Joint Position Bounds in Resolved-Acceleration Control: a Comparison

Andrea Testa, Luigi Raiano, Marco Laghi, Arash Ajoudani, Enrico Mingo Hoffman

Joint Position Bounds in Resolved-Acceleration Control: a Comparison

Andrea Testa, Luigi Raiano, Marco Laghi, Arash Ajoudani, and Enrico Mingo Hoffman*

Abstract The implementation of human-friendly robots is based on the deployment of robots that can safely and effectively work with humans in various environments. To this end, enforcing joint limits in planning and control play a fundamental role in avoiding the robot to exceed its physical constraint and preventing joint damages or failures that could lead to unpredictable behavior or compromised safety. However, the implementation of such limitations in instantaneous controllers is not trivial when position, velocity, and acceleration limits are all considered together. In this work, we compare three State-of-the-Art methods, namely the *P-Step Ahead Predictor*, the *Control Barrier Function*, and *Invariance*. Finally, we select the most performing one applied in a real use case based on a UR5e manipulator for a picking task where hitting joint limits may represent an issue.

Key words: Quadratic Programming, Optimization, Collaborative Robots

1 Introduction

In recent years, the use of optimization tools for controlling robotics systems has become well-established in both academic research and industry practice [10, 11].One commonly used approach is Quadratic Programming (QP) optimization, which enables the solution of differential inverse control

* All the Authors are with Leonardo S.p.A., Via Raffaele Pieragostini, 80, 16149 Genova, Italy and Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. E-mail: {name.surname.ext}@leonardo.com, enrico.mingohoffman@leonardo.com, arash.ajoudani@iit.it

The Authors would like to acknowledge Dr. Matteo Parigi Polverini for the implementation of the joint position constraint based on invariance and Dr. Niels Dehio for his valuable help in reviewing the paper.

problems. In this approach, tasks are typically specified as least squares, and constraints, including equalities and inequalities, are linear.

Researchers have utilized constraints to model a range of limitations, including those associated with the robotic hardware [4], safety [5], and balancing in floating-based systems [3]. Constraints that apply to the integrals of optimization variables, such as limiting joint positions or end-effector poses, are of significant importance in numerous practical applications, particularly those related to human-robot coexistence and collaboration. However, simply constraining the integrals of the controlled variables is insufficient to guarantee that the integral's outcome remains within acceptable limits. This approach can ensure feasibility between control bounds and imposed integral constraints only when they have a relative degree of 1. Resolved rate control, which focuses on joint positions and velocities, exemplifies this point. When a joint is at maximum velocity, it can reach its limits with zero speed in one step, assuming arbitrary deceleration. However, if we include constraints on joint accelerations, one step is insufficient to achieve zero speed unless the joint velocity is restricted to allow for a single deceleration step. This issue is typical in resolved-acceleration control, also known as acceleration-based control, in which joint acceleration/torques are usually limited. This implies a pitfall in instantaneous controllers, requiring reasoning in terms of control horizon and not on a single control step.

The literature presents several methods to solve this issue. In [12], a method to early detect the joint position limits is presented, named *P-Step-Ahead Predictor* (PSAP), based on a rescaling of the time by a scalar constant $p \geq 1$. While this method is very simple to implement, it requires hand-tuning and results in highly conservative behavior. A more advanced approach is based on *Control Barrier Functions* (CBF) applied to generic types of geometric constraints other than joint limits, e.g. self-collision [8] and Cartesian box constraints [14]. However, CBF-based techniques require a certain level of tuning as well. Other interesting methods are based on *viability* and *invariance* control approach [4, 13, 16]. These methods do not require tuning at all. However, they present chattering at the boundaries due to the digital implementation of the switching control law [14].

This paper conducts a comprehensive comparison of the three aforementioned method families to identify the most suitable approach for implementing joint limits in QP-based resolved-acceleration controllers. The evaluation focuses on assessing the qualitative and quantitative behavior of the resulting control input. The effectiveness of the chosen method is demonstrated in a practical scenario where a collaborative robot is tasked with picking up a box positioned significantly below the base link. This task presents potential challenges related to joint limits, which could lead to robot failures and jeopardize task execution. However, with the selected method, the task is successfully and safely executed, overcoming potential joint limit issues.

2 Background

In this section, we shortly introduce the methods we intend to compare. We consider a manipulator with n Degrees of Freedom (DoFs) with $\mathbf{q} \in \mathbb{R}^n$ its generalized coordinates, and $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ the joint velocities and accelerations, respectively. We assume that joint positions, velocities, and accelerations are bounded with given compatible bounds $[\mathbf{q}_m, \mathbf{q}_M]$, $[\dot{\mathbf{q}}_m, \dot{\mathbf{q}}_M]$, and $[\ddot{\mathbf{q}}_m, \ddot{\mathbf{q}}_M]$.

In QP-based resolved-acceleration control, the variables of the optimization problem are the joint accelerations, therefore constraints and cost functions are expressed at the acceleration level. The controller is discretized and at the generic control loop k, the resolution of the QP returns the optimal solution $\ddot{\mathbf{q}}_k$. Given the control loop time as dt, for which $\ddot{\mathbf{q}}_k$ is constant, with backward Euler it is possible to compute the joint velocities and positions:

$$\dot{\mathbf{q}}_k = \dot{\mathbf{q}}_{k-1} + dt\ddot{\mathbf{q}}_k; \tag{1}$$

$$\mathbf{q}_k = \mathbf{q}_{k-1} + dt\dot{\mathbf{q}}_{k-1} + \frac{1}{2}dt^2\ddot{\mathbf{q}}_k. \tag{2}$$

Using (1) and (2), it is possible to derive a *naive* version of joint velocity and position limits:

$$\frac{\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_{k-1}}{dt} \leq \ddot{\mathbf{q}}_k \leq \frac{\dot{\mathbf{q}}_M - \dot{\mathbf{q}}_{k-1}}{dt}; \tag{3}$$

$$2\frac{\mathbf{q}_m - \mathbf{q}_{k-1} - dt\dot{\mathbf{q}}_{k-1}}{dt^2} \leq \ddot{\mathbf{q}}_k \leq 2\frac{\mathbf{q}_M - \mathbf{q}_{k-1} - dt\dot{\mathbf{q}}_{k-1}}{dt^2}. \tag{4}$$

However, by imposing (3) and (4), together with

$$\ddot{\mathbf{q}}_m \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_M, \tag{5}$$

it is not possible to ensure the feasibility of all the imposed constraints, due to the instantaneous nature of the controller.

