

Robust Satisfaction of Joint Position and Velocity Bounds in Discrete-Time Acceleration Control of Robot Manipulators

Erik Zanolli¹ and Andrea Del Prete¹

Abstract—This paper deals with the robust control of fully-actuated robots subject to joint position, velocity and acceleration bounds. Robotic systems are subject to *disturbances*, which may arise from modeling errors, sensor noises or communication delays. This work presents mathematical and computational tools to ensure the robust satisfaction of joint bounds in the control of robot manipulators. We consider a system subject to *bounded additive disturbances* on the control inputs, with constant joint position, velocity and acceleration bounds. We compute the *robust viability kernel*, which is the set of states such that, starting from any such state, it is possible to avoid violating the constraints in the future, despite the presence of disturbances. Then we develop an efficient algorithm to compute the range of feasible accelerations that allow the state to remain inside the robust viability kernel. Our derivation ensures the continuous-time robust satisfaction of the joint bounds, while considering the discrete-time nature of the control inputs. Tests are performed in simulation with a single joint and a 6-DOF robot manipulator, demonstrating the effectiveness of the proposed approach compared to other state-of-the-art methods.

I. INTRODUCTION

One of the main challenges of robotic systems is the fact that they are highly constrained. Robot joints typically have bounded positions. Moreover, actuators have bounded velocity, acceleration and torque/current. Velocity and acceleration may be not directly bounded, but their limits may be the consequence of other bounds, such as on the motor current, or the gear-box torque. This work focuses on the robust control of joints with constant bounds on position, velocity and acceleration. Even though velocity and acceleration bounds may be not constant, they are often approximated by constant values [12, 16, 7, 9, 14, 2, 15].

An additional challenge is that these systems are subject to disturbances, which may arise from modeling errors, sensor noises, or communication delays. Therefore we tackle the problem of robust control, considering bounded additive disturbances on the system inputs. The non-robust version of this problem has been already investigated in the literature [2, 3, 15]. Our main contribution is to extend the algorithm of [3], introducing robustness to bounded additive disturbances on the control inputs.

A. Notation

Let us introduce our notation:

- \wedge and \vee denote the logical quantifiers AND and OR.
- $t \in \mathbb{R}^+$ denotes time.

- $i \in \mathbb{N}$ denotes discrete time steps.
- δt is the time-step duration of the controller.
- $w(t)$ is the disturbance in continuous-time systems, while w_i is the disturbance in discrete-time systems.
- $q(t), \dot{q}(t), \ddot{q}(t) \in \mathbb{R}$ are the joint position, velocity and acceleration at time t .
- $q_i \triangleq q(i\delta t)$, $\dot{q}_i \triangleq \dot{q}(i\delta t)$, $\ddot{q}_i \triangleq \ddot{q}(i\delta t)$.
- q^{min}, q^{max} are the joint position boundaries.
- $\dot{q}^{max}, \ddot{q}^{max}$ are the maximum velocity and acceleration, implying the assumption of symmetric velocity and acceleration bounds.¹

B. State of the art

Assuming a constant acceleration throughout each time step, the future position q^+ and velocity \dot{q}^+ are functions of the current acceleration \ddot{q} :

$$\begin{aligned} q^+ &= q + \delta t \dot{q} + \frac{1}{2} \delta t^2 \ddot{q} \\ \dot{q}^+ &= \dot{q} + \delta t \ddot{q} \end{aligned} \quad (1)$$

A naive approach to bound accelerations is to compute the maximum and minimum \ddot{q} such that q^+ and \dot{q}^+ are within their bounds:

$$\begin{aligned} \ddot{q} &\leq \min \left(\ddot{q}^{max}, \frac{1}{\delta t} (\dot{q}^{max} - \dot{q}), \frac{2}{\delta t^2} (q^{max} - q - \delta t \dot{q}) \right) \\ \ddot{q} &\geq \max \left(-\ddot{q}^{max}, \frac{1}{\delta t} (-\dot{q}^{max} - \dot{q}), \frac{2}{\delta t^2} (q^{min} - q - \delta t \dot{q}) \right) \end{aligned} \quad (2)$$

This simple approach is unsatisfactory because the resulting bounds may be incompatible, leading to an unfeasible problem [12]. Several improvements have been proposed.

