathbf{R}$ penalizes actuator effort to prevent excessive torque usage, and the terminal weight $\mathbf{Q}_N$ provides a Lyapunov-like endpoint penalty that promotes closed-loop stability.

3.2. Parameter selection

The prediction horizon N and the weight matrices were selected following the systematic tuning framework of [23], [24]. A grid search over $N \in \{3, 5, 8, 10\}$ showed that $N = 5$ provides the optimal cost-computation tradeoff: horizons shorter than 5 produced near-sighted torque commands with persistently elevated steady-state error, whereas extending to $N > 5$ reduced RMSE by under 2% at a disproportionate computational cost [4]. Initial diagonal entries of $\mathbf{Q}$ were set according to Bryson's inverse-square rule [24], using the reciprocal of the squared admissible deviation for each state. This was followed by iterative simulation-based refinement, yielding $\mathbf{Q} = \mathrm{diag}(55, 55, 0.1, 0.1)$ and $\mathbf{R} = \mathrm{diag}(0.001, 0.001)$. The elevated Q_{11}/Q_{22} entries prioritize angular accuracy; the smaller Q_{33}/Q_{44} values accept moderate velocity deviations; and the small R entries allow adequate torque authority for tracking demanding trajectories without unnecessary energy consumption [23].

3.3. System dynamics

Defining the state vector as $\mathbf{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^\top$ and the input vector as $\mathbf{u} = [\tau_1, \tau_2]^\top$, the arm's continuous-time dynamics take the compact form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{19}$$

where the right-hand side f encapsulates the complete Lagrangian model, covering configuration-varying inertia, Coriolis velocity coupling, and gravity loading as derived in section 2.

Each predicted state in the horizon is advanced by one sampling step through the classical fourth-order Runge-Kutta (RK4) integrator [22]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \tag{20}$$

where $\mathbf{k}_1 = f(\mathbf{x}_k, \mathbf{u}_k)$, $\mathbf{k}_2 = f(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_1, \mathbf{u}_k)$, $\mathbf{k}_3 = f(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_2, \mathbf{u}_k)$, $\mathbf{k}_4 = f(\mathbf{x}_k + \Delta t\,\mathbf{k}_3, \mathbf{u}_k)$.

3.4. Constraints

At each of the N shooting nodes $k \in \{0, \ldots, N-1\}$ the NLP enforces three constraint groups: bilateral bounds on the state vector $\mathbf{x}_{\min} \le \mathbf{x}_k \le \mathbf{x}_{\max}$, bilateral bounds on the control input $\mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max}$, and continuity equations $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$ that couple successive nodes. Bound values are listed in Table 1.

Table 1. Simulation parameters and configuration

Parameter	Symbol	Value
Link masses	M_1, M_2	1.0 kg
Link lengths	L_1, L_2	1.0 m
Gravity	g	9.81 m/s^2
Simulation time	–	50 s
Sampling time	T	0.1 s
Prediction horizon	N	5
State weight	$\mathbf{Q}$	$\mathrm{diag}(55, 55, 0.1, 0.1)$
Control weight	$\mathbf{R}$	$\mathrm{diag}(0.001, 0.001)$
PID gains (K_p, K_i, K_d)	–	500, 100, 20

3.5. Implementation details

The complete NLP is assembled using CasADi's symbolic API within MATLAB [3], [25], which constructs exact first- and second-order derivatives of both the objective and the constraints via automatic differentiation. The resulting sparse derivative structures are passed directly to IPOPT [26], an interior-point solver that exploits this sparsity for efficient matrix factorization. State discretization uses the direct multiple-shooting transcription [27], [28]: each node's state $\mathbf{x}_k$ is introduced as an independent optimization variable and coupled to its successor through the RK4 equality constraint, which yields a well-conditioned problem compared with single-shooting alternatives. At each control cycle, only the leading torque vector of the computed optimal sequence $\mathbf{U}_{\text{opt}}$ is dispatched to the plant; the whole procedure then repeats from the updated state measurement.