2.1 P-Steps Ahead Predictor (PSAP)

The main idea of the PSAP method is based on activating the preventive motion in advance by $p - steps$ by approximating:

$$\dot{\mathbf{q}}_{p+k} \approx \dot{\mathbf{q}}_{k-1} + p\,dt\ddot{\mathbf{q}}_k; \tag{6}$$

$$\mathbf{q}_{p+k} \approx \mathbf{q}_{k-1} + p\,dt\dot{\mathbf{q}}_{k-1} + \frac{1}{2}(p\,dt)^2\ddot{\mathbf{q}}_k, \tag{7}$$

with $p \geq 1$ [12]. The constraints in (3) and (4) are modified such that:

$$\frac{\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_{k-1}}{p\,dt} \leq \ddot{\mathbf{q}}_k \leq \frac{\dot{\mathbf{q}}_M - \dot{\mathbf{q}}_{k-1}}{p\,dt}; \tag{8}$$

$$2\frac{\mathbf{q}_m - \mathbf{q}_{k-1} - p\,dt\dot{\mathbf{q}}_{k-1}}{(p\,dt)^2} \leq \ddot{\mathbf{q}}_k \leq 2\frac{\mathbf{q}_M - \mathbf{q}_{k-1} - p\,dt\dot{\mathbf{q}}_{k-1}}{(p\,dt)^2}. \tag{9}$$

The pseudo-code in Algorithm 1 presents a simple implementation of the PSAP constraint using (9) and (8), taking into account the maximum joint accelerations $[\ddot{\mathbf{q}}_m,\ \ddot{\mathbf{q}}_M]$ as well. The function *checkFlippingBounds()* depicted

Algorithm 1 Joint-Limits-PSAP

function JLPSAP($\mathbf{q}_{k-1}, \dot{\mathbf{q}}_{k-1}, \mathbf{q}_m, \mathbf{q}_M, \dot{\mathbf{q}}_m, \dot{\mathbf{q}}_M, \ddot{\mathbf{q}}_m, \ddot{\mathbf{q}}_M, p, dt$)
  $p > 1, \quad dt > 0$
  $[\ddot{\mathbf{q}}_{m,p}, \ddot{\mathbf{q}}_{M,p}] = (9)$
  $[\ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_{M,v}] = (8)$
  $\mathbf{b}_l = \max([\ddot{\mathbf{q}}_{m,p}, \ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_m])$
  $\mathbf{b}_u = \min(\left[\ddot{\mathbf{q}}_{M,p}, \ddot{\mathbf{q}}_{M,v}, \ddot{\mathbf{q}}_M\right])$
  for $i = 0, \ldots, n-1$:
    $\left[b_{l,i}, b_{u,i}\right] = \text{checkFlippingBounds}(b_{l,i}, b_{u,i}, \ddot{q}_{m,i}, \ddot{q}_{M,i})$
  return $[\mathbf{b}_l, \mathbf{b}_u]$

in Algorithm 2 flips the bounds if not compatible, which is a situation appearing when the constraints are not respected. By employing this approach, it is possible to ensure that the algorithm generates constraints that maintain the problem's feasibility and swiftly bring the joint positions back within the permissible range.

Algorithm 2 Check-Flipping-Bounds

function CHECKFLIPPINGBOUNDS(v_1, v_2, v_m, v_M)
  if $v_1 \geq v_2$ **then**
    $b_u = v_1,\ b_l = v_2$
  else
    $b_u = v_2,\ b_l = v_1$
  return $[\max(b_l, v_m), \min(b_u, v_M)]$

2.2 Control Barrier Functions (CBF)

CBFs provide a general framework for guaranteeing the safety of robotic systems, leveraging its non-linear dynamics to constrain the robot's state and prevent it from leaving a safe set [2]. Safety is ensured through forward invariance, which relies on a continuously differentiable scalar function $h(\mathbf{q}) \geq 0$. This is done by imposing the following constraint on $\dot{h}(\mathbf{q}, \dot{\mathbf{q}})$:

$$\dot{h}(\mathbf{q}, \dot{\mathbf{q}}) \geq -\alpha_1 h(\mathbf{q}), \tag{10}$$

with $\dot{h}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial h(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}}$, and $\alpha_1 \in \mathbb{R}_+$. However, CBF constraints are typically applied only at the velocity level. To overcome this limitation, a solution called Exponential-CBFs (E-CBFs) is introduced in [7]. E-CBFs enable the imposition of acceleration constraints by introducing a new barrier function, represented as $h_E(\mathbf{q}, \dot{\mathbf{q}}) = \dot{h}(\mathbf{q}, \dot{\mathbf{q}}) + \alpha_1 h(\mathbf{q})$. Therefore $\dot{h}_E(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \ddot{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \alpha_1 \dot{h}(\mathbf{q}, \dot{\mathbf{q}})$, with $\ddot{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \frac{\partial \dot{h}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \ddot{\mathbf{q}}$, and the final constraint become:

$$\dot{h}_E(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \geq -\alpha_2 h_E(\mathbf{q}, \dot{\mathbf{q}}). \tag{11}$$

