

# On the Performance of Jerk-Constrained Time-Optimal Trajectory Planning for Industrial Manipulators

Jee-eun Lee<sup>1</sup>, Andrew Bylard<sup>2</sup>, Robert Sun<sup>2</sup>, and Luis Sentis<sup>1</sup>

**Abstract**— Jerk-constrained trajectories offer a wide range of advantages that collectively improve the performance of robotic systems, including increased energy efficiency, durability, and safety. In this paper, we present a novel approach to jerk-constrained time-optimal trajectory planning (TOTP), which follows a specified path while satisfying up to third-order constraints to ensure safety and smooth motion. One significant challenge in jerk-constrained TOTP is a non-convex formulation arising from the inclusion of third-order constraints. Approximating inequality constraints can be particularly challenging because the resulting solutions may violate the actual constraints. We address this problem by leveraging convexity within the proposed formulation to form conservative inequality constraints. We then obtain the desired trajectories by solving an  $n$ -dimensional Sequential Linear Program (SLP) iteratively until convergence. Lastly, we evaluate in a real robot the performance of trajectories generated with and without jerk limits in terms of peak power, torque efficiency, and tracking capability.

**Index Terms**— Time-Optimal Trajectory Planning, Jerk Constraints, Smooth Trajectory Generation, Industrial Robots

## I. INTRODUCTION

In automation, time-optimal motion planning plays a crucial role in maximizing productivity while ensuring efficiency and safety. However, solving the complete trajectory-planning problem is often difficult to perform in real time [1], [2], [3], [4], [5]. To obtain satisfactory results using reasonable computational resources, trajectory planning is often solved in two stages [6], [7], [8]. As shown in Fig. 1, during the initial stage, the path generator creates a geometric path that incorporates task specifications and obstacle avoidance. In the subsequent stage, a time-optimal trajectory is calculated along the specified geometric path, enforcing constraints such as permissible joint velocity and torque.

This paper focuses on solving the second-stage motion planning sub-problem, also known as time-optimal trajectory planning (TOTP) or path tracking (TOPT) [9], [10]. Although TOTP under first- and second-order constraints (such as velocity, acceleration, and torque limits) has been thoroughly explored, third-order constraints such as jerk or torque rate bounds are often overlooked. This is primarily because their inclusion lacks clear guidance (for instance, these bounds may not be specified in motor usage requirements), and they introduce non-convex terms into the formulation, making problem-solving much more challenging. However, without

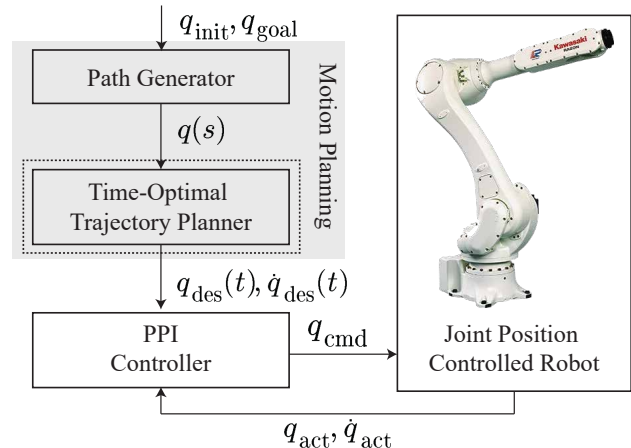


Fig. 1. Two-Stage Motion Planning and Control Architecture: The path generator creates a collision-free path, which is fed into the time-optimal trajectory planner to compute the desired trajectory satisfying robot hardware limits. Subsequently, the predictive proportional-integral (PPI) controller generates high-frequency joint commands with feedback.

considering jerk limits, solution trajectories may include large acceleration jumps, resulting in various issues when applied to real robots [11].

In this paper, we propose a new approach for TOTP with practical third-order constraints. By incorporating jerk limits into trajectory planning and control strategies, robotic systems can achieve higher reliability and effectiveness. Some well-known advantages include: 1) Enhanced path tracking: a smoother trajectory allows the robot’s movements to more closely match the desired trajectory, minimizing jerk-induced vibrations. 2) Reduced wear and tear: Sudden changes in acceleration can subject mechanical components to higher stress levels. By constraining jerk, the robot’s motor gears undergo gentler transitions, extending component lifespans. 3) Energy efficiency: Jerk-limited motions prevent unnecessary spikes in power consumption, leading to decreased energy usage and reduced operational costs.

Finally, we perform real robot experiments to confirm the benefits of including jerk limits. We analyze factors such as peak power and torque, which affect the performance, durability, and maintainability of the robot. The proposed motion planning and control framework is depicted in Fig. 1.

### A. Related Works

Traditionally, there are two widely adopted approaches for addressing time-optimal trajectory planning (TOTP): numerical integration (NI), and convex optimization (CO). NI-based approaches [9], [10], [12] solve the problem based on the insight that one can obtain a time-optimal trajectory

\*This work was supported by Dexterity, Inc.

<sup>1</sup> J. Lee and L. Sentis are with The University of Texas at Austin, Austin, TX, USA, {jelee, lsentis}@utexas.edu

<sup>2</sup> A. Bylard, and R. Sun are with Dexterity, Inc., Redwood City, CA, USA, {andrew.bylard, robert}@dexterity.ai

through bang-bang control. However, this method requires identifying switch points where acceleration changes, and it is prone to failure in certain cases. Conversely, optimization provides a general and robust means to formulate and solve the problem [13], [14]. However, it increases the problem's complexity by at least an order of magnitude, resulting in longer computation times. Other works solve the problem using dynamic programming [15], [16] or by leveraging specific conditions which may be too constraining, such as imposing constant first- or second-order limits in the path parameter space [17], [18], which may not map to desired (e.g., constant) limits in the joint space.