3.5.1. PID controller

The baseline controller is a per-joint decentralized PID [1], [2], in which each joint independently computes its torque as $\tau_i(t) = K_{pi}\, e_i(t) + K_{ii} \int_0^t e_i\, d\xi + K_{di}\, \dot{e}_i(t)$, with $e_i = \theta_{i,\text{ref}} - \theta_i$ the signed angular tracking error for joint i.

3.5.2. PID tuning

Initial gains were determined by the Ziegler-Nichols ultimate-gain procedure [29], [30] and further refined through simulation trials for each of the two sampling periods tested (dt = 0.01 s and dt = 0.1 s); the final values are compiled in Table 1. The requirement for independent re-tuning at each sampling rate, combined with the controller's inherent inability to account for Coriolis cross-coupling, underlines a fundamental limitation of the decentralized PID architecture.

4. SIMULATION SETUP AND RESULTS

4.1. Simulation parameters

The NMPC framework was validated on a two-link arm with parameters and configuration summarised in Table 1. Constraints bound joint angles to $\pm\pi$ rad, velocities to ± 10 rad/s, and torques to ± 20 Nm.

4.2. Control system architecture

Both controllers operate within the closed-loop architecture depicted in Figure 2, where the plant output (measured joint angles and velocities) feeds back to the controller at every sampling instant.

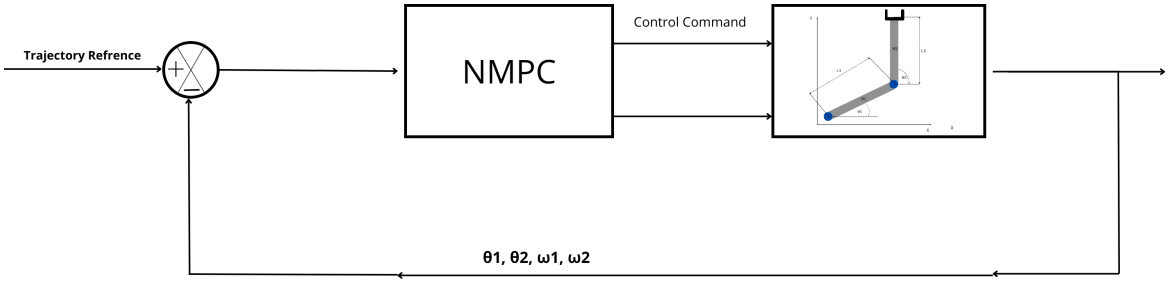


Figure 2. Closed-loop NMPC

4.3. Simulation results

Full-range sinusoidal reference signals spanning the entire joint workspace were issued to both controllers, creating a stringent evaluation of transient agility and long-run steady-state performance under realistic operating conditions.

4.3.1. Performance analysis

The quantitative advantage of the NMPC visible in Table 2 originates from three structural properties that distinguish it from any reactive controller.

- Anticipatory torque generation: The NMPC computes an optimal torque sequence by minimizing the cost J over a rolling five-step horizon, so the actuator is adjusted in advance of predicted trajectory curvature rather than after it produces an error. This pre-emptive capability is architecturally absent from every reactive control scheme.
- Coupled dynamics compensation: Because the prediction model incorporates the full Coriolis matrix $\mathbf{C}(\dot{\theta}_1, \dot{\theta}_2, \theta_2)$, the cross-joint velocity interactions are accounted for when computing optimal torques. The PID handles each joint independently; Coriolis effects then appear as unmeasured disturbances that grow with joint speed and manifest as persistent tracking offset.
- Built-in constraint satisfaction: The inequality constraints $|\tau_i| \leq 20\,\text{Nm}$, $|\theta_i| \leq \pi\,\text{rad}$, and $|\dot{\theta}_i| \leq 10\,\text{rad/s}$ are embedded inside the NLP, so the optimizer is mathematically prohibited from returning limit-violating torques. The PID relies on an external saturation element which, when active, interacts with the integrator state and can erode stability margins.

