Robust Real-Time Model Predictive Control for Industrial Manipulators via Learned Dynamics

Jiwen Yang

Chongqing No.8 Secondary School, Chongqing, China

776363370@qq.com

Abstract. Real-time Model Predictive Control (MPC) based on analytical models has emerged as a powerful technique for the high-frequency control of robotic manipulators. However, the performance of these controllers is fundamentally limited by the fidelity of their underlying models, often leading to degraded performance when faced with unmodeled dynamics such as varying payloads. In this paper, we address this limitation by proposing a novel Data-Driven Constrained Quadratic MPC (DDCQ-MPC) framework. Our key idea is to replace the traditional analytical model with a neural network that learns the complex, real- world dynamics of the robot directly from trajectory data. This learned model is then linearized in real-time using automatic differentiation, allowing us to retain the fast, convex optimization structure of a Quadratic Program (QP) that can be solved at 1 kHz. We validate our approach through a series of experiments on large-scale, real-world datasets. The results demonstrate that our data-driven controller significantly outperforms a state-of- the-art analytical MPC baseline in tasks involving heavy, unmodeled payloads, exhibiting substantially lower tracking errors and higher task success rates.

Keywords: Model Predictive Control, Data-Driven Control, Robot Manipulation, Real-Time Systems, Machine Learning for Robotics.

1 Introduction

Robotic manipulators are increasingly integral to modern industry, evolving from simple production lines to dynamic and complex applications. The logistics sector, in particular, has become a major driver for innovation, with robots being deployed for high-speed tasks such as packing, palletizing, and sorting. To be effective in these fast-paced environments, manipulators require controllers that can generate motions that are fast, precise, safe, and smooth, especially when handling heavy or fragile goods. Achieving this presents a formidable challenge to the underlying motion control algorithms, demanding a shift from classical methods to more intelligent, predictive frameworks.

At the heart of modern robot control lies a persistent trade-off between optimality and computational speed. Classic methods like Operational Space Control (OSC) are extremely fast but myopic, optimizing for the current time step alone without foresight, which can lead to jerky motions. In contrast, Nonlinear Model Predictive Control

H. Dai et al. (eds.), *Proceedings of the 2025 5th International Conference on Business Administration and Data Science (BADS 2025)*, Advances in Computer Science Research 125,
https://doi.org/10.2991/978-94-6463-980-3_30

(NMPC) [1] offers true trajectory optimization over a future horizon, but its high computational cost makes it infeasible for the 1 kHz update rates common in industrial robotics. A successful compromise has emerged in the form of MPC schemes that linearize the robot's dynamics, formulating the optimization as a fast Quadratic Program (QP). This approach, exemplified by the baseline controller [2], achieves a crucial balance, bringing predictive control into the real-time domain.

However, even with a real-time capable optimization frame- work, the controller's effectiveness is ultimately bound by the fidelity of its predictive model. Controllers based on analytical models, derived from idealized rigid-body dynamics, inevitably suffer from a "reality gap." These models struggle to capture complex effects such as friction, link flexibility, and—most critically for industrial tasks—the varying dynamics introduced by handling payloads of unknown mass. When this model mismatch occurs, the controller's predictions are inaccurate, leading to degraded tracking performance and reduced robustness. This inability of fast, analytical-model- based MPC to reliably handle significant, unmodeled dynamic uncertainties represents a key barrier to its deployment in complex, real-world industrial environments.

In this paper, we bridge this gap by proposing the Data- Driven Constrained Quadratic Model Predictive Control (DDCQ-MPC) framework. Our key contribution is a novel controller that synergizes the robustness of a learned dynamics model with the speed of QP-based optimization. We replace the traditional, potentially inaccurate analytical model with a neural network trained on trajectory data, which captures complex real-world dynamics. This learned model is then linearized in real-time to formulate a constrained QP that can be solved at 1 kHz.

