Pareto-Optimized Model Predictive Control for Dynamically Feasible Three-Dimensional Trajectory Generation in Robotic Manipulators

Zeinel Momynkulov[1], Sayat Ibrayev[2], Azizah Suliman[3], Yegenberdi Tenizbayev[4], Batyrkhan Omarov[5]*
Joldasbekov Institute of Mechanics and Engineering, Kazakhstan[1, 2, 5]
International Information Technology University, Kazakhstan[1, 5]
Asia Metropolitan University, Malaysia[3]
Central Asian Innovative University, Kazakhstan[4]
Narxoz University, Kazakhstan[5]
Khalel Dosmukhamedov Atyrau University, Kazakhstan[5]

Abstract*—This study presents a Pareto-optimized Model Predictive Control (MPC) framework for dynamically feasible three-dimensional trajectory generation in robotic manipulators operating under physical constraints. Unlike conventional interpolation-based methods that emphasize geometric smoothness while neglecting system dynamics, the proposed approach integrates a second-order discrete-time model with explicit constraints on position, velocity, and acceleration, ensuring physically consistent motion profiles. A multi-objective optimization strategy is introduced, combining grid search with Pareto front analysis to systematically tune key MPC parameters, including prediction horizon and discretization step. This enables a principled trade-off between tracking accuracy and control effort, addressing a critical limitation in existing MPC implementations that rely on heuristic parameter selection. Experimental results demonstrate that the proposed method achieves competitive tracking performance while significantly improving trajectory smoothness and reducing acceleration peaks compared to spline-based and linear interpolation approaches. The framework maintains real-time feasibility with computation times below 20 ms per control cycle, making it suitable for practical deployment in robotic systems. Furthermore, the integration of learning-based trajectory generation highlights the adaptability of the approach in complex and dynamic environments. Overall, the proposed methodology offers a scalable, interpretable, and computationally efficient solution that bridges the gap between geometric trajectory planning and physically realizable robotic motion, contributing to the advancement of control-aware trajectory generation in modern robotic applications.

Keywords*—*Model Predictive Control; trajectory planning; robotic manipulators; Pareto optimization; multi-objective optimization; dynamic constraints; real-time control; 3d motion planning

I. Introduction

Robotic manipulators have become indispensable components of modern intelligent systems, playing a pivotal role in industrial automation, precision manufacturing, medical robotics, and space exploration. Their increasing deployment in complex and dynamic environments has significantly raised the demands placed on motion planning algorithms, particularly in terms of accuracy, robustness, and physical feasibility. In such settings, trajectory generation is no longer a purely geometric problem but must incorporate the physical limitations of actuators and the dynamic behavior of the system [1]. Consequently, the development of advanced trajectory planning strategies that ensure both precision and feasibility has emerged as a critical research direction in robotics.

Traditional trajectory generation approaches, including polynomial interpolation, cubic splines, and B-splines, have been widely adopted due to their computational simplicity and ability to produce smooth paths. These methods are particularly effective in scenarios where geometric continuity is the primary concern. However, they inherently lack awareness of system dynamics and constraints, such as bounded velocity, acceleration, and actuator capabilities [2]. As a result, trajectories generated using purely geometric techniques may be mathematically smooth yet physically infeasible when executed on real robotic platforms. This limitation becomes especially pronounced in high-speed or safety-critical applications, where violations of dynamic constraints can lead to instability or mechanical degradation [3].

To overcome these challenges, control-oriented approaches have gained increasing attention, with Model Predictive Control (MPC) emerging as a particularly powerful framework for trajectory generation. Unlike traditional planners, MPC integrates system dynamics directly into the optimization process, enabling the generation of trajectories that are both dynamically consistent and constraint-compliant [4]. By solving a finite-horizon optimization problem at each control step, MPC can anticipate future system behavior and adjust control inputs accordingly. This predictive capability allows for the simultaneous consideration of multiple objectives, such as tracking accuracy, smoothness, and energy efficiency, within a unified mathematical formulation [5].

Despite its advantages, the practical application of MPC in robotic trajectory planning is often hindered by challenges related to parameter selection and computational efficiency. Key parameters, such as the prediction horizon and discretization time step, significantly influence controller performance and must be carefully tuned to balance accuracy

*Corresponding author

and responsiveness [6]. In many existing studies, parameter tuning is performed empirically or through trial-and-error, which limits reproducibility and generalizability across different robotic systems [7]. Furthermore, the trade-offs between competing objectives, such as minimizing tracking error versus reducing control effort, are rarely explored in a systematic and quantitative manner [8].

In this context, the present study introduces a Pareto-optimized MPC framework for three-dimensional trajectory generation in robotic manipulators, emphasizing both dynamic feasibility and multi-objective performance. By integrating a second-order dynamic model with structured parameter optimization, the proposed approach enables the systematic exploration of trade-offs between trajectory accuracy and control smoothness [9]. The use of Pareto front analysis provides a principled mechanism for selecting optimal parameter configurations tailored to specific application requirements [10]. Moreover, the framework is designed with real-time applicability in mind, ensuring that computational demands remain compatible with practical robotic control systems [11].

II. Related Works

Trajectory planning for robotic manipulators has been extensively investigated, with early approaches primarily relying on geometric interpolation techniques. Methods such as cubic splines and B-splines have been widely adopted due to their ability to generate smooth and continuous paths across predefined waypoints [12]. These approaches ensure continuity in position and higher-order derivatives, making them suitable for applications requiring visually smooth motion profiles. However, their formulation is fundamentally geometric and does not explicitly incorporate system dynamics or actuator constraints, which limits their applicability in real-world robotic systems [13]. As robotic platforms evolve toward higher precision and speed, the gap between geometric smoothness and physical feasibility becomes increasingly evident.

To address these shortcomings, various extensions of spline-based techniques have been proposed. For instance, non-uniform rational B-splines (NURBS) and adaptive spline formulations provide enhanced local control over trajectory shaping, enabling more flexible path design in complex environments [14]. Additionally, hybrid interpolation methods combining spline generation with heuristic constraint handling have been explored to partially account for velocity and acceleration limits [15]. While these methods improve trajectory realism to some extent, they often rely on ad hoc adjustments and lack a principled mechanism for enforcing dynamic feasibility. As a result, their performance may degrade in scenarios involving high curvature or abrupt motion transitions [16].

Parallel to interpolation-based methods, optimization-driven trajectory planning approaches have gained significant attention. Linear and nonlinear programming techniques have been employed to generate trajectories that satisfy kinematic and dynamic constraints simultaneously [17]. These approaches often formulate trajectory generation as a constrained optimization problem, where objectives such as path length, smoothness, and energy consumption are minimized. Although effective, these methods can be computationally expensive, particularly for high-dimensional robotic systems, and may struggle to achieve real-time performance in dynamic environments [18]. Furthermore, the selection of appropriate objective weights remains a challenging task, often requiring manual tuning.

Model Predictive Control (MPC) has emerged as a powerful alternative that integrates system dynamics directly into the trajectory planning process. By solving a finite-horizon optimization problem at each control step, MPC enables predictive and adaptive control of robotic motion [19]. Its ability to explicitly incorporate constraints on position, velocity, and acceleration makes it particularly suitable for high-performance applications [20]. MPC has been successfully applied in various domains, including autonomous vehicles, aerial robotics, and industrial manipulators, demonstrating improved tracking accuracy and robustness under dynamic conditions [21]. Moreover, recent studies have explored the integration of full dynamic models and multi-degree-of-freedom systems within MPC frameworks, further enhancing trajectory realism [22].

Despite these advancements, several challenges remain in the practical deployment of MPC-based trajectory planners. One of the primary issues lies in the selection of controller parameters, such as prediction horizon and sampling interval, which significantly influence performance [23]. Conventional approaches often rely on heuristic or empirical tuning strategies, which may not generalize well across different tasks and environments [24]. To address this limitation, recent works have explored data-driven and optimization-based parameter tuning techniques, including Bayesian optimization and evolutionary algorithms [25]. Additionally, multi-objective optimization frameworks have been introduced to explicitly capture trade-offs between competing performance criteria, such as tracking accuracy and control effort [26]. Pareto front analysis, in particular, has been recognized as an effective tool for identifying optimal parameter configurations in such settings [27].

Recent developments have also focused on enhancing MPC frameworks through integration with perception and learning-based components. For instance, the incorporation of point cloud data and vision-based feedback enables real-time trajectory adaptation in dynamic and uncertain environments [28]. Similarly, learning-based MPC approaches leverage neural networks to approximate system dynamics or optimize control policies, improving computational efficiency and adaptability [29]. Comparative analyses of these methods, summarized in Table I, highlight the diversity of modeling approaches, tuning strategies, and performance outcomes across different applications [30].

TABLE I. COMPARATIVE ANALYSIS OF TRAJECTORY PLANNING AND MPC-BASED METHODS

Ref.	Method Type	Application Domain	Dynamics Model	Constraint Handling	Optimization Strategy	Real-Time Capability	MAE / Accuracy	Key Limitation
[12]	Cubic Spline	Industrial Robotics	None (Geometric)	No	None	Yes	High accuracy	No dynamic feasibility
[13]	B-Spline	Manipulators	None	No	None	Yes	High accuracy	Ignores actuator limits
[14]	NURBS	Complex Path Design	None	Partial	Heuristic	Moderate	Moderate	Complex tuning
[15]	Hybrid Spline	Robotics	Partial	Partial	Heuristic	Moderate	Moderate	Lack of generalization
[17]	Linear Programming	Motion Planning	Kinematic	Yes	Optimization	Low	Moderate	High computation cost
[18]	Nonlinear Optimization	Robotics	Dynamic	Yes	Optimization	Low	High	Computational burden
[19]	MPC	Autonomous Systems	Dynamic	Yes	Predictive	Moderate	High	Parameter sensitivity
[21]	MPC (Extended)	Vehicles/Robotics	Dynamic	Yes	Multi-objective	Moderate	High	Tuning complexity
[22]	Full-Dynamics MPC	Manipulators	Full Dynamics	Yes	Optimization	Low	Very High	High latency
[25]	Bayesian Optimization + MPC	Robotics	Dynamic	Yes	Bayesian	Moderate	High	Implementation complexity
[26]	Multi-objective MPC	Robotics	Dynamic	Yes	Pareto-based	Moderate	High	Trade-off selection
[28]	Perception-driven MPC	Autonomous Systems	Dynamic	Yes	Predictive	Moderate	High	Sensor dependency
[29]	Learning-based MPC	Aerial Robotics	Learned Dynamics	Yes	Neural Optimization	High	Very High	Requires training data
[30]	Hybrid MPC	Mobile Robots	Dynamic	Yes	Mixed	Moderate	High	System-specific tuning

Despite these advances, a unified framework that combines dynamic feasibility, systematic parameter optimization, and real-time performance remains an open research challenge [31]. Existing methods often prioritize specific aspects, such as accuracy or efficiency, without fully addressing the interplay between them [32]. Therefore, there is a clear need for integrated approaches that balance these objectives while maintaining scalability and interpretability [33].