Following the methodology in [8], joint position constraints can be imposed using E-CBF by choosing the following functions for the $i-th$ joint[2]:

$$\begin{aligned} h_m(q_i) &= q_i - q_{m,i} \geq 0; \\ h_M(q_i) &= q_{M,i} - q_i \geq 0, \end{aligned} \tag{12}$$

with the following the time derivatives:

$$\begin{aligned} \dot{h}_m(q_i, \dot{q}_i) &= \dot{q}_i, \; \ddot{h}_m(q_i, \dot{q}_i, \ddot{q}_i) = \ddot{q}_i; \\ \dot{h}_M(q_i, \dot{q}_i) &= -\dot{q}_i, \; \ddot{h}_M(q_i, \dot{q}_i, \ddot{q}_i) = -\ddot{q}_i. \end{aligned} \tag{13}$$

Therefore by imposing

$$\begin{aligned} h_{E,m}(q_i, \dot{q}_i) &= \dot{q}_i + \alpha_1(q_i - q_{m,i}); \\ h_{E,M}(q_i, \dot{q}_i) &= -\dot{q}_i + \alpha_1(q_{M,i} - q_i), \end{aligned} \tag{14}$$

and

$$\begin{aligned} \dot{h}_{E,m}(q_i, \dot{q}_i, \ddot{q}_i) &= \ddot{q}_i + \alpha_1 \dot{q}_i; \\ \dot{h}_{E,M}(q_i, \dot{q}_i, \ddot{q}_i) &= -\ddot{q}_i - \alpha_1 \dot{q}_i, \end{aligned} \tag{15}$$

is possible to derive the constraint:

$$\begin{aligned} \ddot{q}_i &\geq -(\alpha_1 + \alpha_2)\dot{q}_i - \alpha_1\alpha_2(q_i - q_{m,i}); \\ \ddot{q}_i &\leq -(\alpha_1 + \alpha_2)\dot{q}_i + \alpha_1\alpha_2(q_{M,i} - q_i), \end{aligned} \tag{16}$$

which in vector form for the k step become:

$$-(\alpha_1+\alpha_2)\dot{\mathbf{q}}_{k-1} - \alpha_1\alpha_2(\mathbf{q}_{k-1} - \mathbf{q}_m) \leq \ddot{\mathbf{q}}_k \leq -(\alpha_1+\alpha_2)\dot{\mathbf{q}}_{k-1} + \alpha_1\alpha_2(\mathbf{q}_M - \mathbf{q}_{k-1}). \tag{17}$$

It is worth noticing that the bounds in (17) have the exact same form of the bounds in (9) with:

$$(\alpha_1 + \alpha_2) = \frac{2}{p\,dt}, \quad \alpha_1\alpha_2 = \frac{2}{(p\,dt)^2}. \tag{18}$$

[2] For the seek of clarity, we here remove the subscript $k-1$, which is applied to joint positions and velocities, and k which is applied to joint accelerations

However, the system of equations in (18) does not have solutions in $\mathbb{R}$ because solving for α_1 results in a second-order polynomial $\alpha_2^2 - \alpha_2 \frac{2p}{dt} + \frac{2p^2}{dt^2}$ with discriminant $\Delta = -\frac{4p^2}{dt^2} < 0$. Therefore, it is not possible to simplify the CBF constraint to the PSAP constraint.

A constraint on the joint velocities, based on (10) can be computed by simply choosing $h(\dot{q}_i) = \dot{q}_i - \dot{q}_{m,i} \geq 0$. Consequently, $\dot{h}(\dot{q}_i, \ddot{q}_i) = \ddot{q}_i$, leading to the constraint[3]:

$$\alpha_3(\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_k) \leq \ddot{\mathbf{q}}_k \leq \alpha_3(\dot{\mathbf{q}}_M - \dot{\mathbf{q}}_k). \tag{19}$$

By imposing $\alpha_3 = \frac{1}{p\,dt}$, the constraint (19) is equivalent to (8). Algorithm 3 reports the pseudo-code for the CBF-based constraint.

Algorithm 3 Joint-Limits-CBF

procedure JLCBF($\mathbf{q}_m, \mathbf{q}_M, \dot{\mathbf{q}}_m, \dot{\mathbf{q}}_M, \ddot{\mathbf{q}}_m, \ddot{\mathbf{q}}_M, \alpha_1, \alpha_2, \alpha_3$)
  $\alpha_1 > 0, \quad \alpha_2 > 0, \quad \alpha_3 > 0$
  $[\ddot{\mathbf{q}}_{m,p}, \ddot{\mathbf{q}}_{M,p}] = (17)$
  $[\ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_{M,v}] = (19)$
  $\mathbf{b}_l = \max([\ddot{\mathbf{q}}_{m,p}, \ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_m])$
  $\mathbf{b}_u = \min([\ddot{\mathbf{q}}_{M,p}, \ddot{\mathbf{q}}_{M,v}, \ddot{\mathbf{q}}_M])$
  for $i = 0, \ldots, n-1$:
    $[b_{l,i}, b_{u,i}] = \text{checkFlippingBounds}(b_{l,i}, b_{u,i}, \ddot{q}_{m,i}, \ddot{q}_{M,i})$
  return $[\mathbf{b}_l, \mathbf{b}_u]$

2.3 Invariance

The invariance or viability control approach [4,13,16] permits an acceleration constraint that guarantees the respect of joint position and velocity limits. As a result, the feasible space $\mathcal{L}$ defined by the constraints:

$$q_{m,i} \leq q_i \leq q_{M,i}, \quad \dot{q}_{m,i} \leq \dot{q}_i \leq \dot{q}_{M,i}, \tag{20}$$

for the ith joint becomes a positive invariance region, i.e.,

$$\text{if } \xi_{i,0} \in \mathcal{L} \implies \xi_i(t) \in \mathcal{L} \; \forall \; t > 0. \tag{21}$$