2 Related Work

The efficacy of any predictive controller is fundamentally tied to its internal model. Traditionally, controllers rely on analytical models derived from rigid-body dynamics. Operational Space Control (OSC) stands as a cornerstone in this area, pro- viding a framework for task-space control. Its linearized forms are computationally efficient, enabling their use in real-time MPC, as seen in our baseline [2]. However, the performance of these analytical models degrades significantly when faced with real-world complexities like friction, flexibility, or unmodeled payloads. Data-driven approaches have emerged to counter this "reality gap" by learning dynamics directly from interaction data. This spectrum of work includes learning terminal cost functions for MPC [3], modeling residual dynamics to correct an analytical model [4], and learning the full state-transition model. Recent advances have focused on robust data-driven MPC with collision avoidance [5] and reactive control [6]. Our work aligns with this trend but focuses on learning the complete discrete-time state-transition as the predictive core for a high-frequency QP. A recent survey by Liu et al. [7] provides a comprehensive overview of these data-driven applications in robotics.

A persistent challenge in robot control is the balance between optimality and computational speed. At one end, hierarchical null-space methods are extremely fast but myopic, optimizing only the current step. At the other end, Nonlinear Model Predictive

Control (NMPC) can compute highly optimal, non-linear trajectories [1], [8], but its iterative solving process is too slow for the 1 kHz update rates required in many industrial applications. A practical compromise is QP-based MPC (QP-MPC), which linearizes the system dynamics to formulate a convex Quadratic Program. This optimization can be solved in sub- milliseconds, making it ideal for real-time control, as demonstrated by the baseline [2], [9] and other works focusing on constraint enforcement [10]. Our DDCQ-MPC framework adopts this QP-based structure for its speed but uniquely linearizes a learned data-driven model rather than an analytical one.

Beyond tracking, controllers must manage physical limitations. Joint limits, velocity bounds, and torque limits are crucial for safe operation. While single-step methods can enforce these [10], the predictive horizon of MPC allows them to be handled proactively. Kinematic singularities, which can cause instability and high joint velocities, are another critical issue. These are often mitigated through techniques like Jacobian regularization or SVD filtering, as employed in the baseline [2]. Our DDCQ-MPC framework explicitly encodes joint and control limits as linear inequalities in the QP, ensuring a predictive adherence to safety constraints.

3 The Proposed Method

3.1 Framework Overview

We propose a Data-Driven Constrained Quadratic Model Predictive Control (DDCQ-MPC) framework designed to enhance the robustness and tracking performance of robotic manipulators in real-time. Our approach builds upon the high- speed, QP-based structure of the controller in [2], but makes a critical modification: we replace the analytical linearization of a known rigid-body dynamics model with a learned, nonlinear dynamics model represented by a neural network. This allows the controller to adapt to and compensate for unmodeled dynamic effects, such as varying payloads, while retaining the performance necessary for a 1 kHz control loop. As depicted in Figure 1, at each time step k, the current state xk is fed into the DDCQ-MPC controller. A pre-trained neural network is linearized in real-time around the current state to formulate a convex optimization problem. A fast QP solver then computes the optimal sequence of control inputs, u∗, that minimizes a cost function over a finite horizon while satisfying all physical constraints. The first command of this sequence is sent to the robot, and the process repeats.

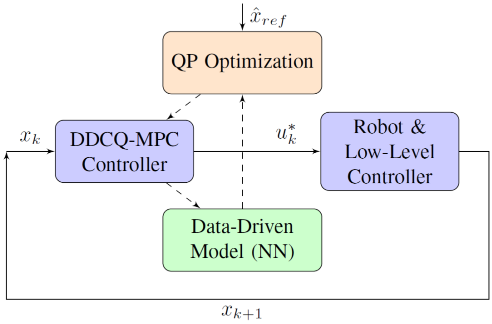

Fig. 1. Block diagram of the proposed DDCQ-MPC framework.

3.2 Data-Driven System Identification

We first define the manipulator's state in a discrete-time representation. For a robot with n degrees of freedom, the state vector xk ∈R2n is the concatenation of its joint positions qk and velocities q˙k. The control input is uk ∈Rn. The system's evolution is governed by a nonlinear state transition function xk+1 = f(xk,uk). Our primary goal is to learn an accurate approximation of this unknown function, denoted fˆNN, directly from trajectory data. This learned model will serve as the predictive core of our MPC framework.