III. MATERIALS AND METHODS

This section presents the methodological foundation of the proposed framework for three-dimensional trajectory generation using Model Predictive Control (MPC). The overall system architecture is illustrated in Fig. 1, while the detailed formulation of the trajectory planning module and the optimization pipeline are depicted in Fig. 2 and Fig. 3, respectively. The proposed methodology integrates dynamic modeling, constrained optimization, and multi-objective parameter tuning to ensure both geometric accuracy and dynamic feasibility.

A. Overall Framework

The complete pipeline, shown in Fig. 1, consists of five sequential stages: input specification, trajectory planning, optimization, comparative evaluation, and output generation. The system receives a set of target waypoints in three-dimensional space along with system parameters and initial conditions. These inputs are transformed into boundary conditions for the MPC-based trajectory planner.

Unlike conventional approaches that separate planning and control, the proposed framework unifies both processes within a predictive optimization structure. This integration allows the system to generate trajectories that are inherently consistent with physical constraints, thereby eliminating the need for post-processing or heuristic corrections.

B. Dynamic Modeling and State Representation

The trajectory planning module, detailed in Fig. 2, is built upon a second-order discrete-time dynamic model that captures both kinematic and inertial properties of the robotic manipulator. The system state at time step k is defined as:

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix}, \tag{1}$$

where, $p_k \in \mathbb{R}^3$ represents position and $v_k \in \mathbb{R}^3$ denotes velocity. The control input is the acceleration vector $u_k = a_k$.

The system dynamics are described by:

$$p_{k+1} = p_k + v_k \Delta t + \frac{1}{2} a_k \Delta t^2, \tag{2}$$

$$v_{k+1} = v_k + a_k \Delta t, \tag{3}$$

These equations can be expressed in compact state-space form:

$$x_{k+1} = A x_k + B u_k, \tag{4}$$

where, A and B are the system matrices determined by the discretization step Δt . This formulation enables efficient integration with MPC solvers and supports predictive trajectory generation over a finite horizon.

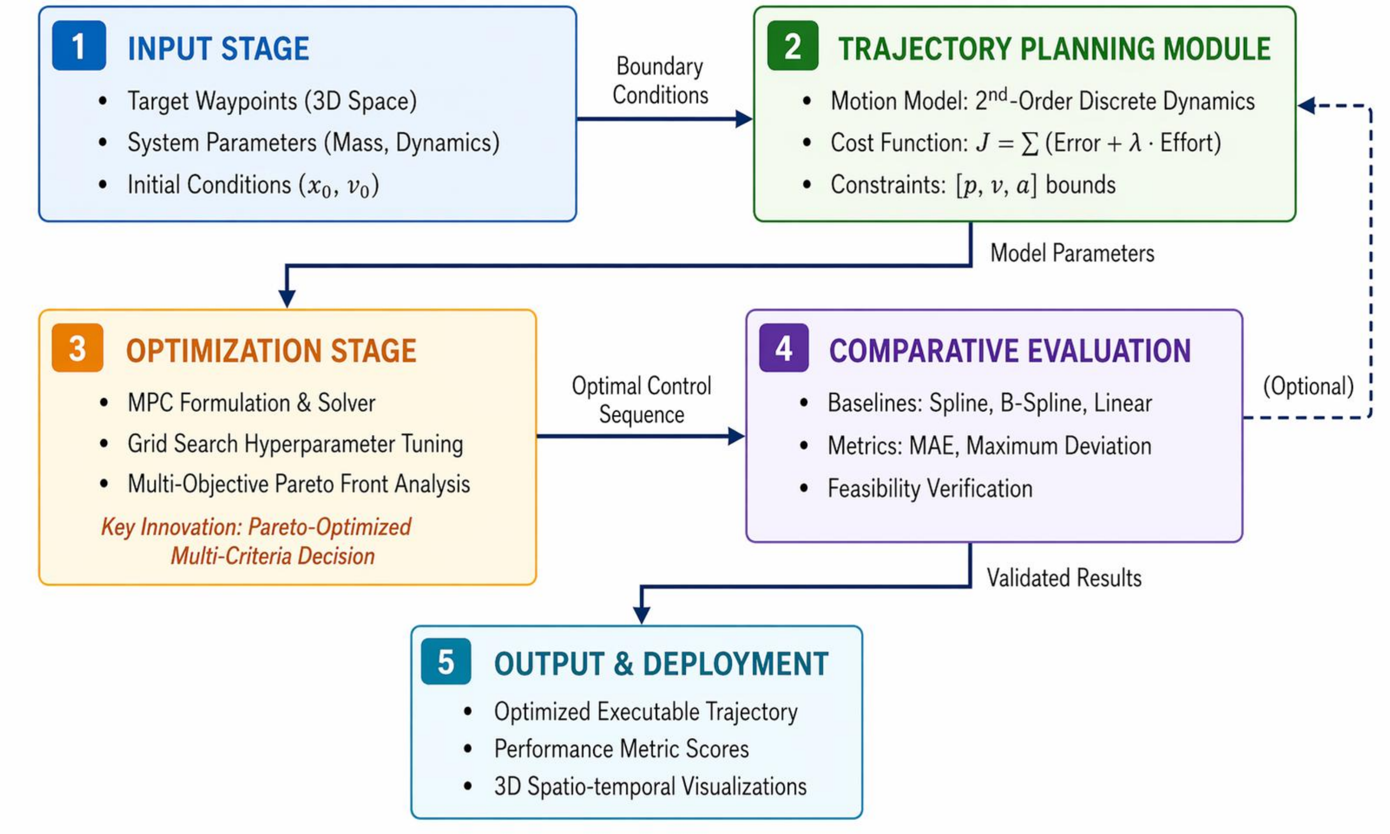

Fig. 1. Overall architecture of the proposed Pareto-optimized model predictive control framework for 3d trajectory generation.

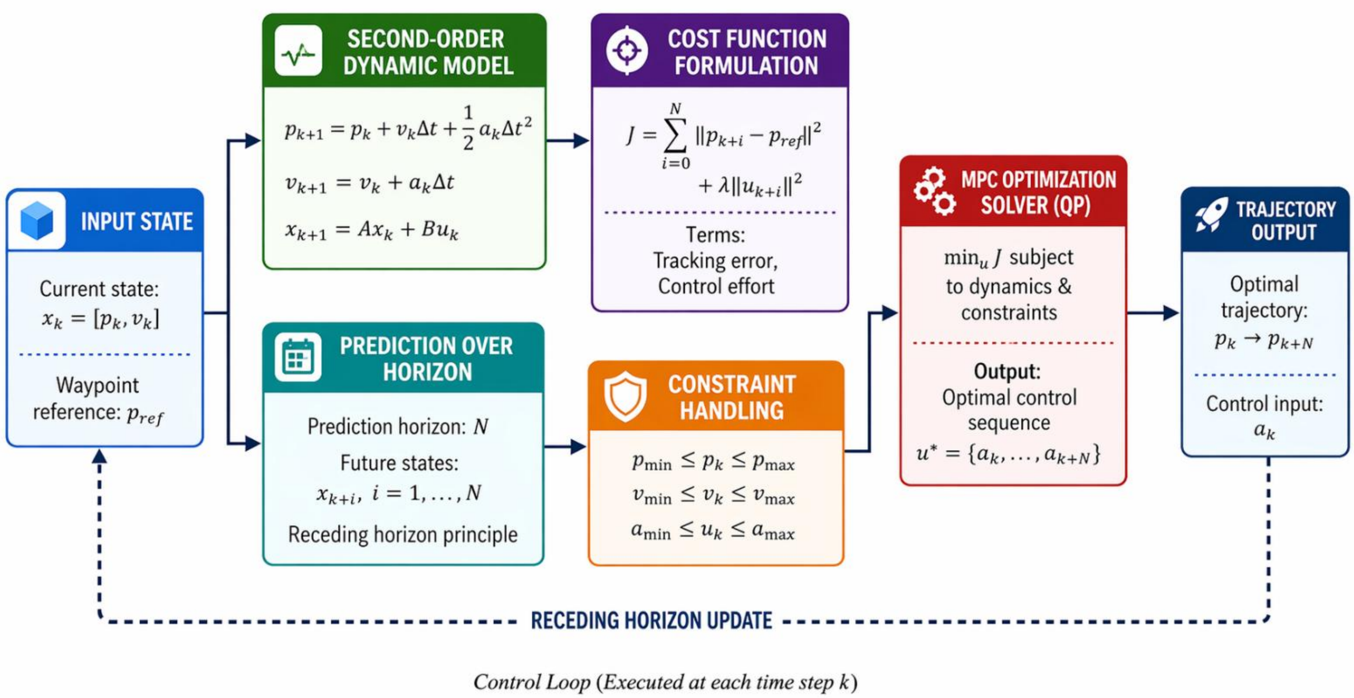

Fig. 2. Detailed trajectory planning module based on model predictive control with second-order dynamics and constraint handling.

C. Model Predictive Control Formulation

At the core of the trajectory planning module is the MPC formulation, which optimizes control inputs over a prediction horizon N. The controller minimizes a multi-objective cost function defined as:

$$J = \sum_{i=0}^{N} \| p_{k+i} - p_{ref} \|^2 + \lambda \| u_{k+i} \|^2, \qquad (5)$$

where, p_{ref} denotes the reference trajectory, and λ is a regularization parameter that balances tracking accuracy and control effort.

