Journal of Intelligent & Robotic Systems (2026) 112:16
https://doi.org/10.1007/s10846-025-02335-z

REGULAR PAPER

Ensuring Viability: A QP-based Inverse Kinematics for Handling Joint Range, Velocity and Acceleration Limits, as Well as Whole-body Collision Avoidance

Yachen Zhang[1] · Ryo Kikuuwe[1]

Received: 20 January 2025 / Accepted: 14 November 2025

Abstract
This paper proposes a new quadratic programming (QP) based inverse kinematics (IK) method to simultaneously handle physical joint limits and whole-body collision avoidance, including self-collision and collisions with static obstacles. These constraints may conflict with each other and result in infeasible solutions of IK, which can subsequently produce unpredictable motions. The proposed method incorporates an additional linear constraint to ensure that the robot state is *viable*, guaranteeing the existence of solutions that do not violate the given constraints. Our IK method consists of two stages: the offline construction stage and the online computation stage. In the offline construction stage, the parameters of the proposed constraint for ensuring viable states are calculated. In the online computation stage, the proposed constraints are updated in realtime based on the robot's state, and the IK is solved as a QP problem. The proposed method can effectively handle most robots with DOFs below 10 and can also accommodate some robots with higher DOFs under simpler constraints. This marks a significant advancement compared to previous studies. The validity of the proposed method is illustrated through some simulation results.

Keywords Inverse kinematics · Collision avoidance · Quadratic programming · Mixed-integer linear programming · Viability kernel

1 Introduction

The inverse kinematics (IK) is one of the essential and foundational aspects of robotics. It is the process of calculating the corresponding joint angles in the joint space based on the given target position and orientation of the end effector in the task space. A significant challenge in IK is managing the physical joint limits of the robot, specifically the joint range, velocity, and acceleration limits. Pseudoinverse-based IK can strictly satisfy all these physical limits by employing task scaling techniques, which directly modify solutions that exceed the limits [1], or compensate the solution through projection onto the null space [2]. Nevertheless, merely satisfying physical joint limits is typically insufficient to ensure safe motion. Collision avoidance, including the prevention of both self-collisions and collisions with external obstacles, is also essential. Optimization-based IK methods are more suitable than pseudoinverse-based methods for addressing physical joint limits in combination with collision avoidance, as they are capable of explicitly handling multiple inequality constraints. Collision avoidance can be translated into various forms of constraints, such as inequality criteria involving uniquely defined Jacobian matrices [3], differential inequalities for barrier functions [4], and velocity field inequalities at the acceleration level [5].

Unfortunately, optimization-based IK may encounter infeasible solutions, even when it only handles the physical joint limits [6]. For example, when a single joint approaches its boundary at a high velocity, it may be unable to avoid surpassing the boundary under an acceleration limit. Park et al. [6] proposed an approach to estimate the joint angle over several control loops, allowing buffer time for joint deceleration. This method results in stricter bounds and does not guarantee constraint compatibility. To provide formal guarantees of satisfying all the physical joint limits forever, additional constraints are required to ensure that the robot's state remains *viable* [7]. Starting from a viable state guarantees the exis-

✉ Yachen Zhang
yachenzhang@mdl.hiroshima-u.ac.jp

Ryo Kikuuwe
kikuuwe@hiroshima-u.ac.jp

[1] Machinery Dynamics Laboratory, Hiroshima University, Hiroshima, Japan

Published online: 05 January 2026

tence of a sequence of solutions that will satisfy all the constraints in the future. Various constraints to ensure viable states are proposed based on the fact that physical joint limits are expressed as decoupled joint constraints, which independently relate to each joint. Decre et al. [8] predicted the future position trajectory under maximum deceleration to ensure viable states, but their method may lead to conflicts between velocity and acceleration constraints. Rubrecht et al. [9] employed a conservative but safe constraint to ensure viable states to avoid all the constraint conflicts. The optimal constraints were proposed by [10] to obtain the maximum number of viable states. But this method becomes numerically unstable near the static state of robot and requires fine-tuning of the sampling time to maintain stability.

When collision avoidance and physical joint limits are considered together, computing constraints to ensure viable states becomes more challenging. A general method is viability kernel algorithm [11], which involves gridding the state space to find an alternate set of constraints ensuring the viability of the state. Consequently, this algorithm faces the curse of dimensionality, meaning the problem size grows exponentially as the dimensionality of the state space increases. Some researchers have proposed more efficient methods tailored to IK. To avoid collision with obstacles, Rubrecht et al. [9] use the minimum distance between the robot and obstacles to trigger deceleration motion of joints, governed by their aforementioned constraints of physical joint limits. This method was improved in [12] by adding a prediction process with mixable joint deceleration to eliminate potential oscillation during deceleration. Faroni et al. [13] developed time-varying joint bounds to guarantee viable solutions for hierarchical QP-based IK when tracking a known collision-free trajectory. These approaches prevent collisions by dynamically adjusting joint limits, but causing the corresponding set of viable states to vary over time. This adds the complexity of computation of optimization.

This paper proposes a new QP-based IK method to simultaneously handle physical joint limits and whole-body collision avoidance, including self-collision and collisions with static obstacles. In the proposed method, whole-body collision avoidance and joint range limits are approximated as constant, linear and coupled inequalities in joint space, referred to as compound joint constraints. A new additional constraint to ensure the viability is employed to make the whole IK problem computationally tractable. Our IK method consists of two stages: the offline construction stage and the online computation stage. In the offline construction stage, the determination of constant parameters relating to the proposed additional constraint are formalized as mixed-integer linear programming (MILP) and QP problems. In the online computation stage, the proposed constraints are updated in realtime based on the robot's state, and the IK is solved as a simple QP problem.

The contribution of this paper is summarized as follows:

- A new QP-based IK method is proposed to ensure viable solutions while handling both physical joint limits and whole-body collision avoidance.
- Unlike previous studies [9, 12, 13] that realize collision avoidance by adjusting join limits, we integrate both types of constraints into a unified form that yields a static set of viable states. This enables lower computation cost of the optimization.
- For scenarios where only physical joint limits are required, the proposed method guarantees high approximation accuracy while maintaining superior numerical stability compared to [10].
- In scenarios also requiring collision avoidance, the proposed method efficiently determines appropriate parameters to guarantee the solution is viable. It is scalable to robots with at least 10 DOFs, whereas the classical viability kernel algorithm [11] is typically restricted to systems with at most 3 DOFs.

The paper is organized as follows. Section 2 provides mathematical preliminaries and the problem formulation. Section 3 gives the derivation of the constraints to ensure viable states. Section 4 describes the whole procedure of the proposed IK method. Section 5 validates the proposed IK by numerical simulations. Finally, Section 6 draws the concluding remarks.

2 Preliminaries

2.1 Mathematical Preliminaries

Throughout this paper, $\mathbb{R}$ and $\mathbb{Z}$ denote the sets of all real numbers and integers, respectively, $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ denote the sets of all positive and non-negative real numbers, respectively, and $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ denote the sets of all positive and non-negative integers, respectively. Inequalities between vectors are interpreted element-wise. The operator $\odot$ denotes the element-wise multiplication of vectors, and $\mathbf{min}(), \mathbf{max}()$, and $\sqrt{()}$ applied to vectors are read as element-wise functions. The bold-face $\mathbf{0}$ and $\mathbf{1}$ represent the vector of zeros and ones with appropriate dimensions, respectively.

For conciseness, we define the following functions:

$$\mathcal{H}(\boldsymbol{a}) \triangleq \left\{\boldsymbol{x} \in \mathbb{R}^n \mid -\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{a}\right\} \tag{1a}$$

$$\boldsymbol{\beta}(\boldsymbol{v}, \boldsymbol{V}) \triangleq [\beta(v_1, V_1), \cdots, \beta(v_n, V_n)]^T \tag{1b}$$

$$\beta(v, V) \triangleq \begin{cases} \emptyset \text{ if } v > V \\ 1 \text{ if } v = V \\ 0 \text{ if } v < V, \end{cases} \tag{1c}$$

where $\boldsymbol{a} \in \mathbb{R}^n_{\geq 0}$, $\boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{V} \in \mathbb{R}^n$, $v \in \mathbb{R}$, and $V \in \mathbb{R}_{\geq 0}$. With a positive definite matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, the weighted Euclidean norm $\|z\|_{\boldsymbol{W}}$ of $z \in \mathbb{R}^n$ is defined as $\|z\|_{\boldsymbol{W}} \triangleq \sqrt{z^T \boldsymbol{W} z}$.

This paper deals with a second-order system whose state and input are $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in \mathbb{R}^{2n}$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$, respectively, where $\boldsymbol{q} \in \mathbb{R}^n$. To discuss the problems in the discrete-time domain, we assume that the discrete-time counterparts of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ have the following relations:

$$\boldsymbol{q}^k = \boldsymbol{q}^{k-1} + T(\dot{\boldsymbol{q}}^{k-1} + \dot{\boldsymbol{q}}^k)/2 \tag{2a}$$
$$\dot{\boldsymbol{q}}^k = \dot{\boldsymbol{q}}^{k-1} + T\ddot{\boldsymbol{q}}^k \tag{2b}$$

where $k \in \mathbb{Z}_{\geq 0}$ denotes the discrete-time index and T is the sampling interval. We also define the following map $\mathcal{F}$ based on (2a):

$$\mathcal{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathcal{C}) \triangleq \left\{ \dot{\boldsymbol{q}}^* \;\middle|\; \begin{bmatrix} \boldsymbol{q} + T(\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^*)/2 \\ \dot{\boldsymbol{q}}^* \end{bmatrix} \in \mathcal{C} \right\}. \tag{3}$$

Here, $\mathcal{C}$ can be a subset of the state space $\mathbb{R}^{2n}$. The following relations can be given due to the definition (3) of $\mathcal{F}$:

$$(2a) \wedge \dot{\boldsymbol{q}}^k \in \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}) \Rightarrow [(\boldsymbol{q}^k)^T, (\dot{\boldsymbol{q}}^k)^T]^T \in \mathcal{C} \tag{4a}$$
$$\dot{\boldsymbol{q}}^k \in \left(\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_1) \cap \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_2)\right) \Leftrightarrow \dot{\boldsymbol{q}}^k \in \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_1 \cap \mathcal{C}_2) \tag{4b}$$
$$\dot{\boldsymbol{q}}^k \in \left(\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_1) \cup \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_2)\right) \Leftrightarrow \dot{\boldsymbol{q}}^k \in \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C}_1 \cup \mathcal{C}_2) \tag{4c}$$

As revealed by (4a), $\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C})$ provides all possible velocities $\dot{\boldsymbol{q}}^k$ for states within $\mathcal{C}$ that are reached from $[(\boldsymbol{q}^{k-1})^T, (\dot{\boldsymbol{q}}^{k-1})^T]^T$.

By observing the definition (3) of $\mathcal{F}$, one can see that when a set $\mathcal{C}$ and a constant $\hat{\boldsymbol{q}} \in \mathbb{R}^n$ are given, all the states $[(\boldsymbol{q}^{k-1})^T, (\dot{\boldsymbol{q}}^{k-1})^T]^T$ satisfying

$$\hat{\boldsymbol{q}} = \boldsymbol{q}^{k-1} + T\dot{\boldsymbol{q}}^{k-1}/2 \tag{5}$$

will cause the same $\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C})$. From a geometric perspective, (5) indicates the intersection of hyperplanes in $\mathbb{R}^{2n}$ state space. On the other hand, obtaining $\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{C})$ can be achieved by substituting (2a) into the expression for $\mathcal{C}$ and solving it. Geometrically, this corresponds to finding the intersection point between the hypersurface of the boundary of $\mathcal{C}$ and the hyperplanes associated with

$$\dot{\boldsymbol{q}}^k = (2/T)(\boldsymbol{q}^k - \hat{\boldsymbol{q}}), \tag{6}$$

which is obtained by replacing $\boldsymbol{q}^{k-1}$ and $\dot{\boldsymbol{q}}^{k-1}$ in (2a) with $\hat{\boldsymbol{q}}$ based on (5). Therefore, when $\hat{\boldsymbol{q}}$ is given, every state $[(\boldsymbol{q}^{k-1})^T, (\dot{\boldsymbol{q}}^{k-1})^T]^T$ satisfying (5) will reach the state satisfying (6) in the next timestep. For a one-DOF system, graphically obtaining $\mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{C})$ can be described as follows, which is also illustrated in Fig. 1:

1. Drawing a line with the slope of $-2/T$ through $(q^{k-1}, \dot{q}^{k-1})$ and intersecting it with the horizontal axis to obtain $(\hat{q}, 0)$. This line corresponds to (5).
2. According to (6), another line with the slope of $2/T$ can be plotted by passing through the obtained point $(\hat{q}, 0)$.
3. Intersecting the line obtained in the previous step with the boundary of $\mathcal{C}$ to obtain all the intersection points $\cdots, (x_i, y_i), \cdots$. Then, $\mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{C})$ is determined by the vertical coordinates y_i of these intersection points.