The statistical tracking metrics and computation times across both sampling rates are compiled in Table 2. At dt = 0.01 s, the NMPC recorded mean errors of 0.0035 rad and 0.0014 rad for Joints 1 and 2, against 0.0099 rad and 0.0033 rad for the PID, corresponding to reductions of 64.65% and 57.58% respectively. Peak deviations dropped from 0.1347 rad to 0.0058 rad for Joint 1 (-95.69%) and from 0.0510 rad to 0.0019 rad for Joint 2 (-96.27%). At the coarser dt = 0.1 s rate, analogous improvements of 56.00% and 63.89% were obtained with no adjustment to the NMPC weight matrices. The mean solve time remained between 0.053 s and 0.056 s across both rates, safely inside the 0.1 s budget.

Table 2. NMPC and PID performance comparison

dt (lr)3-5 (lr)6-8 (lr)9-10	Parameters	Joint 1			Joint 2			Timing (s)	
		Mean	Max	RMS	Mean	Max	RMS	Mean	Max
0.010 (lr)2-11	**NMPC**	0.0035	0.0058	0.0038	0.0014	0.0019	0.0015	0.053	0.261
	PID	0.0099	0.1347	0.0170	0.0033	0.0510	0.0050	3.61×10^{-6}	0.001595
	Improvement (%)	64.65	95.69	77.65	57.58	96.27	70.00	-	-
	NMPC	$Q = 500$	$R = 0.001$		$Q = 55$	$R = 0.0001$			
	PID	Kp = 400 Ki = 100 Kd = 20							
0.100 (lr)2-11	**NMPC**	0.0055	0.0076	0.0055	0.0013	0.0022	0.0015	0.056	0.246
	PID	0.0125	0.1374	0.0177	0.0036	0.0493	0.0056	1.08×10^{-5}	0.001315
	Improvement (%)	56.00	94.47	68.93	63.89	95.54	73.21	-	-
	For NMPC	$Q = 500$	$R = 0.001$		$Q = 55$	$R = 0.0001$			
	PID	Kp = 30 Ki = 15 Kd = 1.2							

4.3.2. Real-time performance and parameter sensitivity

All tests ran on an Intel Core i7-8550U laptop under MATLAB R2023a. Average NMPC solve times were 0.053 s at dt = 0.01 s and 0.056 s at dt = 0.1 s, both keeping comfortably inside the 0.1 s control budget. The single peak of 0.261 s occurred only at the first iteration due to a cold-start initialization; once a warm-start guess was available, every subsequent solve sat well below the budgeted 0.1 s. The horizon sensitivity study over $N \in \{3, 5, 8, 10\}$ verified that $N = 5$ strikes the best balance between accuracy and computation load: smaller windows led to myopic torque strategies with noticeable steady-state offset, while $N = 10$ trimmed RMSE by no more than 2% at a $1.8\times$ increase in average solve time. The weight matrices selected during tuning remained insensitive to changes in trajectory shape because the receding-horizon correction mechanism automatically compensates for plant-model discrepancies at each control step, eliminating the need for gain scheduling.

The workspace trajectory traced by the end-effector under both controllers is shown in Figure 3. The NMPC path closely follows the reference locus, whereas the PID trajectory exhibits notable deviations, particularly at trajectory turning points.

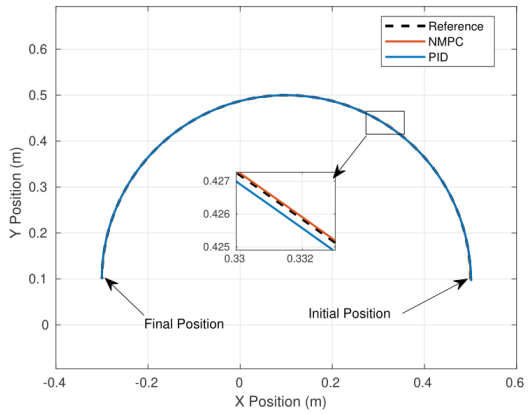


Figure 3. Comparison of the end effector trajectory of NMPC vs PID

Figure 4 presents the angle time histories for each joint under both controllers. The NMPC tracks the sinusoidal reference with near-zero residual error throughout the full 50 s run; proactive torque planning keeps the controller ahead of reference curvature changes. The PID eventually converges but shows noticeable phase lag during fast transitions and persistent oscillation on Joint 2, which is a characteristic symptom of uncompensated Coriolis coupling from the other joint.