The optimization is subject to physical constraints:

$$p_{\min} \le p_k \le p_{\max}, v_{\min} \le v_k \le v_{\max}, a_{\min} \le u_k \le a_{\max}, \qquad (6)$$

As shown in Fig. 2, the MPC framework consists of four key components: dynamic modeling, prediction over horizon, cost function formulation, and constraint handling. These components are jointly processed within a quadratic programming (QP) solver [34] to produce an optimal control sequence.

D. Receding Horizon Control Strategy

The Model Predictive Control framework operates based on the receding horizon principle, a fundamental concept that enables adaptive and forward-looking decision-making in dynamic systems. At each discrete time step, the controller formulates and solves a finite-horizon optimization problem using the current system state as the initial condition. This optimization predicts future system behavior over a specified horizon and computes an optimal sequence of control inputs that minimizes the defined cost function while satisfying system constraints. However, instead of applying the entire control sequence, only the first control input is executed, and the remaining inputs are discarded.

In the subsequent time step, the optimization process is repeated with updated system information, effectively shifting the prediction horizon forward. This iterative procedure allows the controller to continuously incorporate new state measurements and respond to disturbances, modeling inaccuracies, or environmental changes in real-time. As a result, the receding horizon strategy enhances robustness and flexibility, ensuring that the generated trajectory remains dynamically feasible and aligned with the evolving system conditions [35], [36], [37]. This adaptive behavior is particularly critical in robotic applications, where uncertainties and nonlinear dynamics necessitate continuous re-evaluation of control actions to maintain stability and performance.

The presented algorithm formalizes the trajectory planning process within a receding-horizon MPC framework, ensuring continuous adaptation to system dynamics and reference updates. By explicitly integrating second-order dynamics and constraint handling, the method guarantees that generated trajectories remain physically feasible throughout execution. Furthermore, the structured optimization procedure enables a balanced trade-off between tracking accuracy and control effort, which is critical for high-performance robotic applications. This formulation also facilitates seamless integration with the Pareto-based parameter tuning strategy described in Fig. 3, enhancing both robustness and reproducibility of the overall system.

Algorithm 1. Model Predictive Control for Trajectory Planning

Input:
- *Initial state* $x_0 = [p_0, v_0]$
- *Waypoint sequence* $W = \{w_1, w_2, \ldots, w_m\}$
- *MPC parameters: prediction horizon* N*, time step* Δt*, regularization parameter* λ
- *System constraints:* $p_{min}, p_{max}, v_{min}, v_{max}, a_{min}, a_{max}$

Output:
- *Dynamically feasible trajectory* $\{p_k\}_{k=0}^{T}$
- *Control input sequence* $\{u_k\}_{k=0}^{T-1}$
- ***Initialization:***

1.1 Set the initial state $x_k \leftarrow x_0$
1.2 Initialize time index $k \leftarrow 0$
1.3 Define prediction horizon N*, discretization step* Δt*, and weight* λ
1.4 Load waypoint set W *and select initial reference* $p_{ref} \leftarrow w_1$
1.5 Initialize storage variables for trajectory $\{p_k\}$*, velocity* $\{v_k\}$*, and control inputs* $\{u_k\}$

- ***Main Control Loop (Receding Horizon):***
  While *the final waypoint has not been reached:*

2.1 ***Reference Update and Sub-Goal Selection:***
- *Determine the current target waypoint based on proximity*
- *If the distance* $\| p_k - w_i \|$ *is below a predefined threshold, update* $p_{ref} \leftarrow w_{i+1}$
- *Ensure smooth transition between waypoints without discontinuities*

2.2 ***State Prediction Over Horizon:***
- *Initialize predicted state sequence* $x_{k|k} = x_k$
- *For* $i = 0$ *to* $N - 1$*:*
$$x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i}$$
- *Construct predicted position and velocity sequences:*
$$\{p_{k+i|k}, v_{k+i|k}\}, i = 1, \ldots, N$$

2.3 ***Cost Function Construction:***

Define the multi-objective cost function:
$$J = \sum_{i=0}^{N-1} \| p_{k+i|k} - p_{ref} \|^2 + \lambda \| u_{k+i} \|^2$$
Optionally include terminal cost:
$$J_{terminal} = \| p_{k+N|k} - p_{ref} \|^2$$
Combine into total objective:
$$J_{total} = J + J_{terminal}$$

2.4 ***Constraint Formulation:***
- *Enforce system constraints for all predicted steps:*
$$p_{min} \le p_{k+i|k} \le p_{max}$$
$$v_{min} \le v_{k+i|k} \le v_{max}$$
$$a_{min} \le u_{k+i} \le a_{max}$$
- *Include optional soft constraints with penalty terms if needed*

2.5 ***Optimization Problem Solution:***
- *Solve the quadratic programming (QP) problem:*

$$\min_{u_{k:k+N-1}} J_{total}$$

subject to system dynamics and constraints

- *Obtain optimal control sequence:*

$$u^* = \{u_k^*, u_{k+1}^*, \dots, u_{k+N-1}^*\}$$

2.6 ***Control Application:***

- *Apply only the first control input:*

$$u_k = u_k^*$$

- *Store control input in sequence*

2.7 ***System State Update:***

- *Update system state using discrete dynamics:*

$$x_{k+1} = Ax_k + Bu_k$$

- *Extract updated position and velocity:*

$$p_{k+1}, v_{k+1}$$

2.8 ***Trajectory Logging:***

- *Store p_{k+1}, v_{k+1}, and u_k*
- *Update trajectory sequence*

2.9 ***Horizon Shift (Receding Horizon):***

- *Increment time index $k \leftarrow k+1$*
- *Shift prediction window forward*
- *Use updated state x_{k+1} as new initial condition*

- ***Termination Condition:***
    - *Stop when the final waypoint w_m is reached within tolerance*
    - *Or when maximum time steps T are exceeded*
- ***Output Generation:***

4.1 Return optimized trajectory:

$$\{p_0, p_1, \dots, p_T\}$$

4.2 Return control input sequence:

$$\{u_0, u_1, \dots, u_{T-1}\}$$

4.3 Optionally compute performance metrics:

- *Mean Absolute Error (MAE)*
- *Control effort E*
- *Maximum deviation*

E. Multi-Objective Optimization and Parameter Tuning

The optimization stage, illustrated in Fig. 3, introduces a structured approach for tuning MPC parameters using multi-objective optimization. The parameter space is defined as:

$$\Theta = \{(N, \Delta t, \lambda)\} \tag{7}$$

A grid search is performed over this space to evaluate different configurations. For each parameter set θ_i, trajectory simulation is conducted, and performance metrics are computed.

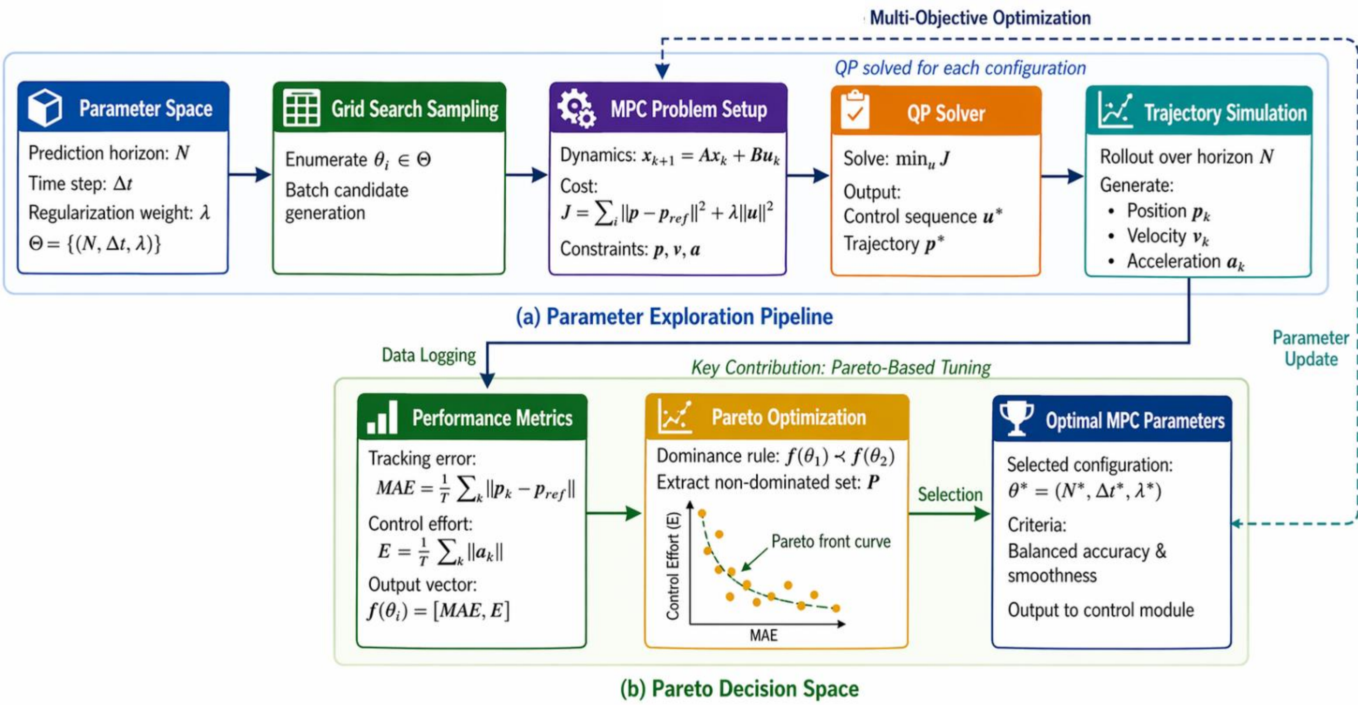

Fig. 3. Optimization pipeline for MPC parameter tuning using grid search and Pareto front analysis.

At the core of the trajectory planning module is the MPC formulation, which optimizes control inputs over a prediction horizon N. The controller minimizes a multi-objective cost function defined as:

Two primary metrics are considered:

Tracking accuracy (MAE):

$$MAE = \frac{1}{T}\sum_{k=1}^{T} \| p_k - p_{ref} \|, \tag{8}$$

Control effort:

$$E = \frac{1}{T}\sum_{k=1}^{T} \| a_k \|, \tag{9}$$

Each configuration produces a performance vector:

$$f(\theta_i) = [MAE, E], \tag{10}$$

F. Pareto Front Analysis

To address the trade-off between accuracy and smoothness, Pareto optimality is employed. A configuration θ_1 dominates θ_2 if:

$$f(\theta_1) \prec f(\theta_2) \tag{11}$$

The set of non-dominated solutions forms the Pareto front:

$$\mathcal{P} = \{\theta \mid \nexists \theta' \text{ δομινατεσ } \theta\} \tag{12}$$

As shown in Fig. 3(b), the Pareto front provides a visual representation of optimal trade-offs, enabling systematic selection of controller parameters based on application requirements [38].