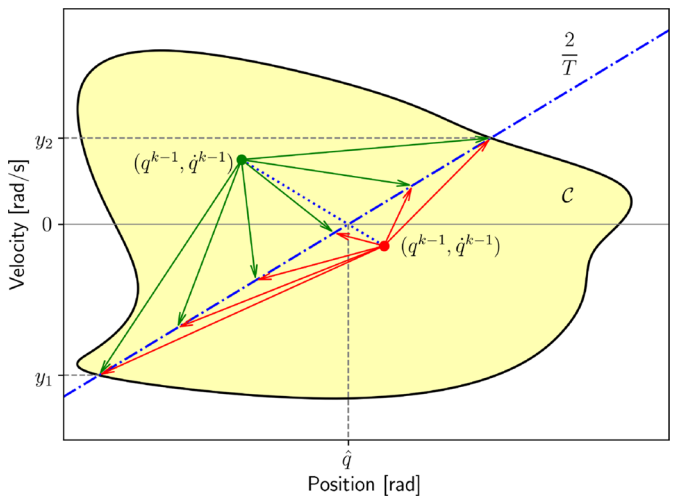

Fig. 1 Illustration of obtaining $\mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{C})$, where $[q^{k-1}, \dot{q}^{k-1}]^T \in \mathbb{R}^2$. The yellow area represents $\mathcal{C}$, while the blue dotted and dash-dotted lines correspond to (5) and (6), respectively. In the case of this figure, the vertical coordinates of intersection points between the blue dash-dotted line and the boundary of $\mathcal{C}$ are y_1 and y_2, which determine $\mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{C})$ as $\{\dot{q}^k | y_1 \leq \dot{q}^k \leq y_2\}$

2.2 Viability Theory

Viability theory [7] is an area of mathematics to study dynamical systems under various constraints that are imposed on their state and inputs at every moment. In this paper, the state and the input are subject to the constraint $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in \mathcal{X} \wedge \ddot{\boldsymbol{q}} \in \mathcal{A}$ where $\mathcal{X}$ is a subset of the state space and $\mathcal{A}$ is a subset of the input space. According to the viability theory, a state $[(\boldsymbol{q}^*)^T, (\dot{\boldsymbol{q}}^*)^T]^T$ is said to be *viable* if there exists a temporal pattern of $\ddot{\boldsymbol{q}} \in \mathcal{A}$ that keeps the state within $\mathcal{X}$ forever once it starts from $[(\boldsymbol{q}^*)^T, (\dot{\boldsymbol{q}}^*)^T]^T$. The set of all viable states can be formally written as

$$\mathcal{K}_{\mathrm{c}}(\mathcal{A}, \mathcal{X}) \triangleq \left\{ \begin{bmatrix} \boldsymbol{q}^* \\ \dot{\boldsymbol{q}}^* \end{bmatrix} \in \mathcal{X} \;\middle|\; \exists \ddot{\boldsymbol{q}} : \mathbb{R}_{\geq 0} \to \mathcal{A} \text{ s.t.} \right.$$
$$\left. \begin{bmatrix} \boldsymbol{q}(0) \\ \dot{\boldsymbol{q}}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}^* \\ \dot{\boldsymbol{q}}^* \end{bmatrix} \wedge \begin{bmatrix} \boldsymbol{q}(t) \\ \dot{\boldsymbol{q}}(t) \end{bmatrix} \in \mathcal{X} \; \forall t \in \mathbb{R}_{\geq 0} \right\}. \tag{7}$$

Here, $\mathcal{K}_{\mathrm{c}}(\mathcal{A}, \mathcal{X})$ can be said to be the *viability kernel* of the second-order integrator on the set $\mathcal{X}$ under the constraint $\ddot{\boldsymbol{q}} \in \mathcal{A}$. Based on (2), we can define a discrete-time analogue $\mathcal{K}$ of the viability kernel $\mathcal{K}_{\mathrm{c}}$ as follows:

$$\mathcal{K}(\mathcal{A}, \mathcal{X}) \triangleq \left\{ \begin{bmatrix} \boldsymbol{q}^* \\ \dot{\boldsymbol{q}}^* \end{bmatrix} \in \mathcal{X} \mid \exists \{\ddot{\boldsymbol{q}}^k \in \mathcal{A}\}_{k \in \mathbb{Z}_{\geq 0}} \text{ s.t.} \right.$$
$$\left. \begin{bmatrix} \boldsymbol{q}^0 \\ \dot{\boldsymbol{q}}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}^* \\ \dot{\boldsymbol{q}}^* \end{bmatrix} \wedge (2) \wedge \begin{bmatrix} \boldsymbol{q}^k \\ \dot{\boldsymbol{q}}^k \end{bmatrix} \in \mathcal{X} \; \forall k \in \mathbb{Z}_{\geq 0} \right\}. \tag{8}$$

It should be noted that $\mathcal{K}_{\mathrm{c}}(\mathcal{A}, \mathcal{X}) \subseteq \mathcal{X}$ and $\mathcal{K}(\mathcal{A}, \mathcal{X}) \subseteq \mathcal{X}$ are always satisfied by definition.

The viability kernel $\mathcal{K}(\mathcal{A}, \mathcal{X})$ is commonly challenging to obtain. One approach is to compute a conservative approximation (a subset) of $\mathcal{K}(\mathcal{A}, \mathcal{X})$. We consider using another set $\mathcal{V}$ to trim $\mathcal{X}$ so that the set $\mathcal{V} \cap \mathcal{X}$ is a subset of $\mathcal{K}(\mathcal{A}, \mathcal{X})$, as illustrated in Fig. 2. We name such a set $\mathcal{V}$, which satisfies

$$\mathcal{V} \cap \mathcal{X} \subseteq \mathcal{K}(\mathcal{A}, \mathcal{X}) \subseteq \mathcal{X}, \tag{9}$$

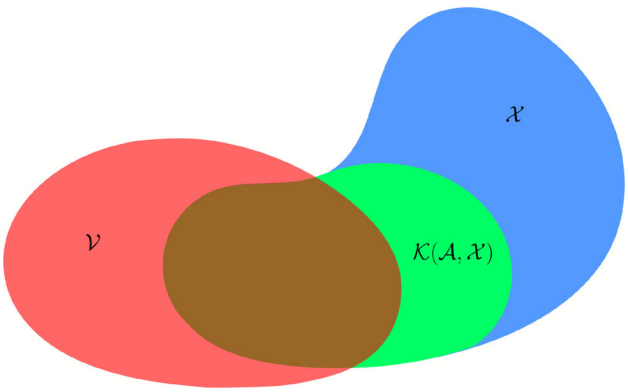

Fig. 2 Illustration of the subset $\mathcal{X}$ of the state space, the viability kernel $\mathcal{K}(\mathcal{A}, \mathcal{X})$ and an auxiliary viability set $\mathcal{V}$

as an *auxiliary viability set* corresponding to $\mathcal{X}$ and $\mathcal{A}$. There can be various methods to derive an auxiliary viability set. In this paper, we consider a set $\mathcal{V}^*$ whose representation includes unknown parameters. These parameters are then determined on the basis of the following lemma to ensure that $\mathcal{V}^*$ serves as an auxiliary viability set:

Lemma 1 *Consider the discrete-time system* (2) *and subsets* $\mathcal{A} \subseteq \mathbb{R}^n$, $\mathcal{X} \subseteq \mathbb{R}^{2n}$, *and* $\mathcal{V}^* \subseteq \mathbb{R}^{2n}$. *Assume that the following condition holds:*

$$\forall [\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in (\mathcal{V}^* \cap \mathcal{X}), \; \exists \ddot{\boldsymbol{q}} \in \mathcal{A} \tag{10a}$$
$$s.t. \;\; (\dot{\boldsymbol{q}} + T\ddot{\boldsymbol{q}}) \in \mathcal{F}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathcal{V}^*\right) \tag{10b}$$
$$\wedge \, (\dot{\boldsymbol{q}} + T\ddot{\boldsymbol{q}}) \in \mathcal{F}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathcal{X}\right), \tag{10c}$$

then $\mathcal{V}^*$ *serves as an auxiliary viability set under* $\mathcal{X}$ *and* $\mathcal{A}$.

Proof According to (2), (4a) and (4b), if (10) is satisfied, then $\forall k \in \mathbb{Z}_{>0}$ and $\forall [(\boldsymbol{q}^{k-1})^T, (\dot{\boldsymbol{q}}^{k-1})^T]^T \in (\mathcal{V}^* \cap \mathcal{X})$, there exists $\ddot{\boldsymbol{q}}^k \in \mathcal{A}$ such that $[(\boldsymbol{q}^k)^T, (\dot{\boldsymbol{q}}^k)^T]^T \in (\mathcal{V}^* \cap \mathcal{X})$. Thus, every state in $\mathcal{V}^* \cap \mathcal{X}$ is viable as indicated by (8), which implies $\mathcal{V}^* \cap \mathcal{X} \subseteq \mathcal{K}(\mathcal{A}, \mathcal{X})$. This confirms that $\mathcal{V}^*$ serves as an auxiliary viability set under $\mathcal{X}$ and $\mathcal{A}$ due to (9). □

2.3 Problem Formulation

Consider a robot with n DOFs, and let $\boldsymbol{q} \in \mathbb{R}^n$ be the joint variable vector of the robot. The position and orientation in task space are represented by $\boldsymbol{p}(\boldsymbol{q}) \in \mathbb{R}^3$ and $\boldsymbol{R}(\boldsymbol{q}) \in \mathbb{R}^{3\times 3}$, respectively, both of which can be obtained via forward kinematics from $\boldsymbol{q}$. The desired position and orientation are denoted by $\boldsymbol{p}_d \in \mathbb{R}^3$ and $\boldsymbol{R}_d \in \mathbb{R}^{3\times 3}$. We assume that $\boldsymbol{p}_d$, $\boldsymbol{R}_d$ and $\boldsymbol{q}$ are functions of time t.

In general, a robot may be required to simultaneously satisfy multiple task-space objectives. For example, the positions and orientations of multiple frames attached to different links, such as the end-effectors, cameras, and force sensors, may need to track corresponding desired values. Let $\boldsymbol{p}_i(\boldsymbol{q})$ and $\boldsymbol{R}_i(\boldsymbol{q})$ denote the position and orientation of the i-th task-space frame, and let $\boldsymbol{p}_{di}$ and $\boldsymbol{R}_{di}$ denote their desired values.

We consider the problem of determining $\boldsymbol{q}$ or either of its derivatives, such that a set of task-space variables $\{\cdots, \boldsymbol{p}_i(\boldsymbol{q}), \boldsymbol{R}_i(\boldsymbol{q}), \cdots\}$ closely track their respective desired values under various constraints. The *pure* inverse kinematics (IK) problem is the problem to solve the following nonlinear equation with respect to $\boldsymbol{q}$:

$$\boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{0}, \tag{11}$$

where the track error $\boldsymbol{e}(\boldsymbol{q}) \in \mathbb{R}^l$ is defined as $\boldsymbol{e}(\boldsymbol{q}) \triangleq [\cdots, \boldsymbol{e}_i(\boldsymbol{q}), \cdots]^T$, with each component $\boldsymbol{e}_i(\boldsymbol{q}) \in \mathbb{R}^3$ given

by

$$\boldsymbol{e}_i(\boldsymbol{q}) = \begin{cases} \boldsymbol{p}_{di} - \boldsymbol{p}_i(\boldsymbol{q}) & \text{for position variable} \\ (\ln \boldsymbol{R}_{di} \boldsymbol{R}_i^T(\boldsymbol{q}))^\vee & \text{for orientation variable.} \end{cases} \tag{12}$$

Here, the notation $(\ln \boldsymbol{R}_a \boldsymbol{R}_b^T)^\vee$ represents the rotation vector from the attitude $\boldsymbol{R}_b$ to the attitude $\boldsymbol{R}_a$, which is explained in detail in Section 2.2.7 of [14] .

