2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Ensuring Joint Constraints of Torque-Controlled Robot Manipulators under Bounded Jerk

Dongwoo Ko, Jonghyeok Kim, and Wan Kyun Chung, *Fellow, IEEE*

***Abstract*— This paper proposes an optimization-based control framework for the torque-controlled robot, which can satisfy the joint position, velocity, and acceleration constraints under a bounded jerk. The optimization filter is incorporated as a module to modify the nominal controller output to ensure joint constraints. To formulate the optimization problem as a QP, the torque optimization problem is converted to the jerk optimization problem using the augmented state, and the constraints are reformulated to be affine in the jerk. Here, the viable constraints are derived using the time-optimal braking policy to guarantee the feasibility of the QP. The proposed method was validated in simulation and with a 6-DOF robot manipulator.**

I. Introduction

One of the practical challenges in operating robot manipulators is that they must comply with constraints. Typically, robot joints have bounded physical quantities such as position, velocity, and acceleration constraints. Acceleration constraints may be indirectly constrained by the consequences of other constraints, such as the torque constraint [1]. In addition, a bounded jerk is often desirable for smooth trajectories, which can reduce mechanical wear and stress in the actuation system [2]. A recent robot manipulator, Panda by Franka Emika, monitors the violation of the jerk limit in the internal controller.

Joint constraints with bounded jerk are usually treated at the trajectory planning level [2] so that the time-optimal tracking can be achieved under the given path. Recently, as robot manipulators share the workspace with the human worker, the Online Trajectory Generation (OTG) has been proposed to replan the trajectory [3]–[5] under the jerk limit for the safety and comfort of the human. Based on the braking strategies [6] (with task scaling methods [2]), the desired velocity or velocity constraints could be computed online. However, the methods at the planning level require a motion planner that can generate the desired waypoints and may not be valid for the general controller, e.g., the low-gain task space PD controller, not a high-precision motion controller.

To satisfy the joint constraints with more general robot controllers, they should be considered at the control level. Optimization-based approaches, particularly Quadratic Programming (QP), have been used to ensure robot constraints [7], [8]. In this optimization problem, robot constraints are often reformulated in terms of the optimization variable (control input) using the Control Barrier Function (CBF) [9]. The CBF-based safety filter is a widely used modular approach to secure robot constraints that modifies the dangerous nominal control input in a minimally invasive manner [10], [11]. However, the CBF may suffer from constraint conflicts unless the CBF is obtained by constructing the control invariant set [12]. The admissible sets [13] or sufficient conditions [14] for only the joint position, velocity, and acceleration constraints could be obtained and included in the QP without conflict. Similarly, if the jerk constraints are newly incorporated, the feasibility of QP must be considered again.

This paper proposes an optimization-based control framework that can satisfy the joint constraints of torque-controlled robot manipulators under bounded jerk. First, the torque optimization problem is transformed into the jerk optimization problem by augmenting the system's state. Other methods that utilize control dynamics include [15], [16] where input constraints are regarded as state constraints. However, it is unclear whether these can include the derivative of the input (i.e., jerk) without constraint conflicts. To ensure feasibility, in this paper, viable constraints are derived in closed form using two types of braking policy with the time-optimal braking policy. To the best of the authors' knowledge, this paper is the first study that could satisfy the joint constraints, including the jerk constraints, at the control level with a feasibility guarantee. The proposed method was also validated with simulation and a 6-DOF industrial collaborative robot.

The paper is organized as follows. Section II states the problem with the preliminary. The proposed method is presented in section III. It is validated in simulation and experiments in sections IV and V, respectively. The paper concludes with a discussion in section VI.

II. Problem Definition & Preliminary

A. Problem Definition

Consider the robot manipulator dynamics, which can be modeled as follows:

$$\tau_c = M(q)\ddot{q} + b(q, \dot{q}), \tag{1}$$

This work was supported by the Industrial Strategic technology development program (20009396, General purpose multi mode robot teaching device for difficult task where 0.1mm resolution of position and vel acc force teachings are essential) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea). This work was supported by the Alchemist Project 1415180884 (20012378, Development of Meta Soft Organ Module Manufacturing Technology without Immunity Rejection and Module Assembly Robot System) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea). *(Dongwoo Ko and Jonghyeok Kim contributed equally to this work.) (Corresponding author : Wan Kyun Chung).*

All authors are associated with the Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), 37673, Gyeongbuk, South Korea (E-mail: {kdw2917, sentojh, wkchung}@postech.ac.kr),

where $\tau_c \in \mathbb{R}^n$ is the joint torque, $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, and $b(q,\dot{q}) \in \mathbb{R}^n$ is the lumped force that includes the centripetal, Coriolis, and gravitational forces. The torque τ_c is designed to achieve the desired performance of the robot but may cause the joint constraints to be violated.

This paper aims to synthesize an optimization-based controller that modifies the nominal control input when it tries to violate the joint constraints. To be used seamlessly with any nominal controller, the filter (also known as a safety filter) could be integrated as a module, as shown in Fig. 1(a). It can ensure the joint constraints by solving the optimization problem formulated as follows:

$$\tau_c^* = \underset{\tau_c}{\operatorname{argmin}} ||\tau_c - \tau_{\mathrm{d}}|| \tag{2}$$

$$\begin{aligned} \text{s.t.} \; -q_{\lim} \le q \le q_{\lim} \\ -\dot{q}_{\lim} \le \dot{q} \le \dot{q}_{\lim} \\ -\ddot{q}_{\lim} \le \ddot{q} \le \ddot{q}_{\lim} \\ -\dddot{q}_{\lim} \le \dddot{q} \le \dddot{q}_{\lim} \end{aligned} \tag{3}$$

where $\tau_d \in \mathbb{R}^n$ is the desired joint torque which comes from the nominal controller and $q, \dot{q}, \ddot{q}, \dddot{q} \in \mathbb{R}^n$ are the joint position, velocity, acceleration, and jerk variables, respectively.