Here, $\xi_{i,0} = (q_{i,0}, \dot{q}_{i,0}) \in (\mathbb{R} \times \mathbb{R})$ is the initial point of the ith joint state trajectory $\xi_i(t) = (q_i(t), \dot{q}_i(t))$. Considering the maximum and minimum limits $[\ddot{q}_{m,i}, \ddot{q}_{M,i}]$ for the joint acceleration $\ddot{q}_i$, it can be demonstrated that the joint state $(\dot{q}_i, q_i)$ is also bounded for the entire time interval $t \in [0, e]$ for some $e > 0$ [16]. This can be achieved by utilizing the polynomials $p_{M,r}(t, \xi_{i,0})$ and $p_{m,r}(t, \xi_{i,0})$, where r represents the number of integrators that relate $\ddot{q}_i$ to $\dot{q}_i$ and q_i. The bounds can be defined as follows:

[3] We skip the computation for the upper bound which is trivial

Algorithm 4 Joint-Limits-Position-Invariance

function JLPINV($q_{m,i}, q_{M,i}, \ddot{q}_{m,i}, \ddot{q}_{M,i}, dt$)
  $a = dt^2$
  $b = dt(2\dot{q}_i - \ddot{q}_{m,i}dt)$
  $c = \dot{q}_i^2 + 2\ddot{q}_{m,i}(q_{M,i} - q_i - dt\dot{q}_i)$
  $\Delta = b^2 - 4ac$
  if $\Delta \geq 0$ **then**
    $b1_{u,i} = \frac{-b+\sqrt{\Delta}}{2a}$, $b1_{l,i} = \frac{-b-\sqrt{\Delta}}{2a}$
  else
    $b1_{u,i} = \frac{-b}{2a}$, $b1_{l,i} = \frac{-b}{2a}$
  $b = dt(2\dot{q}_i - \ddot{q}_{M,i}dt)$
  $c = \dot{q}_i^2 - 2\ddot{q}_{M,i}(q_i + dt\dot{q}_i - q_{m,i})$
  $\Delta = b^2 - 4ac$
  if $\Delta \geq 0$ **then**
    $b2_{u,i} = \frac{-b+\sqrt{\Delta}}{2a}$, $b2_{l,i} = \frac{-b-\sqrt{\Delta}}{2a}$
  else
    $b2_{u,i} = \frac{-b}{2a}$, $b2_{l,i} = \frac{-b}{2a}$
  $\mathbf{b}_{l,i} = \max(\left[b1_{l,i}, b2_{l,i}\right])$
  $\mathbf{b}_{u,i} = \min([b1_{u,i}, b2_{u,i}])$
  return $\left[b_{l,i}, b_{u,i}\right]$

Algorithm 5 Joint-Limits-Invariance

procedure JLINV($\mathbf{q}_m, \mathbf{q}_M, \dot{\mathbf{q}}_m, \dot{\mathbf{q}}_M, \ddot{\mathbf{q}}_m, \ddot{\mathbf{q}}_M$)
  for $i = 0, \ldots, n-1$:
    $\left[\ddot{q}_{m,p}, \ddot{q}_{M,p}\right]$ = JLPINV($q_{m,i}, q_{M,i}, \ddot{q}_{m,i}, \ddot{q}_{M,i}, dt$)
  $[\ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_{M,v}]$ = (3)
  $\mathbf{b}_l = \max([\ddot{\mathbf{q}}_{m,p}, \ddot{\mathbf{q}}_{m,v}, \ddot{\mathbf{q}}_m])$
  $\mathbf{b}_u = \min(\left[\ddot{\mathbf{q}}_{M,p}, \ddot{\mathbf{q}}_{M,v}, \ddot{\mathbf{q}}_M\right])$
  for $i = 0, \ldots, n-1$:
    $\left[b_{l,i}, b_{u,i}\right]$ = checkFlippingBounds($b_{l,i}, b_{u,i}, \ddot{q}_{m,i}, \ddot{q}_{M,i}$)
  return $[\mathbf{b}_l, \mathbf{b}_u]$

$$
\begin{aligned}
\ddot{q}_{m,i} \leq & \ddot{q}_i \leq \ \ddot{q}_{M,i}; \\
t\ddot{q}_{m,i} + \dot{q}_{i,0} = p_{m,1}(t, \xi_{i,0}) \leq & \ \dot{q}_i \leq p_{M,1}(t, \xi_{i,0}) = t\ddot{q}_{M,i} + \dot{q}_{i,0}; \\
\frac{t^2}{2}\ddot{q}_{m,i} + t\dot{q}_{i,0} + q_{i,0} = p_{m,2}(t, \xi_{i,0}) \leq & \ q_i \leq p_{M,2}(t, \xi_{i,0}) = \frac{t^2}{2}\ddot{q}_{M,i} + t\dot{q}_{i,0} + q_{i,0}.
\end{aligned} \tag{22}
$$

The upper and lower admissible accelerations to ensure compliance with the velocity limits are straightforward. The constraint takes the form (3). Concerning the position bounds, let $\Psi_{M,2}$ and $\Psi_{m,2}$ denote the maximum and the minimum of the bounding polynomials $p_{M,2}(t, \xi_{i,0})$ and $p_{m,2}(t, \xi_{i,0})$ for $t \geq 0$, i.e.,

$$\begin{aligned} \Psi_{M,2} &:= \max_{t\geq 0} |p_{M,2}(t,\xi_{i,0})| \quad \Rightarrow \quad \Psi_{M,2} = \begin{cases} q_{i,0} & \text{if } \dot{q}_{i,0} < 0 \\ -\frac{\dot{q}_{0,i}^2}{2\ddot{q}_{M,i}} + q_{i,0} & \text{if } \dot{q}_{i,0} > 0 \end{cases}; \\ \Psi_{m,2} &:= \min_{t\geq 0} |p_{m,2}(t,\xi_{i,0})| \quad \Rightarrow \quad \Psi_{m,2} = \begin{cases} q_{i,0} & \text{if } \dot{q}_{i,0} > 0 \\ -\frac{\dot{q}_{0,i}^2}{2\ddot{q}_{m,i}} + q_{i,0} & \text{if } \dot{q}_{i,0} < 0 \end{cases}. \end{aligned} \tag{23}$$