Directly solving the pure IK (11) at every moment, however, may be practically inconvenient because (11) may not have solutions when it is infeasible to achieve all desired values $\boldsymbol{p}_{di}$ and $\boldsymbol{R}_{di}$. Moreover, even if the pure IK (11) has a solution $\boldsymbol{q}$, it may be outside the physical motion limits of the joints, which can be described in the following form:

$$\boldsymbol{q}^{\min} \le \boldsymbol{q} \le \boldsymbol{q}^{\max} \tag{13}$$

where $\boldsymbol{q}^{\min} \in \mathbb{R}^n$ and $\boldsymbol{q}^{\max} \in \mathbb{R}^n$. In addition, there may be other various geometric constraints on the robot, including contact with the external environment and self-collisions. Such geometric constraints can generally be written in the following form:

$$g(\boldsymbol{q}) \ge 0. \tag{14}$$

Here, $g : \mathbb{R}^n \to \mathbb{R}$ is a function that returns the minimum of the distances from all potential collisions, which is negative if there is a collision or a penetration.

In order to track desired values, we also need to care about the joint velocity and joint acceleration, which should satisfy constraints of the following form:

$$\dot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{v}^{\lim}) \tag{15a}$$

$$\ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{15b}$$

where $\boldsymbol{v}^{\lim} \in \mathbb{R}^n_{>0}$ and $\boldsymbol{a}^{\lim} \in \mathbb{R}^n_{>0}$ are the vectors of velocity and acceleration limits, respectively. Moreover, the state $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T$ should be kept viable, in the sense that it should remain satisfying the constraint (13)∧(14)∧(15). The viability condition can be written as follows:

$$[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in \mathcal{K}_{\mathrm{c}}\left(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X}\right) \tag{16}$$

where

$$\mathcal{X} \triangleq \{[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \,|\, (13) \wedge (14) \wedge (15a)\}. \tag{17}$$

Because the condition (16) implies (13)∧(14)∧(15a), we can see that (15b) ∧ (16) is the constraint that should be imposed on $\boldsymbol{q}$ and its time derivatives.

Under these constraints, we consider determining $\boldsymbol{q}$ and its derivatives so that they satisfy the following:

$$\tau \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} - \boldsymbol{e}(\boldsymbol{q}) \approx 0 \tag{18}$$

where $\boldsymbol{J}(\boldsymbol{q}) \triangleq -\partial \boldsymbol{e}(\boldsymbol{q})/\partial \boldsymbol{q} \in \mathbb{R}^{l \times n}$. As long as (18) is satisfied, it becomes $\tau \dot{\boldsymbol{e}}(\boldsymbol{q}) + \boldsymbol{e}(\boldsymbol{q}) \approx \boldsymbol{0}$, which means that the track error $\boldsymbol{e}(\boldsymbol{q})$ would converge to $\boldsymbol{0}$ with the lag of the time constant τ. We refer to this problem also as an IK problem, in the same light as previous studies such as [15, 16].

In the discrete-time domain, we assume that $\boldsymbol{q}$ and its derivatives are connected with one another by (2). Then, we need to obtain $\dot{\boldsymbol{q}}^k$ that satisfies (18) under the constraints (15b) and (16), whose discrete-time counterparts are

$$(\dot{\boldsymbol{q}}^k - \dot{\boldsymbol{q}}^{k-1})/T \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{19a}$$

$$[(\boldsymbol{q}^k)^T, (\dot{\boldsymbol{q}}^k)^T]^T \in \mathcal{K}\left(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X}\right). \tag{19b}$$

That is, the IK can be formulated as the following optimization problem:

minimize

$$\dot{\boldsymbol{q}}^k \ \|\boldsymbol{J}^{k-1}\dot{\boldsymbol{q}}^k - \boldsymbol{b}^{k-1}\|^2_{\boldsymbol{W}_1} + \|\dot{\boldsymbol{q}}^k\|^2_{\boldsymbol{W}_2} \tag{20a}$$

$$\text{s.t. } \dot{\boldsymbol{q}}^k \in \mathcal{F}\left(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{K}(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X})\right) \tag{20b}$$

$$(\dot{\boldsymbol{q}}^k - \dot{\boldsymbol{q}}^{k-1})/T \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{20c}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{l \times l}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{n \times n}$ are symmetric positive definite matrices and

$$\boldsymbol{J}^{k-1} \triangleq \boldsymbol{J}(\boldsymbol{q}^{k-1}) \in \mathbb{R}^{l \times n} \tag{21a}$$

$$\boldsymbol{b}^{k-1} \triangleq \boldsymbol{e}(\boldsymbol{q}^{k-1})/\tau \in \mathbb{R}^l. \tag{21b}$$

Here, the second term of (20a) is intended to reduce the velocity $\dot{\boldsymbol{q}}^k$ if some redundancy remains even after the first term becomes small enough, and (20b) ensures that (19b) is met according to (4a).

The problem (20) is not convenient because it involves the nonconvex constraint (14) in $\mathcal{X}$. Therefore, this paper approximates the constraint (13) ∧ (14) by a compound joint constraint as

$$\boldsymbol{A}\boldsymbol{q} \le \tilde{\boldsymbol{q}}^{\lim}, \tag{22}$$

which ensures $\{\boldsymbol{q} \,|\, (22)\} \subseteq \{\boldsymbol{q} \,|\, (13) \wedge (14)\}$. Here, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\tilde{\boldsymbol{q}}^{\lim} \in \mathbb{R}^m$. Such an approximation can be derived by approaches proposed in, e.g., [17] and [18]. Based on the linear approximation (22) of the nonconvex constraint (13) ∧ (14), now we approximate the set $\mathcal{X}$ by $\mathcal{X}^*$ defined as follows:

$$\mathcal{X}^* \triangleq \{[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \,|\, (22) \wedge (15a)\} \subseteq \mathcal{X}. \tag{23}$$

In order to deal with the set $\mathcal{K}(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X})$ appearing in (20b), which is not easily obtained, we consider replacing it by a conservative approximation using the auxiliary viability set $\mathcal{V}$ introduced in Section 2.2. The combination of these approximation results in the following:

$$\mathcal{V} \cap \mathcal{X}^* \subseteq \mathcal{K}(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X}^*) \subseteq \mathcal{K}(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X}). \tag{24}$$

Thus, the original IK problem (20) can be approximated in the following form:

$$\underset{\dot{\boldsymbol{q}}^k}{\text{minimize}} \ \|\boldsymbol{J}^{k-1}\dot{\boldsymbol{q}}^k - \boldsymbol{b}^{k-1}\|^2_{\boldsymbol{W}_1} + \|\dot{\boldsymbol{q}}^k\|^2_{\boldsymbol{W}_2} \tag{25a}$$

$$\text{s.t. } \dot{\boldsymbol{q}}^k \in \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{V}) \tag{25b}$$

$$\dot{\boldsymbol{q}}^k \in \mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{X}^*) \tag{25c}$$

$$(\dot{\boldsymbol{q}}^k - \dot{\boldsymbol{q}}^{k-1})/T \in \mathcal{H}(\boldsymbol{a}^{\lim}). \tag{25d}$$

Here, (25c) is a linear constraint of $\dot{\boldsymbol{q}}^k$ and (25b) $\wedge$ (25c) ensures that the obtained state $[(\boldsymbol{q}^k)^T, (\dot{\boldsymbol{q}}^k)^T]^T$ is within $\mathcal{K}(\mathcal{H}(\boldsymbol{a}^{\lim}), \mathcal{X})$ due to (4b) and (24). The remaining problem, which will be discussed in the subsequent sections, is to derive an appropriate $\mathcal{V}$ that transforms the problem (25) into a QP problem.

3 Derivation of Auxiliary Viability Set

3.1 Auxiliary Viability Set for One-dimensional Position Constraint

This section discusses the simplest case that includes a single joint subject to angle, velocity, and acceleration constraints, which can be written as follows:

$$q \le q^{\lim} \tag{26a}$$

$$\dot{q} \in \mathcal{H}(v^{\lim}) \tag{26b}$$

$$\ddot{q} \in \mathcal{H}(a^{\lim}). \tag{26c}$$

Here, $q^{\lim} \in \mathbb{R}$, $v^{\lim} \in \mathbb{R}_{>0}$, and $a^{\lim} \in \mathbb{R}_{>0}$.

The viability kernel of this discrete system can be written as $\mathcal{K}\left(\mathcal{H}(a^{\lim}), \{(26a)\wedge(26b)\}\right)$, and, because of the simplicity of the system, it can be analytically obtained as follows:

$$\mathcal{K}_p \triangleq \mathcal{K}\left(\mathcal{H}(a^{\lim}), \{(26a)\wedge(26b)\}\right) = \mathcal{V}_p(q^{\lim}, a^{\lim}) \cap \{(26a)\wedge(26b)\} \tag{27}$$

where

$$\mathcal{V}_p(q^{\lim}, a^{\lim}) \triangleq \left\{ [q, \dot{q}]^T \,\middle|\, \dot{q} \le \sqrt{2a^{\lim}\max(0, q^{\lim} - q)} \right\}. \tag{28}$$

This expression is derived based on the requirement that the angle q must stop with full deceleration (i.e., $\ddot{q} = -a^{\lim}$) to avoid exceeding the limit $q^{\lim}$. It should be emphasized that the set $\mathcal{V}_p(q^{\lim}, a^{\lim})$ serves as an auxiliary viability set due to (27). The illustration of $\mathcal{K}_p$ and $\mathcal{V}_p(q^{\lim}, a^{\lim})$ are shown in Fig. 3(a).

Since $\mathcal{V}_p(q^{\lim}, a^{\lim})$ serves as an auxiliary viability set under the constraints (26), it can be applied in the IK problem (25) for a system constrained solely by physical joint limits (13) $\wedge$ (15), which can be decoupled into the combination of (26). In this case, each component of the IK solution must satisfy

$$\dot{q}^k \in \mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{V}_p(q^{\lim}, a^{\lim})) \tag{29a}$$

$$\dot{q}^k \in \mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \{(26a)\wedge(26b)\}), \tag{29b}$$

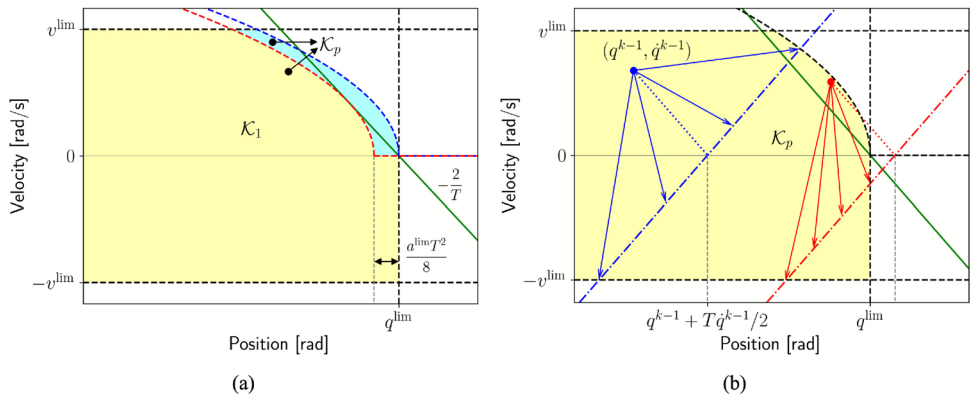

Fig. 3 (**a**) The illustration of $\mathcal{K}_p$ and $\mathcal{K}_1$. The yellow area represents $\mathcal{K}_1$, and the cyan area combining yellow area indicates $\mathcal{K}_p$. The dashed lines of different colors express the different boundaries of sets, with black indicating $\{(26a)\wedge(26b)\}$, blue indicating $\mathcal{V}_p(q^{\lim}, a^{\lim})$, and red indicating $\mathcal{V}_1(q^{\lim}, a^{\lim})$, respectively. (**b**) A state satisfying $q^{k-1} + T\dot{q}^{k-1}/2 \le q^{\lim}$ (indicated in blue) can come to a stop at the next timestep (i.e., $\dot{q}^k = 0$) when required, while a state with $q^{k-1} + T\dot{q}^{k-1}/2 > q^{\lim}$ (indicated in red) cannot. The yellow area expresses $\mathcal{K}_p$. The green line in (**a**) and (**b**) corresponds to the boundary of $\mathcal{B}$ defined in (33)

which are derived according to (25b) and (25c), respectively. According to the definition (3) of $\mathcal{F}$, the two conditions can be obtained as

$$\dot{q}^k \le \sqrt{2a^{\lim} \max(0, q^{\lim} - (q^{k-1} + T(q^{k-1} + \dot{q}^k)/2)} \tag{30a}$$

$$\dot{q}^k \le (2/T)(q^{\lim} - q^{k-1} - T\dot{q}^{k-1}/2) \tag{30b}$$

$$\dot{q}^k \in \mathcal{H}(v^{\lim}). \tag{30c}$$

Here, (30a) and (30b) $\wedge$ (30c) are derived from (29a) and (29b), respectively, and (30a) can be solved analytically according to Appendix A:

$$\dot{q}^k \le g_p(q^{k-1}, \dot{q}^{k-1}, q^{\lim}, a^{\lim}), \tag{31}$$

where $g_p : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}_{\ge 0}$ is defined as

$$g_p(q, \dot{q}, q^{\lim}, a^{\lim}) \triangleq \left(-a^{\lim}T + \sqrt{(a^{\lim}T)^2 + 8a^{\lim}\max(0, q^{\lim} - q - T\dot{q}/2)}\right)/2. \tag{32}$$

Here, the output of g_p is always nonnegative due to the term of $\max(0, *)$. Please notice that (30b), (30c) and (31) are linear inequalities of $\dot{q}^k$, thereby transforming the problem (25) into a QP problem.