However, there are two challenges in handling the optimization problem when considering the jerk constraint. First, we cannot analytically represent the jerk constraint as an algebraic function of the input (torque or acceleration) because the jerk constraint is of higher order than the input. To formulate the optimization problem as a QP that allows for real-time computation, the jerk constraint must be affine with respect to the optimization variable.

Second, the feasibility of the optimization problem must be ensured: a possible optimization variable must exist that satisfies all optimization constraints. It is well-known that conflicts between constraints can arise in optimization-based control problems, particularly when input constraints are involved [12]. Consider a scenario in which a car is braking to avoid a collision. The driver must initiate braking while the distance from the obstacle is still greater than the required stopping distance, as the braking force, determined by the friction between the tires and the road, is inherently limited. If the braking initiation time is delayed, collision avoidance becomes impossible because a braking force greater than the maximum available would be required. Similarly, under limited input, even if the current system satisfies the constraints, there may be no feasible input in the future to secure the constraints. To ensure feasibility, the constraint must be reformulated into a viable form that does not conflict with the input constraints.

B. Preliminary

Before explaining the proposed method, the set invariance approaches, including Nagumo's theorem and the CBF, were briefly reviewed. The set invariance means the trajectory starting in the invariant set will never leave the set [17]. Thus, the constraint insurance can be mathematically formulated based on the set invariance approach by defining the set based on the constraints.

For a dynamical system $\dot{x} = f(x)$, assume that the set $\mathcal{C}$ is the superlevel set of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$, i.e., $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \ge 0\}$ with $\frac{\partial h}{\partial x}(x) \neq 0$ for all x such that $h(x) = 0$ satisfying

$$\begin{cases} h(x) & \ge 0 \;\; \text{for } x \in \mathcal{C} \\ h(x) & = 0 \;\; \text{for } x \in \partial\mathcal{C} \\ h(x) & > 0 \;\; \text{for } x \in \text{Int}(\mathcal{C}). \end{cases} \tag{4}$$

Then, Nagumo's theorem states the necessary and sufficient condition for set invariance based upon the derivative of a function h on the boundary of the set $\mathcal{C}$ as below:

$$\mathcal{C} \text{ is invariant } \Leftrightarrow \;\; \dot{h}(x) \ge 0 \;\; \forall x \in \partial\mathcal{C}.$$

The above condition matches directly to prevent the system from escaping the set $\mathcal{C}$.

In this manner, extending Nagumo's Theorem, the CBF has been researched as an extension of the "control" version that takes a similar intuition about the set invariance. Consider a nonlinear affine control system as

$$\dot{x} = f(x) + g(x)u, \tag{5}$$

where f and g are Lipschitz, x is the state, and u is the control input of the system.

If it can be guaranteed that $x(t) \in \mathcal{C}$ for all $t > 0$ for any $x_0 \in \mathcal{C}$, then the set $\mathcal{C}$ is said to be forward invariant. Here, the Lipschitz function $h(x)$ is a CBF if there exists extended class $\mathcal{K}_\infty$ function[1] α

$$\sup_{u\in\mathcal{U}} \underbrace{\dot{h}}_{\frac{\partial h}{\partial x}(f(x)+g(x)u)} + \alpha(h) \ge 0. \tag{6}$$

If the control input $u \in \mathcal{U}$ satisfies the above condition, i.e., $\frac{\partial h}{\partial x}(f(x) + g(x)u) + \alpha(h) \ge 0$, then the set $\mathcal{C}$ is forward invariant (refer to [9] for a more rigorous analysis).

III. Proposed method

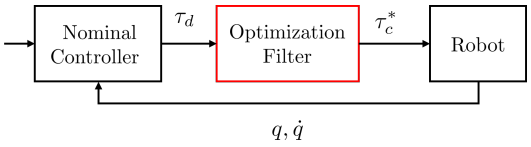
This section introduces the proposed method. First, the overall control structure is explained, and the optimization filter is proposed to ensure joint constraints.

A. Overall control structure

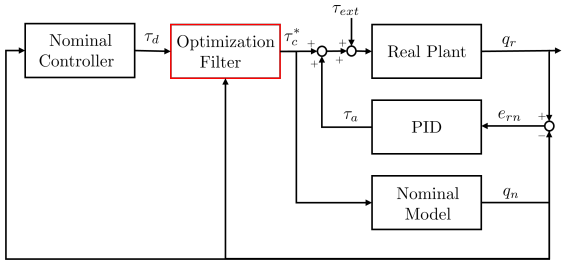
For the simulation environment, the proposed method consists of the nominal controller and the optimization filter, as shown in Fig. 1(a). The nominal controller can be designed to satisfy the desired robot control performance, and the optimization filter will reshape the nominal controller output to satisfy the joint constraints.

However, the proposed method would not be valid in real-world applications for two reasons. First, many robot manipulators usually suffer from the uncertainty of the model in (1), such as friction. Under the model uncertainty, the constraints in (3) could not be satisfied since the robot dynamics is used when converting the constraints into the optimization variable. Second, the constraints in (3) would

[1] A extended class $\mathcal{K}_\infty$ function is a strictly increasing function $\alpha : \mathbb{R} \to \mathbb{R}$ such that $\alpha(0) = 0$ and $\lim_{r\to\infty} \alpha(r) = \infty$.



(a) Optimization-based safety filter



(b) The NRIC structure [18] is used with the optimization filter for the real world application.

Fig. 1. Overall control framework.

require the current joint acceleration and its derivative, i.e., the jerk, which is often immeasurable or noisy.

