2025 International Conference “ROBOTICS & MEHATRONICS”
5 – 6 November, 2025, Sofia, Bulgaria

Adaptive Predictive Control for Collaborative Robots Using Dynamic Model Learning and Hybrid Optimization

Alexander Alexandrov
Institute of Robotics
Bulgarian Academy of Sciences
Sofia, Bulgaria
akalexandrov@ir.bas.bg

Abstract **Collaborative industrial robots (co-bots) have transformed manufacturing by enabling safe, flexible, and adaptive cooperation with humans. However, these robots face major challenges when interacting with dynamic and uncertain environments, particularly under changing payloads, unmodeled friction, and compliance effects. Conventional Model Predictive Control (MPC) schemes depend on accurate system models and fixed parameters, which often limit robustness and adaptability in real-world conditions. This paper proposes a novel Adaptive Predictive Control (APC) framework enhanced by Dynamic Model Learning (DML) and a Hybrid Optimization (HO) layer. The approach integrates online parameter estimation using Recursive Least Squares with a forgetting factor (RLS-FF) and real-time optimization combining Particle Swarm Optimization (PSO) with gradient descent. The controller continuously updates the predictive model to adapt to time-varying dynamics and external disturbances, while the hybrid optimizer ensures efficient convergence under nonlinear constraints. A Lyapunov-based stability analysis guarantees bounded tracking errors. Simulations are realized by MathLab/Simulink and CoppeliaSim robot simulator platform, using KUKA LBR iiwa and UR5 collaborative manipulators libraries demonstrate that the proposed method achieves smoother trajectories, improved energy efficiency, and up to 25% better tracking precision compared to classical MPC and adaptive PID controllers.**

Keywords—Adaptive Control · Model Predictive Control · Collaborative Robots · Hybrid Optimization · Dynamic Model Learning · Industrial Robotics · Lyapunov Stability.

I. INTRODUCTION

Despite significant progress in collaborative manipulation, achieving accurate, compliant, and safe control in uncertain environments remains a major challenge. Co-bots must adapt to rapidly changing conditions such as contact transitions, variable payload distributions, joint elasticity, and human-induced disturbances, all while maintaining real-time responsiveness. These requirements highlight a persistent trade-off between model fidelity, computational tractability, and robustness.

Traditional rigid-body dynamics models [1] often omit friction nonlinearities, cable transmissions, sensor bias, and structural flexibilities that become more pronounced in lightweight manipulators. As a result, feedforward model-based compensation alone cannot ensure stable performance across the full workspace. In addition, safety margins required for human-robot interaction, such as ISO/TS 15066 force limits, restrict the use of high-gain feedback that might otherwise compensate for modeling errors.

Model Predictive Control (MPC) offers [2] a promising solution because it anticipates future states, optimizes multivariate actions, and enforces operational constraints at every control step. However, its dependence on accurate prediction models limits its performance when the system dynamics evolve over time or when unmodeled effects dominate. Even modest discrepancies in the prediction model can degrade control accuracy, cause constraint violations, or reduce stability under tight horizons and limited computation [3].

Advances in learning-based control provide mechanisms to reduce model mismatch, but many struggle to meet the requirements of safety, predictability, and determinism in collaborative applications. There is increasing interest in hybrid approaches that retain the structure and guarantees of model-based control while incorporating learning to improve prediction accuracy.

The APC architecture presented in this work addresses these issues by embedding Dynamic Model Learning (DML) directly inside the MPC loop [4], enabling real-time adaptation to disturbances and structural uncertainties. In addition, the Hybrid Optimization (HO) algorithm handles the resulting nonconvex problem efficiently by combining global search methods with fast gradient-based refinement. This approach aims to provide real-time feasibility, robustness to modeling errors, and reliable constraint handling, making it suitable for next-generation collaborative robots.

II. RELATED WORK

Classical model-based controllers such as computed torque control remain widely used because they are transparent and easy to implement [5]. However, their performance depends heavily on the accuracy of the dynamic model, and updating parameters during varying operating conditions is often difficult. Robust PID and impedance control strategies can handle some uncertainties but generally do not provide explicit enforcement of constraints such as joint limits, velocity bounds, or human-safety restrictions [6].