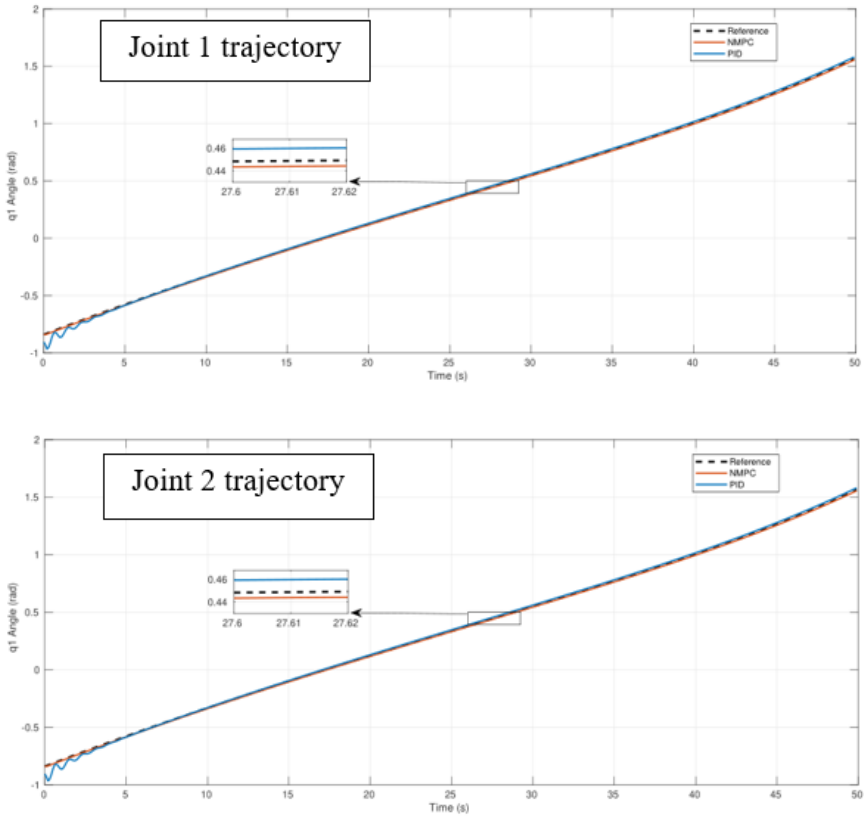


Figure 4. Comparison of joint trajectories: PID vs NMPC

The error time series in Figure 5 make the performance difference quantitatively clear. After a brief initial transient, the NMPC error settles to a narrow, near-stationary band, whereas the PID error shows recurring oscillatory spikes whose amplitude and frequency track variations in trajectory curvature. The NMPC's consistently low error is direct evidence that anticipatory torque planning prevents the error bursts that are unavoidable with purely reactive, error-driven control.