Unfortunately, applying (30b), (30c) and (31) in (25) may lead to a numerical issue. A state $[q^{k-1}, \dot{q}^{k-1}]^T$ belonging to the following set always causes (30b) into $\dot{q}^k < 0$:

$$\mathcal{B} \triangleq \left\{ [q, \dot{q}]^T \,|\, q + T\dot{q}/2 > q^{\lim} \right\}, \tag{33}$$

where the boundary of $\mathcal{B}$ is a line with the slope of $-2/T$ passing through $(q^{\lim}, 0)$, as shown in Fig. 3(a) and (b). Therefore, a state $[q^{k-1}, \dot{q}^{k-1}]^T$ within $\mathcal{B}$ cannot come to a stop at the next timestep (i.e., $\dot{q}^k = 0$) when required, which can be graphically shown in Fig. 3(b) based on the discussion in Section 2.2. This may cause the obtained solution to oscillate around $q^{\lim}$, which is also reported in [10], and is illustrated in Section 5.1.

To address this issue, one approach is to ensure that the states will never belong to $\mathcal{B}$. For this purpose, we define a new set by shifting $\mathcal{V}_p(q^{\lim}, a^{\lim})$ to the left by $T^2a^{\lim}/8$, which is given by

$$\mathcal{V}_1(q^{\lim}, a^{\lim}) \triangleq \mathcal{V}_p(q^{\lim} - T^2a^{\lim}/8, a^{\lim}). \tag{34}$$

As shown in Fig. 3(a), the new set is still an auxiliary viability set due to $\mathcal{K}_1 \subseteq \mathcal{K}_p$, where

$$\mathcal{K}_1 \triangleq \mathcal{V}_1(q^{\lim}, a^{\lim}) \cap \{(26a) \wedge (26b)\}. \tag{35}$$

The boundary of $\mathcal{B}$ is tangent to the boundary of $\mathcal{V}_1(q^{\lim}, a^{\lim})$, ensuring that $\mathcal{K}_1 \cap \mathcal{B} = \emptyset$. Consequently, $\mathcal{V}_1(q^{\lim}, a^{\lim})$ is the closest shifted set derived from $\mathcal{V}_p(q^{\lim}, a^{\lim})$ that can resolve the aforementioned issue. Since the shift magnitude $T^2a^{\lim}/8$ is very small in practice, $\mathcal{K}_1$ typically serves as a close approximation to $\mathcal{K}_p$. The corresponding condition instead of (31) is given by

$$\dot{q}^k \le g_1(q^{k-1}, \dot{q}^{k-1}, q^{\lim}, a^{\lim}) \tag{36}$$

where $g_1 : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}_{\ge 0}$ is defined as

$$g_1(q^{k-1}, \dot{q}^{k-1}q^{\lim}, a^{\lim}) \triangleq g_p(q^{k-1}, \dot{q}^{k-1}, q^{\lim} - T^2a^{\lim}/8, a^{\lim}). \tag{37}$$

In conclusion, (25b) $\wedge$ (25c) corresponding to $\mathcal{K}_1$ can be obtained as

$$\dot{q}^k \in \mathcal{F}(q^{k-1}, \dot{q}^{k-1}, \mathcal{V}_1(q^{\lim}, a^{\lim})) \wedge (29b) \tag{38a}$$

$$\iff (36) \wedge ((30b) \wedge (30c)) \tag{38b}$$

$$\iff (36) \wedge (30c). \tag{38c}$$

Here, (38b) can be simplified to (38c) as proved in Appendix B, under the condition $q^{\lim} - q^{k-1} - T\dot{q}^{k-1}/2 \ge 0$. This condition always holds when $[q^{k-1}, \dot{q}^{k-1}]^T \in \mathcal{K}_1$ because of $\mathcal{K}_1 \cap \mathcal{B} = \emptyset$.

3.2 Auxiliary Viability Set for Single Row of Compound Joint Constraint

In this section, we consider (15) combining one row of the compound joint constraint (22) as the hard joint constraints:

$$\boldsymbol{c}^T\boldsymbol{q} \le \tilde{q}^{\lim} \tag{39a}$$

$$\dot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{v}^{\lim}) \tag{39b}$$

$$\ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}). \tag{39c}$$

Here, $\boldsymbol{c} \in \mathbb{R}^n$ and $\tilde{q}^{\lim} \in \mathbb{R}$. Note that (39a) can be interpreted as the one-dimensional constraint (26a) because it actually limits the scalar $\boldsymbol{c}^T\boldsymbol{q}$. Based on this fact, a new set in $\mathbb{R}^{2n}$ can be constructed by extending $\mathcal{V}_1(\tilde{q}^{\lim}, \tilde{a}^{\lim})$ in Section 3.1 as follows:

$$\mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim}) \triangleq \{[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \,|\, [\boldsymbol{c}^T\boldsymbol{q}, \boldsymbol{c}^T\dot{\boldsymbol{q}}]^T \in \mathcal{V}_1(\tilde{q}^{\lim}, \tilde{a}^{\lim})\}. \tag{40}$$

Here, $\tilde{a}^{\lim} \in \mathbb{R}_{>0}$ is a constant to be determined and $\mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim})$ is the set of states ensuring that $\boldsymbol{c}^T\dot{\boldsymbol{q}}$ can reduce to 0 before $\boldsymbol{c}^T\boldsymbol{q}$ reaching $\tilde{q}^{\lim} - \tilde{a}^{\lim}T^2/8$ under $\boldsymbol{c}^T\ddot{\boldsymbol{q}} = -\tilde{a}^{\lim}$. In other words, (39a) and $\mathcal{V}_2$ can be derived from (26a) and $\mathcal{V}_1$ by substituting q, $\dot{q}$, $q^{\lim}$ and $a^{\lim}$ with

$\boldsymbol{c}^T\boldsymbol{q}$, $\boldsymbol{c}^T\dot{\boldsymbol{q}}$, $\tilde{q}^{\lim}$ and $\tilde{a}^{\lim}$, respectively. Thus, applying $\mathcal{F}$ for $\mathcal{V}_2$ and (39a) can be derived as follows by referencing the derivation of (36) and (30b), respectively:

$$\boldsymbol{c}^T\dot{\boldsymbol{q}}^k \le g_1(\boldsymbol{c}^T\boldsymbol{q}^{k-1}, \boldsymbol{c}^T\dot{\boldsymbol{q}}^{k-1}, \tilde{q}^{\lim}, \tilde{a}^{\lim}) \tag{41a}$$

$$\boldsymbol{c}^T\dot{\boldsymbol{q}}^k \le (2/T)(\tilde{q}^{\lim} - \boldsymbol{c}^T\boldsymbol{q}^{k-1} - T\boldsymbol{c}^T\dot{\boldsymbol{q}}^{k-1}/2). \tag{41b}$$

Please notice that (41a) $\wedge$ (41b) can be simplified to (41a) for the same reasons as (38). Finally, the constraints corresponding to $\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim}) \cap \{(39a) \wedge (39b)\})$ can be given by

$$(41a) \wedge \dot{\boldsymbol{q}}^k \in \mathcal{H}(\boldsymbol{v}^{\lim}). \tag{42}$$

To apply (42) in IK (25), $\tilde{a}^{\lim}$ must be determined to ensure that $\mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim})$ serves as an auxiliary viability set under (39). This requires the following condition to be satisfied, as stated in Lemma 1:

$$\forall [\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in (\mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim}) \cap \{(39a) \wedge (39b)\}),\ \exists \ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{43a}$$

$$\text{s.t. } \boldsymbol{c}^T\ddot{\boldsymbol{q}} \le (g_1(\boldsymbol{c}^T\boldsymbol{q}, \boldsymbol{c}^T\dot{\boldsymbol{q}}, \tilde{q}^{\lim}, \tilde{a}^{\lim}) - \boldsymbol{c}^T\dot{\boldsymbol{q}})/T \tag{43b}$$

$$\wedge\ (-\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \le \ddot{\boldsymbol{q}} \le (\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T. \tag{43c}$$

Here, (43b) $\wedge$ (43c) is obtained by substituting (2b) into (42). Satisfaction of the above condition requires verifying the existence of $\ddot{\boldsymbol{q}}$ by traversing every $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T$ specified in (43a). This traversal can be represented as a two-level nested loop: the outer loop iterates over $\dot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{v}^{\lim})$, while the inner loop traverses each $\boldsymbol{q}$ that satisfies

$$\boldsymbol{c}^T\boldsymbol{q} \le \tilde{q}^{\lim} \wedge [\boldsymbol{c}^T\boldsymbol{q}, \boldsymbol{c}^T\dot{\boldsymbol{q}}]^T \in \mathcal{V}_1(\tilde{q}^{\lim}, \tilde{a}^{\lim}). \tag{44}$$

Here, the second condition in (44) is rewritten from $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in \mathcal{V}_2(\tilde{q}^{\lim}, \tilde{a}^{\lim})$ according to the definition of $\mathcal{V}_2$.

Due to the complexity of this nested loop, it needs to be equally simplified. Since $\dot{\boldsymbol{q}}$ is within $\mathcal{H}(\boldsymbol{v}^{\lim})$ in the outer loop, the left-hand side and right-hand side of (43c) are always nonpositive and nonnegative, respectively. When $\boldsymbol{c}^T\dot{\boldsymbol{q}} < 0$, the right-hand side of (43b) will always be positive because the output of g_1 is nonnegative. This implies that (43) is always satisfied in the case of $\ddot{\boldsymbol{q}} = \boldsymbol{0}$ when $\boldsymbol{c}^T\dot{\boldsymbol{q}} < 0$. Thus, the outer loop only need to iterate over $\dot{\boldsymbol{q}} \in \mathcal{D}$ to verify the existence of $\ddot{\boldsymbol{q}}$, where

$$\mathcal{D} \triangleq \{\dot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{v}^{\lim}) |\ \boldsymbol{c}^T\dot{\boldsymbol{q}} \ge 0\}. \tag{45}$$

Furthermore, as shown in Appendix C, the following condition holds for all $\boldsymbol{q}$ satisfying (44) when $\boldsymbol{c}^T\dot{\boldsymbol{q}} \ge 0$:

$$g_1(\boldsymbol{c}^T\boldsymbol{q}, \boldsymbol{c}^T\dot{\boldsymbol{q}}, \tilde{q}^{\lim}, \tilde{a}^{\lim}) \ge \max(0, \boldsymbol{c}^T\dot{\boldsymbol{q}} - \tilde{a}^{\lim}T). \tag{46}$$

During the inner loop that traverses each $\boldsymbol{q}$ that satisfies (44), if the following condition is met:

$$\boldsymbol{c}^T\ddot{\boldsymbol{q}} \le (\max(0, \boldsymbol{c}^T\dot{\boldsymbol{q}} - \tilde{a}^{\lim}T) - \boldsymbol{c}^T\dot{\boldsymbol{q}})/T, \tag{47}$$

then (43b) is always satisfied due to (46). As a result, the entire inner loop can be omitted because both (47) and (43c) are independent of $\boldsymbol{q}$. In conclusion, the necessary and sufficient condition of (43) can be given by

$$\forall \dot{\boldsymbol{q}} \in \mathcal{D},\ \exists \ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{48a}$$

$$\text{s.t. } \boldsymbol{c}^T\ddot{\boldsymbol{q}} \le -\boldsymbol{c}^T\dot{\boldsymbol{q}}/T \quad \text{if } 0 \le \boldsymbol{c}^T\dot{\boldsymbol{q}} < \tilde{a}^{\lim}T \tag{48b}$$

$$\wedge\ \boldsymbol{c}^T\ddot{\boldsymbol{q}} \le -\tilde{a}^{\lim} \quad \text{if } \boldsymbol{c}^T\dot{\boldsymbol{q}} \ge \tilde{a}^{\lim}T \tag{48c}$$

$$\wedge\ (-\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \le \ddot{\boldsymbol{q}} \le (\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T. \tag{48d}$$

Here, (48b) $\wedge$ (48c) is rewritten from (47).