To address the issues, this paper used the nonlinear robust internal-loop compensator (NRIC) structure [18], which could bring nonlinear $\mathcal{H}_\infty$ optimality to robot controllers such as PD controller and impedance controller. As shown in Fig. 1(b), the nominal plant is newly introduced, which is simulated with each control step, and the nominal plant can easily achieve the desired performance because it is not affected by any model uncertainty (when the optimization filter is not used). Then, the robustness can be imparted to the real plant by adding a simple PID-type auxiliary input to the real plant as

$$\tau_a = -K_r(\dot{e}_{rn} + K_p e_{rn} + K_i \int e_{rn}), \tag{7}$$

where $e_{rn} = q_r - q_n$ and q_r, q_n are the displacement vector of the real plant and nominal model, respectively. Namely, combining the NRIC structure can address model uncertainty issues. In addition, the optimization problem is designed for the nominal plant whose acceleration and jerk can be accurately calculated. By the NRIC structure, under the model uncertainty and noisy signal, the real plant would try to behave as the nominal plant that satisfies the joint constraints. In this paper, the optimization filter will use the nominal plant signal, omitting the subscript "n" for simplicity.

The following two subsections reformulate the optimization filter in (2). First, the optimization problem is reformulated by changing the optimization variable from the joint torque to the joint jerk. Then, each joint constraint is replaced by a viable constraint to guarantee the feasibility of the optimization. The optimization constraints derived from the viable constraints become affine with respect to the optimization variable (jerk), which leads to QP. Finally, the allowable torque input can be derived from the solution of the optimization problem, which is the optimal jerk.

B. Objective function reformulation

In this subsection, the objective function in the optimization problem is reformulated in terms of joint jerk instead of joint torque. For this purpose, the system state is extended by introducing the internal control dynamics. First, the state of the system is denoted as $x = [q^T, \dot{q}^T]^T$ and the dynamics can be written as

$$\dot{x} = \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \underbrace{\begin{pmatrix} \dot{q} \\ -M^{-1}(q)b(q,\dot{q}) \end{pmatrix}}_{f(x)} + \underbrace{\begin{pmatrix} 0 \\ M^{-1}(q) \end{pmatrix}}_{g(x)} \tau_c. \tag{8}$$

Then, the control dynamics is modeled as

$$\dot{x}_c = \underbrace{(0_{n\times n})}_{f_c(x_c)} + \underbrace{(I_{n\times n})}_{g_c(x_c)} u_c \tag{9}$$

$$\tau_c = h_c(x, x_c) = M(q)x_c + b(q,\dot{q}), \tag{10}$$

where $x_c = \ddot{q}$, $u_c = \dddot{q}$. The augmented state equation is given as

$$\dot{z} = \begin{pmatrix} \dot{x} \\ \dot{x}_c \end{pmatrix} = \underbrace{\begin{pmatrix} f(x) + g(x)h_c(x,x_c) \\ f_c(x_c) \end{pmatrix}}_{f_z(z)} + \underbrace{\begin{pmatrix} 0 \\ g_c(x_c) \end{pmatrix}}_{g_z(z)} u_c, \tag{11}$$

where $z = [q^T, \dot{q}^T, \ddot{q}^T]^T$ is an augmented state. With a simple derivation, it can be easily seen that the augmented state equation is the triple integrator system as $f_z = [\dot{q}^T, \ddot{q}^T, 0_{n\times 1}^T]^T$, $g_z = [0_{n\times 1}^T, 0_{n\times 1}^T, 1_{n\times 1}^T]^T$.

Based on the equation of the augmented state, the objective function (2) needs to be rewritten in terms of the joint jerk u_c. First, using the system dynamics (1), the objective function is equivalent to

$$\tau_c^*, \ddot{q}^* = \underset{\tau_c, \ddot{q}}{\operatorname{argmin}} ||M(\ddot{q} - \ddot{q}_d)||, \tag{12}$$

where $\ddot{q}_d = M^{-1}(\tau_d - b)$ and τ_d is the output of the nominal controller. Then, the surrogate[2] represented with the jerk u_c is suggested such that minimizing it tends to minimize the objective function in (12) as

$$u_c^* = \underset{u_c}{\operatorname{argmin}} ||M(u_c - u_{c,d})||, \tag{13}$$

$$\text{where } u_{c,d} = \dddot{q}_d - \gamma(\ddot{q} - \ddot{q}_d). \tag{14}$$

It is noted that minimizing this objective function tends to minimize the (12) as the following proposition.

Proposition 1. *Under the jerk $u_c = u_{c,d}$, it follows that*

$$\lim_{t\to\infty} (\ddot{q}(t) - \ddot{q}_d(t)) = 0, \tag{15}$$

which minimizes (12).

Proof: Since $u_c = u_{c,d}$, the error $e_u(t) = \ddot{q}(t) - \ddot{q}_d(t)$ satisfies $\dot{e}_u(t) + \gamma e_u = 0$ from (14). Thus, $\ddot{q}$ converges exponentially to $\ddot{q}_d$ which minimizes (12). □

Remark 1. *It can be difficult to analytically obtain the desired jerk $\dddot{q}_d$ in* (14). *Instead, if the control time duration*

[2]The surrogate is the specific case of the one proposed in [16, IV-B].

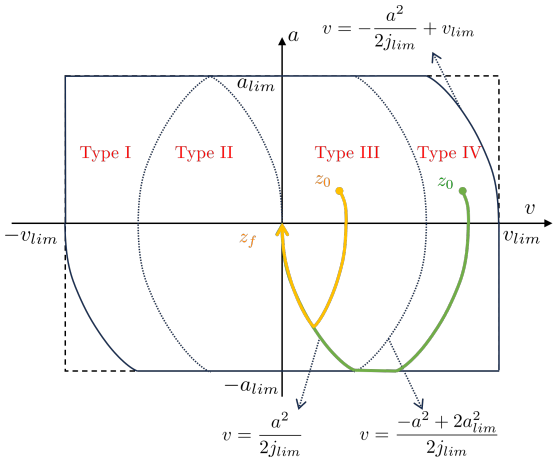
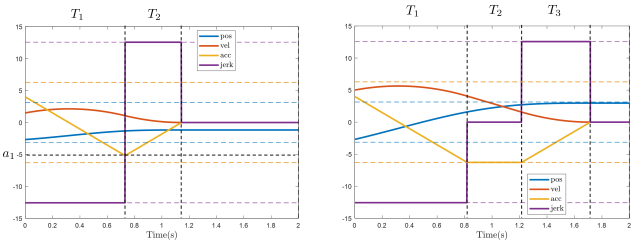


Fig. 2. The velocity and acceleration trajectory using the time-optimal stationary braking policy. It uses four types of jerk profiles depending on the current velocity and acceleration.