A common approach [12, 16] is to use a larger value of δt in (2), which helps reducing the acceleration when approaching a bound, but does not guarantee constraint compatibility. Another approach [7] is to bound velocities with a hand-tuned linear function of the distance to the position limit. While this is a sensible idea, this method does not explicitly account for acceleration limits. Control barrier functions provide a general framework for handling constraints [11, 13]. However, these methods do not deal with constraint conflicts, which are the key issue when considering bounds on position and acceleration [14].

Decre et al. [2] have been the first ones trying to provide formal guarantees of constraint satisfaction. Their method does not require any hand tuning, but it has two critical issues. First, they assumed constant velocity throughout the

¹The authors are with the Industrial Engineering Department of the University of Trento, Via Sommarive 11, 38123 Trento, Italy andrea.delprete@unitn.it

¹Our discussion can be easily extended to asymmetric velocity/acceleration bounds, but for the sake of clarity we prefer to focus on symmetric bounds.

time step, so they do not bound the acceleration, but a pseudo-acceleration, defined as $(\dot{q}^+ - \dot{q})/\delta t$. Second, their method may lead to conflicts between velocity and acceleration limits when getting close to the position bounds. This second issue was later addressed by Rubrecht et al. [14, 15]. However, they introduced some conservatism in the solution, and they also dealt with pseudo-acceleration bounds.

The viability-based approach developed in [3] solved the above-mentioned limitations. It is exact and does not introduce any type of arbitrary conservatism. It assumes constant acceleration between time steps, so it bounds the real acceleration. However, it cannot deal with disturbances. Our contribution is to develop a *robust* version of this method.

The problem of robust constraint satisfaction has been thoroughly investigated in the field of Robust Model Predictive Control (RMPC) [10], with most work dealing with linear *discrete-time* dynamical systems, subject to linear constraints [5, 6]. The system and the constraints considered in this paper are linear, so we could apply methods from RMPC. However, these methods can only guarantee the satisfaction of the constraints in *discrete time*, meaning that violations could still occur in-between time steps. Our method instead guarantees that constraints be robustly satisfied in *continuous time*, even though the control inputs can only be changed at discrete time steps. Moreover, while RMPC methods typically rely on complex *polytope projection* techniques (e.g. to compute Minkowski sums), which require the use of advanced (and often numerically brittle) software libraries [4], our approach boils down to a simple algebraic algorithm, easy to implement and fast to execute.

II. PROBLEM STATEMENT

A. Feasible states

Considering a robot with joint position and acceleration limits, the set of feasible states for a single joint is:

$$X = \{(q, \dot{q}) \in \mathbb{R}^2 : q^{\min} \leq q \leq q^{\max}, |\dot{q}| \leq \dot{q}^{\max}\} \quad (3)$$

As previously stated, our control inputs are the joint accelerations, which are bounded: $|\ddot{q}| \leq \ddot{q}^{\max}$.

B. Disturbance definition

We assume our system is subject to bounded additive disturbances on the inputs: $w(t) \in [-\bar{w}, \bar{w}]$. For the control of real-world systems we use generally discrete-time models. The disturbance in discrete time still holds the property of being bounded, $w_i \in [-\bar{w}, \bar{w}]$, but it remains constant throughout the whole time step i . Assuming constant acceleration and constant disturbance through the time step, the discrete-time dynamics are:

$$\begin{aligned} q_{i+1} &= q_i + \delta t \dot{q}_i + \frac{1}{2} \delta t^2 (\ddot{q}_i + w_i) \\ \dot{q}_{i+1} &= \dot{q}_i + \delta t (\ddot{q}_i + w_i) \end{aligned} \quad (4)$$

Notice that w_i does not influence the acceleration bounds: \ddot{q} is the commanded acceleration, to which we add w .

C. Problem formulation

Finding the maximum and minimum accelerations such that the state constraints can be satisfied in the future can be formulated as an infinite-horizon optimal control problem:

$$\begin{aligned} \ddot{q}_0^{\max} &= \underset{\ddot{q}_0, \ddot{q}_1, \dots}{\text{maximize}} \quad \ddot{q}_0 \\ \text{subject to} \quad & q(i\delta t + t) = q_i + t \dot{q}_i + \frac{t^2}{2} (\ddot{q}_i + w_i) \\ & \dot{q}(i\delta t + t) = \dot{q}_i + t (\ddot{q}_i + w_i) \\ & (q(i\delta t + t), \dot{q}(i\delta t + t)) \in X \\ & |\ddot{q}_i| \leq \ddot{q}^{\max} \\ & (q(0), \dot{q}(0)) \text{ fixed} \\ & \forall i \geq 0, \forall t \in [0, \delta t], \forall w_i \in [-\bar{w}, \bar{w}] \end{aligned} \quad (5)$$

The problem has an infinite number of constraints and variables, therefore it cannot be solved directly.