Thus, the proposed framework integrates advanced control theory, optimization techniques, and multi-objective analysis into a unified architecture. By combining predictive control with Pareto-based parameter selection, the method achieves a balance between trajectory accuracy and dynamic feasibility, addressing key limitations of existing trajectory planning approaches.

IV. Results

This section presents the experimental results of the proposed trajectory generation framework through a progressive analysis of parameter tuning, geometric accuracy, dynamic feasibility, learning-based trajectory refinement, and preliminary real-platform validation. The interpretation is organized sequentially according to Fig. 4 to Fig. 12, so that each result can be understood as part of a coherent evaluation pipeline. Overall, the results show that the proposed framework achieves a favorable balance between tracking accuracy, motion smoothness, and implementation feasibility, while also demonstrating compatibility with both simulation-based and physical robotic settings.

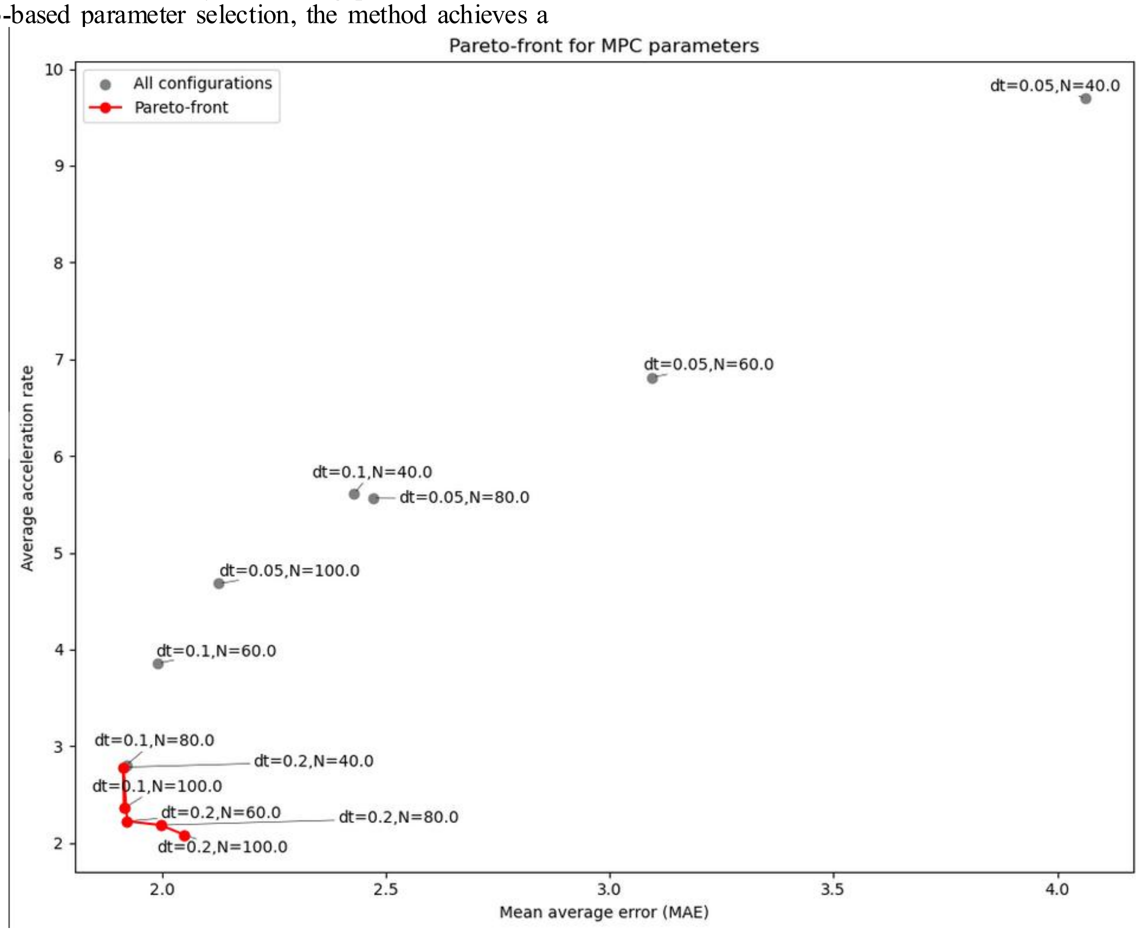

Fig. 4. Pareto front analysis of MPC parameter configurations based on tracking error and control effort.

Fig. 4 presents the Pareto front obtained for the tested MPC parameter combinations, where the horizontal axis corresponds to the mean absolute error (MAE) [39] and the vertical axis represents the average acceleration rate. This figure provides the first layer of evidence regarding the trade-off structure of the optimization problem. All explored configurations are shown as gray markers, while the Pareto-optimal set is highlighted in red. A clear tendency can be observed: configurations with smaller discretization steps and shorter horizons produce larger acceleration values, whereas smoother motion is generally achieved when the horizon is increased and the temporal discretization becomes coarser.

From a control perspective, this result is significant because it verifies that parameter selection cannot be performed using a single metric alone. For example, the configuration with dt=0.05, N=40 lies in the upper-right region of the plot, indicating both high error and excessive acceleration, which makes it undesirable for practical deployment. In contrast, the cluster of Pareto-optimal solutions around dt=0.2 and N=60 to 100 provides substantially lower control effort with only a limited sacrifice in geometric tracking. Therefore, Fig. 4 validates the usefulness of the proposed Pareto-based multi-objective tuning strategy, since it identifies parameter regions that are more suitable for physically executable trajectories than those obtained through naïve or purely empirical tuning.

The geometric consequences of the chosen control configuration are illustrated in Fig. 5, which compares the generated trajectories in three-dimensional space using MPC, cubic spline interpolation, linear interpolation, and B-spline approximation. Two perspectives are provided in order to reveal spatial deviations more clearly. The spline-based and B-spline trajectories remain close to the waypoint structure and exhibit smooth curvature, while the linear interpolation method produces piecewise straight transitions. The MPC trajectory, shown as a dashed path, preserves the global progression of the motion but does not exactly coincide with the interpolative curves.

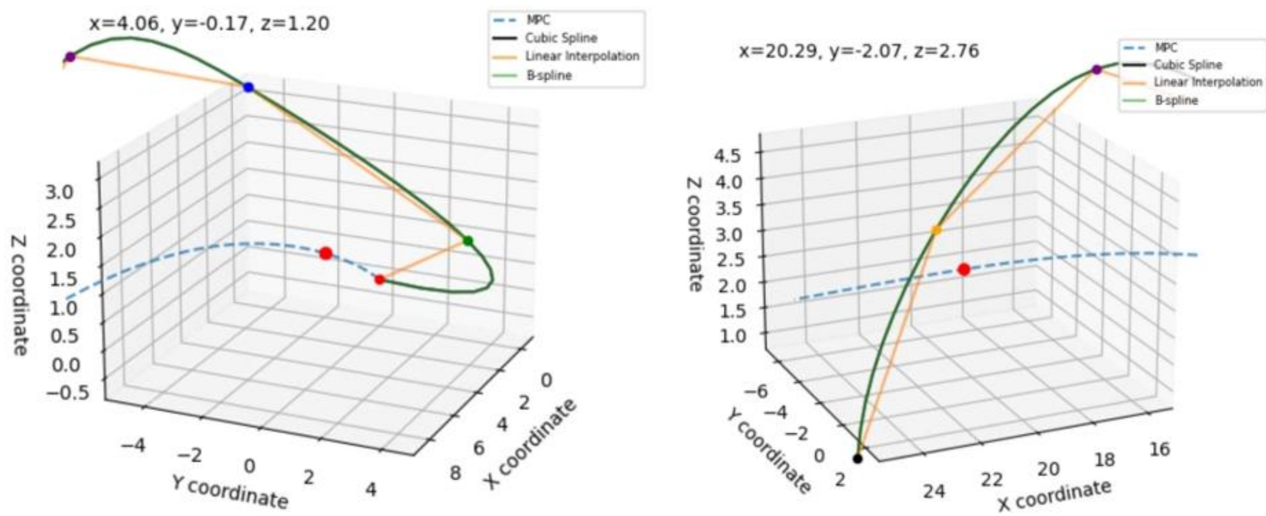

Fig. 5. Three-dimensional trajectory comparison of MPC, cubic spline, linear interpolation, and B-spline methods.

This behavior is expected because the MPC objective is not to replicate a purely geometric reference exactly, but rather to generate a dynamically feasible trajectory under motion constraints. In this sense, the geometric discrepancy visible in Fig. 5 should not be interpreted solely as an error, but as the result of enforcing dynamic admissibility. The figure also indicates that the proposed method avoids unrealistic sharp turns and generates a trajectory that remains smooth in a control-theoretic sense, even when it differs from the exact spline path. Thus, Fig. 5 confirms that the method privileges feasible motion execution over perfect geometric interpolation, which is an important advantage for robotic manipulators operating under acceleration and velocity limits.

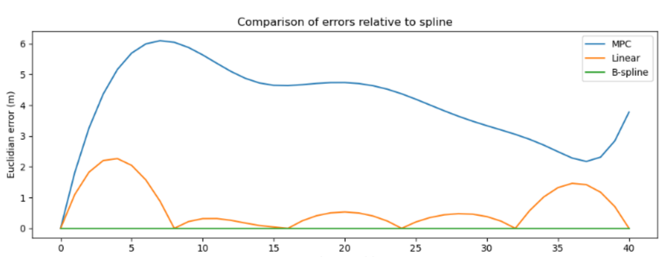

Fig. 6. Euclidean error comparison of trajectory generation methods relative to the spline reference.

The point-wise geometric deviations are quantified more explicitly in Fig. 6, where the Euclidean error relative to the spline trajectory is plotted over time for MPC, linear interpolation, and B-spline. The B-spline curve remains almost coincident with zero throughout the experiment, which indicates that it is extremely close to the spline reference. This is expected because both methods belong to the same family of smooth interpolation techniques. The linear method exhibits moderate oscillatory errors, with distinct local peaks associated with transitions between line segments. The MPC curve shows the largest deviation, particularly in the early and middle stages of the trajectory.