Considering (20), (22) and (23), we can derive

$$\begin{aligned} \Psi_{m,2} - q_{M,i} \leq q_i - q_{M,i} \leq \Psi_{M,2} - q_{M,i} \leq 0, \\ 0 \leq \Psi_{m,2} - q_{m,i} \leq q_i - q_{m,i} \leq \Psi_{M,2} - q_{m,i}. \end{aligned} \tag{24}$$

To ensure the respect of the upper joint position limit, i.e., $q_i - q_{M,i} \leq 0$, the condition $\Psi_{M,2} - q_{M,i} \leq 0$ is sufficient. It guarantees that even when accelerating at the maximum rate $\ddot{q}_{M,i}$, we remain within the feasible set $\mathcal{L}$. However, this condition is overly strict. In fact, when approaching the upper limit $q_{M,i}$, our primary concern is whether a feasible acceleration $\ddot{q}_i$ exists that complies with the boundary. To address this, we can consider the necessary condition $\Psi_{m,2} - q_{M,i} \leq 0$ to identify an acceleration $\ddot{q}_i$ that satisfies this inequality even in the subsequent iteration, for the state $(q_i + dt\ \dot{q}_i + \frac{dt^2}{2}\ddot{q}_i, \dot{q}_i + dt\ddot{q}_i)$, as described by (2) and (1) [13]. Assuming $\ddot{q}_{m,i} < 0$, when $\ddot{q}_i \leq -\frac{\dot{q}_i}{dt}$, the necessary condition $\Psi_{m,2} - q_{M,i} \leq 0$ becomes

$$(\dot{q}_i + dt\ddot{q}_i)^2 \leq -2\ddot{q}_{m,i}\left(q_{M,i} - q_i - dt\ \dot{q}_i - \frac{dt^2}{2}\ddot{q}_i\right), \tag{25}$$

which simplifies to $a\ddot{q}_i^2 + b\ddot{q}_i + c \leq 0$. This quadratic inequality allows us to determine both an upper and lower bound for $\ddot{q}_i$. If $\ddot{q}_i \geq -\frac{\dot{q}_i}{dt}$, the necessary condition is

$$q_i + dt\ \dot{q}_i + \frac{dt^2}{2}\ddot{q}_i - q_{M,i} \leq 0, \quad \ddot{q}_i \leq \frac{2(q_{M,i} - q_i - dt\dot{q}_i)}{dt^2}. \tag{26}$$

For small sampling times dt, this case occurs for $\ddot{q}_i > 0$, and the term $\frac{2(q_{M,i} - q_i - dt\dot{q}_i)}{t^2}$ is significantly larger than $\ddot{q}_{M,i}$, making this limit uninteresting to be introduced. Similarly, for the lower position limit, we consider the necessary condition $\Psi_{M,2} - q_{m,i} \geq 0$. Given $\ddot{q}_{M,i} > 0$, when $\ddot{q}_i \geq -\frac{\dot{q}_i}{dt}$, this condition can be expressed as

$$(\dot{q}_i + dt\ddot{q}_i)^2 \leq 2\ddot{q}_{M,i}\left(q_i + dt\ \dot{q}_i + \frac{dt^2}{2}\ddot{q}_i - q_{m,i}\right). \tag{27}$$

This inequality leads to additional upper and lower bounds for $\ddot{q}_i$. When $\ddot{q}_i \leq -\frac{\dot{q}_i}{dt}$, equivalent considerations to the upper limit case apply. The extension of this approach to a larger sampling time is discussed in [4]. The implementation details can be found in Appendix 5. The pseudo-code Algorithm 4

implements the computation of the bounds related to the position part. Combining the position and velocity constraints, we obtain Algorithm 5.

3 Numerical Evaluation and Comparison

This section numerically evaluates and compare the constraints introduced in Section 2 on the basis of the tracking of a reference joint position trajectory. For a robot with n Degrees of Freedom (DoFs), we formulate a QP problem to be solved in the following form:

$$\begin{aligned} &\min_{\ddot{\mathbf{q}}} \ \|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_r\|_2^2 \\ &\text{s.t.} \quad \mathbf{b}_l \leq \ddot{\mathbf{q}} \leq \mathbf{b}_u, \end{aligned} \tag{28}$$

with $\ddot{\mathbf{q}} \in \mathbb{R}^n$ the joint accelerations, and $\mathbf{b}_l, \mathbf{b}_u \in \mathbb{R}^n$ the lower and upper bounds, respectively, computed using one of the methods introduced in Section 2. The acceleration references $\ddot{\mathbf{q}}_r \in \mathbb{R}^n$ are computed as

$$\ddot{\mathbf{q}}_r = \mathbf{K}(\mathbf{q}_d - \mathbf{q}) - \mathbf{D}\dot{\mathbf{q}}, \tag{29}$$

with $\mathbf{K} = 5000\mathbf{I}$ and $\mathbf{D} = 2\sqrt{\mathbf{K}}$. $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix.