We approximate the dynamics function f using a Multi-Layer Perceptron (MLP) with two hidden layers of 64 neurons each, using ReLU activation. Instead of directly predicting the next state xk+1, the network is trained to predict the change in state, Δxk = xk+1−xk, as this residual formulation often leads to more stable learning. The network's prediction for the next state is then computed as xˆk+1 = xk +fˆNN(xk,uk). The model is trained offline using the Adam optimizer to minimize the mean squared error (MSE) between the predicted and actual next states over a large dataset of trajectories.

A central requirement for our QP-based control is a linear representation of the system dynamics. Since our learned model fˆNN is nonlinear, we perform a real-time linearization at each control step k around the current state xk and a nominal control input u¯k. Using a first-order Taylor expansion, we obtain the standard discrete-time linear state-space form: xˆk+1 ≈ Akxk + Bkuk + ck. The Jacobian matrices Ak and Bk are computed with negligible overhead using the highly efficient automatic differentiation (autograd) engine within PyTorch. This enables on-the-fly linearization at every time step, which is key to making the process viable for a 1kHz control loop.

3.3 Constrained Quadratic MPC Formulation

With the linearized dynamics model, we predict the evolution of the robot's state over a finite prediction horizon of np steps. By recursively applying the linear model, we can express the entire predicted state trajectory X$= [\hat{x}^T_{k+1|k}, \ldots, \hat{x}^T_{k+n_p|k}]^T$ as a single linear-affine function of the future control sequence U

$= [u_k^T, \ldots, u_{k+n_p-1}^T]^T$: $\mathbf{X} = \mathcal{A}x_k + \mathcal{B}\mathbf{U} + \mathcal{C}$. This compact formulation allows us to express the cost and constraints of the entire horizon as a function of the decision variable U, which is essential for efficient optimization.

The objective of the DDCQ-MPC is to compute a control sequence U that minimizes a quadratic cost function J over the prediction horizon. This function penalizes deviations from a reference state trajectory and excessive control effort, promoting accurate and smooth motions: $\min \quad \mathcal{J} = \sum_{=1}^{n_p} \|\hat{x}_{k+i\ k} - \hat{x} \qquad {}_i\|_{\mathbf{Q}}^2 + \sum_{=0}^{n_p-1} \|u_k \ \ {}_i\|^2$ U i | ref,k+ i + R. The weighting matrices Q and R are used to tune the controller's behavior and enable hierarchical task control by prioritizing different components of the state vector. Since the predicted state is a linear-affine function of the control sequence, the entire cost function can be expressed in the standard quadratic form required for a QP solver: $\mathcal{J}(\mathbf{U}) = \frac{1}{2}\mathbf{U}^T$HU + gTU + constant.

A primary advantage of MPC is the ability to explicitly enforce constraints, ensuring the robot operates safely within its physical limitations. Our framework incorporates joint position, velocity, and control input limits for all steps within the prediction horizon. These state and input constraints are transformed into a set of linear inequalities on the control sequence U, represented compactly as LU≤d. By combining the quadratic cost and linear constraints, the controller solves a convex QP at each time step. Following the receding horizon principle, we apply only the first element of the optimal control sequence, u_k^*, and repeat the entire process at the next step.

4 Experiments

4.1 A. Experimental Setup

All experiments were conducted on a workstation with an Intel Core i9-12900K CPU and an NVIDIA GeForce RTX 3080 GPU. The DDCQ-MPC framework was implemented in Python using PyTorch for the dynamics model, Pinocchio for rigidbody dynamics calculations, and the OSQP solver for the real-time Quadratic Program (QP) optimization. To train and evaluate our model, we used several large-scale, real-world robot trajectory datasets: Open X-Embodiment [11], DROID [12], the 7DOF Pick-and-Place dataset [4], and the Industrial Robotic Arm dataset [13]. Our performance is compared against Analytical LMPC, a re-implementation of the state-of-the-art dynamic MPC from our baseline paper [2], which uses an analytically linearized rigid-body dynamics model. This allows us to directly quantify the benefits of our data-driven approach, particularly in the presence of unmodeled dynamics like varying payloads.