III. PROBLEM SOLUTION

The concept of viability will help us reformulate problem (5). A state is defined as viable if starting from that state there exists a sequence of control inputs that allows for the satisfaction of all constraints in the future. Formally, a state $(q(0), \dot{q}(0))$ belongs to the robust viability kernel \mathcal{V} if and only if, using that state as initial conditions for problem (5), the problem admits a solution. The main interest in the introduction of the robust viability kernel \mathcal{V} is that ensuring the existence of a feasible future trajectory (i.e., our original problem) is equivalent to ensuring that the next state belongs to \mathcal{V} . However, this definition of \mathcal{V} does not immediately provide practical utility: verifying its membership amounts to finding an infinite sequence of accelerations that results in a feasible trajectory, which is too computationally demanding. In the following, we derive an equivalent definition of \mathcal{V} that allows us to check membership easily. Thanks to this, we reformulate the hard problem of satisfying the position-velocity-acceleration limits as the simpler problem of ensuring that the next state is viable.

A. Continuous time control

In the beginning, let us assume to deal with a continuous system, in which \ddot{q} and w can change at any instant. This results in a set of viable states \mathcal{V}^C that is a superset of the previous one: $\mathcal{V} \subset \mathcal{V}^C$. It is obvious that a viable state is also feasible, but not all the feasible states are viable. For a given initial position q_0 , we can find the maximum initial velocity $\dot{q}_M^{\mathcal{V}}$ that allows us to satisfy the position limits in the future:

$$\begin{aligned} \dot{q}_M^{\mathcal{V}}(q_0) &= \underset{\dot{q}_0, \ddot{q}(t)}{\text{maximize}} \quad \dot{q}_0 \\ \text{subject to} \quad & \frac{d q(t)}{d t} = \dot{q}(t) \\ & \frac{d \dot{q}(t)}{d t} = \ddot{q}(t) + w(t) \\ & (q(t), \dot{q}(t)) \in X \\ & |\ddot{q}(t)| \leq \ddot{q}^{\max} \\ & q(0) = q_0, \quad \dot{q}(0) = \dot{q}_0 \\ & \forall t \geq 0, \forall w(t) \in [-\bar{w}, \bar{w}] \end{aligned} \quad (6)$$

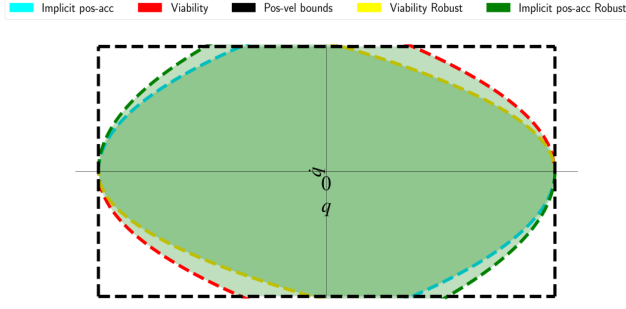


Fig. 1: Example of robust viability kernel with $\bar{w} = 0$ and with $\bar{w} = \frac{1}{3}\dot{q}^{max}$.

In the case without disturbance the solution of this problem is rather intuitive: the maximum initial velocity is such that, if we constantly apply the maximum deceleration, we end up exactly at q^{max} with zero velocity. Taking into account the disturbance $w(t)$, we can consider applying the maximum deceleration $-\ddot{q}^{max}$ and the maximum disturbance \bar{w} : this represents the worst case scenario because the deceleration is reduced by the disturbance. So we can:

- 1) write the position trajectory for the maximum deceleration and maximum disturbance $\ddot{q}(t) = -\ddot{q}^{max} + \bar{w}$,
- 2) compute the time at which the velocity of this trajectory is zero $t^0 = \dot{q}_0 / (\ddot{q}^{max} - \bar{w})$,
- 3) compute the initial velocity such that $q(t^0) = q^{max}$.