Recently, there have been efforts to simplify TOTP algorithms through methods based in reachability analysis (RA) [19], [20]. These methods, instead of solving a high-dimensional optimization problem (with  $n$  optimization variables, one for each timestep), can achieve the optimal solution by solving a series of  $2n$  2-D sub-problems. However, it's worth noting that these reachability-based algorithms are restricted to second-order constraints, since they cannot handle situations where second-order constraints violate the third-order constraints, known as third-order singularity [21].

Many works have attempted to incorporate third-order constraints into time-optimal trajectory planning (TOTP). Many of them rely on the NI-based approaches [21], [22], [23], [24]. In [22], however, this formulation is limited to bounding the third-order derivative of the path parameter ( $\ddot{s}$ ) rather than directly limiting the robot jerk or torque rate, and the third-order singularity issue is not addressed. Instead, one can carefully bridge the maximum and minimum jerk profile to avoid failure from singularity [21]. However, this approach requires handling numerous exceptional cases, making the algorithm overly complex when adding first and second-order constraints, which are often more critical [24].

Our approach provides a more general formulation within a convex optimization framework. A key challenge is enforcing nonlinear third-order inequality constraints, as approximating them without careful consideration can lead to solutions that violate real constraints. An alternative is to minimize both jerk and time [25], but this requires careful tuning of the relative weight for jerk and time in the cost function. Since the solutions of the optimization problem are often very sensitive to the weight, the solution can frequently sacrifice motion time or smoothness excessively depending on the input path given, even under the same weight. In this paper, we formulate the problem as an SLP using conservative linearization of the nonlinear constraints at each iteration to ensure solution feasibility. The work in [26] also used conservative approximation for third-order constraints, based on the fact that torque obtained without third-order constraints is always bigger than the one with third-order constraints. Thus, their approach is limited to a one-time approximation, which can lead to suboptimal solutions, with stricter limits often widening the optimality gap.

## B. Contribution

- We formulate a unique TOTP with third-order constraints as a Sequential Linear Program (SLP), exceeding the performance of existing methods.
- We introduce a method for approximating general third-order constraints, enabling iterative optimization to converge to the optimal solution.
- We demonstrate that imposing jerk limits significantly reduces peak power, enhances energy efficiency, and improves tracking performance.

## C. Organization

The paper is organized as follows. Section II revisits the concepts of TOTP and the corresponding parameterization methods. Section III formulates the jerk constraints for TOTP and outlines the overall optimization process with the approximated jerk constraints. In Section IV, we discuss the numerical and experimental results of the proposed jerk-bounded time-optimal trajectory planning. Finally, in Section V we discuss conclusions and future work.

## II. PRELIMINARIES

### A. Problem Formulation

We formulate time-optimal trajectory planning (TOTP) as follows: Given a path  $\mathbf{q}(s)$  as a function of a scalar path coordinate  $s \in [0, 1]$ , find a monotonically increasing time scaling  $s(t) : [0, T] \rightarrow [0, 1]$  that

- satisfies an initial state  $(s_0, \dot{s}_0) = (0, 0)$  and a final state  $(s_{\text{end}}, \dot{s}_{\text{end}}) = (1, 0)$ ,
- minimizes the total travel time  $T$  along the path,
- enforces kinematics and dynamics constraints imposed by robot hardware limitations.

### B. Time-Optimal Trajectory Planning (TOTP)

For the purpose of solving the TOTP problem to track the prescribed path  $\mathbf{q}(s)$  for a robot manipulator, we express the first- and second-order time derivatives of  $\mathbf{q}(s)$  as

$$\dot{\mathbf{q}}(s) = \mathbf{q}'(s)\dot{s}, \quad \ddot{\mathbf{q}}(s) = \mathbf{q}''(s)\dot{s}^2 + \mathbf{q}'(s)\ddot{s}. \quad (1)$$

We also transform the dynamics of the manipulator into

$$\begin{aligned} \boldsymbol{\tau} &= M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) \\ &= M(\mathbf{q})(\mathbf{q}''\dot{s}^2 + \mathbf{q}'\ddot{s}) + C(\mathbf{q}, \mathbf{q}')\mathbf{q}'\dot{s}^2 + g(\mathbf{q}) \\ &:= \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s). \end{aligned} \quad (2)$$

Note that from the linear nature of the Coriolis matrix, we have  $C(\mathbf{q}, \mathbf{q}')\mathbf{q}'\dot{s} = C(\mathbf{q}, \mathbf{q}')\mathbf{q}'\dot{s}^2$  above.

Furthermore, it is known that a second-order time differential equation can be transformed into a first-order differential equation in  $(\dot{s}^2, s)$  based on the relation introduced in [10],

$$\ddot{s} = \frac{d\dot{s}}{dt} = \frac{d\dot{s}}{ds} \frac{ds}{dt} = \dot{s}'\dot{s} = \frac{1}{2}(\dot{s}^2)'. \quad (3)$$

This is beneficial for solving TOTP problems, as it allows writing all the equations in linear form by setting  $(\dot{s}^2, s)$  as optimization variables [19], [13].