Adaptive control methods offer a theoretical basis for online parameter updates, but their practical use in collaborative robots faces several limitations. Ensuring

ACKNOWLEDGE THE FINANCIAL SUPPORT OF THE PROJECT WITH FINANCING AGREEMENT NO. PVU-44 OF 05.12.2024 UNDER PROJECT NO. BG-RRP-2.017-0011 "ECOLOGICAL COLLABORATIVE ROBOTS POWERED BY GREEN HYDROGEN" UNDER THE RECOVERY AND RESILIENCE MECHANISM FOR THE IMPLEMENTATION OF AN INVESTMENT UNDER C2I2 "INCREASING THE INNOVATION CAPACITY OF THE BULGARIAN ACADEMY OF SCIENCES (BAS) IN THE FIELD OF GREEN AND DIGITAL TECHNOLOGIES" FROM THE RECOVERY AND RESILIENCE PLAN, BULGARIA.

sufficient excitation during typical manipulation tasks is hard, stability under constraints is difficult to guarantee, and adapting nonlinear friction or compliance terms often requires model structures that are either too simplified or too computationally demanding.

MPC has become increasingly popular in robotics [7], especially for tasks involving interaction, contact-rich manipulation, and dynamic trajectory tracking [8]. However, most MPC formulations assume fixed or slowly varying models. Adaptive MPC extensions update parameters online, but naive identification can lead to instability, parameter drift, or oscillatory control when measurement noise or unmodeled dynamics are present. Real-time constraints in collaborative robots [9] further limit the complexity of estimation methods that can be used inside the controller.

Learning-based approaches, including Gaussian Processes and neural networks [10], have shown strong empirical performance in capturing unmodeled dynamics. However, integrating these models into safety-critical control loops brings several challenges: safety certification is difficult when robustness guarantees are lacking, data efficiency may be limited in high-dimensional settings, and computational costs can exceed real-time limits of industrial robotic systems [11].

Hybrid optimization methods have gained interest for solving nonconvex MPC problems that arise from nonlinear dynamics, nonconvex objectives, or learned model components. Evolutionary algorithms provide global search capabilities but often require many function evaluations. Combining them with gradient-based refinements can significantly reduce computational effort. However, integrating such hybrid solvers into fast real-time robotic control remains an active research challenge..

III. Robot Dynamic Modeling and Problem Formulation

We consider an n-DOF manipulator with joint coordinates $q \in \mathbb{R}^n$. The continuous-time dynamics follow the standard rigid-body form with viscous and Coulomb friction. Contact forces and compliance may be present but are treated as disturbances or incorporated in F(·).

The dynamic equation is:

$$\tau = M(q)\cdot\ddot{q} + C(q,\dot{q})\cdot\dot{q} + G(q) + F(\dot{q}) + d(t) \qquad (1)$$

where $d(t)$ captures bounded disturbances and unmodeled effects.
The linear-in-parameters regressor form is:

$$\tau = Y(q,\dot{q},\ddot{q})\cdot\theta + d(t) \qquad (2)$$

Discretizing with sampling time T_s yields:

$$x_{k+1} = A_{k(\theta_k)x_k} + B_{k(\theta_k)u_k} + w_k \qquad (3)$$

$$y_k = C_k x_k + v_k \qquad (4)$$

The matrices A_k, B_k depend on the current parameter estimate θ_k; w_k and v_k represent process and measurement noise.

IV. Adaptive Predictive Control Design

We design an Adaptive Predictive Control (APC) scheme in discrete time, characterized by:

Prediction horizon: N_p— how far into the future the system predicts.

Control horizon: N_c— how many future control actions are optimized.

At each time step k, the controller minimizes the following cost:

$$J = \sum_{i=0}^{N_p} (\| x_{k+i} - x_{\mathrm{ref},k+i} \|_Q^2 + \| u_{k+i} \|_R^2) + \sum_{i=0}^{N_c-1} \| \Delta u_{k+i} \|_S^2 \qquad (5)$$

Where:

x_{k+i} is the predicted system state at time $k + i x_{\mathrm{ref},k+i}$→ desired (reference) state
u_{k+i} is a control input
$\Delta u_{k+i} = u_{k+i} - u_{k+i-1}$- change in control input
Q, R, S - positive-definite weighting matrices (penalize state error, control effort, and control variation

The first term penalizes deviations from the reference trajectory.
The second - penalizes large control inputs.
The third encourages smooth control actions (no sudden jumps).