Even though (48) is a simplified version of (43), determining $\tilde{a}^{\lim}$ remains challenging. Thus, we further simplify (48) to the following sufficient condition of it, based on the fact that $\boldsymbol{c}^T\ddot{\boldsymbol{q}} \le -\tilde{a}^{\lim} < -\boldsymbol{c}^T\dot{\boldsymbol{q}}/T$ holds in (48b) due to $\boldsymbol{c}^T\dot{\boldsymbol{q}} < \tilde{a}^{\lim}T$:

$$\forall \dot{\boldsymbol{q}} \in \mathcal{D},\ \exists \ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{49a}$$

$$\text{s.t. } -\boldsymbol{c}^T\ddot{\boldsymbol{q}} \ge \tilde{a}^{\lim} \tag{49b}$$

$$\wedge\ (-\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \le \ddot{\boldsymbol{q}} \le (\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T. \tag{49c}$$

Here, (49b) is a more conservative condition of (48b) $\wedge$ (48c). Please notice that as T gradually decreases to 0, the conditions (48b) $\wedge$ (48c) and (49b) converge until they serve as the same condition. This convergence indicates that when T is sufficiently small, (49) can be regarded as both a necessary and sufficient condition for (48).

Since (49) is a sufficient condition for (43) and (48), the value of $\tilde{a}^{\lim}$ ensuring them can be determined based on (49). For each $\dot{\boldsymbol{q}}$ in (49), $\tilde{a}^{\lim}$ must be smaller than the maximum of $-\boldsymbol{c}^T\ddot{\boldsymbol{q}}$ with the condition $\ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \wedge$ (49c) to ensure the existence of a feasible $\ddot{\boldsymbol{q}}$. Since a larger $\tilde{a}^{\lim}$ provides better deceleration performance, $\tilde{a}^{\lim}$ should be chosen as the maximum of $-\boldsymbol{c}^T\ddot{\boldsymbol{q}}$ for each $\dot{\boldsymbol{q}}$. Such process will be considered while traversing all $\dot{\boldsymbol{q}} \in \mathcal{D}$ to determine the final value $\tilde{a}^{\lim}$, which can be described by the following linear

bilevel optimization problem:

$$\tilde{a}^{\lim} = \min_{\dot{\boldsymbol{q}} \in \mathcal{D}} \left(\begin{array}{ll} \max\limits_{\ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim})} & -\boldsymbol{c}^T \ddot{\boldsymbol{q}} \\ \text{s.t.} & (-\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \le \ddot{\boldsymbol{q}} \le (\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \end{array} \right). \quad (50)$$

This problem possesses a unique structure that simplifies its resolution compared to general problems. Since the optimal value of the inner maximization problem in (50) can be explicitly obtained, (50) can be derived as

$$\tilde{a}^{\lim} = \min_{\dot{\boldsymbol{q}} \in \mathcal{D}} \left(\sum_{i=1}^{n} \begin{cases} -c_i \max\left((-v_i^{\lim} - \dot{q}_i)/T, -a_i^{\lim}\right) & \text{if } c_i \ge 0 \\ -c_i \min\left((v_i^{\lim} - \dot{q}_i)/T, a_i^{\lim}\right) & \text{if } c_i < 0 \end{cases} \right) \quad (51a)$$

$$= \min_{\dot{\boldsymbol{q}} \in \mathcal{D}} \left(\sum_{i=1}^{n} |c_i| \min\left((v_i^{\lim} + c_i \dot{q}_i/|c_i|)/T, a_i^{\lim}\right) \right) \quad (51b)$$

$$= \min_{\dot{\boldsymbol{q}} \in \mathcal{D}} \left(\sum_{i=1}^{n} \min\left((|c_i| v_i^{\lim} + c_i \dot{q}_i)/T, |c_i| a_i^{\lim}\right) \right). \quad (51c)$$

Comparing the three equations in (51), the form of (51a) is overly complex, while (51b) may encounter zero division errors when $c_i \approx 0$. Thus, (51c) is chosen as the basis for further derivation.

The above problem (51c) can be further reformulated into a MILP problem:

$$\tilde{a}^{\lim} = \min_{z, \dot{\boldsymbol{q}} \in \mathcal{D}, \boldsymbol{\delta} \in \{0,1\}^n} \mathbf{1}^T z \quad (52a)$$

$$\text{s.t. } z \le |\boldsymbol{c}| \odot \boldsymbol{a}^{\lim} \quad (52b)$$

$$z \ge |\boldsymbol{c}| \odot \boldsymbol{a}^{\lim} - M(\mathbf{1} - \boldsymbol{\delta}) \quad (52c)$$

$$z \le (|\boldsymbol{c}| \odot \boldsymbol{v}^{\lim} + \boldsymbol{c} \odot \dot{\boldsymbol{q}})/T \quad (52d)$$

$$z \ge (|\boldsymbol{c}| \odot \boldsymbol{v}^{\lim} + \boldsymbol{c} \odot \dot{\boldsymbol{q}})/T - M\boldsymbol{\delta}. \quad (52e)$$

Here, $z \in \mathbb{R}^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ are auxiliary variables and $M \in \mathbb{R}$ is a large constant satisfying the following condition for all $\dot{\boldsymbol{q}} \in \mathcal{D}$:

$$|\boldsymbol{c}| \odot \boldsymbol{a}^{\lim} - M\mathbf{1} \le (|\boldsymbol{c}| \odot \boldsymbol{v}^{\lim} + \boldsymbol{c} \odot \dot{\boldsymbol{q}})/T \quad (53a)$$

$$\wedge \ (|\boldsymbol{c}| \odot \boldsymbol{v}^{\lim} + \boldsymbol{c} \odot \dot{\boldsymbol{q}})/T - M\mathbf{1} \le |\boldsymbol{c}| \odot \boldsymbol{a}^{\lim}. \quad (53b)$$

In this paper, we simply determine M as the maximum element of the vector $\left(|\boldsymbol{c}| \odot \mathbf{max}(\boldsymbol{a}^{\lim}, 2\boldsymbol{v}^{\lim}/T - \boldsymbol{a}^{\lim})\right)$ to satisfy (53) for all $\dot{\boldsymbol{q}} \in \mathcal{D}$. With this appropriately determined M, the constraints (52b)∼(52e) ensure that the obtained z_i equals to the inner minimum term $\min(*)$ in (51c) for every i form 1 to n. More precisely, (52b)∼(52e) lead to $z_i = (|c_i| v_i^{\lim} + c_i \dot{q}_i)/T \wedge z_i \le |c_i| a_i^{\lim}$ when $\delta_i = 0$, and become $z_i = |c_i| a_i^{\lim} \wedge z_i \le (|c_i| v_i^{\lim} + c_i \dot{q}_i)/T$ when $\delta_i = 1$.

The MILP problem (52) can be efficiently solved by various optimization solvers, such as SCIP [19] and GLPK [20].

3.3 Auxiliary Viability Set for Entire Compound Joint Constraint

Finally, we consider the general case, which is the constraints discussed in Section 2.3:

$$\boldsymbol{A}\boldsymbol{q} \le \tilde{\boldsymbol{q}}^{\lim} \quad (54a)$$

$$\dot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{v}^{\lim}) \quad (54b)$$

$$\ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}). \quad (54c)$$

One can see that the only difference between the above constraints and the case of Section 3.2 is that (54a) is composed of multiple (39a). Thus, $\mathcal{V}_2$ defined by (40) can be extended as follows corresponding to (54):

$$\mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim}) \triangleq \{[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \mid [\boldsymbol{A}_i \boldsymbol{q}, \boldsymbol{A}_i \dot{\boldsymbol{q}}]^T \in \mathcal{V}_1(\tilde{q}_i^{\lim}, \tilde{a}_i^{\lim}),\ i = 1, \cdots, m\}. \quad (55)$$

Here, $\boldsymbol{A}_i \in \mathbb{R}^{1 \times n}$ is the i-th row of $\boldsymbol{A}$, and $\tilde{q}_i^{\lim}$ and $\tilde{a}_i^{\lim}$ are the i-th component of $\tilde{\boldsymbol{q}}^{\lim}$ and $\tilde{\boldsymbol{a}}^{\lim}$, respectively, where $\tilde{\boldsymbol{a}}^{\lim} \in \mathbb{R}^m_{>0}$ is a constant vector to be determined. By referencing the derivation of (42), the constraints corresponding to $\mathcal{F}(\boldsymbol{q}^{k-1}, \dot{\boldsymbol{q}}^{k-1}, \mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim} \cap \{(54a)) \wedge (54b)\})$ can be given by

$$\boldsymbol{A}_i \dot{\boldsymbol{q}}^k \le g_1(\boldsymbol{A}_i \boldsymbol{q}^{k-1}, \boldsymbol{A}_i \dot{\boldsymbol{q}}^{k-1}, \tilde{q}_i^{\lim}, \tilde{a}_i^{\lim}), \quad i = 1, \cdots, m \quad (56a)$$

$$\wedge \ \dot{\boldsymbol{q}}^k \in \mathcal{H}(\boldsymbol{v}^{\lim}). \quad (56b)$$

To ensure $\mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim})$ serves as an auxiliary viability set, $\tilde{\boldsymbol{a}}^{\lim}$ must be determined to satisfy the following condition according to Lemma 1:

$$\forall [\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T \in (\mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim}) \cap \{(54a)) \wedge (54b)\}),\ \exists \ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \quad (57a)$$

$$\text{s.t. } \boldsymbol{A}_i \ddot{\boldsymbol{q}} \le (g_1(\boldsymbol{A}_i \boldsymbol{q}, \boldsymbol{A}_i \dot{\boldsymbol{q}}, \tilde{q}_i^{\lim}, \tilde{a}_i^{\lim}) - \boldsymbol{A}_i \dot{\boldsymbol{q}})/T, \quad i = 1, \cdots, m \quad (57b)$$

$$\wedge \ (-\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T \le \ddot{\boldsymbol{q}} \le (\boldsymbol{v}^{\lim} - \dot{\boldsymbol{q}})/T. \quad (57c)$$

Here, (57b)∧(57c) is obtained by substituting (2b) into (56). This condition is challenging to ensure fully for all cases. To address this, we propose a method for determining $\tilde{\boldsymbol{a}}^{\lim}$ that satisfies (57) in most practical scenarios, even for high-dimensional cases such as $m = 1000 \wedge n = 10$. The proposed method is based on the discussions in Section 3.2. We define another constant vector $\tilde{\boldsymbol{a}}_u^{\lim} \in \mathbb{R}^m$, where each component $\tilde{a}_{ui}^{\lim}$ is computed by substituting the corresponding $\boldsymbol{A}_i^T$ into

$\boldsymbol{c}$ of (52). It is evident that (43) is a necessary condition for (57). Thus, $\tilde{\boldsymbol{a}}_u^{\lim}$ is an upper bound for $\tilde{\boldsymbol{a}}^{\lim}$, as only values of $\tilde{\boldsymbol{a}}^{\lim}$ smaller than $\tilde{\boldsymbol{a}}_u^{\lim}$ can potentially satisfy (57).