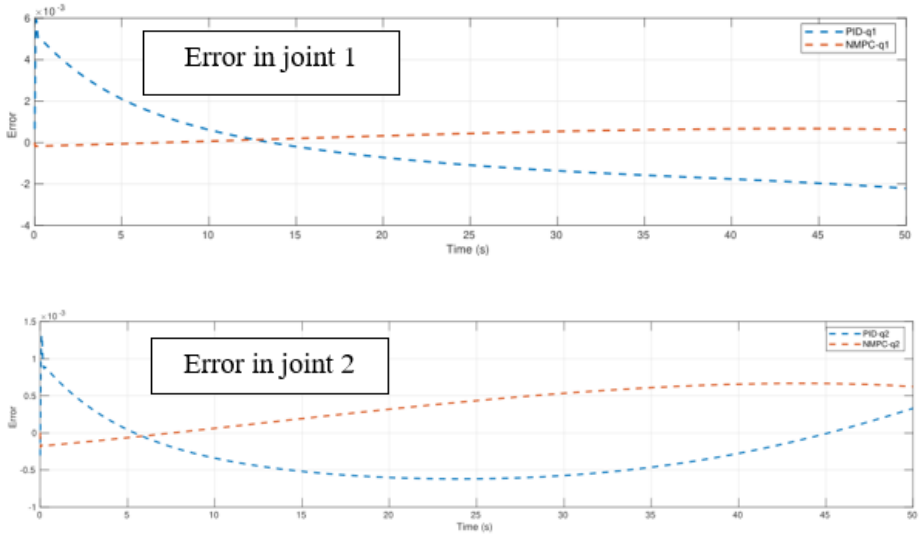


Figure 5. Tracking error comparison: PID vs NMPC

Figure 6 display the torque waveforms delivered to Joint 1. The NMPC torque stays smooth and remains strictly within $\pm 20\,\mathrm{Nm}$ at all times because the constraint is built into the NLP; the optimizer spreads corrective effort across the whole horizon rather than issuing single large impulses. The PID signal is larger at trajectory onset, exhibits abrupt step changes at curvature inflections, and requires an external saturation block to clip limit violations-a reactive strategy that is fundamentally incapable of preventing constraint breaches before they occur.

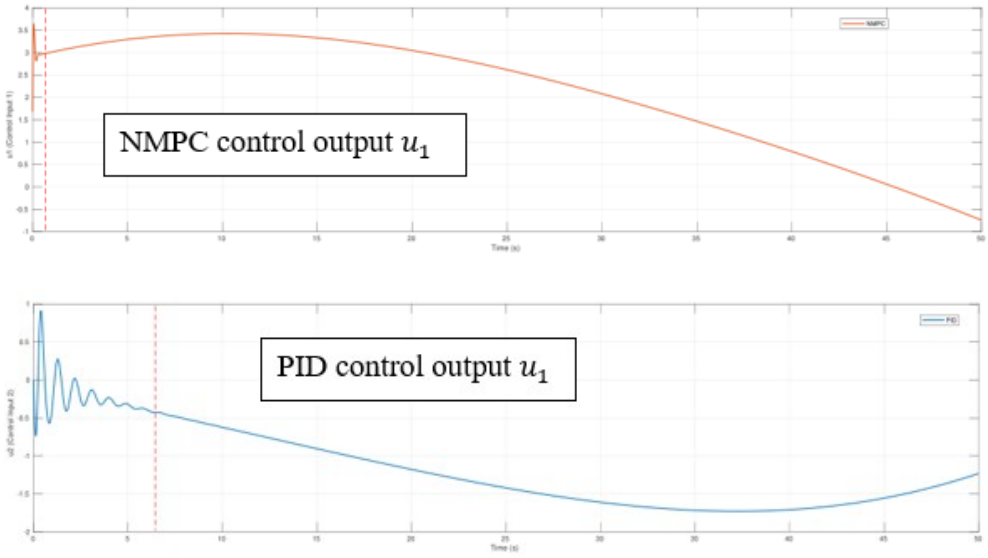


Figure 6. Control torque for joint 1: NMPC vs PID

Figure 7 extend the torque comparison to Joint 2. The NMPC optimizes τ_1 and τ_2 simultaneously as a coupled two-dimensional control problem, so the Joint 2 torque is naturally coordinated with Joint 1 to share Coriolis loads between the two joints. The output is smooth and energy-efficient. Without access to Joint 1's state, the PID Joint 2 loop produces more irregular torque; the oscillation frequency in its output tracks Joint 1 angular speed directly, confirming that unmodeled Coriolis forces are driving the disturbance.