(a) The profile for the type III (b) The profile for the type IV

Fig. 3. The jerk profile of the time-optimal stationary braking policy depends on the type.

Δt is sufficiently small, we can approximate the desired jerk numerically as $\dddot{q}_{d,k} \approx \frac{\ddot{q}_{d,k} - \ddot{q}_{d,k-1}}{\Delta t}$ where k is the control time step.

C. Feasible constraints derivation

To guarantee the feasibility of the QP, a possible jerk should exist that satisfies all the optimization inequalities arising from the joint constraints. First, the (augmented) state constraints, including position, velocity, and acceleration, should not induce conflict with the input (jerk) constraints. Second, the state constraints should be compatible with each other. In this subsection, the viable constraints for state constraints will be derived by considering input constraints and then converting them into input constraints using the CBF condition without conflict.

Without loss of generality, the viable constraints are analyzed for the 1-DOF system, and the position, velocity, acceleration, and jerk will be denoted as $p(t), v(t), a(t)$, and $j(t)$, respectively. As a motivating example, consider only the position and acceleration constraints. In this case, it is well known that the viable constraints are given for the maximum position as [12], [13]:

$$p_{\text{lim}} - p - \frac{v^2}{2a_{\text{lim}}} \geq 0, \tag{16}$$

which includes the minimum stopping distance in the constraints. It can be derived from the inequality $p_f \leq p_{\text{lim}}$, where p_f is the final position when stopped under the maximum deceleration $a = -a_{\text{lim}}$. In other words, the condition is derived using the trajectory driven from the current state $x(t_0) = x_0 = [p_0, v_0]^T$ to the stationary state $x(t_f) = x_f = [p_f, 0]^T$ by the maximum deceleration.

Similarly, considering the time-optimal stationary braking policy from the current state $z(t_0) = z_0 = [p_0, v_0, a_0]^T$ to the final state $z(t_f) = z_f = [p_f, 0, 0]^T$ under the limited jerk, the viable constraints can be derived. There are four types of jerk profiles depending on the current velocity and acceleration, as shown in Fig. 2. Since the types I and II are symmetric to IV and III, respectively, we will only explain the types III and IV. The profile uses the piecewise constant jerk, $j_i(t) \in \{-j_{\text{lim}}, 0, j_{\text{lim}}\}$. For the constant jerk, the position, velocity, and acceleration can be computed as

$$\begin{aligned} p_i &= p_{i-1} + v_{i-1}T_i + \frac{1}{2}a_{i-1}T_i^2 + \frac{1}{6}j_iT_i^3 \\ v_i &= v_{i-1} + a_{i-1}T_i + \frac{1}{2}j_iT_i^2 \\ a_i &= a_{i-1} + j_iT_i \end{aligned} \tag{17}$$

where $i = 1, 2$ corresponds to type III, $i = 1, 2, 3$ to type IV, and T_i represents the time duration during which the constant jerk is applied. Accordingly, the system reaches the final state z_f after $T_1 + T_2$ for type III and after $T_1 + T_2 + T_3$ for type IV, as shown in Fig. 3. The time durations T_i are given in Table I depending on the types of jerk profiles. As shown in Fig. 3, it is easy to see that the difference between the types III and IV is whether the acceleration reaches its minimum value. Using this policy, we can obtain the viable form of each constraint in the closed form as follows.

1) Position constraints: The types III and IV currently have or will eventually have a positive velocity. Thus, the final position p_f (when $v_f = a_f = 0$) should be smaller than the maximum position p_{lim}. Using (17), the final position p_f can be calculated, and it is observed that the final position is a function of the current position, velocity, and acceleration, and the feasible constraint is given by

$$h_p(p_0, v_0, a_0) = p_{\text{lim}} - p_f(p_0, v_0, a_0) \geq 0. \tag{18}$$

On the other hand, the minimum position constraint does not need to be considered for types III and IV if it has already been addressed when the system is in types I and II, and vice versa. The scenario of the greatest position deviation in the negative direction for type III under the optimal stationary braking policy occurs when the trajectory follows the boundary between types II and III. Since the state must cross the boundary between types II and III to reach type III, the worst-case scenario for the minimum position in type III is already satisfied.

2) Velocity constraints: Likewise, the types III and IV only need to consider the maximum velocity constraints. In both cases, it is specified as

$$h_v(v_0, a_0) = v_{\text{lim}} - v_0 - \frac{a_0^2}{2j_{\text{lim}}} \geq 0, \tag{19}$$

which is similar to (16).

TABLE I

THE INFORMATION TO COMPUTE THE TRAJECTORY BY TIME-OPTIMAL STATIONARY BRAKING POLICY

	T_1	T_2	T_3 or a_1	$\frac{dp_f}{dp_0}$	$\frac{dp_f}{dv_0}$	$\frac{dp_f}{da_0}$
Type I	$\frac{a_{\lim}-a_0}{j_{\lim}}$	$-\frac{v_0}{a_{\lim}}-\frac{a_{\lim}}{j_{\lim}}+\frac{a_0^2}{2j_{\lim}a_{\lim}}$	$T_3=\frac{a_{\lim}}{j_{\lim}}$	1	$T_1+T_2+\frac{1}{2}T_3$	$\frac{1}{2}T_1(T_1+2T_2+T_3)$
Type II	$\frac{a_1-a_0}{j_{\lim}}$	$\frac{a_1}{j_{\lim}}$	$a_1=\sqrt{\frac{a_0^2-2j_{\lim}v_0}{2}}$	1	$T_1+\frac{1}{2}T_2$	$\frac{1}{2}T_1(T_1+T_2)$
Type III	$\frac{a_0-a_1}{j_{\lim}}$	$-\frac{a_1}{j_{\lim}}$	$a_1=-\sqrt{\frac{a_0^2+2j_{\lim}v_0}{2}}$	1	$T_1+\frac{1}{2}T_2$	$\frac{1}{2}T_1(T_1+T_2)$
Type IV	$\frac{a_{\lim}+a_0}{j_{\lim}}$	$\frac{v_0}{a_{\lim}}-\frac{a_{\lim}}{j_{\lim}}+\frac{a_0^2}{2j_{\lim}a_{\lim}}$	$T_3=\frac{a_{\lim}}{j_{\lim}}$	1	$T_1+T_2+\frac{1}{2}T_3$	$\frac{1}{2}T_1(T_1+2T_2+T_3)$