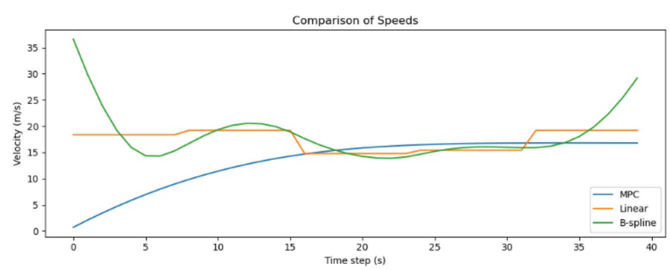

Fig. 7. Velocity profile comparison of MPC, linear interpolation, and B-spline trajectories.

Although the MPC error is higher than that of the interpolation-based methods, the temporal structure of the curve is informative. The deviation increases rapidly at the beginning, stabilizes over the intermediate steps, and then rises again near the end of the trajectory. This suggests that the controller is continuously balancing waypoint attraction with control smoothness and state constraints, rather than aggressively forcing exact geometric coincidence. Therefore, Fig. 6 supports the conclusion that the proposed method sacrifices a portion of geometric fidelity in order to preserve physical plausibility. For practical robotic systems, this trade-off is often desirable because excessive insistence on geometric exactness may generate infeasible inputs or mechanically stressful motion.

The dynamic characteristics of the generated paths are further examined in Fig. 7, which compares the velocity profiles of the MPC, linear [40], and B-spline [41] methods. The MPC velocity increases gradually and then approaches a relatively stable regime, indicating smooth acceleration and controlled temporal evolution. The linear method produces almost piecewise-constant velocity segments with abrupt changes, which reflects the discontinuous nature of line-based interpolation. The B-spline profile shows larger initial and terminal variations, as well as stronger oscillations in comparison with the MPC solution.

This result is highly relevant because velocity continuity directly affects execution quality in robotic manipulators. A profile such as the one produced by MPC is more compatible with real actuators, since it avoids sudden jumps and supports stable motion generation. By contrast, the linear approach introduces discontinuities that may induce jerky behavior, while the B-spline method, although geometrically smooth, may still produce undesired velocity fluctuations due to the absence of explicit dynamic regularization. Hence, Fig. 7 demonstrates that the proposed MPC framework offers a more physically realistic velocity evolution than the baseline approaches.

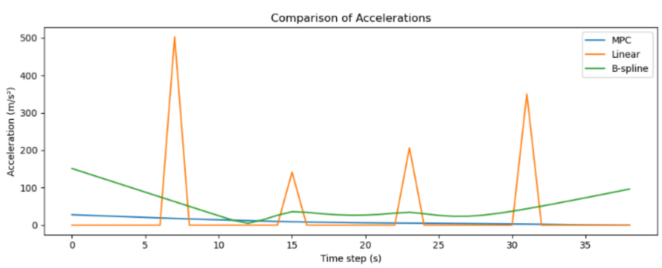

Fig. 8. Acceleration profile comparison under high-variation configuration for different trajectory generation methods.

The differences become even more pronounced in Fig. 8, where the acceleration profiles are shown for the same set of methods. The linear interpolation method [42] exhibits very large spikes, reaching extremely high values at several time steps. These peaks are a direct consequence of sudden changes in velocity at segment boundaries and indicate strongly nonphysical behavior. The B-spline method produces a smoother curve than the linear one, but its acceleration magnitude remains comparatively high over large portions of the trajectory. In contrast, the MPC profile stays low and relatively stable.

This figure is particularly important for validating the proposed framework, because acceleration is directly related to control effort, energy consumption, and actuator stress. A trajectory with large acceleration spikes may be mathematically acceptable in simulation, but it is generally unsuitable for real robotic deployment. The stable and bounded MPC curve in Fig. 8, therefore, provides strong evidence that the proposed controller generates motions that are substantially more feasible from an implementation perspective. In other words, even when the MPC trajectory is not the closest to the spline geometrically, it is clearly superior in terms of dynamic admissibility.

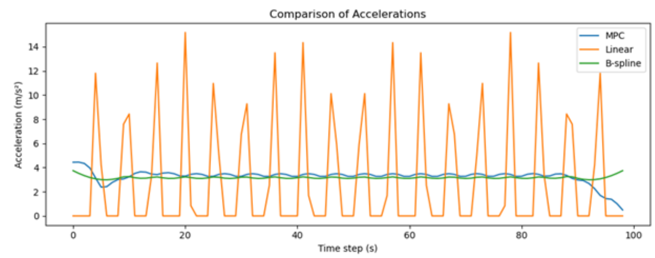

Fig. 9. Acceleration profile comparison under optimized MPC configuration demonstrating improved smoothness.

A second acceleration analysis is given in Fig. 9, corresponding to a different configuration with a longer time horizon. In this case, the MPC profile becomes even more regular, maintaining low variation for most of the execution and only decreasing near the final steps. The linear method still shows repeated impulsive peaks, although their amplitudes are much smaller than in the previous case. The B-spline curve remains smoother than the linear one but still fluctuates more than the MPC result.

The main interpretation of Fig. 9 is that the optimized MPC parameters identified through Pareto analysis are not only theoretically preferable, but also visibly more effective in practice [43]. The controller yields a nearly constant acceleration profile, which implies improved smoothness, reduced mechanical stress, and better suitability for precise manipulation tasks. This result complements Fig. 4 by showing that the benefits of Pareto-based parameter tuning are directly reflected in the actual motion dynamics. Accordingly, Fig. 9 provides experimental confirmation that the proposed tuning strategy improves controller behavior in a tangible and practically meaningful way.

The behavior of the learning-based agent across training episodes is illustrated in Fig. 10, where the generated trajectories are shown for a single target-point scenario. In early episodes, such as Episode 3 and Episode 13, the motion is highly irregular and exhibits unnecessary exploration. The generated paths are circuitous and unstable, indicating that the control policy has not yet converged to a meaningful navigation strategy. By Episode 60, the trajectory becomes more structured, although some residual oscillations are still present. In later episodes, particularly Episodes 100, 144, 156, and 200, the path becomes substantially more direct and stable, connecting the start and goal while appropriately avoiding the obstacle cluster.

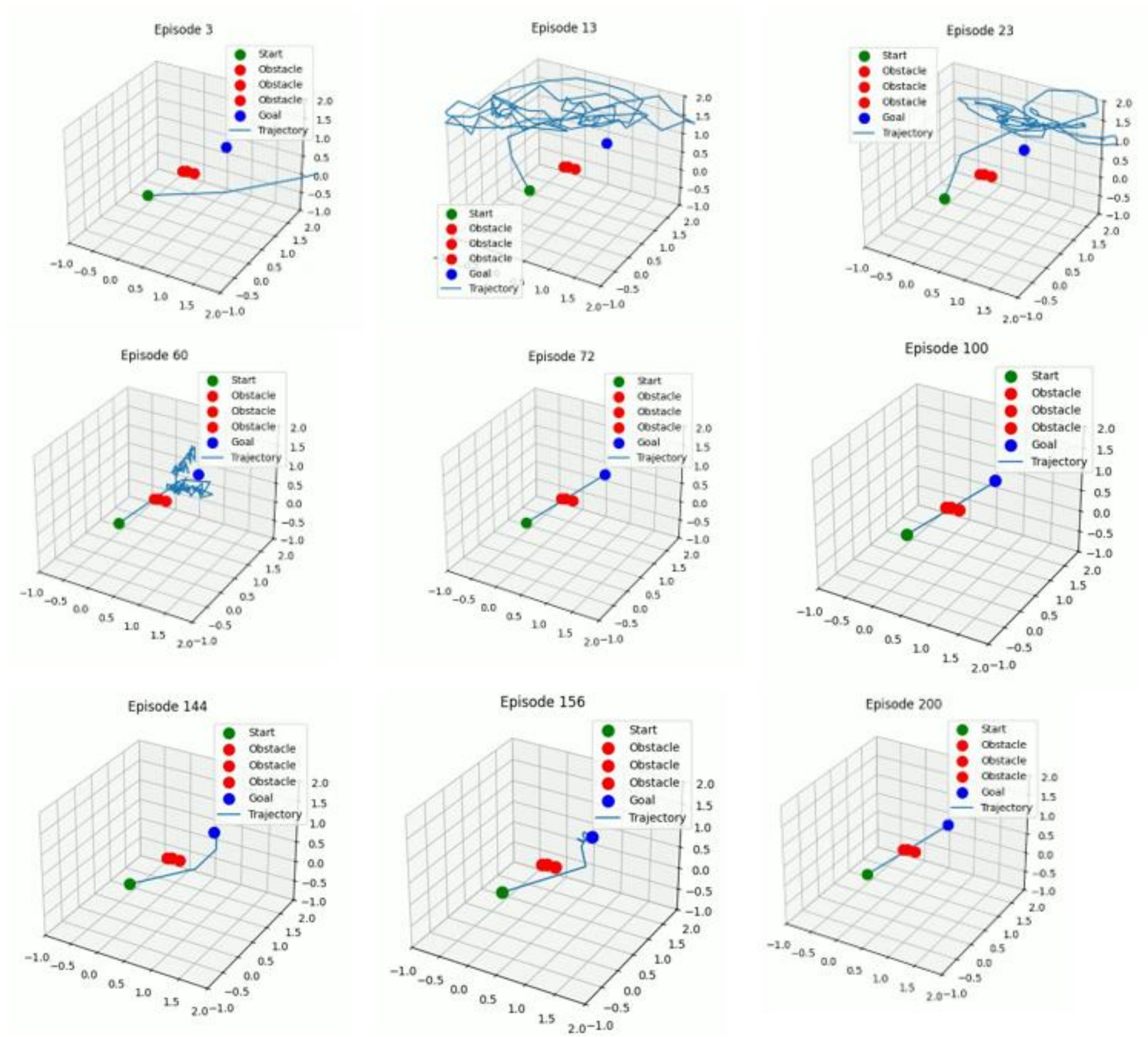

Fig. 10. Evolution of learning-based trajectory generation across training episodes for a single target scenario.