Following these steps we find:

$$\dot{q}_M^{\mathcal{V}}(q) = \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q)} \quad (7)$$

Following the same steps, we can define also the minimum velocity to ensure viability:

$$\dot{q}_m^{\mathcal{V}}(q) = -\sqrt{2(\ddot{q}^{max} - \bar{w})(q - q^{min})} \quad (8)$$

So the set \mathcal{V}^C of viable states can be written as:

$$\mathcal{V}^C = \{(q, \dot{q}) : (q, \dot{q}) \in X, \dot{q}_m^{\mathcal{V}}(q) \leq \dot{q} \leq \dot{q}_M^{\mathcal{V}}(q)\} \quad (9)$$

This definition of \mathcal{V}^C allows us to check easily the viability of a state by just verifying three inequalities. Fig. 1 shows how the viability kernel varies with the presence of disturbances.

B. Discrete-time control

While in continuous time the acceleration and the disturbance can change at any instant, in discrete time they remain constant for the whole time step. This means that if q reaches q^{max} with zero velocity, in continuous-time control we can immediately switch to zero also \ddot{q} . Instead, in discrete-time control we must keep applying a constant deceleration for the whole time step. Theoretically, this could lead to a violation of the lower position-velocity bounds. Let us understand in which cases this is possible. The worst case is represented by the system reaching the state $(q_0, \dot{q}_0) = (q^{max}, 0)$ a moment after the beginning of the time step with the maximum deceleration $\ddot{q}_0 = -\ddot{q}^{max}$ and the disturbance $w_0 = -\bar{w}$ that

increases the deceleration. The resulting deceleration is kept for the whole time step, resulting in:

$$\begin{cases} q_1 = q^{max} - 0.5\delta t^2(\ddot{q}^{max} + \bar{w}) \\ \dot{q}_1 = -\delta t(\ddot{q}^{max} + \bar{w}) \end{cases} \quad (10)$$

First of all, it is necessary that the value of \dot{q}_1 does not violate the lower velocity bound, so:

$$-\dot{q}^{max} \leq -\delta t(\ddot{q}^{max} + \bar{w}) \quad (11)$$

We can see this as a bound on \bar{w} :

$$\bar{w} \leq \frac{\dot{q}^{max} - \delta t\ddot{q}^{max}}{\delta t} \quad (12)$$

After the first time step the joint is approaching the lower position bound with velocity \dot{q}_1 . To ensure that the lower bound is not going to be reached, we must reach $\dot{q} \geq 0$. Therefore we must apply maximum acceleration $\ddot{q}_1 = \ddot{q}^{max}$, while the worst-case disturbance that can occur is $w_1 = -\bar{w}$, which decreases the acceleration.

$$\begin{cases} q_2 = q_1 + \dot{q}_1\delta t + 0.5\delta t^2(\ddot{q}^{max} - \bar{w}) \\ \dot{q}_2 = \dot{q}_1 + \delta t(\ddot{q}^{max} - \bar{w}) \end{cases} \quad (13)$$

After the substitution of q_1, \dot{q}_1 from (10) we get:

$$\begin{cases} q_2 = q^{max} - 0.5\delta t^2(\ddot{q}^{max} + 2\bar{w}) \\ \dot{q}_2 = -2\delta t\bar{w} \end{cases} \quad (14)$$

Due to the disturbances the velocity is still negative, so we apply again maximum acceleration $\ddot{q}_2 = \ddot{q}^{max}$, with disturbance $w_2 = -\bar{w}$, and analyze in first place the resulting velocity:

$$\dot{q}_3 = \delta t(\ddot{q}^{max} - 3\bar{w}) \quad (15)$$

After 3 time steps the velocity could be positive, depending on the value of \bar{w} . In the following we are going to assume that this is the case. This is not necessary, but it is reasonable because it results in a rather large upper bound for \bar{w} :

$$\delta t(\ddot{q}^{max} - 3\bar{w}) \geq 0 \Rightarrow \bar{w} \leq \frac{1}{3}\ddot{q}^{max} \quad (16)$$

The assumption of a positive \dot{q} after 3 time steps implies assuming that disturbances are not greater than one third of the maximum acceleration. It would be possible to deal with greater values of \bar{w} considering a change of velocity sign after a higher number of time steps, but we do not think this is necessary in practice. Analyzing the \dot{q} trajectory in continuous time in the third time step we can calculate the time when the velocity is null, i.e. the time when the position reaches its minimum value.