### C. TOTP Constraints in the Discretized System

We now show that we can express all constraints up to second-order as linear in  $x = s^2$  for the discretized system. By dividing the path interval  $[0, 1]$  into  $N$  segments, we have

$$0 =: s_0, s_1, \dots, s_{N-1}, s_N := 1.$$

Then from the relation in (3),  $\dot{s}_k, \ddot{s}_k$  can be represented as:

$$\dot{s}_k^2 := x_k, \quad \ddot{s}_k = \frac{x_{k+1} - x_k}{2\Delta_k} \quad \text{or} \quad \frac{x_k - x_{k-1}}{2\Delta_{k-1}}, \quad (4)$$

where  $\Delta_k := s_{k+1} - s_k$  and  $\Delta_{k-1} := s_k - s_{k-1}$ .

For representational convenience, we simplify the notation of each quantity at  $s_k$  as  $(\cdot)(s_k) = (\cdot)_k$ . Instead of introducing another optimization variable  $u = \ddot{s}$  like other works [19], we express all quantities as linear in  $x$ , which is sufficient. Finally, we formulate the following constraints:

- **Velocity limits (1st-order constraints)** First, the joint velocity limit can be represented as

$$\dot{\mathbf{q}}_k \circ \dot{\mathbf{q}}_k = (\mathbf{q}'_k \circ \mathbf{q}'_k)x_k \leq \dot{\mathbf{q}}_k^{max} \circ \dot{\mathbf{q}}_k^{max}, \quad (5)$$

where  $\mathbf{a} \circ \mathbf{b}$  represents the Hadamard product (i.e., the element-wise product) of two vectors.

- **Acceleration limits (2nd-order constraints)** Similarly, from the parameterized expression of acceleration

$$\ddot{\mathbf{q}}_k = \mathbf{q}''_k \dot{s}_k^2 + \mathbf{q}'_k \ddot{s}_k = \mathbf{q}''_k x_k + \mathbf{q}'_k \left( \frac{x_{k+1} - x_k}{2\Delta_k} \right),$$

we can constrain two consecutive  $x_k$  by

$$\left| \frac{\mathbf{q}'_k}{2\Delta_k} x_{k+1} + \left( \mathbf{q}''_k - \frac{\mathbf{q}'_k}{2\Delta_k} \right) x_k \right| \leq \ddot{\mathbf{q}}_k^{max}$$

or

$$\left| \left( \mathbf{q}''_k + \frac{\mathbf{q}'_k}{2\Delta_{k-1}} \right) x_k - \frac{\mathbf{q}'_k}{2\Delta_{k-1}} x_{k-1} \right| \leq \ddot{\mathbf{q}}_k^{max}. \quad (6)$$

- **Torque limits (2nd-order constraints)** The parameterized dynamics can also be rewritten with respect to  $x$  by substituting (4) into (2):

$$\left| \frac{\mathbf{m}_k}{2\Delta_k} x_{k+1} + \left( \mathbf{c}_k - \frac{\mathbf{m}_k}{2\Delta_k} \right) x_k + \mathbf{g}_k \right| \leq \boldsymbol{\tau}_k^{max}$$

or

$$\left| \left( \mathbf{c}_k + \frac{\mathbf{m}_k}{2\Delta_{k-1}} \right) x_k - \frac{\mathbf{m}_k}{2\Delta_{k-1}} x_{k-1} + \mathbf{g}_k \right| \leq \boldsymbol{\tau}_k^{max}. \quad (7)$$

By stacking up all the constraints derived above, we can simply express first-order constraints as  $\boldsymbol{\omega}_k x_k \leq \boldsymbol{\nu}_k$ , where  $\boldsymbol{\omega}_k = \mathbf{q}'_k \circ \mathbf{q}'_k$  and  $\boldsymbol{\nu}_k = \dot{\mathbf{q}}_k^{max} \circ \dot{\mathbf{q}}_k^{max}$ . Similarly, second-order constraints can be represented in the form of  $\boldsymbol{\alpha}_k^0 x_k + \boldsymbol{\alpha}_k^1 x_{k+1} \leq \boldsymbol{\beta}_k$ .

## III. JERK-CONSTRAINED TIME-OPTIMAL TRAJECTORY PLANNING

### A. Jerk Constraint Formulation

Suppose we want to enforce joint-level jerk bounds

$$-\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_{max}. \quad (8)$$

In this paper, we define jerk as change in acceleration between adjacent trajectory points. While this formula is

more forgiving than an instantaneous rate, it is still effective in preventing abrupt changes in acceleration and simplifies the formulation. Additional details can be found in Section V. Then joint jerk given the discretized system can be formulated as:

$$\begin{aligned} \ddot{\mathbf{q}}_k &= \frac{\Delta \ddot{\mathbf{q}}}{\Delta t}(s_k) = \frac{(\ddot{\mathbf{q}}_{k+1} - \ddot{\mathbf{q}}_k)}{\frac{1}{2}(\Delta t_{k+1} + \Delta t_k)} \\ &= \frac{\mathbf{j}_k^2 x_{k+2} + \mathbf{j}_k^1 x_{k+1} + \mathbf{j}_k^0 x_k}{h_k(\mathbf{x}_{k:k+2})}, \end{aligned} \quad (9)$$

where

$$\begin{aligned} \mathbf{x}_{k:k+2} &= [x_k \ x_{k+1} \ x_{k+2}]^\top \\ h_k(\mathbf{x}_{k:k+2}) &= \frac{\Delta_{k+1}}{\sqrt{x_{k+2}} + \sqrt{x_{k+1}}} + \frac{\Delta_k}{\sqrt{x_{k+1}} + \sqrt{x_k}} \\ \mathbf{j}_k^2 &= \frac{\mathbf{q}'_{k+1}}{2\Delta_{k+1}}, \quad \mathbf{j}_k^1 = \mathbf{q}''_{k+1} - \frac{\mathbf{q}'_{k+1}}{2\Delta_{k+1}} - \frac{\mathbf{q}'_k}{2\Delta_k}, \quad \mathbf{j}_k^0 = -\mathbf{q}''_k + \frac{\mathbf{q}'_k}{2\Delta_k}. \end{aligned}$$