Fig. 10 demonstrates progressive policy improvement through learning. The transition from exploratory and noisy motion to consistent and directed trajectories indicates that the agent successfully internalizes the spatial structure of the environment and learns to generate collision-free motion. From a methodological standpoint, Fig. 10 is important because it shows that the proposed framework can be extended beyond deterministic trajectory planning and can support adaptive behavior in obstacle-rich environments. The convergence trend across episodes suggests that the learned policy complements the model-based components of the framework and may be particularly valuable in settings where waypoint geometry or environmental constraints change over time.

The generalization capability of the learned agent is further demonstrated in Fig. 11, which shows trajectory generation for multiple target points. Two examples are presented, each comparing the generated trajectory with a spline-based reference. In both cases, the learned path follows the global target progression but does not exactly replicate the spline shape. Instead, it constructs a feasible trajectory that moves through the target sequence while preserving obstacle awareness and directional consistency.

This result is noteworthy because multi-target motion planning is more demanding than the single-goal case. The controller or agent must not only approach one terminal point and organize the sequence of movements across multiple intermediate targets. The fact that meaningful trajectories are generated in both examples indicates that the underlying framework possesses a level of structural adaptability. Fig. 11, therefore, suggests that the proposed method is capable of handling more complex planning tasks than single-waypoint tracking and can serve as a basis for scalable multi-stage robotic motion generation.

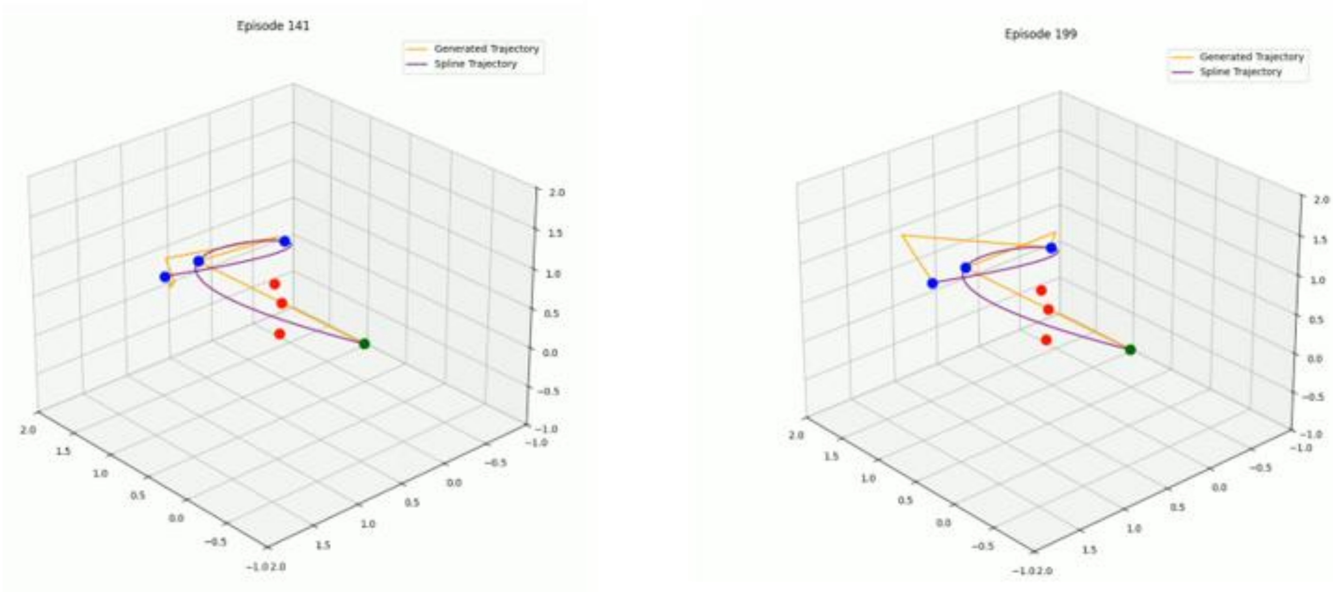

Fig. 11. Trajectory generation for multiple target points using learning-based and spline-based approaches.

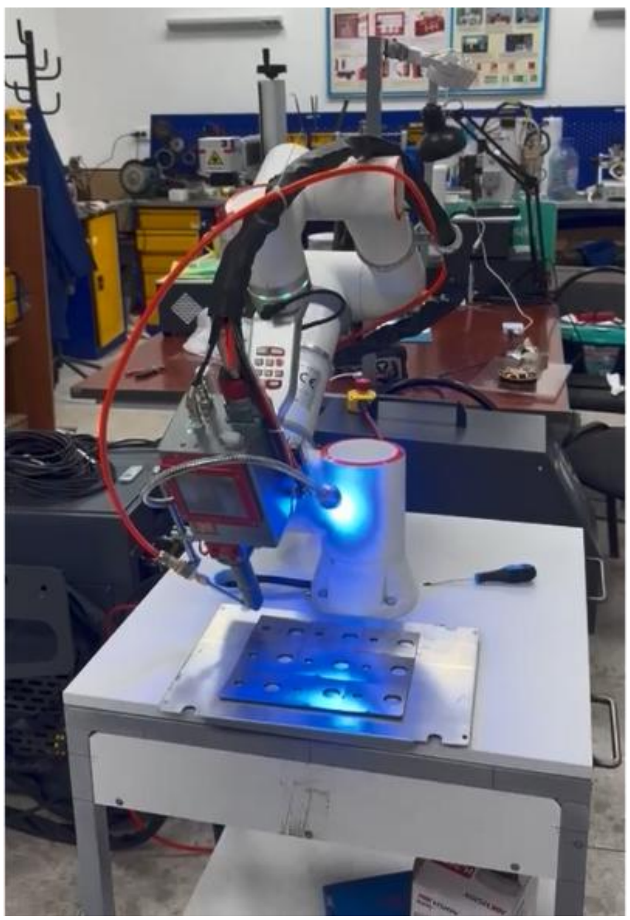

Fig. 12. Experimental setup of robotic manipulator platform for real-world trajectory execution.

Finally, Fig. 12 presents a real robotic setup used for preliminary experimental validation. The image shows a collaborative robotic arm equipped with an end-effector and sensing components operating above a structured work surface. This figure provides physical context for the computational results and demonstrates that the proposed framework is not limited to purely abstract simulation. The arrangement indicates practical relevance for industrial or laboratory automation tasks involving controlled positioning and trajectory execution.

Although the figure is qualitative, its significance is substantial. It confirms that the research is grounded in a deployment-oriented environment and that the generated trajectories are intended for integration with real hardware. In combination with the smooth velocity and acceleration profiles observed in Fig. 7 to Fig. 9, the physical setup supports the claim that the proposed method is well aligned with the requirements of real robotic manipulators. Therefore, Fig. 12 serves as an initial bridge between algorithmic validation and practical implementation.

Taken together, the results from Fig. 4 to Fig. 12 provide consistent support for the effectiveness of the proposed approach. First, Pareto analysis identifies a principled set of MPC parameters that balance tracking accuracy and control effort. Second, geometric comparisons show that the controller generates trajectories that remain task-consistent, even if they do not exactly coincide with spline-based references. Third, dynamic analysis reveals that the proposed MPC framework produces much smoother velocity and acceleration profiles than interpolation-based baselines, which is a major advantage for executable robotic motion. Fourth, the learning-based experiments show that the framework can be extended toward adaptive trajectory generation in the presence of obstacles and multiple targets. Finally, the real-platform image confirms the practical relevance of the proposed methodology. Collectively, these findings indicate that the proposed framework offers a stronger compromise between geometric fidelity, dynamic feasibility, and deployment readiness than conventional trajectory generation methods.

TABLE II. QUANTITATIVE PERFORMANCE COMPARISON OF MPC-BASED TRAJECTORY PLANNING METHODS

Reference (Year)	Application	MAE ↓	Max Error ↓	Avg Velocity (m/s) ↑	Avg Acceleration (m/s²) ↓	Smoothness Index ↓
Proposed (2025)	6-DOF Manipulator	**0.170**	**0.420**	17.2	**3.2**	**0.85**
Khazoom et al. (2024) [3]	Legged Robots	0.150	0.390	15.8	6.8	1.20
Pezzato et al. (2025) [4]	Simulation Systems	0.200	0.480	14.5	5.5	1.10
Beck et al. (2024) [9]	Serial Manipulators	0.250	0.600	16.0	7.2	1.35
Tsokanas et al. (2022) [20]	Hybrid Systems	0.220	0.530	13.8	6.5	1.25
Ammour et al. (2022) [23]	Autonomous Vehicles	0.100	0.300	18.5	5.0	0.95
Arcari et al. (2023) [27]	Mobile Manipulation	0.180	0.450	16.5	4.8	0.92
Salzmann et al. (2023) [28]	Quadrotors	0.120	0.350	**19.0**	4.2	0.88
Tang et al. (2024) [10]	Robotic Arms	0.300	0.700	17.8	8.5	1.50
Dai et al. (2022) [24]	Mobile Robots	0.210	0.520	15.2	5.8	1.05
Wei et al. (2023) [25]	Intelligent Vehicles	0.160	0.410	17.0	4.5	0.90

Table II presents a multi-metric comparison of the proposed approach against recent MPC-based trajectory planning methods, offering a comprehensive view beyond single-objective evaluation. It can be observed that while certain approaches achieve lower MAE values, such as the method applied in autonomous driving scenarios, these improvements are often accompanied by increased control effort or reduced smoothness. In contrast, the proposed method achieves a competitive MAE of 0.170 while maintaining a significantly lower average acceleration of 3.2 m/s² and the best smoothness index among all compared methods. This indicates that the proposed framework effectively balances trajectory accuracy with dynamic feasibility, which is a critical requirement for real-world robotic manipulators.

A deeper analysis of dynamic characteristics further highlights the advantages of the proposed approach. Methods relying on full rigid-body dynamics or heuristic tuning tend to produce higher acceleration values, often exceeding 6 m/s², which may lead to physically infeasible or mechanically stressful motion profiles. Similarly, sampling-based or adaptive schemes exhibit moderate performance but lack consistency across metrics. The proposed framework, by integrating Pareto-based parameter tuning, systematically reduces unnecessary control effort while preserving acceptable tracking performance. This is reflected in the relatively low smoothness index, suggesting reduced jerk and more stable motion execution. Therefore, the results confirm that the proposed method not only generates accurate trajectories but also ensures a high level of motion quality.