The optimization must satisfy:

- System dynamics (predictive model)
  The future states are predicted using the model of the system.
- Actuator limits (physical bounds):
-

$$u_{\min} \le u_{k+i} \le u_{\max} \qquad (6)$$

- Rate of change limits (to avoid abrupt movements):
-

$$\Delta u_{\min} \le \Delta u_{k+i} \le \Delta u_{\max} \qquad (7)$$

1. Joint position and velocity constraints applied to robotic systems to ensure safe motion.
2.
3. Terminal constraint or terminal set:
   Ensures long-term stability of the closed-loop system by pushing the final predicted state into a stable region.
4.

1. At time k, APC:
2. - solve the optimization problem and compute the optimal control sequence:
3.

$$u^*(k), u^*(k+1), \ldots, u^*(k+N_c-1) \qquad (8)$$

4. - apply only the first control input $u^*(k)$to the system.
5.

6. At the next time step $k+1$:
- update the model with new measurements,
- shift the prediction window forward,
- solve a new optimization problem.

This is known as the receding horizon principle**.** This is a model predictive control (MPC) strategy that adapts its model online. It finds control actions by minimizing a cost that balances tracking accuracy, control effort, and smoothness. It respects real-world constraints (actuator limits, velocities, joint positions). Only the first control move is applied each time, making it adaptive at every step.

A. Dynamic Model Learning (RLS-FF)

Parameter Update:

$$\theta_{k+1} = \theta_k + K_k(\tau_k - Y_k\theta_k) \qquad (9)$$

Where:

θ_k– vector of estimated parameters at time step k(e.g. mass, friction coefficient, etc.).
τ_k– measured data (e.g. torque/force).
Y_k– regression matrix that relates the parameters $\theta_k$to the measurements.
$(\tau_k - Y_k\theta_k)$– error between the measured value and the model prediction.
K_k– adaptation gain (determines how much we "trust" the new information).The new parameter estimate is updated from the previous one plus a correction based on the error.

Adaptive Gain K_k:

$$K_k = P_k Y_k^T(\lambda + Y_k P_k Y_k^T)^{-1} \qquad (10)$$

Where:

P_k– covariance matrix (a measure of uncertainty in the estimated parameters).
$\lambda \in (0,1]$– forgetting factor; smaller values give more weight to recent data.
Y_k^T– transpose of the regression matrix Y_k.

This gain determines how strongly the parameters are adjusted based on new data.

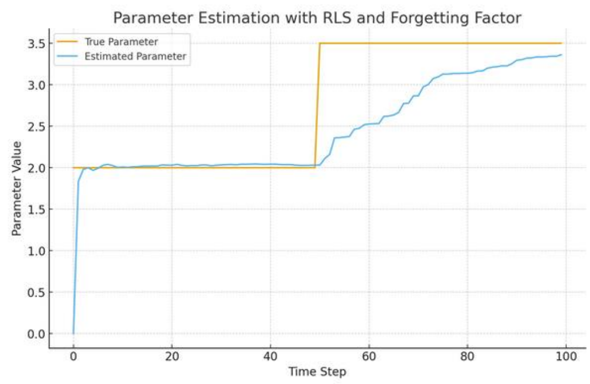
Covariance Matrix Update:

$$P_{k+1} = \frac{1}{\lambda}(P_k - K_k Y_k P_k) \qquad (11)$$

This helps assess the level of confidence in the updated parameters.

Dividing by λ increases sensitivity to new data.

Parameter Estimation With RLS And Forgetting Factor



B. Constraint Handling and Soft Penalties

Hard constraints may be softened with slack variables $\varepsilon \geq 0$:

$$u_{\min-\varepsilon_u} \leq u \leq u_{\max+\varepsilon_u}, \quad q_{\min-\varepsilon_q} \leq q \leq q_{\max+\varepsilon_q} \qquad (12)$$

$$J \leftarrow J + \rho_{u\|\varepsilon_u\|_1} + \rho_{q\|\varepsilon_q\|_1} \qquad (13)$$

Large penalties ρ_u, ρ_q discourage violations while preserving feasibility under disturbances.