Unfortunately, unlike second- or lower-order constraints, a nonlinear term such as  $h_k(\mathbf{x}_{k:k+2})$  in Eqn. (9) always remains when expressing jerk in the TOTP problem. We address this issue by linearizing the nonlinear term  $h_k(\mathbf{x}_{k:k+2})$  and leveraging its convexity to ensure the solution always satisfies the original constraints, i.e.,

$$\begin{aligned} h_k(\mathbf{x}_{k:k+2}) &\geq h_k(\bar{\mathbf{x}}_{k:k+2}) + \nabla h_k(\bar{\mathbf{x}}_{k:k+2})(\mathbf{x}_{k:k+2} - \bar{\mathbf{x}}_{k:k+2}) \\ &:= \bar{h}_k + \bar{h}_k^2 x_{k+2} + \bar{h}_k^1 x_{k+1} + \bar{h}_k^0 x_k, \end{aligned}$$

where

$$\begin{aligned} \bar{h}_k^i &= \left. \frac{\partial h_k(\mathbf{x}_{k:k+2})}{\partial x_{k+i}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}, \quad i \in \{0, 1, 2\}, \\ \bar{h}_k &= h_k(\bar{\mathbf{x}}_{k:k+2}) - \bar{h}_k^2 \bar{x}_{k+2} - \bar{h}_k^1 \bar{x}_{k+1} - \bar{h}_k^0 \bar{x}_k. \end{aligned}$$

We show the convexity of this function in the Appendix. From this, we can reformulate the joint jerk constraints as

$$\begin{aligned} &\left| \mathbf{j}_k^2 x_{k+2} + \mathbf{j}_k^1 x_{k+1} + \mathbf{j}_k^0 x_k \right| \\ &\leq \ddot{\mathbf{q}}_{max} (\bar{h}_k + \bar{h}_k^2 x_{k+2} + \bar{h}_k^1 x_{k+1} + \bar{h}_k^0 x_k) \\ &\leq \ddot{\mathbf{q}}_{max} h_k(\mathbf{x}_{k:k+2}), \end{aligned}$$

which finally leads to the linear form of 3rd-order constraints:

$$\boldsymbol{\gamma}_k^0 x_k + \boldsymbol{\gamma}_k^1 x_{k+1} + \boldsymbol{\gamma}_k^2 x_{k+2} \leq \boldsymbol{\eta}_k.$$

We can now combine all first- to third-order constraints as a linear matrix inequality:

$$\begin{cases} \text{1st order constraints:} & \boldsymbol{\omega}_k x_k \leq \boldsymbol{\nu}_k \\ \text{2nd order constraints:} & \boldsymbol{\alpha}_k^0 x_k + \boldsymbol{\alpha}_k^1 x_{k+1} \leq \boldsymbol{\beta}_k \\ \text{3rd order constraints:} & \boldsymbol{\gamma}_k^0 x_k + \boldsymbol{\gamma}_k^1 x_{k+1} + \boldsymbol{\gamma}_k^2 x_{k+2} \leq \boldsymbol{\eta}_k \end{cases} \quad (10)$$

### B. Trajectory Optimization

The cost function for minimizing the total time required to follow the given path can be expressed as

$$f(\mathbf{x}) = \sum_{i=0}^{N-1} \frac{s_{i+1} - s_i}{\sqrt{x_i} + \sqrt{x_{i+1}}}, \quad (11)$$

which can be linearized along the nominal trajectory  $\bar{\mathbf{x}}$ .