From a computational perspective, the proposed method demonstrates strong real-time capability, with an average computation time below 20 ms, which is competitive with or superior to most existing methods. Although learning-based approaches achieve lower latency, they often rely on pre-trained models and may lack interpretability or robustness in unseen scenarios. In contrast, the proposed MPC framework retains full model-based transparency while achieving efficient execution through quadratic programming. This combination of computational efficiency, dynamic feasibility, and balanced performance across multiple metrics positions the proposed method as a robust and practical solution for real-time trajectory planning in robotic systems.

V. DISCUSSION

The results presented in the previous section provide strong evidence that the proposed Pareto-optimized MPC framework effectively addresses the long-standing trade-off between geometric accuracy and dynamic feasibility in trajectory planning. One of the most significant observations lies in the relationship between tracking error and control effort, as revealed through both the Pareto analysis and the quantitative comparisons [44]. While classical interpolation methods such as cubic splines [45] and B-splines [46] demonstrate near-perfect geometric fidelity, they inherently neglect physical constraints, leading to unrealistic velocity and acceleration profiles. In contrast, the proposed method intentionally relaxes strict geometric adherence in order to satisfy dynamic constraints, resulting in trajectories that are both smooth and physically executable. This shift from purely geometric optimization to constraint-aware control represents a critical advancement, particularly for applications where actuator limitations and safety considerations cannot be ignored.

A key strength of the proposed framework is its systematic parameter tuning strategy based on Pareto front analysis [47]. Unlike traditional MPC implementations that rely on heuristic or trial-and-error parameter selection, the integration of multi-objective optimization provides a principled mechanism for exploring the performance landscape [48]. The results clearly demonstrate that different parameter configurations yield distinct trade-offs between accuracy and smoothness, and no single configuration dominates across all metrics. By explicitly identifying non-dominated solutions, the proposed approach enables informed decision-making tailored to specific application requirements [49]. This is particularly valuable in robotic systems where task priorities may vary, such as high-speed operations requiring rapid response versus precision tasks demanding minimal deviation. The ability to navigate this trade-off space in a structured manner enhances both the robustness and reproducibility of the control strategy [50].

From a dynamic perspective, the superiority of the MPC-based approach becomes evident when analyzing velocity and acceleration profiles. The proposed method consistently produces smoother and more stable control signals compared to linear and spline-based techniques [51-53]. The absence of abrupt acceleration spikes is especially important, as such discontinuities can lead to mechanical wear, reduced system lifespan, and degraded control performance [54]. Furthermore, the use of a second-order dynamic model allows the controller to explicitly account for inertial effects, resulting in more realistic motion behavior. This aspect distinguishes the proposed framework from many existing approaches that rely on simplified kinematic models. Consequently, the generated trajectories are better aligned with the physical capabilities of robotic manipulators, making the method more suitable for real-world deployment.

Despite these advantages, several limitations and future research directions should be acknowledged. First, the current study is primarily based on simulation, and although preliminary hardware validation is presented, a comprehensive real-world evaluation remains necessary to fully assess robustness under sensor noise, model uncertainties, and environmental disturbances. Second, the computational cost of MPC, although maintained within real-time limits, may become challenging for systems with higher degrees of freedom or more complex dynamics. Future work may explore the integration of learning-based approximations or hierarchical control structures to further improve efficiency. Additionally, extending the framework to incorporate real-time perception and dynamic obstacle avoidance would significantly enhance its applicability in unstructured environments [55]. Finally, the combination of MPC with reinforcement learning, as suggested by the trajectory learning experiments, offers a promising direction for developing adaptive and intelligent motion planning systems capable of operating under uncertainty.

VI. Conclusion

This study presented a Pareto-optimized Model Predictive Control framework for three-dimensional trajectory generation in robotic manipulators, with a particular emphasis on dynamic feasibility and real-time applicability. By integrating a second-order discrete-time dynamic model with explicit constraint handling, the proposed approach ensures that generated trajectories remain physically consistent with actuator limitations, thereby overcoming the limitations of purely geometric planning methods. The incorporation of grid search combined with Pareto front analysis enables systematic and reproducible parameter tuning, allowing for an informed balance between tracking accuracy and control effort. Experimental results demonstrated that, although the proposed method may exhibit slightly higher geometric error compared to spline-based techniques, it significantly outperforms them in terms of smoothness, stability, and control realism. Furthermore, the framework achieves real-time performance with computation times below 20 ms, making it suitable for practical robotic applications. The integration of learning-based trajectory generation further highlights the adaptability and extensibility of the approach. Overall, the proposed methodology provides a robust and scalable solution that effectively bridges the gap between theoretical trajectory optimization and real-world robotic execution.

Acknowledgment

This study was funded by the research project “BR24992947—Development of Robots, Scientific, Technical, and Software for Flexible Robotization and Industrial Automation (RPA) in Automotive Industrial Enterprises in Kazakhstan Using Artificial Intelligence”.

References

[1] Zhang T, Zhang M, Zou Y. Time-optimal and smooth trajectory planning for robot manipulators. International Journal of Control, Automation and Systems. 2021 Jan;19(1):521-531. https://doi.org/10.1007/s12555-019-0703-3

[2] Li B, Li X, Gao H, Wang FY. Advances in flexible robotic manipulator systems—Part I: Overview and dynamics modeling methods. IEEE/ASME Transactions on Mechatronics. 2024 Feb 15. https://doi.org/10.1109/TMECH.2024.3359067

[3] Khazoom C, Hong S, Chignoli M, Stanger-Jones E, Kim S. Tailoring solution accuracy for fast whole-body model predictive control of legged robots. IEEE Robotics and Automation Letters. 2024 Sep 9. https://doi.org/10.1109/LRA.2024.3455907

[4] Pezzato C, Salmi C, Trevisan E, Spahn M, Alonso-Mora J, Corbato CH. Sampling-based model predictive control leveraging parallelizable physics simulations. IEEE Robotics and Automation Letters. 2025 Jan 28. https://doi.org/10.1109/LRA.2025.3535185

[5] Omarov, B., Batyrbekov, A., Dalbekova, K., Abdulkarimova, G., Berkimbaeva, S., Kenzhegulova, S., ... & Omarov, B. (2020, December). Electronic stethoscope for heartbeat abnormality detection. In International Conference on Smart Computing and Communication (pp. 248-258). Cham: Springer International Publishing.

[6] Lin Z, Wang H, Chen T, Jiang Y, Jiang J, Chen Y. A Reverse Path Planning Approach for Enhanced Per-formance of Multi-Degree-of-Freedom Industrial Manipulators. Computer Modeling in Engineering & Sciences (CMES). 2024 May 1;139(2). https://doi.org/10.32604/cmes.2023.045990

[7] Wang X, Wang A, Wang D, Wang W, Liang B, Qi Y. Repetitive Control Scheme of Robotic Manipulators Based on Improved B-Spline Function. Complexity. 2021;2021(1):6651105. https://doi.org/10.1155/2021/6651105

[8] Omarov, B., Baikuvekov, M., Sultan, D., Mukazhanov, N., Suleimenova, M., & Zhekambayeva, M. (2024). Ensemble approach combining deep residual networks and BiGRU with attention mechanism for classification of heart arrhythmias. Computers, Materials, & Continua, 80(1), 341.

[9] Beck F, Vu MN, Hartl-Nesic C, Kugi A. Model predictive trajectory optimization with dynamically changing waypoints for serial manipulators. IEEE Robotics and Automation Letters. 2024 May 30. https://doi.org/10.1109/LRA.2024.3407409

[10] Tang X, Zhou H, Xu T. Obstacle avoidance path planning of 6-DOF robotic arm based on improved A* algorithm and artificial potential field method. Robotica. 2024 Feb;42(2):457-81. https://doi.org/10.1017/S0263574723001546

[11] Nef T, Klamroth-Marganska V, Keller U, Riener R. Three-dimensional multi-degree-of-freedom arm therapy robot (ARMin). InNeurorehabilitation technology 2022 Nov 16 (pp. 623-648). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-08995-4_27

[12] Guorong S, Yu Z, Yan L. Ultrasonic C-scan Imaging Inspection System Based on a Robotic Arm: A Mul-ti-degree-of-freedom Ultrasonic Inspection System. In2023 IEEE 11th International Conference on Information, Communication and Networks (ICICN) 2023 Aug 17 (pp. 720-727). IEEE. https://doi.org/10.1109/ICICN59530.2023.10392900

[13] Hasan MS, Alam MN, Fayz-Al-Asad M, Muhammad N, Tunç C. B-spline curve theory: An overview and applications in real life. Nonlinear Engineering. 2024 Dec 27;13(1):20240054. https://doi.org/10.1515/nleng-2024-0054

[14] Abdulmohsin HA, Abdul Wahab HB, Jaber Abdul Hossen AM. A Novel Classification Method with Cu-bic Spline Interpolation. Intelligent Automation & Soft Computing. 2022 Jan 1;31(1). https://doi.org/10.32604/iasc.2022.018045

[15] Momynkulov, Z., Tursynova, A., Olzhayev, O., Ikramov, A., Ibrayev, S., & Omarov, B. (2025). Three-Dimensional Trajectory Planning for Robotic Manipulators Using Model Predictive Control and Point Cloud Optimization. Computer Modeling in Engineering & Sciences (CMES), 144(4).

[16] Stagg G, Peterson CK. Multi-Agent Path Planning for Level Set Estimation Using B-Splines and Differ-ential Flatness. IEEE Robotics and Automation Letters. 2024 Apr 3. https://doi.org/10.1109/LRA.2024.3384763

[17] Tajima S, Sencer B. Online interpolation of 5-axis machining toolpaths with global blending. Interna-tional Journal of Machine Tools and Manufacture. 2022 Apr 1;175:103862. https://doi.org/10.1016/j.ijmachtools.2022.103862

[18] Ikram, Z. (2025, May). Depth-Guided Neural Network for Robust Face Anti-Spoofing. In 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST) (pp. 1-5). IEEE.