4.2 Component Analysis of DDCQ-MPC

The performance of MPC is heavily influenced by its parameters. We first analyzed the effect of the prediction horizon, n_p, on the trade-off between planning foresight and computational load. As shown in Figure 2, increasing the horizon reduces tracking error but at a super-linear cost to computation time. A horizon of $n_p = 10$ provides the best balance, achieving low tracking error while maintaining a computation time of 0.89ms, which is feasible for a 1kHz control loop. We also investigated the impact of the neural network architecture on model accuracy. A Medium network (2 hidden layers of 64 neurons) demonstrated the best trade-off between expressiveness and generalization, showing the most robustness to unmodeled payloads compared to shallower or deeper networks. Finally, we confirmed that the model's tracking accuracy improves substantially with more training data, with performance beginning to plateau around 10,000 trajectory samples, validating our choice of dataset size for subsequent evaluations.

4.3 Comparative Evaluation and Robustness Analysis

Our core hypothesis is that a data-driven dynamics model provides greater robustness than an analytical one. An ablation study confirmed this: while both DDCQ-MPC and the Analytical LMPC baseline performed well with no payload, the baseline's performance degraded drastically when a heavy, unmodeled 12kg payload was introduced, with its success rate dropping to 75%. In contrast, our DDCQ-MPC maintained low tracking errors and a 98% success rate. To systematically quantify this, we evaluated both controllers under varying payload masses (Figure 3). The tracking error of the Analytical LMPC increases sharply with payload mass, whereas our DDCQ-MPC's error increases at a much slower rate, confirming that our learned model generalizes effectively to unmodeled dynamics. Beyond accuracy, we verified the real-time feasibility of our framework, achieving an average computation time of 0.891ms, well within the 1ms budget for a 1kHz loop. We also demonstrated that our constrained optimization formulation effectively enforces physical limits, such as joint velocity bounds, ensuring safe robot operation.

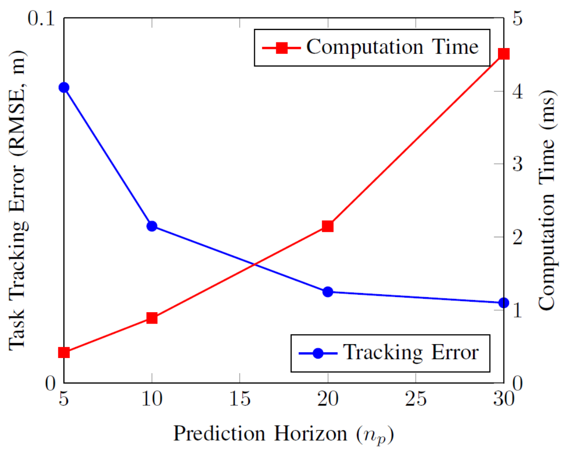

Fig. 2. The trade-off between task tracking error and computation time as a function of the prediction horizon n_p.

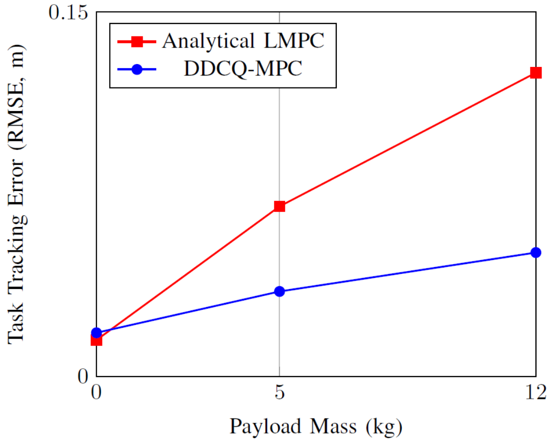

Fig. 3. Tracking error of DDCQ-MPC versus the Analytical LMPC baseline under varying unmodeled payload masses.

4.4 Comparison with State-of-the-Art Methods

To further situate our framework's performance, we conducted a comparative analysis against several state-of-the-art data driven and robust MPC controllers from recent literature. We selected three representative methods:

- Reactive-Contour [6]: A recent method focused on reactive contouring control, adapted for our tracking task.
- Robust-CMPC [5]: A robust convex MPC method that uses set-based uncertainty and collision avoidance.
- Prompt-DT [14]: An open-world MPC framework that leverages digital twins and prompt-based task specification.