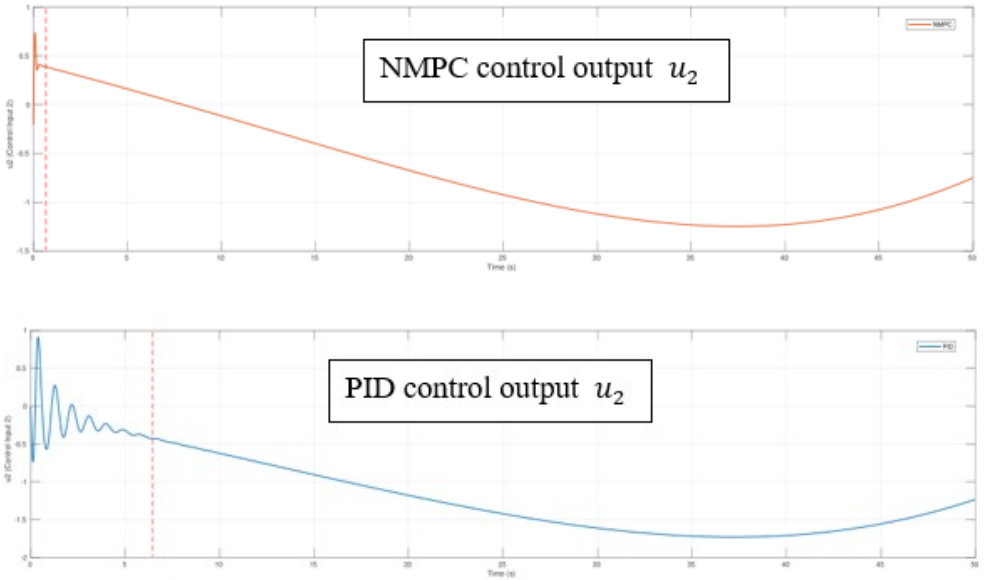


Figure 7. Control torque for joint 2: NMPC vs PID

5. CONCLUSION

This work developed and validated an NMPC framework for joint-space trajectory tracking of a 2-DOF serial robotic arm, built within the CasADi symbolic environment and solved at each control step by IPOPT's interior-point algorithm. The prediction model embeds the complete Lagrangian dynamics-configuration-dependent inertia, Coriolis coupling, and gravitational loading-discretized via RK4 integration, and actuator and state limits are imposed as hard NLP inequalities rather than post-hoc saturations. Compared with a Ziegler-Nichols-tuned decentralized PID across two sampling rates, the NMPC trimmed peak angular error by 95.69% for Joint 1 and 96.27% for Joint 2, and cut mean error by 64.65% and 57.58% respectively. Average solve times of 0.053–0.056 s confirmed that the optimizer consistently fits within the 0.1 s control window on a standard laptop, establishing real-time suitability without needing specialized embedded hardware.

Three structural features of the NMPC formulation-look-ahead torque planning, integrated coupling compensation, and mathematically guaranteed constraint adherence-collectively account for the measured performance advantage over the decentralized PID. Critically, none of these attributes requires problem-specific engineering effort beyond providing the system description to CasADi. Planned extensions include: i) Hardware-in-the-loop verification on a physical two-link arm testbed to assess disturbance rejection and model-mismatch robustness under real operating conditions. ii) Scaling to higher-DOF manipulators with machine-learning surrogate dynamics to reduce online computational demand. iii) Reinforcement-learning-based automated tuning of the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ to remove the manual calibration step. and iv) Exploration of adaptive and robust NMPC formulations capable of handling time-varying parameters such as payload mass changes.

ACKNOWLEDGMENTS

The authors declare that no specific acknowledgments are required for this work.

FUNDING INFORMATION

The authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Lahcen Boulbalah	✓	✓	✓	✓	✓	✓		✓	✓	✓				
Faiza Dib		✓								✓		✓		
Nabil Benaya	✓									✓		✓		
Khaddouj Ben Meziane		✓			✓					✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

REFERENCES

[1] K. J. Astr¨om and R. M. Murray, Feedback systems: an introduction for scientists and engineers, 2nd ed. Princeton University Press, 2021.
[2] K. J. Astr¨om and T. H¨agglund, Advanced PID control. ISA-The Instrumentation, Systems, and Automation Society, 2006.