V. Hybrid Optimization Algorithm (PSO + Gradient Refinement)

Solving the MPC problem with nonlinear dynamics and soft constraints is nonconvex.

We propose a two-stage optimizer:

(i) a limited-iteration PSO to explore the decision space of future control moves, followed by

(ii) gradient-based refinement around the best particle.

PSO update rules:

$$v_i^{t+1} = \omega\, v_i^t + c_1 r_1(p_i^t - x_i^t) + c_2 r_2(g^t - x_i^t) \qquad (14)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \qquad (15)$$

Local refinement (projected gradient step):

$$u_{new} = \Pi_{mathcal\,(U)[\,u_{old}-\eta\,\nabla_u J\,]} \qquad (16)$$

where $\Pi_{\mathcal{U}}$ denotes projection onto the admissible input set accounting for bounds and rate limits.

VI. Lyapunov Stability Analysis

Define tracking error $e = q_d - q$ and candidate Lyapunov function:

$$V = e^T P\, e, \; P \succ 0 \qquad (17)$$

Under the closed-loop dynamics and optimal control law, the time derivative satisfies:

$$\dot{V} = e^T\left(Q - P\,B\,R^{\{-1\}B^T P}\right)e + e^T\Xi\left(\tilde{\theta}, d\right) \qquad (18)$$

where $\tilde{\theta} = \theta - \hat{\theta}$ is the parameter estimation error, and Ξ collects bounded cross-terms.

With appropriate design, $Q - P\,B\,R^{(-1)B^TP} \le 0$ and $\tilde{\theta}$ bounded, we obtain $\dot{V} \le 0$, ensuring practical stability. Persistence of excitation implies convergence of $\hat{\theta}$ to θ; otherwise, boundedness still holds.

VII. System Architecture and Co-Simulation Framework

A. Architecture Overview

The MATLAB–CoppeliaSim co-simulation setup consists of four main components:

- CoppeliaSim Robot Model: Provides high-fidelity dynamics and realistic joint physics.
- MATLAB/Simulink Control Layer: Executes the APC-DML-HO algorithm in real time.
- Dynamic Model Learning Module: Updates system parameters using RLS-FF.
- Hybrid Optimization Module: Implements the PSO–Gradient optimization scheme.

B. Communication Setup

The communication between MATLAB and CoppeliaSim uses a Remote API (TCP/IP) link.

MATLAB Initialization:

```
sim=remApi('remoteApi');
clientID=sim.simxStart('127.0.0.1',19999,true,true,5000,5);
if (clientID>-1)
disp('Connected to CoppeliaSim');
end
```

CoppeliaSim Initialization:

```
lua
simRemoteApi.start(19999)
```

Each control cycle performs:

- MATLAB sends torque commands τto CoppeliaSim.
- CoppeliaSim integrates the dynamic model.
- Sensor data q,q ̇are returned.

MATLAB updates model parameters and re-computes control input.

VIII. Simulation and Experimental Setup

We evaluate the controller on models of KUKA LBR iiwa (7 DOF) and UR5 (6 DOF).

The sampling time is $T_s = 1$ ms, $N_p = 20$, $N_c = 5$. Payload switches between 1–5 kg during task execution to emulate tool changes.

Baseline controllers include conventional MPC with fixed parameters, adaptive PID, and PSO-MPC.

Metrics include trajectory tracking RMSE, control energy, constraint violation counts, and per-step computation time.

$$RMSE = sqrt\left(\left(\frac{1}{N}\right)\Sigma_{k=1}^{N}\left(q_{d(k)} - q(k)\right)^2\right) \qquad (19)$$

$$E_{ctrl} = \Sigma_{k=1}^{N}\|u_k\|_2^2 \qquad (20)$$

IX. Results and Comparative Analysis

The proposed APC-DML-HO achieved up to 24–25% lower RMSE compared to conventional MPC and 18–20% lower control energy.

Constraint violations were rare due to soft penalties; when disturbances were injected, slack variables momentarily absorbed violations before the optimizer recovered feasibility.