3) Acceleration constraints: The acceleration constraints are given as

$$h_{a,\max}(a_0)=a_{\lim}-a_0\geq 0, h_{a,\min}(a_0)=a_0+a_{\lim}\geq 0. \tag{20}$$

4) Optimization constraints: Since all the above state constraints depend on the current position, velocity, and acceleration, they must be converted into the conditions affine in the input (jerk) to formulate the QP. First, for the position and velocity constraints, the CBF condition (6) is used. The time derivative for the position constraints (18) can be calculated by the chain rule. For example, the maximum position can be obtained by

$$\dot{h}_p=\frac{dh_p}{dt}=-\frac{dp_f}{dp_0}v_0-\frac{dp_f}{dv_0}a_0-\frac{dp_f}{da_0}j \tag{21}$$

where the derivative terms, given in Table I, can be obtained through a tedious calculation using (17). For the acceleration constraints, the following constraints can be easily considered:

$$\begin{cases} j\leq 0 & \text{if } a_0=a_{\lim} \\ j\geq 0 & \text{else if } a_0=-a_{\lim}. \end{cases} \tag{22}$$

Despite the state (position, velocity, and acceleration) constraints being reformulated under the input (jerk) constraints, it is still unclear whether conflicts between the state constraints might occur. Note that the viable constraints are only conflict-free with the input constraint. However, the feasibility of the optimization problem can be ensured since the above viable constraints are derived using the same braking policy, as shown below.

Proposition 2. *The optimization problem with the constraints of viable position* (18) *and velocity* (19) *constraints using the CBF condition in* (6)*, the acceleration constraints* (22)*, and the jerk constraints in* (3) *are always feasible.*

Proof: Without loss of generality, it is shown for the 1-DOF system and only if the state is of type III or IV. The proof can be shown by checking whether there exists a possible jerk that can satisfy all the constraints. Indeed, the time-optimal stationary braking policy can provide a feasible jerk given by

$$\begin{cases} j=0 & \text{if } a_0=-a_{\lim} \\ j=j_{\lim} & \text{else if } v_0=\frac{a_0^2}{2j_{\lim}} \text{ and } a_0\leq 0 \\ j=-j_{\lim} & \text{else} \end{cases} \tag{23}$$

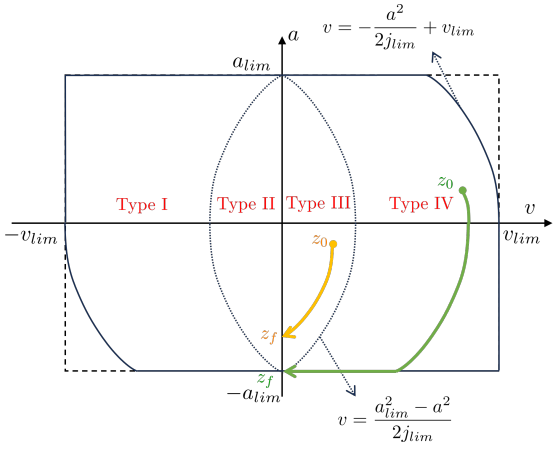


Fig. 4. The velocity and acceleration trajectory using the time-optimal velocity-zeroing braking policy. It also uses four types of jerk profiles depending on the current velocity and acceleration.

for type III or IV. Under the policy, the time derivative of the CBF of the (maximum) position constraint (18) becomes zero, so the CBF condition (6) is satisfied if the current state satisfies the viable position constraint (18). In addition, the time derivative of the CBF of the velocity constraint (19), $\dot{h}=-a_0(1+\frac{j}{j_{\lim}})$, is equal to or greater than zero with the above jerk policy. The acceleration and jerk constraints are straightforward. □

In summary, the final optimization formulation to consider the joint constraints under the bounded jerk is as follows:

$$u_c^*=\underset{u_c\in\mathcal{U}}{\operatorname{argmin}}\,||M(u_c-u_{c,d})||$$

subject to (18), (19), and (22). Note that (18) and (19) should follow the CBF condition, i.e., $\dot{h}\geq-\alpha(h)$ in (6).

Remark 2. *One can think of the time-optimal velocity-zeroing braking policy that drives the current state to the final state $z(t_f)=[p_f,0,a_f]^T$ where a_f is not necessarily zero instead of the proposed method. This policy also has four types of jerk profiles according to the current velocity and acceleration, as shown in Fig. 4. This policy could achieve a shorter stopping distance than the proposed one. Still, it could generate the oscillatory motion near the position limit under the limited jerk, as shown in the simulation, where the final position and its derivative are calculated based on Table II. Other constraints for velocity and acceleration are set the same.*

TABLE II

THE INFORMATION UNDER TIME-OPTIMAL VELOCITY-ZEROING BRAKING POLICY

	T_1	T_2	$\frac{dp_f}{dp_0}$	$\frac{dp_f}{dv_0}$	$\frac{dp_f}{da_0}$
Type I	$\frac{a_{\lim}-a_0}{j_{\lim}}$	$\frac{-v_1}{a_{\lim}}$	1	T_1+T_2	$\frac{1}{2}T_1^2+T_1T_2$
Type II	$\frac{-a_0+\sqrt{a_0^2-2j_{\lim}v_0}}{j_{\lim}}$		1	T_1	$\frac{1}{2}T_1^2$
Type III	$\frac{a_0+\sqrt{a_0^2+2j_{\lim}v_0}}{j_{\lim}}$		1	T_1	$\frac{1}{2}T_1^2$
Type IV	$\frac{a_{\lim}+a_0}{j_{\lim}}$	$\frac{v_1}{a_{\lim}}$	1	T_1+T_2	$\frac{1}{2}T_1^2+T_1T_2$

where $v_1 = v_0 + a_0T_1 + \frac{1}{2}j_1T_1^2$, and $j_1 = j_{\lim}$ or $j_1 = -j_{\lim}$.