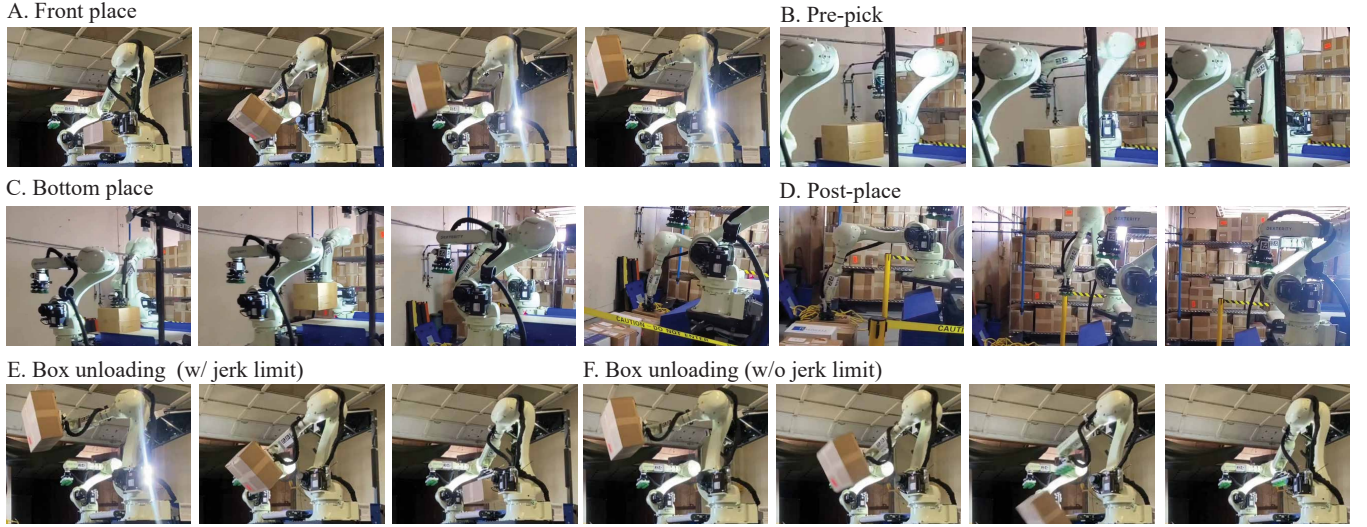


Fig. 2. Test motion snapshots. A: Front place motion to move boxes from a conveyor to the top of a virtual stack of packages. B: Pre-pick motion preparing to pick a box from a conveyor. C: Bottom place motion to place boxes onto the bottom of the truck. D: Post-place motion to return to the ready pose after placing boxes. E and F show box unloading motions, which return the box from a stack of packages to the conveyor with and without jerk constraints. Without jerk constraints, the robot's suction gripper is also more likely to lose grasp of the box.

Now we formulate the trajectory optimization problem as a Sequential Linear Program (SLP), where the optimization variables are  $\mathbf{x} = [x_1, \dots, x_{N-1}]^\top$  as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq b, \end{aligned} \quad (12)$$

where

$$\begin{aligned} \mathbf{c} &= \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}, \quad A = \begin{bmatrix} A^1 \\ A^2 \\ A^3 \end{bmatrix}, \quad b = \begin{bmatrix} b^1 \\ b^2 \\ b^3 \end{bmatrix} \\ A^1 &= \begin{bmatrix} \omega_1 & 0 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & 0 & \omega_{N-1} \end{bmatrix}, \quad b^1 = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_{N-1} \end{bmatrix}, \\ A^2 &= \begin{bmatrix} \alpha_0^1 & 0 & \dots & 0 \\ \alpha_1^1 & \alpha_1^1 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \alpha_{N-2}^0 & \alpha_{N-2}^1 \\ 0 & \dots & \dots & \alpha_{N-1}^0 \end{bmatrix}, \quad b^2 = \begin{bmatrix} \beta_0 - \alpha_0^0 x_0 \\ \beta_1 \\ \vdots \\ \beta_{N-2} \\ \beta_{N-1} - \alpha_{N-1}^1 x_N \end{bmatrix}, \\ A^3 &= \begin{bmatrix} \gamma_0^2 & 0 & 0 & 0 & \dots & 0 \\ \gamma_1^2 & \gamma_1^2 & 0 & 0 & \dots & 0 \\ \gamma_2^2 & \gamma_2^2 & \gamma_2^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \gamma_{N-3}^0 & \gamma_{N-3}^1 & \gamma_{N-3}^2 \\ 0 & \dots & 0 & \gamma_{N-2}^0 & \gamma_{N-2}^1 & \gamma_{N-2}^2 \end{bmatrix}, \quad (13) \end{aligned}$$

$$b^3 = \begin{bmatrix} \eta_0 - \gamma_0^0 x_0 - \gamma_0^1 x_1 \\ \eta_1 - \gamma_1^0 x_1 \\ \eta_2 \\ \vdots \\ \eta_{N-3} \\ \eta_{N-2} - \gamma_{N-2}^2 x_N \end{bmatrix}. \quad (14)$$

Note that due to the linearization, we need to iteratively update the nominal trajectory to reformulate the coefficients for cost function and 3rd-order constraints. The steps of the resulting algorithm are shown in Algorithm 1.

---

#### Algorithm 1 TOTP with 3rd-order constraints (TOTP3)

---

**Require:** Nominal variables  $\bar{\mathbf{x}}_{1:N}$

**while** True **do**

    Update  $\mathbf{c}, A^3, b^3$  ▷ Eqn. (13),(14)

$\mathbf{x}_{1:N} = \text{solve LP}$  ▷ Eqn. (12)

**if**  $\|\mathbf{x}_{1:N} - \bar{\mathbf{x}}_{1:N}\| < \epsilon$  **then break**

**end if**

    Update  $\bar{\mathbf{x}}_{1:N}$

**end while**

---

## IV. EXPERIMENT RESULTS

In this section, we provide experimental results to validate the effectiveness of the proposed approach.