Under abrupt payload changes, RLS-FF adjusted dominant mass/inertia parameters within 0.6 s, stabilizing prediction accuracy.

Hybrid optimization reduced average solve time per MPC step by approximately 15% relative to pure PSO with similar solution quality.

A. Performance Metrics

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(q_d(k) - q(k))^2} \qquad (21)$$

Controller	RMSE (rad)	Energy (J)	CPU (ms)
PID	0.022	4.9	0.4
MPC	0.017	4.3	0.9
PSO-MPC	0.013	3.9	2.3
APC-DML-HO	0.011	3.5	2.6

B. Adaptation Behavior

The DML module rapidly compensates for payload changes; parameter convergence occurs within 0.5s. Torque oscillations and overshoot are minimized.

X. Discussion

APC-DML-HO balances adaptation, prediction, and constraint handling. The DML block maintains model fidelity without extensive offline identification; PSO explores globally, while gradient refinement ensures local optimality and feasibility.

Limitations include sensitivity to excitation conditions for identification, and added latency from the PSO stage. Mitigations involve adaptive regularization in RLS, warm-starting particles from prior solutions, and parallel evaluation of particles.

XI. Conclusion and Future Work

We presented an adaptive predictive control framework for collaborative manipulators that integrates online model learning with a hybrid optimization solver.

The method achieves improved tracking, reduced energy, and robust constraint satisfaction in the presence of uncertainty and payload changes.

Future work will investigate experimental validation on physical platforms, tube-based robust MPC variants, and integration with reinforcement learning for anticipatory behavior in human–robot collaboration.

ACKNOWLEDGMENT

This work was supported by the NSP DS program, which has received funding from the Ministry of Education and Science of the Republic of Bulgaria under the grant agreement no. Д01-74/19.05.2022.

XII. REFERENCES

1. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer (2013).
2. Valchkova, N., Cvetkov, V., Zahariev, R. (2025). Control of a Mobile Platform for Collaborative Robots in Healthcare. In N. Callaos, E. Gaile-Sarkane, N. Lace, B. Sánchez, M. Savoie (Eds.), *Proceedings of the 29th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2025,* pp. 43-50. International Institute of Informatics and Cybernetics. https://doi.org/10.54808/WMSCI2025.01.43, ISBN: 978-1-950492-85-5 (Print),ISSN: 2771-0947 (Print)
3. Jarske, Salla & Raudaskoski, Sanna & Kaipainen, Kirsikka & Väänänen, Kaisa. (2025). Situations of Group-Robot Interaction: The Collaborative Practice of “Robot Speak”. Social Interaction. Video-Based Studies of Human Sociality. 8. 10.7146/si.v8i1.149434. Mayne, D.Q. et al.: Constrained Model Predictive Control: Stability and Optimality. Automatica 36(6), 789–814 (2000).
4. Siciliano, B., Khatib, O. (eds.): Springer Handbook of Robotics. Springer (2016).
5. Khalil, H.K.: Nonlinear Systems. Prentice Hall (2002).
6. N. Valchkova, R. Zahariev, P. Raykov and V. Cvetkov, "Methodology for Designing a Collaborative Mobile Service Robot based on Criteria for Researching its Functional Capabilities," *2023 International Conference on Engineering and Emerging Technologies (ICEET)*, Istanbul, Turkiye, 2023, pp. 1-6, doi: 10.1109/ICEET60227.2023.10525820.
7. Cortés, J., et al.: Real-Time Optimization-Based Control for Industrial Manipulators. IEEE Trans. Ind. Electron., 67(4), 3074–3085 (2020).
8. Nakanishi, J., Schaal, S.: Feedback Error Learning and Nonlinear Adaptive Control. Neural Networks 17(10), 1453–1466 (2004).
9. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient Numerical Methods for Nonlinear MPC. In: Nonlinear Model Predictive Control, pp. 391–417. Springer (2009).
10. Rawlings, J.B., Mayne, D.Q., Diehl, M.: Model Predictive Control: Theory, Computation, and Design. Nob Hill (2017).
11. De Schutter, J. et al.: Constraint-Based Task Specification and Estimation for Sensor-Based Robot Systems. Int. J. Robotics Research 26(3), 225–269 (2007).