IV. SIMULATION

In this section, we validate the proposed method in simulation for a 1-DOF robot manipulator compared to other approaches. The low-gain PD control is used, and the desired waypoints are predefined so that the robot violates the joint constraints. Then, the robot dynamics are as follows:

$$m\ddot{q} = k_p(q_d - q) - k_d\dot{q} \tag{24}$$

where $m \in \mathbb{R}^+$ is the mass, $k_p, k_d \in \mathbb{R}^+$ are the proportional and derivative gains. For simplicity, the mass is set to one, and then the desired acceleration is given as $\ddot{q}_d = k_p(q_d - q) - k_d\dot{q}$. Note that the joint variables in the simulation are scalar (1-D), i.e., $q, \dot{q}, \ddot{q}, \dddot{q} \in \mathbb{R}$. Consequently, the desired jerk can be obtained as

$$u_{c,d} = \dddot{q}_d - \gamma\left(\ddot{q} - \ddot{q}_d\right) = -k_p\dot{q} - k_d\ddot{q} - \gamma\left(\ddot{q} - \ddot{q}_d\right). \tag{25}$$

The joint constraints are enforced as follows:

$$q_{\lim} = \pi, \dot{q}_{\lim} = 2\pi, \ddot{q}_{\lim} = 3\pi, \dddot{q}_{\lim} = 4\pi \tag{26}$$

with proper unit (rad). The control gain was set as $k_p = 5, k_d = 1, \gamma = 100$ and the class $\mathcal{K}_\infty$ function is set as a linear function as $\alpha(x) = k_\alpha x$ with $k_\alpha = 5$. Two simulation studies are examined, and the results are summarized below.

1) Comparison with existing methods: First, the proposed method is compared with the existing method. We compare the cases below:

- S1: Without any constraint,
- S2: With position, velocity, and acceleration constraints,
- S3: With position, velocity, acceleration, and jerk constraints without considering the feasibility,
- S4: With position, velocity, acceleration, and jerk constraints considering the feasibility (proposed method).

S3 uses the exponential CBF that has the same negative real poles, that is, $-k_\alpha$ [9, IV-B]. The desired waypoint was generated by alternating 2π and -2π every 2 s.

The result is shown in Fig. 5. First, the joint variables are not bounded for S1 since it does not consider any constraint. The joint position, velocity, and acceleration are bounded for S2 and S4, but the joint jerk is only satisfied by the proposed method (S4). In particular, S2 (see the orange line) requires a giant jerk to use the maximum deceleration so as not to violate the position limit at about $t = 1.2$ s and to follow the desired acceleration, which changes abruptly when the desired waypoint alternates at $t = 2$ s. Meanwhile, S3 shows an infeasible optimization result so that it cannot solve QP (see the yellow line after $t = 1$ s in Fig. 5). The feasibility issue in QP originates from the enforcement of constraints without viability. Note that the proposed method can guarantee the feasibility of QP due to the viability consideration based on the proposed braking policy.

2) Comparison depending on the braking policy: The second simulation was studied to compare the results depending on the braking policy between the time-optimal stationary braking policy and the time-optimal velocity-zeroing braking policy (see remark 2). The desired waypoints have been specified as 2π for all time. The behavior at the constraint boundary can be captured since the desired trajectory continues to violate the joint position constraint. We compare the following cases:

- S4: Viable constraints using time-optimal stationary braking policy (TOSBP, see section III-C)
- S5: Viable constraints using time-optimal velocity-zeroing braking policy (TOVZBP, see remark 2).

The result is shown in Fig. 6. S5 shows an oscillatory motion at the boundary of the joint position constraint. Instead, the proposed method (S4), which constructs the viable constraint using TOSBP, can stay at the constraint boundary. Since TOVZBP only sets the current velocity to zero, unnecessary acceleration can cause oscillation, while TOSBP tries to make the system stationary.

V. EXPERIMENT & RESULTS

A 6-DOF collaborative robot (Indy7, Neuromeka) was used to verify the proposed control framework in the real world. In particular, the unilateral teleoperation experiment was conducted since the joint constraints may be violated easily by the remote side user command. The motion command was generated by a 6-DOF remote controller (Vive Pro controller, HTC) during the teleoperation. To solve the optimization in real-time (1kHz), qpOASES [19] was used.

A task space PD controller was used as the nominal controller to validate whether the proposed method could ensure the joint constraints at the control level. The control torque of the task space PD controller was calculated as

$$\tau_{pd} = J^TKe + J^TD\dot{e}, \tag{27}$$

where $e = r_{\text{des}} - r$, $\dot{e} = \dot{r}_{\text{des}} - \dot{r} \in \mathbb{R}^6$ are task space pose error and velocity error, respectively. The matrices $K, D \in \mathbb{R}^{6\times6}$ denote task space stiffness and damping gain. Then, the resulting acceleration can be obtained as

$$\ddot{q}_d = M^{-1}(\tau_{pd} - b). \tag{28}$$

Furthermore, the desired joint jerk is calculated by (14) as

$$u_{c,d} = \dddot{q}_d - \gamma\left(\ddot{q} - \ddot{q}_d\right), \tag{29}$$

where γ is a proportional gain of acceleration error. During the experiment, the control parameters are set as $K = \text{diag}\{600, 600, 600, 160, 160, 160\}$, $D = \text{diag}\{60, 60, 60, 16, 16, 16\}$, and $\gamma = 20$. For the nominal