### A. Experiment Setup

For practical evaluation, we used an example motion from a real-world box-loading application. We performed the test motions depicted and described in Fig. 2 with and without the jerk limit. We used the output from TOPP-RA [19] as the initial nominal trajectory for our algorithm and as a comparison target for the optimal trajectory without the jerk limit. Finally, the jerk limits were heuristically chosen to be  $100 \sim 1000 \text{ rad/s}^3$ , balancing the trade-off between motion

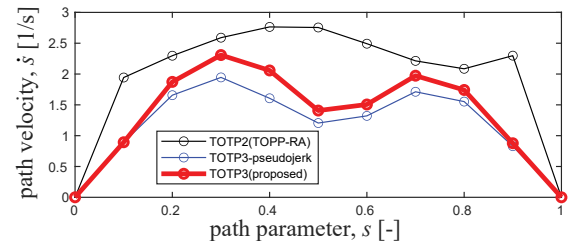
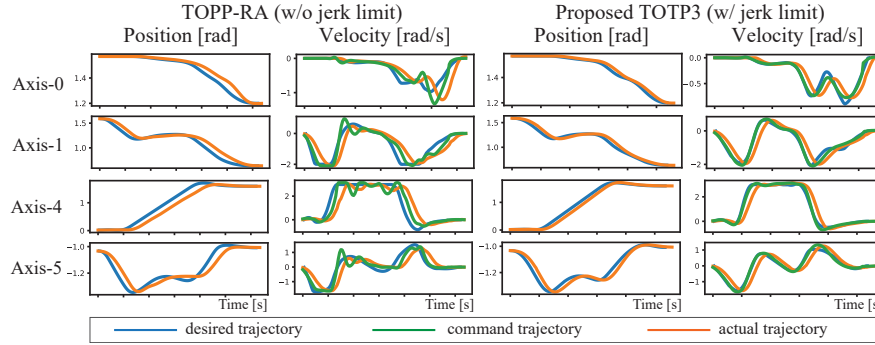


Fig. 3. Comparison of the velocity curve for the pre-pick motion (Fig. 2B) computed from TOPP-RA [19] without jerk limits (black), [26] with pseudo jerk limits (blue) and the proposed approach with jerk limits (red).

## A. Tracking Performance



## B. Energy Performance

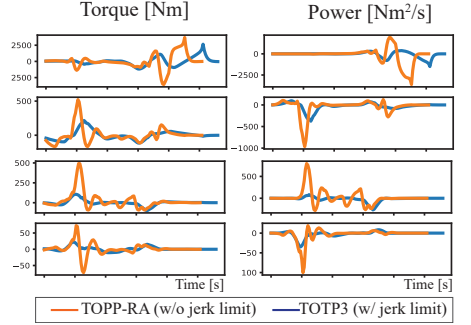


Fig. 4. Algorithm performance comparison on real-robot for the front place motion (Fig. 2A): **A** shows tracking performance of the algorithms with and without jerk limits by comparing the desired, commanded, and actual joint position and velocity. **B** shows the corresponding estimated torque and power.

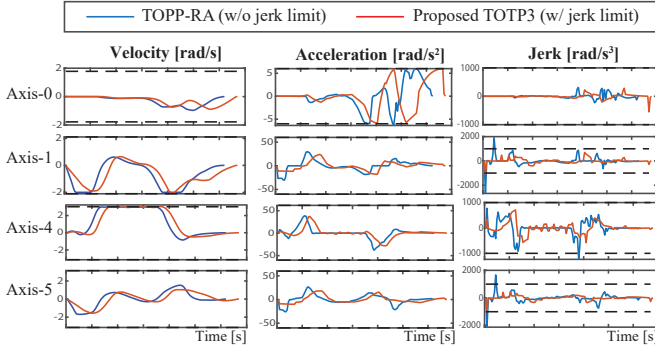


Fig. 5. Time-optimal trajectory generated for the front place motion (Fig. 2A) without jerk limits (blue) and with jerk limits (red).

duration and overall performance metrics such as energy efficiency and tracking (note that some applications may require more specific strict limitations on jerk). Lastly, we used 7-DOF robotic system composed of a 6-DOF RS020N Kawasaki arm mounted on an additional revolute joint for our experiment.

## B. Comparison With TOPP-RA

1) *Simulation Results*: We compare the TOTP outcomes obtained by three different methods in the phase plane. These methods include TOPP-RA [19] without jerk limits, TOTP3 [26] with pseudo jerk limits, and our proposed approach with jerk limits. While the original work in [26] focused on torque rate constraints, we adapted it for our experiment by introducing jerk limits. We employed a similar methodology to the one used for pseudo-torque rate to establish the pseudo-jerk limit. Fig. 3 shows that the curve tends to have a smoother shape when jerk limits are enforced. Note that since we formulate jerk limits in joint space and then map them to the space of the path parameter  $s$ , the smoothness of the joint-space curves along the path may vary at different path points. The result shows that the proposed approach yields faster motion than the pseudo-torque rate method under the same limits. To highlight the optimality gap, we applied a very stringent jerk limit of  $100 \text{ rad/s}^3$ .

Fig. 5 illustrates the computed velocity, acceleration, and jerk with the corresponding limits for the significant axes of the robot. It demonstrates that by enforcing a jerk limit, rapid

TABLE I

RMS TORQUE AND PEAK POWER DURING FRONT PLACE MOTION

		Axis						
		0	1	2	3	4	5	6
RMS Trq [Nm]	TOPP-RA	3624	965	32	73	795	100	28
	TOTP3	1504	377	14	77	267	35	6.4
Peak Pwr [Nm <sup>2</sup> /s]	TOPP-RA	1119	113	57	40	102	18	8.5
	TOTP3	612	67	27	38	34	6.6	1.2

changes in acceleration are mitigated, leading to a smoother velocity profile.

2) *Robot Implementation Results*: While we can observe jerk limits producing a smoother trajectory in simulation, we must still establish their tangible benefits on a real robot. We performed a comparative test on the real Kawasaki robot system described above, running the same path with and without jerk limits. The results are shown in Fig. 4.