$$\begin{aligned} \dot{q}_3(t) &= \dot{q}_2 + t(\ddot{q}^{max} - \bar{w}) = 0 \\ t &= \frac{-\dot{q}_2}{\ddot{q}^{max} - \bar{w}} \Rightarrow t = \frac{2\delta t\bar{w}}{\ddot{q}^{max} - \bar{w}} \triangleq t_0 \end{aligned} \quad (17)$$

We can now substitute t_0 in the position equation and obtain the minimum value of the trajectory. This value is defined as q^{discr} and represents the maximum value of the lower bound such that we can neglect the fact that the controller operates in discrete time.

$$q_3(t_0) \triangleq q_{discr} = q_2 - \frac{4\delta t \bar{w}}{\ddot{q}^{max} - \bar{w}} \delta t \bar{w} + \frac{\delta t^2 \bar{w}^2}{\ddot{q}^{max} - \bar{w}} (\ddot{q}^{max} - \bar{w}) \quad (18)$$

The lower bound q^{min} must be less than or equal to q^{discr} :

$$q_{discr} = q^{max} - \delta t^2 \left(\ddot{q}^{max} + 2\bar{w} + \frac{\bar{w}^2}{\ddot{q}^{max} - \bar{w}} \right) \geq q^{min} \quad (19)$$

$$q^{max} - q^{min} \geq \delta t^2 \left(\ddot{q}^{max} + 2\bar{w} + \frac{\bar{w}^2}{\ddot{q}^{max} - \bar{w}} \right) \quad (20)$$

As expected, (20) is equal to the expression found in [3] when the disturbance $\bar{w} = 0$. Now we want to find the maximum value of \bar{w} that satisfies (20), which is a second-order polynomial of \bar{w} . We express the maximum disturbance as a percentage of \ddot{q}^{max} (i.e. $\bar{w} = \alpha \ddot{q}^{max}$, with $\alpha \in \mathbb{R}^+$), and substituting it in (20) we get:

$$\alpha \leq \alpha_1 \vee \alpha \geq \alpha_2$$

$$\alpha_{1,2} = \frac{k^2 + \ddot{q}^{max} q^r \pm \sqrt{(k^2 + \ddot{q}^{max} q^r)^2 - 4k^2(\ddot{q}^{max} q^r - k^2)}}{2k^2}, \quad (21)$$

where we have defined:

$$k \triangleq \ddot{q}^{max} \delta t, \quad q^r \triangleq q^{max} - q^{min} \quad (22)$$

We can simplify these bounds because the bound of α imposed by (16) can be stricter than both $\alpha_{1,2}$. At the same time, negative alpha values are not admissible. First, notice that $\alpha_2 > 0.5$ because its numerator is the sum of k^2 and a positive value (the square root plus $\ddot{q}^{max} q^r$), while its denominator is $2k^2$. Since $\alpha_2 > 0.5$, and we assumed that $\alpha \leq \frac{1}{3}$, we can discard the range $\alpha \geq \alpha_2$, and consider only the range $[0, \min(\alpha_1, 1/3)]$. Another bound on α comes from the velocity constraint (12):

$$\alpha \leq \frac{\ddot{q}^{max} - \delta t \ddot{q}^{max}}{\delta t \ddot{q}^{max}} \triangleq \alpha_3 \quad (23)$$

We have now considered all the contributions to define our final range for the allowable α :

$$\alpha \in \left[0, \min \left(\alpha_1, \frac{1}{3}, \alpha_3 \right) \right] \quad (24)$$

If α belongs to this range, then we can neglect that the controller works in discrete time. This greatly simplifies the problem, and we expect this assumption to be verified in most practical cases. Therefore, in the following we will assume that robust viability kernels in discrete time and continuous time coincide: $\mathcal{V} = \mathcal{V}^C$.

C. Reformulation in terms of viability

We have now obtained a formulation of \mathcal{V} so we can reformulate problem (5). Starting from the current state $(q(0), \dot{q}(0)) \in \mathcal{V}$ we need to compute the maximum value of \ddot{q} such that:

- 1) the next state $(q(\delta t), \dot{q}(\delta t)) \in \mathcal{V}$;
- 2) the entire trajectory leading to the next state belongs to the feasible set X .