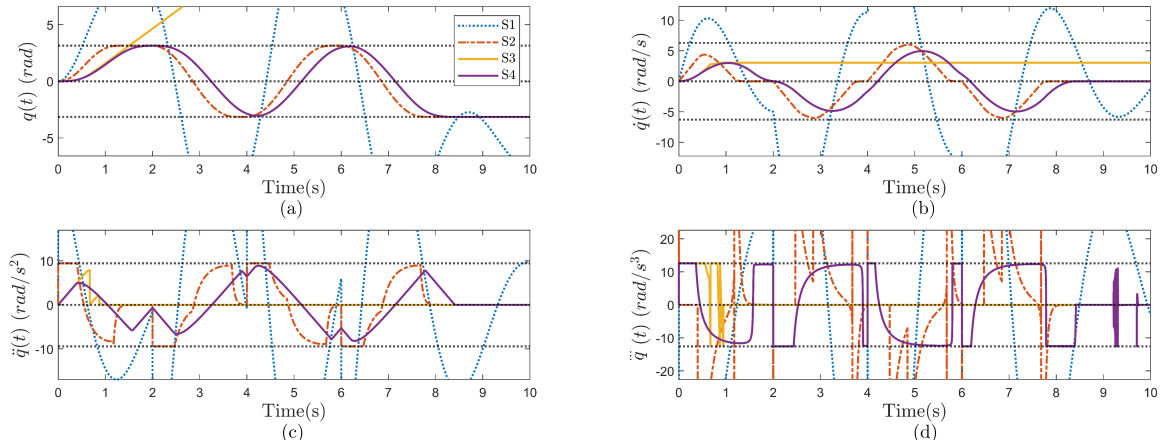


Fig. 5. Results of the first simulation. Plots of (a) joint position q, (b) joint velocity $\dot{q}$, (c) joint acceleration $\ddot{q}$, and (d) joint jerk $\dddot{q}$. The constraints are shown as black dotted lines in each plot as upper and lower bounds. S1 violates the joint constraints, as it simply tracks the desired waypoints. S2 satisfies the joint position, velocity, and acceleration constraints but violates the joint jerk constraint, while S4 (the proposed) simultaneously satisfies the jerk constraint. S3, the enforced constraint without considering the feasibility condition, does not guarantee a feasible optimization. Thus, the simulation is stopped because it cannot solve QP after $t = 1$ s.

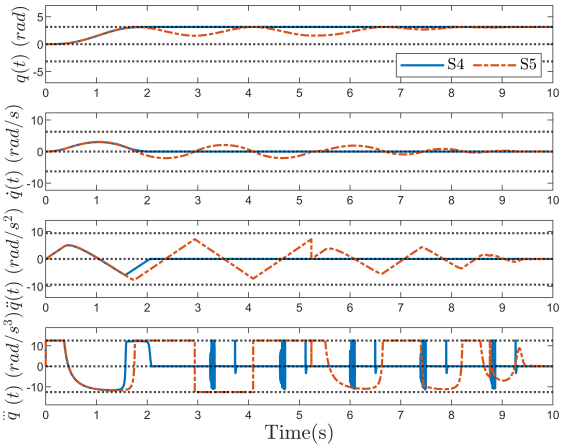


Fig. 6. Result of the second simulation. The constraints are shown as black dotted lines in each plot as upper and lower bounds.

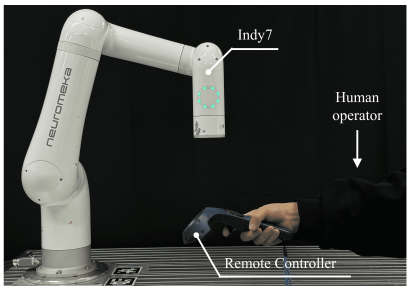


Fig. 7. Experimental setup in real-world applications. A 6-DOF collaborative robot (Indy7, Neuromeka) was used where the desired motion is commanded by the user with a remote controller (Vive Pro controller, HTC).

robot based on the NRIC structure, the control parameters for the auxiliary input torque are $K_\gamma = 600$, $K_p = 20I_{6\times6}$, and $K_i = 8I_{6\times6}$. The joint constraints are imposed as follows:

$$\begin{aligned} -q_{\mathrm{lim},i} \le q_i - q_{h,i} \le q_{\mathrm{lim},i} \quad &(\mathrm{rad}) \\ -2\pi/3 \le \dot{q}_i \le 2\pi/3 \quad &(\mathrm{rad/s}) \\ -4\pi \le \ddot{q}_i \le 4\pi \quad &(\mathrm{rad/s^2}) \\ -12\pi \le \dddot{q}_i \le 12\pi \quad &(\mathrm{rad/s^3}), \end{aligned} \tag{30}$$

for all joints $i \in \{1, 2, \cdots 6\}$. The position constraints were set relative to the home configuration of the robot, $q_h = (0, -\pi/12, -\pi/2, -0, -5\pi/12, 0)$ rad, with the limit $q_{\mathrm{lim}} = (\pi/6, \pi/12, \pi/6, \pi/12, \pi/6, \pi/3)$ rad. We implemented two cases as follows:

- C1: With position, velocity, and acceleration constraints,
- C2: With position, velocity, acceleration, and jerk constraints considering the feasibility (proposed method, TOSBP was used).

During the experiment, the human operator arbitrarily operated the robot using the remote controller for C2 first, and the command motion was recorded and reused for C1. Then, we checked whether the proposed method satisfies the joint constraints. As shown in Fig. 8, C1 violates the joint jerk constraints, but C2 could satisfy all joint constraints by the proposed method.

Furthermore, Fig. 9 presents a comparison of the trajectories between the real robot (illustrated in blue) and the nominal plant (illustrated in yellow) under C2. Due to the numerical computation required for acceleration and jerk, the jerk trajectory of the real robot exhibits significant noise. This makes it difficult to confirm that the real robot met all joint constraints. Nevertheless, it is observed that the performance of the real robot closely aligns with that of the nominal model (see the position and velocity trajectories), which adheres to the joint constraints. This outcome is due to the use of the NRIC structure. The experimental results are also demonstrated in the supplementary video.