Fig. 4A shows that the robot deviated from its desired path during rapid acceleration changes, leading to overshooting and increased fluctuations in the velocity plots. However, applying jerk limits significantly smoothed the motions, resulting in differences in the generated torque and power, as illustrated in Fig. 4B. The jerk-limited motions prevented aggressive velocity spikes, lowering peak power and torque usage, with significant reductions for some joints.

We measured peak power and RMS torque and the results are presented in Table I, showing that peak power was reduced by about 25%, and RMS torque was reduced to half of its original value by limiting jerk.

3) *Algorithm Efficiency*: We ran experiments via a C++ implementation running on a 64-bit system with a 2.35 GHz AMD EPYC 7452 32-core processor. To show the algorithm's efficiency, we measured the computation time of TOTP3 and TOPP-RA by averaging 20 trials of box-place motions. Each trial involved 20-30 path points beginning from slightly different initial box lift poses. TOTP3 (w/ jerk limit) took  $7.533 \pm 5.848 \text{ ms}$  while TOPP-RA (w/o jerk limit) took  $0.273 \pm 0.089 \text{ ms}$  to obtain the solution. The TOTP3 algorithm's runtime was longer than TOPP-RA due to the additional complexity of enforcing third-order constraints. However, it could still be executed within 10 ms, very reasonable for real-time applications. We were

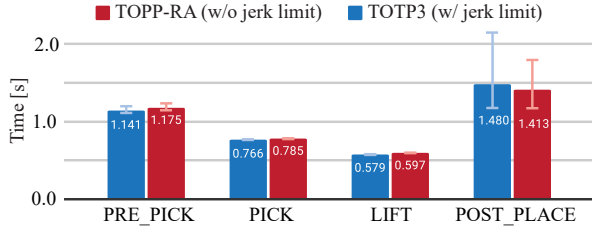


Fig. 6. Motion duration for several motions computed with and without jerk limit.

able to accelerate our computation times by warm-starting our algorithm with the non-jerk-limited TOPP-RA result and recomputing the only changed portion at each iteration.

Lastly, we measured the motion time generated by each algorithm for several different motions. Notably, imposing a jerk limit of  $1000 \text{ rad/s}^3$  increased overall motion duration by just 3-5%. For the “Post-Place” motion, we observed lower motion time with jerk limit, but this is just due to a known case of suboptimality in TOPP-RA, which guarantees optimality only if path points exclude zero-inertia point (see [19] for details), which requires denser path segmentation. On the other hand, the proposed algorithm’s SLP formulation allows it to converge to the optimal solution.

## V. DISCUSSION & CONCLUSIONS

In summary, we introduced a novel approach to solve the time-optimal trajectory planning problem while enforcing third-order constraints. In particular, we constrained the acceleration change between path points to produce smoother and more trackable trajectories. Using the proposed formulation, we removed the need for high-order dynamics variables which are computationally difficult to handle. For instance, the jerk formulation in [19] requires  $\mathbf{q}'''$ , and the torque rate formulation in [26] requires  $\mathbf{M}'$  and  $\mathbf{C}'$ . These quantities can be challenging to compute from standard robot dynamics libraries. Though this paper tested only jerk limits, the provided formulation is versatile and can be extended to other useful third-order constraints such as torque rate.

Furthermore, our study has shown a significant reduction in peak power and torque efficiency from these jerk limits. This comes at the expense of some increased computation time, though it remains well within real-time limits. Additionally, this trade-off can be managed by computing the next step while the robot is executing the preceding path and using warm-starting to recompute solutions as needed. Additionally, although jerk limits result in slightly longer motion durations, this can be compensated by reduced time required for the robot to stabilize and come to a stop, due to reduced tracking fluctuation and overshoot.

During this study, we also observed that the results were quite sensitive to path parameter function discretization and splining. Splines can interpolate the discretized solution to generate robot commands at the desired high control frequency, but some spline types may induce jerk discontinuities between spline segments. This can be mitigated by constraining acceleration changes in transitions between spline segments, which can be formulated as additional

third-order constraints. Furthermore, it is crucial to ensure sufficient discretization resolution to avoid high jerk between knot points where jerk is not directly constrained. In the future, we should explore more optimal and constrained parameterization methods for TOTP’s path parameter function and refine the overall optimization process to accommodate corresponding changes effectively.

## APPENDIX

### A. Proof of convexity

**Proposition 1.**  $f(x) = \frac{a}{\sqrt{x_1} + \sqrt{x_2}} + \frac{b}{\sqrt{x_2} + \sqrt{x_3}}$  is convex in  $x = (x_1, x_2, x_3) \in \mathbb{R}_+^3$ .