We evaluated all methods on the most challenging task: tracking a complex trajectory with an unmodeled 12kg payload.
The results, summarized in Table 1, highlight the advantages of our approach.

Table 1. Comparison with State-Of-The-Art Methods (12KG Payload)

Method	Tracking Error (RMSE, m)	Computation Time (ms)	Task Success Rate (%)
Analytical LMPC [2]	0.125	0.78	75%
Reactive-Contour [6]	0.098	1.85	82%
Robust-CMPC [5]	0.081	4.12	91%
Prompt-DT [14]	0.075	3.55	90%
DDCQ-MPC (Ours)	0.051	0.89	98%

While robust methods like Robust-CMPC [5] and Prompt-DT [14] show improved tracking over the analytical baseline, they incur a significant computational cost, making them unsuitable for 1kHz control. The Reactive-Contour method [6] is faster but less accurate. Our DDCQ-MPC framework achieves the best tracking performance and success rate, while its computation time remains well under the 1ms threshold, demonstrating a superior balance of robustness, accuracy, and real-time capability.

5 Conclusions

In this work, we introduced DDCQ-MPC, a novel framework that integrates a learned dynamics model into a real-time, constrained MPC loop. By replacing an idealized analytical model with a neural network trained on real-world data, our controller achieves superior robustness to unmodeled dynamics, such as varying payloads. Our experiments demonstrate that DDCQ-MPC significantly outperforms a state-of-the-art analytical baseline and other recent methods in challenging tasks, while retaining 1kHz real-time performance and guaranteeing safe operation by respecting physical constraints. This work shows that data-driven predictive control is a viable and effective strategy for the next generation of high-performance industrial robots.

References

1. S. Kleff, A. Meduri, R. Budhiraja, N. Mansard, and L. Righetti, “High-frequency nonlinear model predictive control of a manipulator,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7330–7336.
2. J. Lee, M. Seo, A. Bylard, R. Sun, and L. Sentis, “Real-time model predictive control for industrial manipulators with singularity-tolerant hierarchical task control,” *arXiv preprint arXiv:2209.11880*, 2022.
3. E. Kang, H. Qiao, Z. Chen, and J. Gao, “Tracking of uncertain robotic manipulators using event-triggered model predictive control with learning terminal cost,” *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1050–1062, 2022.
4. J. Nubert, J. Kohler, V. Berenz, F. Allg¨ ower, and S. Trimpe, “Safe and fast tracking on robot manipulators: Robust mpc and neural network control,”¨ *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.
5. B. Wullt, “Robust convex model predictive control with collision avoidance guarantees for robot manipulators,” 2025.
6. J. Yoon, “Reactive model predictive contouring control for robot manipulators,” 2025.
7. H. Liu, “A survey on application of data-driven model predictive control in robot control,” *Applied and Computational Engineering*, vol. 154, pp. 14–19, 2025.
8. J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, “Whole-body model-predictive control applied to the hrp-2 humanoid,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.
9. J. Lee, S. H. Bang, E. Bakolas, and L. Sentis, “Mpc-based hierarchical task space control of underactuated and constrained robots for execution of multiple tasks,” in *2020 IEEE 59th Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5942–5949.
10. D. J. Braun, Y. Chen, and L. Li, “Operational space control under actuation constraints using strictly convex optimization,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 302–309, 2019.

11. O. X.-E. Collaboration, "The open x-embodiment dataset and rt-x models," https://robotics-transformer-x.github.io/, 2023.
12. A. Mohan, S. Aparajit, A. Gupta, V. Kumar, S. Levine, and C. Finn, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2310.04079*, 2023.
13. H. Kayan, "Industrial robotic arm anomaly detection," 2021. [Online]. Available: https://www.kaggle.com/datasets/hkayan/industrial-robotic-arm-anomaly-detection
14. C. Ning, "Prompting with the future: Open-world model predictive control with interactive digital twins," 2025.