VI. Conclusion

This work suggests a method that can satisfy the joint constraints at the control level, allowing it to be used with general nominal controllers. It was validated in simulation by

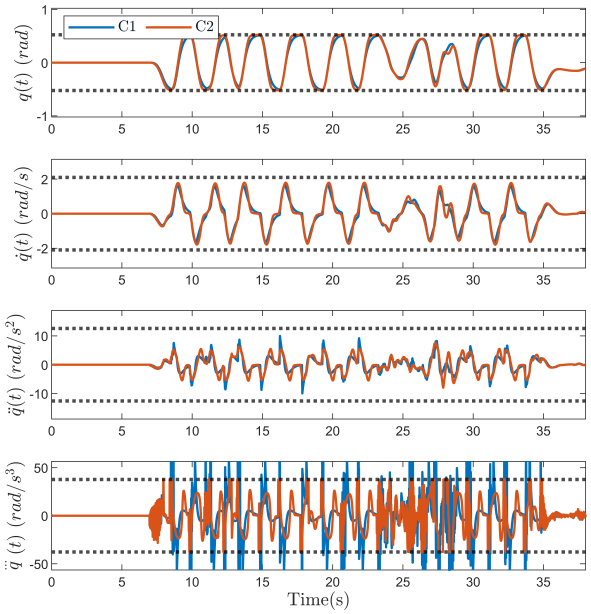


Fig. 8. Trajectories of the first joint of the nominal plant for C1 and C2. The black dotted line is the upper and lower bound for the joint constraints imposed as in (30). The plot shows that C1 violates the joint jerk constraints, while C2 can satisfy all joint constraints.

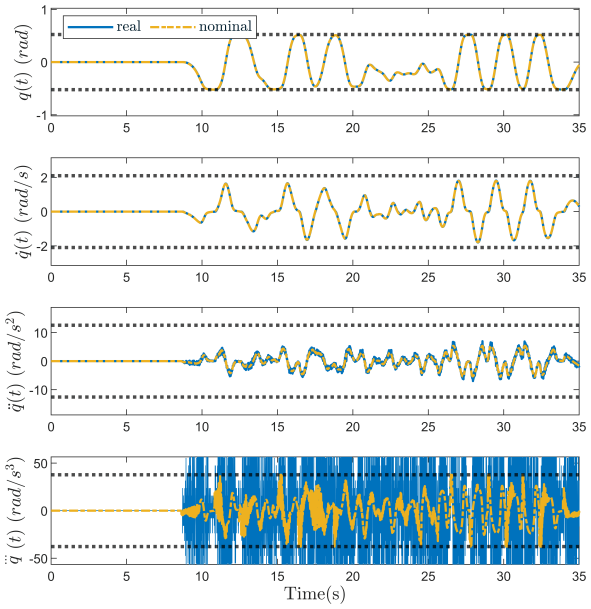


Fig. 9. Trajectories of the first joint of real robot (blue) and nominal robot (yellow) under C2. The jerk trajectory of the real robot exhibits significant noise, but the real robot's performance closely aligns with that of the nominal model, thanks to the NRIC structure.

comparison with existing methods and the real application, i.e., unilateral teleoperation with a PD controller.

However, the proposed method has several limitations that need to be addressed in the future. Firstly, the method only considers the constant constraint. The torque constraints can be included if it is possible to consider the state-dependent constraint. Second, for the safety of the robot and its environment, it would be more useful if the task space constraints, such as collision avoidance, could be included in the QP with the feasibility guarantee. Lastly, although the proposed method could also be used with the interaction controller, further analysis is required when rigid contact occurs.

REFERENCES

[1] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[2] A. Palleschi, M. Garabini, D. Caporale, and L. Pallottino, "Time-optimal path tracking for jerk controlled robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3932–3939, 2019.

[3] K. Desormeaux and D. Sidobre, "Online trajectory generation: Reactive control with return inside an admissible kinematic domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4381–4386.

[4] D. Sidobre and K. Desormeaux, "Smooth cubic polynomial trajectories for human-robot interactions," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3-4, pp. 851–869, 2019.

[5] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," *arXiv preprint arXiv:2105.04830*, 2021.

[6] L. Joseph, J. K. Pickard, V. Padois, and D. Daney, "Online velocity constraint adaptation for safe and efficient human-robot workspace sharing," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 045–11 051.

[7] S. Hong, K. Jang, S. Kim, and J. Park, "Regularized hierarchical quadratic program for real-time whole-body motion generation," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2115–2126, 2020.

[8] M. Djeha, P. Gergondet, and A. Kheddar, "Robust task-space quadratic programming for kinematic-controlled robots," *IEEE Transactions on Robotics*, 2023.

[9] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[10] A. Singletary, S. Kolathaya, and A. D. Ames, "Safety-critical kinematic control of robotic systems," *IEEE Control Systems Letters*, vol. 6, pp. 139–144, 2021.

[11] D. Lee, D. Ko, W. K. Chung, and K. Kim, "Maximal manipulation framework using quadratic programming for a teleoperated robotic system with articulated bodies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9339–9345.

[12] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6835–6841.

[13] A. Del Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 281–288, 2017.

[14] D. Lee, D. Ko, W. K. Chung, and K. Kim, "Quadratic programming-based task scaling for safe and passive robot arm teleoperation," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 1937–1945, 2022.

[15] A. D. Ames, G. Notomista, Y. Wardi, and M. Egerstedt, "Integral control barrier functions for dynamically defined control laws," *IEEE control systems letters*, vol. 5, no. 3, pp. 887–892, 2020.

[16] P. Rabiee and J. B. Hoagg, "Composition of control barrier functions with differing relative degrees for safety under input constraints," *arXiv preprint arXiv:2310.00363*, 2023.

[17] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[18] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear $\mathcal{H}_\infty$ optimality to robot controllers," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 682–698, 2015.

[19] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.