Unlike (43), the condition (57) requires accounting for multiple constraints in (57b). Nevertheless, for any given $[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T]^T$, verifying the existence of a feasible $\ddot{\boldsymbol{q}}$ does not necessitate checking all constraints in (57b). Actually, only a limited number of specific combinations of constraints need to be considered. When $\boldsymbol{A}_i\boldsymbol{q}$ is far from $\tilde{q}_i^{\lim}$, the function g_1 returns a large positive number, rendering the corresponding constraint negligible. As $\boldsymbol{A}_i\boldsymbol{q}$ approaches $\tilde{q}_i^{\lim}$, the right-hand side of the corresponding constraint in (57b) decreases until it reaches the minimum $-\tilde{a}_i^{\lim}$. In practice, each constraint in (57b) only affect the solution when the corresponding $\boldsymbol{A}_i\boldsymbol{q}$ is very close to $\tilde{q}_i^{\lim}$. Therefore, the combination of multiple constraints that need to be considered can often be identified by the vertices of the polytope associated with (54a).

Furthermore, if each constraint in (57b) is ensured to be individually compatible with (57c), then their combination is typically also compatible with (57c). This fact indicates that (57c) can be neglected when $\tilde{\boldsymbol{a}}^{\lim} \le \tilde{\boldsymbol{a}}_u^{\lim}$ is satisfied. This is because $\tilde{\boldsymbol{a}}_u^{\lim}$ ensures that any single constraint in (57b) holds under (57c), as a result of satisfying (43). In conclusion, for most practical scenarios, we can assume that the following condition is the sufficient condition for (57):

$$\forall \boldsymbol{s} \in \mathcal{S},\ \exists \ddot{\boldsymbol{q}} \in \mathcal{H}(\boldsymbol{a}^{\lim}) \tag{58a}$$

$$\text{s.t. } \boldsymbol{s} \odot (\boldsymbol{A}\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{a}}^{\lim}) \le \boldsymbol{0} \tag{58b}$$

$$\tilde{\boldsymbol{a}}^{\lim} \le \tilde{\boldsymbol{a}}_u^{\lim} \tag{58c}$$

where

$$\mathcal{S} \triangleq \{\boldsymbol{s} \in \{0,1\}^m | \hat{\boldsymbol{q}} \in \text{vert}(\{\boldsymbol{A}\boldsymbol{q} \le \tilde{\boldsymbol{q}}^{\max}\}), \boldsymbol{s} = \boldsymbol{\beta}(\boldsymbol{A}\hat{\boldsymbol{q}}, \tilde{\boldsymbol{q}}^{\lim})\}. \tag{59}$$

Here, the set vert() includes all the vertices of the given polytope, which can be efficiently computed using several methods, such as the double-description method [21]. For the most of practical cases with small T and relative large $\boldsymbol{a}^{\lim}$, (58) can ensure that (57) holds.

An appropriate $\tilde{\boldsymbol{a}}^{\lim}$ satisfying (58) can be obtained by solving the following QP problem:

minimize

$$\tilde{\boldsymbol{a}}^{\lim}, \tilde{c}, \cdots, \hat{\ddot{\boldsymbol{q}}}_i, \cdots \|\tilde{\boldsymbol{a}}^{\lim} - \tilde{\boldsymbol{a}}_u^{\lim}\|_{\boldsymbol{W}_a}^2 + \|\tilde{c} - c_0\|_{w_{\tilde{c}}}^2 \tag{60a}$$

$$\text{s.t. } 0 \le \tilde{c} \le c_0 \tag{60b}$$

$$\tilde{c}\tilde{\boldsymbol{a}}_u^{\lim} \le \tilde{\boldsymbol{a}}^{\lim} \le \tilde{\boldsymbol{a}}_u^{\lim} \tag{60c}$$

$$\forall \boldsymbol{s} \in \mathcal{S},$$

$$\begin{cases} \hat{\ddot{\boldsymbol{q}}}_i \in \mathcal{H}(\boldsymbol{a}^{\lim}) & (60d) \\ \boldsymbol{s} \odot (\boldsymbol{A}\hat{\ddot{\boldsymbol{q}}}_i + \tilde{\boldsymbol{a}}^{\lim}) \le \boldsymbol{0}. & (60e) \end{cases}$$

Algorithm 1 Calculation for constant $\tilde{\boldsymbol{a}}^{\lim}$

Require: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \tilde{\boldsymbol{q}}^{\lim} \in \mathbb{R}^m, \boldsymbol{v}^{\lim} \in \mathbb{R}_{>0}^n, \boldsymbol{a}^{\lim} \in \mathbb{R}_{>0}^n$

```
ε := 10^-5
S := {}
ã_u^lim := ZERO_VECTOR(m)
for i = 1 to m do
    ã_ui^lim := MILP(c = A_i^T, v^lim, a^lim)        ▷ Eq. (52)
s := ZERO_VECTOR(m)
for all q̂ ∈ VERT({Aq ≤ q̃^lim}) do
    for i = 1 to m do
        if A_i q̂ ≥ q̃_i^lim − ε then
            s_i := 1
        else
            s_i := 0
    PUSH(S, s)
ã^lim := QP(A, a^lim, ã_u^lim, S)                    ▷ Eq. (60)
return ã^lim
```

Here, each $\hat{\ddot{\boldsymbol{q}}}_i \in \mathbb{R}^n$ corresponds to one element of $\mathcal{S}$, $\tilde{c} \in \mathbb{R}_{\ge 0}$ represents a scaling factor for $\tilde{\boldsymbol{a}}_u^{\lim}$, $c_0 \in \mathbb{R}_{>0}$ is a positive constant that is less than 1, $\boldsymbol{W}_a \triangleq \text{diag}([\cdots, 1/\|\boldsymbol{A}_j\|^2, \cdots]^T) \in \mathbb{R}^{m\times m}$ is a weight matrix making the obtained $\tilde{\boldsymbol{a}}^{\lim}$ optimal on the actual deceleration effect, $w_{\tilde{c}} \in \mathbb{R}_{>0}$ should be a relatively large value, such as 1000. The left-hand side of (60c) can be consider as a soft constraint to avoid obtaining extremely small elements of $\tilde{\boldsymbol{a}}^{\lim}$.

In conclusion, the determination of the constant vector $\tilde{\boldsymbol{a}}^{\lim}$ is summarized in Algorithm 1. With this appropriate $\tilde{\boldsymbol{a}}^{\lim}$, $\mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim})$ is ensured to serve as an auxiliary viability set, which is expected for IK (25).

4 Procedure of Proposed IK Method

The block diagram of the proposed IK is shown in Fig. 4. According to the discussion of Section 3, applying $\mathcal{V}_3(\tilde{\boldsymbol{q}}^{\lim}, \tilde{\boldsymbol{a}}^{\lim})$ to IK (25) results in the following QP problem:

minimize

$$\dot{\boldsymbol{q}}^k \|\boldsymbol{J}^{k-1}\dot{\boldsymbol{q}}^k - \boldsymbol{b}^{k-1}\|_{\boldsymbol{W}_1}^2 + \|\dot{\boldsymbol{q}}^k\|_{\boldsymbol{W}_2}^2$$

$$\text{s.t.} \boldsymbol{A}\dot{\boldsymbol{q}}^k \le \overline{\boldsymbol{q}}_A^{k-1} \tag{61a}$$

$$\underline{\boldsymbol{q}}^{k-1} \le \dot{\boldsymbol{q}}^k \le \overline{\boldsymbol{q}}^{k-1}, \tag{61b}$$

where

$$\overline{\boldsymbol{q}}_A^{k-1} \triangleq -T\tilde{\boldsymbol{a}}^{\lim}/2 + \sqrt{2\tilde{\boldsymbol{a}}^{\lim} \odot \max\left(T^2\tilde{\boldsymbol{a}}^{\lim}/8, \tilde{\boldsymbol{q}}\right)} \tag{62a}$$

$$\tilde{\boldsymbol{q}} \triangleq \tilde{\boldsymbol{q}}^{\lim} - \boldsymbol{A}\boldsymbol{q}^{k-1} - T\boldsymbol{A}\dot{\boldsymbol{q}}^{k-1}/2 \tag{62b}$$

$$\underline{\boldsymbol{q}}^{k-1} \triangleq \max(-\boldsymbol{v}^{\lim}, \dot{\boldsymbol{q}}^{k-1} - T\boldsymbol{a}^{\lim}) \tag{62c}$$

$$\overline{\boldsymbol{q}}^{k-1} \triangleq \min(\boldsymbol{v}^{\lim}, \dot{\boldsymbol{q}}^{k-1} + T\boldsymbol{a}^{\lim}). \tag{62d}$$

Here, (61a) $\wedge$ (61b) is reformulated from (25d) $\wedge$ (56), in which (61a) is the specific matrix form of (56a).

The whole method consists of two stages: the offline construction stage and the online computation stage. In the

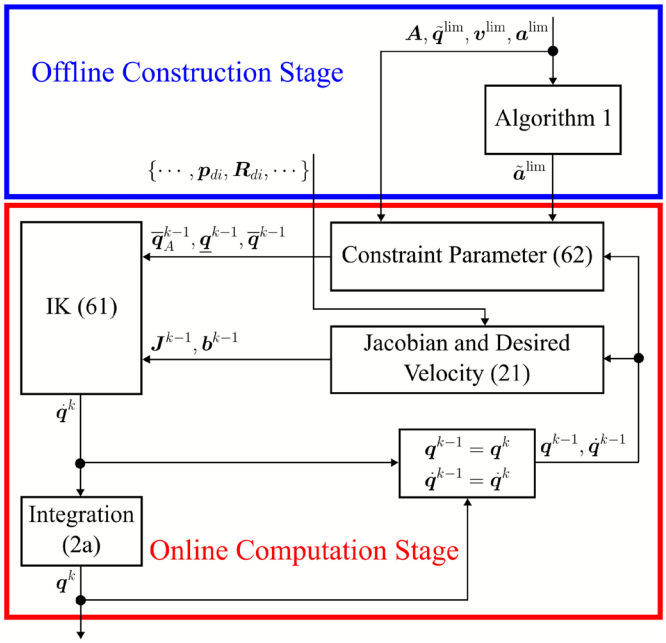

Fig. 4 Block diagram of the proposed IK

offline construction stage, the constant vector $\tilde{\boldsymbol{a}}^{\lim}$ is obtained through Algorithm 1. In the online computation stage, the parameters of the constraints and $\boldsymbol{J}^{k-1}, \boldsymbol{b}^{k-1}$ in the objective function of (61) are updated based on $[(\boldsymbol{q}^{k-1})^T, (\dot{\boldsymbol{q}}^{k-1})^T]^T$ in every loop. One can see that the parameters of (61a) and (61b) can be calculated by simple equations (62), enabling the rapid formulation of the QP in realtime applications. Since the proposed IK method is formulated as a QP with simple linear constraints, it can be easily extended or integrated with additional constraints and objective function terms to address more complex tasks.

5 Numerical Simulation

5.1 Numerical Result 1

We conducted six numerical simulations to compare the proposed method with two related works [9] and [10] on a 1-DOF robot under the sampling interval $T = 0.01$s and 0.1s, respectively. The hard joint constraints of the robot are set as $q \leq 1$, $\dot{q} \in \mathcal{H}(3)$, and $\ddot{q} \in \mathcal{H}(12)$. The target position is given as 1.1m, which lies outside the robot's motion range. The initial state of the robot is $[0.02, 0]^T$.

The results of simulation are shown in Fig. 5. In the phase portraits of Fig. 5(a) and (b), the yellow area indicates the corresponding viability kernel of the robot. It can be seen that all three methods have high approximation accuracy to the viability kernel under small T, e.g., 0.01s, but the accuracy loss of the method of [9] is much greater than that of other two methods at large T, e.g., 0.1s. In Fig. 5(c) and (d), trajectories of position, velocity, and acceleration are given. It can be seen that the method of [10] will cause severe oscillations of acceleration, and the acceleration strategy of method of [9] is more conservative. The above results suggest that the proposed method ensures high approximation accuracy regardless of the sampling interval, while also having better numerical stability.