The numerical comparison has been performed using the OpenSoT [15] control framework. The resulting joint accelerations from the optimization are integrated twice to compute joint velocities and positions. The QP solver we use for the comparison is qpOASES [6] which uses an online active-set strategy to handle constraints. For the seek of simplicity, we consider the comparison taking into account a single joint with $q_m = -0.31$ $[rad]$, $q_M = 2.09$ $[rad]$, $\dot{q}_M = 3.14$ $[\frac{rad}{sec}]$, and $\ddot{q}_M = 20.$ $[\frac{rad}{sec^2}]$. In this numerical example, the reference is a sinusoidal position trajectory in the form:

$$\mathbf{q}_d = 3\sin(k\delta t), \tag{30}$$

with $\delta t = 0.001$ $[s]$, forcing the amplitude to overcome the upper and lower joint position limits.

We first compare the PSAP approach with different values of the p parameter. As shown in Figure 1a, increasing the p value permits the reduction of the oscillations which occur when the position bounds are exceeded, while velocity and acceleration bounds are always kept. It is worth noticing that unfeasibility in this situation is avoided thanks to the checkFlippingBounds() function. Increasing the p parameter, however, leads to a deteriorated tracking of the position reference. We selected $p = 100$ as a value to compare with the other methods, leading to a constraint satisfaction of 0.021 $[rad]$.

The same comparison is made for the CBF approach, considering the following simplification over the parameters: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha$. Figure 1b

reports the same numerical evaluation, according to different parameter α values. It is worth noticing the different behavior of the velocity and acceleration in order to avoid oscillations when the position bound is exceeded (yellow line in Figure 1b), while the similar behavior w.r.t. the PSAP, when the bounds are kept. We selected $\alpha = 15$ as a value to compare with the other methods, leading to a constraint satisfaction $\leq$ 1e$-$4 $[rad]$. Finally, we evaluate the Invariance approach. The basic implementation does not require any tuning, however, chattering in the joint acceleration appears when hitting the joint limits, as shown in Figure 1c. As reported in [4], such oscillations can be reduced by applying the PSAP. It is enough to have $p = 2$ to make chattering disappear entirely, leading to a constraint satisfaction of $\leq$ 1e$-$4 $[rad]$. In Figure 1d are reported the tracking errors of the compared approaches. As expected, the PSAP approach shows the worst performance in tracking the given trajectories, other than not being able to keep the joint position constraints. The CBF performs slightly better in terms of tracking with the advantage of being able to keep the constraint. Finally, the Invariance shows the best performance in both tracking and the ability to keep the constraint.

4 Experimental Results

This section implements in a practical use case the best set of constraints that emerged in Section 3, i.e., the PSAP-aided Invariance approach, with $p = 2$. For this purpose, we implemented a QP controller to perform a pick-and-place task using a UR5e[4] cobot. The primary aim of the controller is to guide the manipulator through a series of four sequential Cartesian poses. The initial pose, referred to as the *home* position, is selected taking into account factors such as the robot's dimensions and workspace. Subsequently, the *picking* pose is determined utilizing an artificial vision system [9] [1]. The selection of the *placing* pose considers the location of a designated storage cell. Finally, the task is concluded by returning the manipulator to the home position. The trajectories followed interpolate the mentioned poses using splines. Additionally, in the transition between the picking and placing positions, we incorporate an intermediate state to facilitate the movement. The step-by-step process involved in the pick and place task is shown in Figure 2.

In this experiment, we consider the following QP problem:

$$\begin{aligned} &\min_{\ddot{\mathbf{q}}} \ \|\mathbf{J}\ddot{\mathbf{q}} - \ddot{\mathbf{x}}_r + \dot{\mathbf{J}}\dot{\mathbf{q}}\|_2^2 \\ &\text{s.t.} \quad \mathbf{b}_l^{inv} \leq \ddot{\mathbf{q}} \leq \mathbf{b}_u^{inv}. \end{aligned} \tag{31}$$

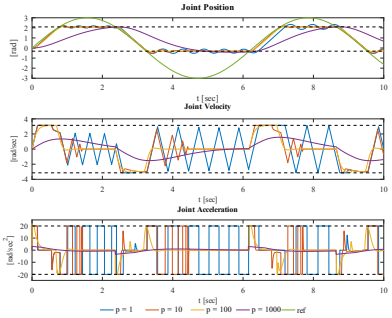
As before, here $\ddot{\mathbf{q}} \in \mathbb{R}^6$ represents the joint accelerations, while $\mathbf{b}_l^{inv} \in \mathbb{R}^6$ and $\mathbf{b}_u^{inv} \in \mathbb{R}^6$ correspond to the lower and upper bounds, respectively, computed

[4] `https://www.universal-robots.com/products/ur5-robot/`

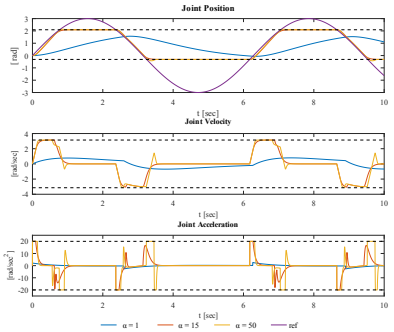
using the Invariance approach described in Section 2. The computation of the Cartesian acceleration reference $\ddot{\mathbf{x}}_r \in \mathbb{R}^6$ is expressed as follows:

$$\ddot{\mathbf{x}}_r = \mathbf{K}(\mathbf{x}_d \ominus \mathbf{x}) - \mathbf{D}\dot{\mathbf{x}}, \tag{32}$$