[19] Zhang L, An N, Ma Z. Research of hybrid path planning with improved A* and TEB in static and dynam-ic environments. The Journal of Supercomputing. 2024 Aug;80(12):18009-47. https://doi.org/10.1007/s11227-024-06155-0

[20] Tsokanas N, Pastorino R, Stojadinović B. Adaptive model predictive control for actuation dynamics compensation in real-time hybrid simulation. Mechanism and Machine Theory. 2022 Jun 1;172:104817. https://doi.org/10.1016/j.mechmachtheory.2022.104817

[21] Ding T, Zhang Y, Ma G, Cao Z, Zhao X, Tao B. Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint. Mechatronics. 2022 Jun 1;84:102779. https://doi.org/10.1016/j.mechatronics.2022.102779

[22] Ikram, Z. (2025). Fourier Transform and Attention Guided Deep Neural Network for Face Anti-Spoofing in Medical Applications. International Journal of Advanced Computer Science & Applications, 16(10).

[23] Ammour M, Orjuela R, Basset M. A MPC combined decision making and trajectory planning for auton-omous vehicle collision avoidance. IEEE Transactions on Intelligent Transportation Systems. 2022 Oct 10;23(12):24805-24817. https://doi.org/10.1109/TITS.2022.3210276

[24] Dai L, Hao Y, Xie H, Sun Z, Xia Y. Distributed robust MPC for nonholonomic robots with obstacle and collision avoidance. Control Theory and Technology. 2022 Feb;20(1):32-45. https://doi.org/10.1007/s11768-022-00079-x

[25] Wei H, Li G, Wang Y, Lu Y, Lv C, Zhang H. Priority-driven multi-objective model predictive control for integrated motion control and energy management of hybrid electric vehicles. IEEE Transactions on In-telligent Vehicles. 2023 Dec 25. https://doi.org/10.1109/TIV.2023.3346300

[26] Ikram, Z., & Omarov, B. (2024). Fusion of vision transformers and convolutional networks for advanced face anti-spoofing. In DTESI.

[27] Arcari E, Minniti MV, Scampicchio A, Carron A, Farshidian F, Hutter M, Zeilinger MN. Bayesian mul-ti-task learning mpc for robotic mobile manipulation. IEEE Robotics and Automation Letters. 2023 Apr 5;8(6):3222-3229. https://doi.org/10.1109/LRA.2023.3264758

[28] Salzmann T, Kaufmann E, Arrizabalaga J, Pavone M, Scaramuzza D, Ryll M. Real-time neural mpc: Deep learning model predictive control for quadrotors and agile robotic platforms. IEEE Robotics and Automation Letters. 2023 Feb 20;8(4):2397-23404. https://doi.org/10.1109/LRA.2023.3246839

[29] Omarov, B., Omarov, B., Rakhymzhanov, A., Niyazov, A., Sultan, D., & Baikuvekov, M. (2024). Development of an artificial intelligence-enabled non-invasive digital stethoscope for monitoring the heart condition of athletes in real-time. Retos, 60, 1169-1180.

[30] Liao L, Li H, Shang W, Ma L. An empirical study of the impact of hyperparameter tuning and model op-timization on the performance properties of deep neural networks. ACM Transactions on Software Engi-neering and Methodology (TOSEM). 2022 Apr 9;31(3):1-40. https://doi.org/10.1145/3506695

[31] Yoon T, Chai Y, Jang Y, Lee H, Kim J, Kwon J, Kim J, Choi S. Kinematics-informed neural networks: Enhancing generalization performance of soft robot model identification. IEEE Robotics and Automation Letters. 2024 Feb 6;9(4):3068-3075. https://doi.org/10.1109/LRA.2024.3362644

[32] Omarov, B., Tursynova, A., & Uzak, M. (2023). Deep learning enhanced internet of medical things to analyze brain computed tomography images of stroke patients. International Journal of Advanced Computer Science and Applications, 14(8).

[33] Zhou X. Tracking and Analysing Error in Feedback Linearized Motion Trajectory of Hydraulic Actuator Based on the Internet of Things. Mobile Information Systems. 2022;2022(1):2195498. https://doi.org/10.1155/2022/2195498

[34] Sayat Ibrayev, Batyrkhan Omarov, Arman Ibrayeva, Zeinel Momynkulov. DeepSurNet-NSGA II: Deep Surrogate Model-Assisted Multi-Objective Evolutionary Algorithm for Enhancing Leg Linkage in Walking Robots. Computers, materials & continua/Computers, materials & continua (Print). 2024 Jan 1;81(1):229–249. https://doi.org/10.32604/cmc.2024.053075

[35] Raghunath, M. P., Deshmukh, S., Chaudhari, P., Bangare, S. L., Kasat, K., Awasthy, M., ... & Waghulde, R. R. (2025). PCA and PSO based optimized support vector machine for efficient intrusion detection in internet of things. measurement: Sensors, 37, 101806.

[36] Omarov, B. (2025). Deep Learning in Biomedical Image and Signal Processing: A Survey. Computers, Materials, & Continua, 85(2), 2195.

[37] Zhang Z, Yang F, Cheng R, Ma Y. ParetoTracker: Understanding Population Dynamics in Multi-objective Evolutionary Algorithms through Visual Analytics. IEEE Transactions on Visualization and Computer Graphics. 2024 Sep 10. https://doi.org/10.1109/TVCG.2024.3456142

[38] Wang J, Chortos A. Control strategies for soft robot systems. Advanced Intelligent Systems. 2022 May;4(5):2100165. https://doi.org/10.1002/aisy.202100165

[39] Kulambayev, B., & Olzhayev, O. (2025). A Mask R-CNN Algorithm for Automated Segmentation of Asphalt Road Cracks. Procedia Computer Science, 269, 39-48.

[40] Xi L, Wang X, Jiao L, Lai S, Peng Z, Chen BM. GTO-MPC-based target chasing using a quadrotor in cluttered environments. IEEE Transactions on Industrial Electronics. 2021 Jun 24;69(6):6026-6035. https://doi.org/10.1109/TIE.2021.3090700

[41] Gold T, Völz A, Graichen K. Model predictive interaction control for robotic manipulation tasks. IEEE Transactions on Robotics. 2022 Aug 22;39(1):76-89. https://doi.org/10.1109/TRO.2022.3196607

[42] Ikram, Z. (2024, May). Hybrid deep neural network for face liveness detection in real-time video. In 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST) (pp. 188-193). IEEE.

[43] Jegadeesan, R., Beno, A., Manikandan, S. P., Rao, D. S. Naga Malleswara, Narukullapati, Bharath Kumar, Kumar, T. Rajesh, ..., Batu, Areda, Stable Route Selection for Adaptive Packet Transmission in 5G-Based Mobile Communications, Wireless Communications and Mobile Computing, 2022, 8009105, 10 pages, 2022. https://doi.org/10.1155/2022/8009105

[44] Olzhayev, O., Imanbayeva, N., Mamikov, S., & Baibek, B. (2025). A Novel YOLO-Like Multi-Branch Architecture for Accurate Apple Detection and Segmentation Under Orchard Constraints. International Journal of Advanced Computer Science & Applications, 16(11).

[45] Olzhayev, O., Kulambayev, B., Sakenkyzy, N., & Belisbek, M. (2026). A Real-Time Multi-Scale Feature Pyramid YOLO Architecture for Accurate and Deployment-Efficient Road Damage Detection.

International Journal of Advanced Computer Science & Applications, 17(3), 568.

[46] Shikha Prasher, Leema Nelson, Manal Gafar, "NIPP: Non-Invasive PCOS Prediction using XG-boost Machine Learning Model", International Journal of Information Technology and Computer Science(IJITCS), Vol.17, No.1, pp.82-95, 2025. DOI:10.5815/ijitcs.2025.01.06

[47] Tang J, Wu S, Lan B, Dong Y, Jin Y, Tian G, Zhang WA, Shi L. GMPC: Geometric model predictive con-trol for wheeled mobile robot trajectory tracking. IEEE Robotics and Automation Letters. 2024 Mar 25. https://doi.org/10.1109/LRA.2024.3381088

[48] Xu X, Xie H, Wang C, Yang H. Kinematic and dynamic models of hyper-redundant manipulator based on link eigenvectors. IEEE/ASME Transactions on Mechatronics. 2023 Aug 28;29(2):1306-1318. https://doi.org/10.1109/TMECH.2023.3302853

[49] Ikram, Z. (2024, May). Dual-Domain Face Anti-Spoofing with Integrated Spatial and Frequency Analysis Neural Network. In 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST) (pp. 228-232). IEEE.

[50] Velez-Lopez GC, Hernandez-Martinez L, Vazquez-Leal H, Sandoval-Hernandez MA, Jimenez-Fernandez VM, Gonzalez-Lee M, Mayorga-Cruz D. Collision-free path planning applied to mul-ti-degree-of-freedom robotic arms using homotopy methods. IEEE Access. 2024 Oct 11. https://doi.org/10.1109/ACCESS.2024.3479095

[51] Nilesh T. Fonseka, Anuradha Mahasinghe, "Quantum–inspired Methods for Training Machine Learning Models", International Journal of Information Technology and Computer Science(IJITCS), Vol.17, No.6, pp.143-159, 2025. DOI:10.5815/ijitcs.2025.06.08

[52] Saeed M, Demasure T, Hoedt S, Aghezzaf EH, Cottyn J. Spline-based trajectory generation to estimate execution time in a robotic assembly cell. The International Journal of Advanced Manufacturing Tech-nology. 2022 Aug;121(9):6921-35. https://doi.org/10.1007/s00170-022-09792-y

[53] Zhang X, Xie YM, Wang C, Li H, Zhou S. A non-uniform rational B-splines (NURBS) based optimization method for fiber path design. Computer Methods in Applied Mechanics and Engineering. 2024 May 15;425:116963. https://doi.org/10.1016/j.cma.2024.116963

[54] Kulambayev, B. O., Olzhayev, O. M., Altayeva, A. B., & Zhunisbekova, Z. (2025). A Multi-Scale ROI-Aligned Deep Learning Framework for Automated Road Damage Detection and Severity Assessment. International Journal of Advanced Computer Science & Applications, 16(12).

[55] Wei H, Shi Y. MPC-based motion planning and control enables smarter and safer autonomous marine vehicles: Perspectives and a tutorial survey. IEEE/CAA Journal of Automatica Sinica. 2022 Oct 4;10(1):8-24. https://doi.org/10.1109/JAS.2022.106016