[3] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019, doi: 10.1007/s12532-018-0139-4.
[4] 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, doi: 10.1016/S0005-1098(99)00214-9.
[5] M. Singh and D. Parhi, "Path optimisation of a mobile robot using an artificial neural network controller," *International Journal of Systems Science*, vol. 42, pp. 107–120, 2011, doi: 10.1080/00207720903470155.
[6] A. Kosari, H. Jahanshahi, and S. Razavi, "Optimal FPID control approach for a docking maneuver of two spacecraft: translational motion," *Journal of Aerospace Engineering*, vol. 30, p. 04017011, 2017, doi: 10.1061/(ASCE)AS.1943-5525.0000720.
[7] A. Kosari, H. Jahanshahi, and S. A. Razavi, "An optimal fuzzy PID control approach for docking maneuver of two spacecraft: Orientational motion," *Engineering Science and Technology, an International Journal*, vol. 20, pp. 293–309, 2017, doi: 1016/j.jestch.2016.07.018.
[8] L. Whitcomb, A. Rizzi, and D. Koditschek, "Comparative experiments with a new adaptive controller for robot arms," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 59–70, 1993, doi: 10.1109/70.210795.
[9] A. Hakimzadeh and V. Ghaffari, "Designing of non-fragile robust model predictive control for constrained uncertain systems and its application in process control," *Journal of Process Control*, vol. 95, pp. 86–97, 2020, doi: 10.1016/j.jprocont.2020.10.004.
[10] J. Wilson, M. Charest, and R. Dubay, "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator," *Robotics and Computer-Integrated Manufacturing* , vol. 41, pp. 23–30, 2016, doi: 10.1016/j.rcim.2016.02.003.
[11] E. H. Guechi, S. Bouzoualegh, Y. Zennir, and S. Bla˘zi˘c, "MPC control and LQ optimal control of a two-link robot arm: A comparative study," *Machines*, vol. 6, no. 3, p. 37, 2018, doi: 10.3390/machines6030037.
[12] A. G. Adane and C. M. Abdissa, "Adaptive fuzzy sliding mode controller of three link robot arm manipulator," *IEEE Access*, vol. 13, 2025, doi: 10.1109/ACCESS.2025. 3607809.
[13] H. G. Dirara, F. T. Yareshe, and C. M. Abdissa, "Design and analysis of adaptive fuzzy super-twisting sliding mode controller for uncertain 2-DOF robotic manipulator," *IEEE Access*, vol. 13, pp. 110 241–110 254, 2025, doi: 10.1109/ACCESS.2025.3581449.
[14] T. B. Arega, Y. M. Tesfa, and C. M. Abdissa, "Three-wheeled mobile robot trajectory tracking control using nonlinear PID controller based neural network combined with backstepping controller," *IEEE Access*, vol. 13, 2025, doi: 10.1109/ACCESS.2025.3577269.
[15] C. M. Abdissa and K. T. Chong, "A modified controller for three-level three-phase voltage source inverter based on Laguerre functions," *International Journal of Computer Applications*, vol. 182, no. 25, pp. 21–28, 2018, doi: 10.5120/ijca2018918081.
[16] C. M. Abdissa and K. T. Chon, "Stabilization and voltage regulation of the Buck DC-DC converter using model predictive of Laguerre functions," *Studies in Informatics and Control*, vol. 26, no. 3, pp. 315–324, 2017, doi: 10.24846/v26i3y201707.
[17] C. M. Abdissa, "Improved model predictive speed control of a PMSM via Laguerre functions," *Mathematical Problems in Engineering*, vol. 2024, p. 5562771, 2024, doi: 10.1155/2024/5562771.
[18] C. A. Kedir and C. M. Abdissa, "PSO based linear parameter varying-model predictive control for trajectory tracking of autonomous vehicles," *Engineering Research Express*, vol. 6, no. 3, p. 035229, 2024, doi: 10.1088/2631-8695/ad722e.
[19] E. A. Gedefaw, N. B. Abera, and C. M. Abdissa, "A review of modeling and control techniques for unmanned aerial vehicles," *Engineering Reports*, vol. 7, no. 6, p. e70215, 2025, doi: 10.1002/eng2.70215.
[20] G. T. Terefa, J. G. Njiri, P. I. Muiruri, and C. M. Abdissa, "Optimal design and analysis of a mixed airfoil blade for small-scale HAWTs," *Engineering, Technology Applied Science Research*, vol. 15, no. 3, pp. 22 538–22 547, 2025, doi: 10.48084/etasr.10275.
[21] J.-L. Lagrange, M´ecanique Analytique. Paris: Desaint, 1788, reprinted by Cambridge University Press, 2009.
[22] J. C. Butcher, Numerical Methods for Ordinary Differential Equations. John Wiley Sons, 2003. doi: 10.1002/0470868279.
[23] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, Model predictive control: Theory, computation, and design, 2nd ed. Madison, WI: Nob Hill Publishing, 2017.
[24] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial and Engineering Chemistry Research*, vol. 49, no. 8, pp. 3505–3515, 2010, doi: 10.1021/ie900323c.
[25] J. Andersson and M. Diehl, User's Guide to CasADi, 2019, [Online]. Available: https://web.casadi.org/.
[26] A. W¨achter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006, doi: 10.1007/s10107-004-0559-y.
[27] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings of the 9th IFAC World Congress*. Elsevier, 1984, pp. 1603–1608, doi: 10.1016/S1474-6670(17)61205-9.
[28] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, "Autogenerated algorithms for nonlinear model predictive control on long and short horizons," in *IFAC-PapersOnLine*, vol. 48, no. 23. Elsevier, 2015, pp. 988–993, doi: 10.1109/CDC.2013.6760692.
[29] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the American society of mechanical engineers*, vol. 64, no. 11, pp. 759–768, 1942, doi: 10.1115/1.4019264.
[30] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *Journal of Process Control*, vol. 13, pp. 291–309, 2003, doi: 10.1016/S0959-1524(02)00062-8.