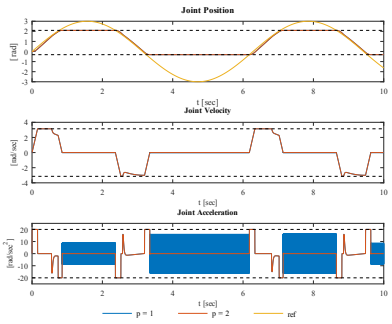
where $\mathbf{K} = 10\mathbf{I}$, $\mathbf{D} = 2\sqrt{\mathbf{K}}$, and $\mathbf{I} \in \mathbb{R}^{6\times 6}$ denotes the identity matrix. The difference operator $\ominus$ determines the algebraic difference between the translation components of the Cartesian poses $\mathbf{x}_d$ and $\mathbf{x}$. It also computes the rotation error denoted as $\theta\phi \in \mathbb{R}^3$, where $\{\phi\ \theta\}$ represents the axis-angle representation of the quaternion difference $\xi_d\xi^*$. The joint accelerations obtained from the optimization are integrated twice to compute joint velocities and positions. These values are subsequently provided to the low-level position controllers of the cobot. The robot has been equipped with a vacuum gripper to ensure safe grasping of the target box. The box has intentionally



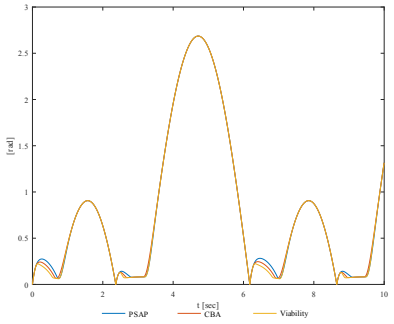
(a) Comparison between different values of parameter p using the PSAP.



(b) Comparison between different values of parameter α using the CBF.



(c) Comparison between Invariance approach ($p = 1$) and PSAP-aided Invariance ($p = 2$).



(d) Comparison between tracking error of PSAP, CBA, and Invariance approaches.

Fig. 1: Comparison between methods for the joint position tracking. Dashed lines highlight the joint position, velocity, and acceleration limits.

(a) Initial pose. (b) Home position. (c) Pick position.

(d) Box transfer. (e) Box release (f) Return home.

Fig. 2: Picture sequence of the pick and place task.

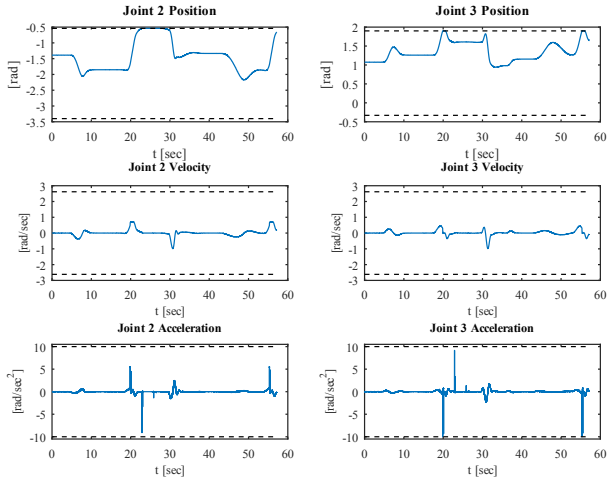


Fig. 3: The state of the second joint (left plots) and the third joint (right plots) throughout the pick and place task. The dashed lines represent conservative joint limits.

been placed in a location that is inconvenient and difficult to access. When moving towards and away from the picking position, there is a risk of self-collisions between the manipulator's links, between the manipulator and the base, or impacts with the picked box. To ensure the safety of the robot, we have carefully adjusted the original joint position limits. The QP controller is

capable of enforcing these limits, thereby preventing any collisions throughout the entire task. During the experiment, the second and third joints reach their upper position limits. The behavior of these joints is illustrated in Figure 3. It is worth noting that the joint boundaries are consistently satisfied throughout the task, with the joints only experiencing slight acceleration chattering during the initial moments when the limit is reached.

5 Conclusions and Future Works

In this work, we presented a comparison of methods for enforcing linear constraints with a relative degree greater than 1 in control problems based on QP optimization. Specifically, we compare the *P-Step Ahead Predictor*, *Control Barrier Function*, and *Invariance* methods. We benchmark these methods in joint limit constraints, including position, velocity, and acceleration, that are essential in tasks involving human-robot coexistence and cooperation. Based on our study, we select the P-step Ahead Predictor-aided Invariance method as the most effective in terms of tracking quality and constraint maintenance. Furthermore, we demonstrate the use of this method in a pick-and-place task performed with a UR5e robot. In future work, we plan to apply the selected method to more complex types of constraints with a relative degree greater than 1, such as virtual walls in Cartesian space and collision avoidance.

Appendix

In [4], the invariance or viability control approach is implemented, extending the method to accommodate larger sampling times dt. In this case, the key difference is that the velocity and position limits computed in Algorithm 5 in Section 2.3 are no longer sufficient to ensure compliance with the joint limits. To highlight this observation, we will derive the acceleration constraints from the position limits stated in (20). It can be demonstrated that for large sampling times dt, these constraints are more stringent compared to the ones obtained using the viability/invariance approach presented in Algorithm 4. Recalling (2), when considering $t \in [0, dt]$, the joint position q_i reaches its extreme, such as the maximum $q_{M,i}$, when its time derivative is zero. i.e., $\frac{\partial q_i}{\partial t} = \dot{q}_i + t\ddot{q}_i = 0$. The time instant $t_{\max}$ at which the maximum is reached is given by

$$t_{max} = -\frac{\dot{q}_i}{\ddot{q}_i}, \quad t \in [0, dt]. \tag{33}$$

If

$$\dot{q}_i > 0 \quad \wedge \quad \ddot{q}_i \leq -\frac{\dot{q}_i}{\delta t}, \tag{34}$$

then the maximum $q_M = q_i + t_{max}\dot{q}_i + \frac{1}{2}t_{max}^2\ddot{q}_i = q_i - \frac{\dot{q}_i^2}{2\ddot{q}_i}$ is reached inside the time step $[0, dt]$, leading to the constraint:

$$\ddot{q}_i \leq -\frac{\dot{q}_i^2}{2(q_{M,i} - q_i)}. \tag{35}$$

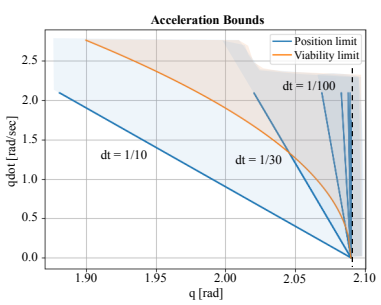
Instead, if the conditions in (34) are not met, the maximum is reached at the boundaries of the time step, in particular, the constraint needs to be enforced in $t = dt$, therefore:

$$\ddot{q}_i \leq \frac{2}{dt^2}(q_{M,i} - q_i - dt\dot{q}_i). \tag{36}$$

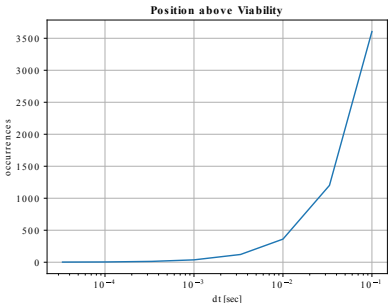
A similar analysis can be performed for the minimum position. The blue regions in Figure 4a represent the areas in the state space where the acceleration constraints take precedence over the ones derived from the viability/invariance approach. The dashed black line corresponds to the upper joint limit $q_{M,i} = 2.09$. The orange region beyond the viability limit is infeasible due to the viability/invariance constraints. The expansion of the blue region within the feasible area indicates that the position limits defined in (20) cannot be satisfied without imposing the conditions described by (35) and (36). Additionally, the acceleration constraints obtained from the alternative necessary condition (26) cannot be disregarded anymore, and it is necessary to incorporate them into the implementation. As the sampling time dt decreases, this region becomes progressively smaller, rendering the measures employed in [4] less significant. Figure 4b provides an illustration of the number of states sampled from the feasible region where the conditions (35) and (36) impose stricter constraints compared to those computed using Algorithm 4.

References

1. Amadio, F., Laghi, M., Raiano, L., Rollo, F., Zunino, A., Raiola, G., Ajoudani, A.: Target-referred dmps for learning bimanual tasks from shared-autonomy telemanipulation. In: IEEE-RAS Int. Conference on Humanoid Robots, pp. 496–503 (2022)
2. Ames, A.D., Grizzle, J.W., Tabuada, P.: Control barrier function based quadratic programs with application to adaptive cruise control. In: IEEE Conference on Decision and Control, pp. 6271–6278 (2014)
3. Caron, S., Kheddar, A., Tempier, O.: Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control. In: IEEE International conference on robotics and automation (ICRA), pp. 277–283 (2019)
4. Del Prete, A.: Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators. IEEE Robotics and Automation Letters **3**(1), 281–288 (2017)
5. Fang, C., Rocchi, A., Hoffman, E.M., Tsagarakis, N.G., Caldwell, D.G.: Efficient self-collision avoidance based on focus of interest for humanoid robots. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 1060–1066 (2015)
6. Ferreau, H., Kirches, C., Potschka, A., Bock, H., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Mathematical Programming Com-



(a) The blue lines delimit the regions where bounds from position inequalities take precedence. The orange line represents the feasible region based on the viability inequality. As highlighted, the viability condition alone may not guarantee compliance with joint position limits.



(b) Number of registered occurrences in which the acceleration limits computed by the position inequalities dominate over the ones coming from the viability inequality. As highlighted, the position inequalities become more relevant when the discretized time increases.

Fig. 4: Study on Viability and position constraints as proposed in [4]

putation **6**(4), 327–363 (2014)
7. Grandia, R., Taylor, A.J., Ames, A.D., Hutter, M.: Multi-layered safety for legged robots via control barrier functions and model predictive control. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 8352–8358 (2021)
8. Khazoom, C., Gonzalez-Diaz, D., Ding, Y., Kim, S.: Humanoid self-collision avoidance using whole-body control with control barrier functions. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 558–565 (2022)
9. Laghi, M., Raiano, L., Amadio, F., Rollo, F., Zunino, A., Ajoudani, A.: A target-guided telemanipulation architecture for assisted grasping. IEEE Robotics and Automation Letters **7**(4), 8759–8766 (2022)
10. Landi, C.T., Ferraguti, F., Costi, S., Bonfè, M., Secchi, C.: Safety barrier functions for human-robot interaction with industrial manipulators. In: European Control Conference (ECC), pp. 2565–2570 (2019)
11. Maderna, R., Casalino, A., Zanchettin, A.M., Rocco, P.: Robotic handling of liquids with spilling avoidance: A constraint-based control approach. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 7414–7420 (2018)
12. Park, K.C., Chang, P.H., Kim, S.H.: The enhanced compact qp method for redundant manipulators using practical inequality constraints. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 1, pp. 107–114 (1998)
13. Polverini, M.P., Nicolis, D., Zanchettin, A.M., Rocco, P.: Implicit robot force control based on set invariance. IEEE Robotics and Automation Letters **2**(3), 1288–1295 (2017)
14. Rauscher, M., Kimmel, M., Hirche, S.: Constrained robot control using control barrier functions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 279–285 (2016)
15. Rocchi, A., Hoffman, E.M., Caldwell, D.G., Tsagarakis, N.G.: Opensot: A whole-body control library for the compliant humanoid robot coman. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 6248–6253 (2015)
16. Wolff, J., Buss, M.: Invariance control design for nonlinear control affine systems under hard state constraints. IFAC Proceedings Volumes **37**(13), 555–560 (2004). IFAC Symposium on Nonlinear Control Systems (NOLCOS)