5.2 Numerical Result 2

We simulated the proposed method with a 2-DOF robot, whose both lengths of arms are 1m. The simulation results are shown in Fig. 6, in which the robot is indicated by purple lines and green circles, the end-effector is denoted by cyan circle, the target position is shown by red point, and the trajectory of the end-effector is expressed by blue solid line. In Fig. 6, the black solid line corresponds to the obstacles, which are 1m and 0.5m tall in Y direction, and its distance between the base point of the robot are 1m and 1.5m in X direction, respectively. The physical joint limits of this robot are given by $[0\ 0]^T \leq \boldsymbol{q} \leq [\pi\ \pi/2]^T$, $\dot{\boldsymbol{q}} \in \mathcal{H}([1\ 2]^T)$ and $\ddot{\boldsymbol{q}} \in \mathcal{H}([15\ 12]^T)$. As illustrated by the black dashed polygon of Fig. 7(a), the compound joint constraints ensuring collision avoidance and compliance with the joint range limit are

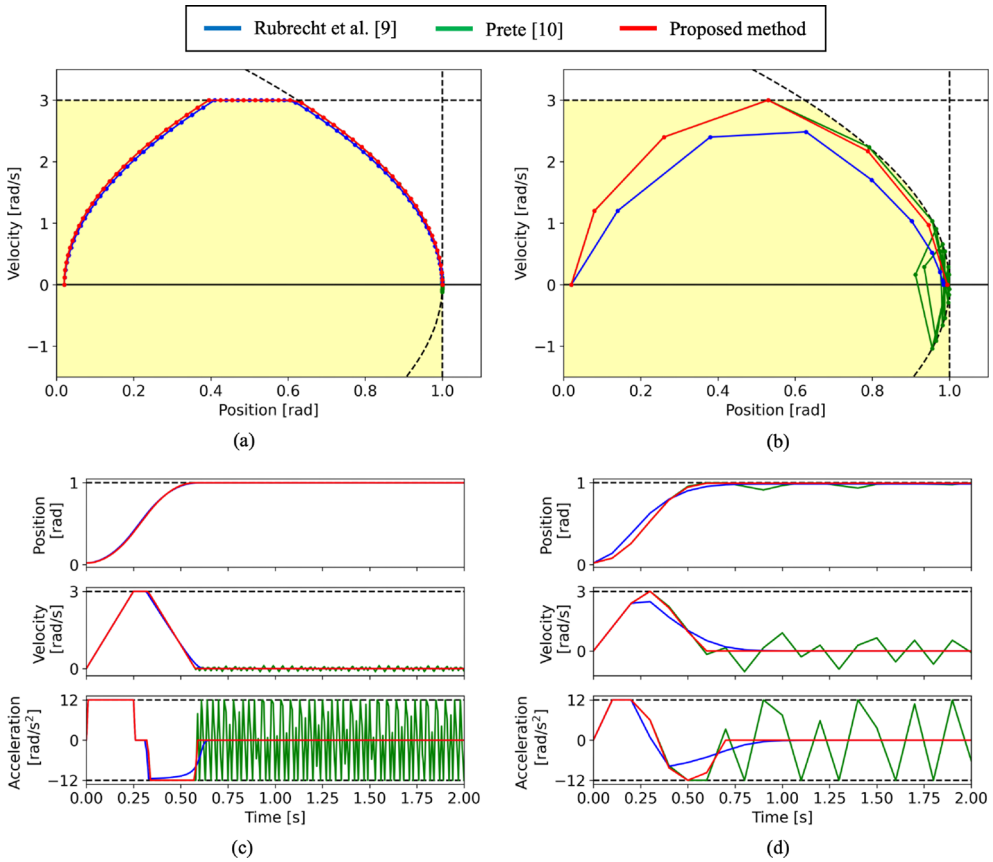

Fig. 5 Numerical result 1. (**a**), (**b**) Phase portraits of one joint obtained by different IK methods with $T = 0.01s$, and $T = 0.1s$, respectively. The yellow area indicates the corresponding viability kernel of the robot. (**c**), (**d**) Trajectories of one joint obtained by different IK methods with $T = 0.01s$, and $T = 0.1s$, respectively

given. The resulting motion range of the robot's end-effector, dictated by these compound joint constraints, is depicted by the black dashed line in Fig. 6.

The simulation is conducted over a duration of 4 seconds. To clearly illustrate the trajectories, the entire process is divided into 4 segments at the time points 0.5, 1.5, and 2.5 seconds, corresponding to the 4 sub-figures in Fig. 6. The joint angle trajectories are depicted in joint space in Fig. 7(a). At each time segment, a new target position is assigned, with the initial position of the end-effector marked

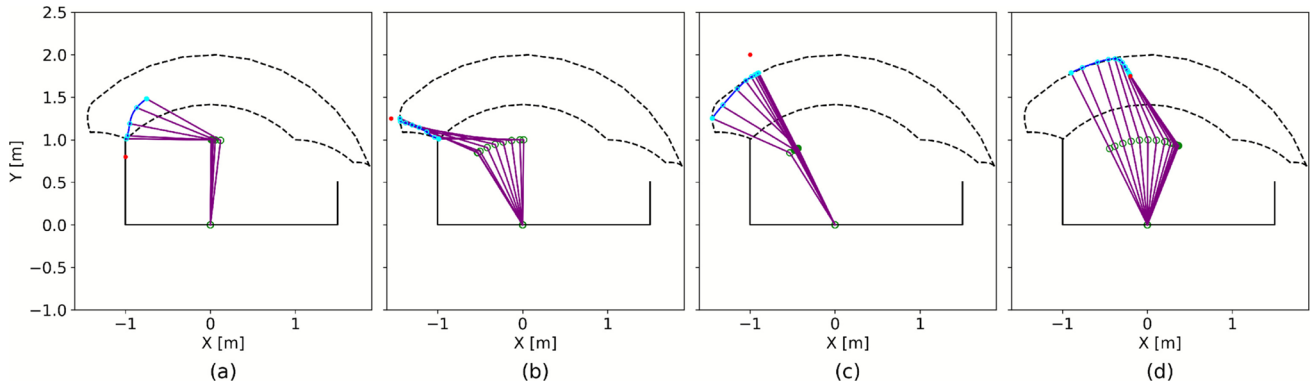

Fig. 6 Trajectory of a 2-DOF robot of numerical result 2

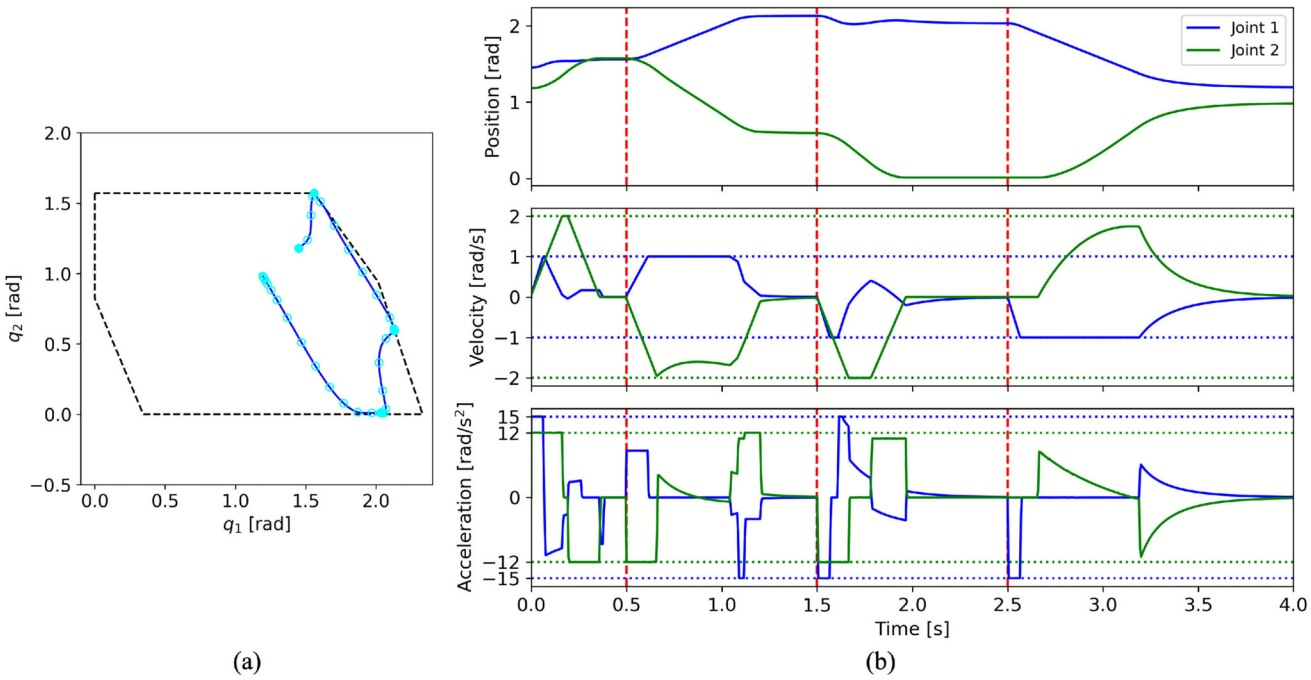

Fig. 7 (**a**) Trajectories of state obtained by the proposed IK method in numerical result 2. (**b**) Trajectories obtained by the proposed IK method of numerical result 2

by solid cyan points. The simulation results are presented in Fig. 7(b), where the red dashed lines represent the aforementioned time points, and the dot-dashed lines correspond to the physical joint limits of the robot. According to Fig. 7, all the given constraints are satisfied, demonstrating the validity of the proposed IK method.

5.3 Numerical Result 3

We measured the computation time required for the offline construction stage of the proposed IK method on robots with DOFs ranging from 2 to 10. All necessary hard constraints (54) were randomly generated within a predefined range. The vertices of the polytopes associated with the position constraints (54a) were randomly distributed on the faces of a hypercube with an edge length of π, ensuring the generated constraints were practically realistic. The number of faces of the polytope, which is also the number of the position constraints' rows m was set to 2^n. This order of magnitude for m can provide a good balance between constraint complexity and approximation accuracy for the nonlinear geometric constraints (14) in practical applications. The velocity constraint parameter $v_i^{\lim}$ and acceleration constraint parameter $a_i^{\lim}$ in (54b) and (54c) are randomly generated within the ranges [0.5, 2] and [1, 5], respectively.

As described in Algorithm 1, the offline construction stage can be divided into three parts: performing m iterations of MILP solving, obtaining the set of vertices VERT of the polytopes associated with (54a) while constructing $\mathcal{S}$ defined in (59), and solving a QP to determine $\tilde{\boldsymbol{a}}^{\lim}$. For each specific n, Algorithm 1 was executed 15 times, and the average computation times for the three parts were recorded and presented in Table 1. The results indicate that the proposed algorithm can efficiently handle high dimensional cases with up to $n = 10$ and $m = 1024$. Moreover, the results do not imply that higher-dimensional cases are entirely infeasible. When the number of rows of the position constraints is reduced, higher-dimensional cases (e.g., $m = 100$ and $n = 20$) can also be processed within an acceptable computation time.

Table 1 Average computation time of the offline construction stage

		Average time [s]			
n	m	MILP	VERT	QP	Total
2	4	0.004747	0.000020	0.001327	0.006094
3	8	0.007952	0.000041	0.001290	0.009283
4	16	0.039792	0.000120	0.002178	0.042090
5	32	0.115859	0.000380	0.004777	0.121015
6	64	0.289074	0.003725	0.015475	0.308273
7	128	0.927218	0.075500	0.077341	1.080060
8	256	2.798633	2.519383	0.337033	5.655049
9	512	9.151099	109.216535	1.869819	120.237453
10	1024	24.749289	1530.569926	7.775779	1563.094994

6 Conclusion

This paper has proposed a QP-based IK method to simultaneously handle physical joint limits and whole-body collision avoidance, including self-collision and collisions with static obstacles. In the proposed method, whole-body collision avoidance and joint range limits are approximated as a linear compound joint constraint. A simple linear constraint corresponding to the auxiliary viability set is derived to ensure the solution of IK remains viable. For scenarios where only physical joint limits are required, the proposed method guarantees high approximation accuracy to the corresponding viability kernel while maintaining superior numerical stability compared to previous works. In scenarios requiring collision avoidance, the proposed method efficiently determines appropriate parameters to guarantee the solution is viable, even in high-dimensional cases such as $n = 10$ and $m \approx 1000$. This marks a significant advancement compared to previous studies.