BIOGRAPHIES OF AUTHORS

Lahcen Boulbalah is a Ph.D. student at Abdelmalek Essaadi University, Morocco. He is conducting his doctoral research in the areas of artificial intelligence, control design, robotic systems, and intelligent control. His academic interests include control theory, machine learning for robotics, autonomous systems, and algorithmic optimization. He can be contacted at email: boulbalah.lahcen@gmail.com.

Faiza Dib holds a faculty position in the Physics Department at the Faculty of Sciences and Techniques (FSTH), Abdelmalek Essaadi University, Tetouan, Morocco. She earned her doctoral degree in Signals, Systems, and Computing from Sidi Mohammed Ben Abdellah University, Fez, Morocco, in 2016, and joined FSTH as an assistant professor in 2019. Her scientific activities span advanced control design, electrical power systems, robotics, fuzzy logic, sliding-mode techniques, wind energy conversion, intelligent control strategies, and metaheuristic optimization. She can be contacted at email: dib.fayza@yahoo.fr.

Nabil Benaya is an engineer-researcher with extensive prior experience in industrial applications before transitioning to academia. He currently serves as a professor in the Physics Department at the Faculty of Sciences and Techniques (FSTH) of Al Hoceima, Abdelmalek Essaadi University, Morocco. He completed his Ph.D. in Automatic Control and Industrial Computing at Abdelmalek Essaadi University in 2019. His research covers automation, control systems theory, fuzzy-logic controllers, sliding-mode control, renewable energy systems, and system dynamic reliability. He can be contacted at email: nbenaya@uae.ac.ma.

Khaddouj Ben Meziane is an assistant professor in the Laboratory of Innovation in Management and Engineering (LIMIE) at the Higher Institute of Engineering and Business (ISGA), Fez, Morocco, a position she has held since 2017. She received her Ph.D. in Signals, Systems, and Computing from Sidi Mohammed Ben Abdellah University, Fez, Morocco, in 2016. Her research interests encompass advanced control design, electrical power systems, fuzzy-logic methods, sliding-mode control strategies, renewable energy systems, wind turbine technology, intelligent control, and computational optimization. She can be contacted at email: khaddouj.benmeziane@isga.ma