*Proof.* To show a twice-differentiable function  $f$  is convex on a convex set, it is sufficient to show that its Hessian matrix of second partial derivatives is positive semi-definite in the interior of the set [27]. The Hessian matrix is

$$H = \frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} s_{11} & s_{12} & 0 \\ s_{12} & s_{22} & s_{23} \\ 0 & s_{23} & s_{33} \end{bmatrix},$$

where

$$\begin{aligned} s_{11} &= \frac{a}{2x_1(\sqrt{x_1} + \sqrt{x_2})^2} \left( \frac{1}{2\sqrt{x_1}} + \frac{1}{\sqrt{x_1} + \sqrt{x_2}} \right) \\ s_{22} &= \frac{a}{2x_2(\sqrt{x_1} + \sqrt{x_2})^2} \left( \frac{1}{\sqrt{x_1} + \sqrt{x_2}} + \frac{1}{2\sqrt{x_2}} \right) \\ &\quad + \frac{b}{2x_2(\sqrt{x_2} + \sqrt{x_3})^2} \left( \frac{1}{2\sqrt{x_2}} + \frac{1}{\sqrt{x_2} + \sqrt{x_3}} \right) \\ s_{33} &= \frac{b}{2x_3(\sqrt{x_2} + \sqrt{x_3})^2} \left( \frac{1}{\sqrt{x_2} + \sqrt{x_3}} + \frac{1}{2\sqrt{x_3}} \right) \\ s_{12} &= \frac{a}{2\sqrt{x_1}\sqrt{x_2}(\sqrt{x_1} + \sqrt{x_2})^3}, \quad s_{23} = \frac{b}{2\sqrt{x_2}\sqrt{x_3}(\sqrt{x_2} + \sqrt{x_3})^3}. \end{aligned}$$

As  $H$  is a real symmetric matrix,  $\lambda_s$  (eigenvalues of  $H$ ) are real, which can be obtained from the characteristic equation:

$$\lambda^3 - \text{Tr}(H)\lambda^2 + \left( \sum_{i \neq j} s_{ii}s_{jj} - s_{12}^2 - s_{23}^2 \right) \lambda - |H| = 0.$$

First, given the domain of  $x_i > 0$ , it is clear that  $\text{Tr}(H) > 0$  and  $\sum_{i \neq j} s_{ii}s_{jj} - s_{12}^2 - s_{23}^2 > 0$ . The determinant of  $H$  is also positive in the given domain:

$$|H| = \frac{3ab(a(4\sqrt{x_2}\sqrt{x_3} + x_2 + 3x_3) + b(4\sqrt{x_1}\sqrt{x_2} + 3x_1 + x_2))}{64\sqrt{x_1^3x_2^3x_3^3}(\sqrt{x_1} + \sqrt{x_2})^4(\sqrt{x_2} + \sqrt{x_3})^4} > 0.$$

When considering the relationship between the roots and coefficients of a cubic equation, it is known that if all coefficients above are positive, then all  $\lambda_s$  satisfying the equation are also positive. Thus we have shown that  $H$  is positive semi-definite, which implies that  $f$  is convex.

## ACKNOWLEDGMENTS

We thank Dexterity and the HCRL personnel for their support of this project. Jee-eun Lee was a robotics intern for Dexterity, Inc. during the summer of 2023, and Luis Sentis was a consultant for Dexterity, Inc. during the summer of 2022.

## REFERENCES

- [1] T. Chettibi, H. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics-A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.
- [2] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," *Fast Motions in Biomechanics and Robotics*, pp. 65–93, 2006.
- [3] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Robotics: Science and Systems*, vol. 9, no. 1. Berlin, Germany, 2013.
- [4] Y. Zhao, H.-C. Lin, and M. Tomizuka, "Efficient trajectory optimization for robot motion planning," in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 260–265.
- [5] X. Zhang, F. Xiao, X. Tong, J. Yun, Y. Liu, Y. Sun, B. Tao, J. Kong, M. Xu, and B. Chen, "Time optimal trajectory planning based on improved sparrow search algorithm," *Frontiers in Bioengineering and Biotechnology*, vol. 10, 2022.
- [6] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.
- [7] Z. Shiller and S. Dubowsky, "Robot path planning with obstacles, actuator, gripper, and payload constraints," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 3–18, 1989.
- [8] —, "On computing the global time-optimal motions of robotic manipulators in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 785–797, 1991.
- [9] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.
- [10] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.
- [11] D. Eager, A.-M. Pendrill, and N. Reistad, "Beyond velocity and acceleration: Jerk, snap and higher derivatives," *European Journal of Physics*, vol. 37, no. 6, 2016.
- [12] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, 2014.
- [13] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [14] K. Hauser, "Fast interpolation and time-optimization with contact," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1231–1250, 2014.
- [15] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, 1986.
- [16] S. Singh and M.-C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," 1987.
- [17] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3248–3253.
- [18] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," *arXiv preprint arXiv:2105.04830*, 2021.
- [19] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, 2018.
- [20] L. Consolini, M. Locatelli, A. Minari, A. Nagy, and I. Vajk, "Optimal time-complexity speed planning for robot manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 790–797, 2019.
- [21] H. Pham and Q.-C. Pham, "On the structure of the time-optimal path parameterization problem with third-order constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 679–686.
- [22] J. Ma, S. Gao, H. Yan, Q. Lv, and G. Hu, "A new approach to time-optimal trajectory planning with torque and jerk limits for robot," *Robotics and Autonomous Systems*, vol. 140, 2021.
- [23] J. Mattmüller and D. Gisler, "Calculating a near time-optimal jerk-constrained trajectory along a specified smooth path," *The International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 1007–1016, 2009.
- [24] M. Wang, J. Xiao, S. Liu, and H. Liu, "A third-order constrained approximate time-optimal feedrate planning algorithm," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2295–2307, 2021.
- [25] S. Lu, J. Zhao, L. Jiang, H. Liu, *et al.*, "Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2017, 2017.
- [26] Q. Zhang, S. Li, and X. Gao, "Practical smooth minimum time trajectory planning for path following robotic manipulators," in *American Control Conference*. IEEE, 2013, pp. 2778–2783.
- [27] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.