Nevertheless, the proposed IK method still has several areas for improvement. The set of solutions obtained by proposed method tends to be more conservative than the viability kernel of the original problem. This conservatism arises from two main factors: 1. The original nonlinear geometric constraints are replaced by conservative linear compound joint constraints, where the gap between the two can sometimes be significant. 2. The constant vector $\tilde{\boldsymbol{a}}^{\lim}$ is determined based on the most extreme cases, making the resulting $\tilde{\boldsymbol{a}}^{\lim}$ overly conservative for most other states. The first issue can be addressed by using the union of multiple compound joint constraints to better approximate the original nonlinear constraints. This would also require an enhancement of the algorithm to handle the union of multiple linear constraints in realtime. The second issue could be mitigated by allowing $\tilde{\boldsymbol{a}}^{\lim}$ to vary dynamically based on target position rather than being fixed. Moreover, the proposed method lacks sufficient generality and is not applicable to many scenarios, including cases with large sampling interval T and higher-dimensional cases. Addressing these limitations would require comprehensive modifications to the algorithm.

Appendix A

Lemma 2 *For $a > 0$, $b \in \mathbb{R}$ and $c > 0$, the following statement holds true:*

$$x \le \sqrt{a\max(0, b - cx)} \iff x \le \frac{-ac + \sqrt{a^2c^2 + 4a\max(0, b)}}{2} \tag{A1}$$

Proof

$$\begin{aligned}
&x \le \sqrt{a\max(0, b - cx)} \\
&\iff \Big(x > 0 \wedge x^2 \le a\max(0, b - cx)\Big) \vee x \le 0 \\
&\iff x^2 \le a\max(0, b - cx) \vee x \le 0 \\
&\iff (b - cx < 0 \wedge x^2 \le 0) \vee \Big(b - cx \ge 0 \wedge x^2 \\
&\le a(b - cx)) \vee x \le 0 \\
&\iff x^2 \le a(b - cx) \vee x \le 0 \\
&\iff \left(\frac{-ac - \sqrt{a^2c^2 + 4ab}}{2} \le x \right. \\
&\left. \le \frac{-ac + \sqrt{a^2c^2 + 4ab}}{2} \wedge ac^2 + 4b \ge 0\right) \vee x \le 0 \\
&\iff x \le \frac{-ac + \sqrt{a^2c^2 + 4a\max(0, b)}}{2}
\end{aligned}$$

□

Appendix B

Lemma 3 *For $a > 0$, $T > 0$ and $x \ge 0$, the following statement holds true:*

$$q \le q_a \wedge q \le q_b \iff q \le q_a \tag{B2}$$

where

$$q_a \triangleq \left(-aT + \sqrt{(aT)^2 + 8a\max(0, x - aT^2/8)}\right) \Big/ 2 \tag{B3a}$$

$$q_b \triangleq (2/T)x \tag{B3b}$$

Proof The sufficient and necessary condition for the above statement is that $q_a \le q_b$ holds for $a > 0$, $T > 0$ and $x \ge 0$.

When $0 \le x \le aT^2/8$, q_a is always 0, thus, $q_a \le q_b$ is satisfied.

When $x > aT^2/8$,

$$\begin{aligned}
q_a - q_b &= \left(-aT + \sqrt{8ax}\right) \Big/ 2 - 2x/T \\
&= \left(2\sqrt{2axT^2} - aT^2 - 4x\right) \Big/ (2T).
\end{aligned} \tag{B4}$$

According to the AM–GM inequality,

$$2\sqrt{4axT^2} \le aT^2 + 4x. \tag{B5}$$

Thus, (B4) remains negative, i.e., $q_a < q_b$, due to $2\sqrt{2axT^2} < 2\sqrt{4axT^2}$ and (B5). □

Appendix C

Lemma 4 *Given $y \geq 0$, $a > 0$, $T > 0$ and $x^{\lim} \in \mathbb{R}$, for all x satisfying $y \leq f(x)$, the following statement holds true:*

$$g(x) \geq \max(0, y - aT), \tag{C6}$$

where

$$f(x) \triangleq \sqrt{2a\max(0, x^{\lim} - aT^2/8 - x)} \tag{C7a}$$

$$g(x) \triangleq \left(-aT + \sqrt{(aT)^2 + 8a\max(0, x^{\lim} - aT^2/8 - x - Ty/2)}\right)/2. \tag{C7b}$$

Proof According to (C7b), the function $g(x)$ alternates between two forms: 0 and

$$g_a(x) \triangleq \left(-aT + \sqrt{8a(x^{\lim} - x - Ty/2)}\right)/2, \tag{C8}$$

where the selection is determined by the condition

$$x^{\lim} - aT^2/8 - x - Ty/2 \geq 0. \tag{C9}$$

On the other hand, $f(x)$ behaves as a concave function when $f(x) > 0$. The two intersection points between $y = f(x)$ and $x^{\lim} - aT^2/8 - x - Ty/2 = 0$ are given by $(x^{\lim} - 5aT^2/8, aT)$ and $(x^{\lim} - aT^2/8, 0)$. This fact implies that for all x and y satisfying $y \leq f(x)$, (C9) always holds when $y \geq aT$. Conversely, for $0 \leq y < aT$, there must exist some values of x where (C9) does not hold.

As a result, the minimum of $g(x)$ is 0 when $y < aT$ holds. For the case of $y \geq aT$, $g(x)$ turns into $g_a(x)$, and $y \leq f(x)$ can be rewritten as

$$x \leq x^{\lim} - aT^2/8 - y^2/2a. \tag{C10}$$

Because (C9) is always satisfied in this case, $g_a(x)$ is a monotonic decreasing function, and its minimum value is attained when x is maximized as specified by (C10). Thus, the minimum value of $g_a(x)$ is given by $y - aT$.

In conclusion, given $y \geq 0$, for all x satisfying $y \leq f(x)$:

$$\min\ g(x) = \begin{cases} y - aT & \text{if } y \geq aT \\ 0 & \text{if } y < aT, \end{cases} \tag{C11}$$

which gives (C6). □

Acknowledgements Open Access funding provided by Hiroshima University. Our source code of the proposed IK method utilizes the open-source libraries SCIP [19], eigen-cddlib [22], CasADi [23], qpOASES [24] and OSQP [25].

Author Contributions Yachen Zhang conceived the core idea, developed the algorithm, performed the simulations, and wrote the manuscript. Ryo Kikuuwe reviewed and edited the manuscript and supervised the project.

Funding Open Access funding provided by Hiroshima University. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Statement The source code corresponding to the proposed method in this paper is available at: https://github.com/zyc1155/Viability_IK.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Competing Interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Antonelli, G., Chiaverini, S., Fusco, G.: A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits. IEEE Trans. Robot. Autom. **19**(1), 162–167 (2003)
2. Flacco, F., De Luca, A., Khatib, O.: Control of redundant robots under hard joint constraints: saturation in the null space. IEEE Trans. Rob. **31**(3), 637–654 (2015)
3. Guo, D., Zhang, Y.: Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators. IEEE Trans. Industr. Electron. **61**(12), 6903–6914 (2014)
4. Suleiman, W.: On inverse kinematics with inequality constraints: new insights into minimum jerk trajectory generation. Adv. Robot. **30**(17–18), 1164–1172 (2016)
5. Quiroz-Omaña, J.J., Adorno, B.V.: Whole-body control with (self) collision avoidance using vector field inequalities. IEEE Robot. Autom. Lett. **4**(4), 4048–4053 (2019)

6. Park, K.C., Chang, P.H., Kim, S.H.: The enhanced compact QP method for redundant manipulators using practical inequality constraints. In: Proceedings of 1998 IEEE International Conference on Robotics and Automation, vol. 1, pp. 107–114 (1998)
7. Aubin, J.-P.: Viability Theory. Birkhäuser Boston, Boston, MA (2009)
8. Decré, W., Smits, R., Bruyninckx, H., De Schutter, J.: Extending iTaSC to support inequality constraints and non-instantaneous task specification. In: Proceedings of 2009 IEEE International Conference on Robotics and Automation, pp. 964–971 (2009)
9. Rubrecht, S., Padois, V., Bidaud, P., De Broissia, M.: Constraints compliant control: constraints compatibility and the displaced configuration approach. In: Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 677–684 (2010)
10. Del Prete, A.: Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators. IEEE Robot. Autom. Lett. **3**(1), 281–288 (2017)
11. Saint-Pierre, P.: Approximation of the viability kernel. Appl. Math. Optim. **29**, 187–209 (1994)
12. Rubrecht, S., Padois, V., Bidaud, P., De Broissia, M., Da Silva Simoes, M.: Motion safety and constraints compatibility for multibody robots. Auton. Robot. **32**, 333–349 (2012)
13. Faroni, M., Beschi, M., Pedrocchi, N.: Inverse kinematics of redundant manipulators with dynamic bounds on joint movements. IEEE Robot. Autom. Lett. **5**(4), 6435–6442 (2020)
14. Kajita, S., Hirukawa, H., Harada, K., Yokoi, K.: Kinematics. In: Introduction to Humanoid Robotics, pp. 19–67. Springer, Berlin, Heidelberg (2014)
15. Nakamura, Y., Hanafusa, H.: Inverse kinematic solutions with singularity robustness for robot manipulator control. Trans. ASME: J. Dyn. Syst. Meas. Control **108**(3), 163–171 (1986)
16. Chiaverini, S.: Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans. Robot. Autom. **13**(3), 398–410 (1997)
17. Dai, H., Amice, A., Werner, P., Zhang, A., Tedrake, R.: Certified polyhedral decompositions of collision-free configuration space. Int. J. Robot. Res. **43**(9), 1322–1341 (2024)
18. Werner, P., Amice, A., Marcucci, T., Rus, D., Tedrake, R.: Approximating robot configuration spaces with few convex sets using clique covers of visibility graphs. In: Proceedings of 2024 IEEE International Conference on Robotics and Automation, pp. 10359–10365 (2024)
19. Bolusani, S., Besançon, M., Bestuzheva, K., Chmiela, A., Dionísio, J., Donkiewicz, T., Doornmalen, J., Eifler, L., Ghannam, M., Gleixner, A., Graczyk, C., Halbig, K., Hedtke, I., Hoen, A., Hojny, C., Hulst, R., Kamp, D., Koch, T., Kofler, K., Lentz, J., Manns, J., Mexi, G., Mühmer, E., Pfetsch, M.E., Schlösser, F., Serrano, F., Shinano, Y., Turner, M., Vigerske, S., Weninger, D., Xu, L.: The SCIP Optimization Suite 9.0. Technical report, Optimization Online (2024). https://optimization-online.org/2024/02/the-scip-optimization-suite-9-0/
20. GLPK (GNU Linear Programming Kit). https://www.gnu.org/software/glpk
21. Fukuda, K., Prodon, A.: Double description method revisited. In: Combinatorics and Computer Science, pp. 91–111. Springer, Berlin, Heidelberg (1996)
22. eigen-cddlib. https://github.com/vsamy/eigen-cddlib
23. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi - A software framework for nonlinear optimization and optimal control. Math. Program. Comput. **11**(1), 1–36 (2019)
24. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Math. Program. Comput. **6**(4), 327–363 (2014)
25. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Math. Program. Comput. **12**(4), 637–672 (2020)

Yachen Zhang received the B.E. degree in mechanical engineering from North China University of Science and Technology, Hebei, China, in 2017, and the M.E. degree in mechanical engineering from Okayama University, Okayama, Japan, in 2021. He is currently pursuing the Ph.D. degree with Hiroshima University, Hiroshima, Japan. His research interests include teleoperation, whole-body planning and control for humanoid robots, and optimal control for robotics.

Ryo Kikuuwe received the B.S., M.S., and Ph.D. (Eng.) degrees in mechanical engineering from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2007, he was an Endowed-Chair Research Associate with the Nagoya Institute of Technology, Nagoya, Japan. From 2007 to 2017, he was an Associate Professor with the Department of Mechanical Engineering, Kyushu University, Fukuoka, Japan. From 2014 to 2015, he was a Visiting Researcher with Inria Grenoble Rhône-Alpes, Saint-Ismier, France. He is currently a Full Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Hiroshima, Japan. His research interests include force control of robot manipulators, control and modeling of hydraulic systems, and engineering applications of differential inclusions. Dr. Kikuuwe is a Member of the IEEE, the Robotics Society of Japan, the Japan Society of Mechanical Engineers, and the Society of Instrument and Control Engineers (Japan). He was a recipient of the Best Paper Award of Advanced Robotics in 2013 and the Young Investigator Excellence Award from the Robotics Society of Japan in 2005.