The first condition alone is not sufficient, similarly to what happens in (17), because the trajectory between two viable states may violate a constraint. We can reformulate (5) as:

$$\begin{aligned} \ddot{q}_0^{max} = \text{maximize } \ddot{q} \\ \text{subject to } q(0) = q, \quad \dot{q}(0) = \dot{q} \\ q(t) = q + t\dot{q} + \frac{t^2}{2}(\ddot{q} + w) \\ \dot{q}(t) = \dot{q} + t(\ddot{q} + w) \\ (q(t), \dot{q}(t)) \in \mathcal{F} \\ \dot{q}_m^{\mathcal{V}}(q(\delta t)) \leq \dot{q}(\delta t) \leq \dot{q}_M^{\mathcal{V}}(q(\delta t)) \\ |\ddot{q}| \leq \ddot{q}^{max} \\ \forall t \in [0, \delta t], \forall w \in [-\bar{w}, \bar{w}] \end{aligned} \quad (25)$$

This problem is much simpler than the previous one: it has a single variable instead of an infinite sequence, and its constraints concern only the trajectory in the $[0, \delta t]$ interval, rather than in $[0, \infty]$. However, problem (25) is still hard to solve because the constraints are infinitely many and nonlinear. In the next three subsections we reformulate the inequality constraints of problem (25). Each one will give us a lower and an upper bound on \ddot{q} , which we will combine in Section III-G.

D. Position inequalities

The position bounds of (25) are:

$$q^{min} \leq q + t\dot{q} + \frac{1}{2}t^2(\ddot{q} + w) \leq q^{max} \quad \forall t \in [0, \delta t], \forall w \in [-\bar{w}, \bar{w}] \quad (26)$$

To be robust we need to guarantee the constraint satisfaction for the worst-case disturbance, which is \bar{w} for the upper bound and $-\bar{w}$ for the lower bound:

$$\begin{aligned} q^{min} \leq q + t\dot{q} + \frac{1}{2}t^2(\ddot{q} - \bar{w}) \quad \forall t \in [0, \delta t] \\ q + t\dot{q} + \frac{1}{2}t^2(\ddot{q} + \bar{w}) \leq q^{max} \quad \forall t \in [0, \delta t] \end{aligned} \quad (27)$$

Let us focus on the upper bound, and introduce a new variable $\gamma \triangleq \ddot{q} + \bar{w}$:

$$q + t\dot{q} + \frac{1}{2}t^2\gamma \leq q^{max} \quad \forall t \in [0, \delta t] \quad (28)$$

In this form, the constraint is equivalent to the associated constraint for the nominal case (i.e., assuming $\bar{w} = 0$). This means that we can use the algorithm developed in [3] (Alg. 1) to compute an upper bound for γ , and then convert it to a bound on \ddot{q} by simply subtracting \bar{w} from it. Similarly, for the lower position limit we can define a new variable $\gamma \triangleq \ddot{q} - \bar{w}$, compute its lower bound using the algorithm for the nominal case [3], and finally convert it to a bound on \ddot{q} by adding \bar{w} to it. The computation is summarized in Alg. 1.

E. Velocity inequalities

The velocity trajectory is a line and so we need to verify only that the bounds are satisfied for $t = \delta t$:

$$|\dot{q} + \delta t(\ddot{q} + w)| \leq \dot{q}^{max} \quad \forall w \in [-\bar{w}, \bar{w}] \quad (29)$$

Since \bar{w} is the worst-case disturbance for the upper bound, rearranging (29) we obtain:

$$\ddot{q} \leq \frac{\dot{q}^{max} - \dot{q}}{\delta t} - \bar{w} \quad (30)$$

As for the position inequalities, considering the disturbance boils down to simply subtracting \bar{w} to the acceleration limit computed without disturbance. This analysis stands also for the lower bound, so we can write both bounds as:

$$\frac{1}{\delta t}(-\dot{q}^{max} - \dot{q}) + \bar{w} \leq \ddot{q} \leq \frac{1}{\delta t}(\dot{q}^{max} - \dot{q}) - \bar{w} \quad (31)$$

F. Viability inequalities

Let us consider the upper bound of the viability inequality.

$$\dot{q}(\delta t) \leq \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q(\delta t))} \quad \forall w \in [-\bar{w}, \bar{w}] \quad (32)$$

Assuming a worst-case disturbance, which for the upper bound is \bar{w} , we get:

$$\dot{q} + \delta t(\ddot{q} + \bar{w}) \leq \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\dot{q} - 0.5\delta t^2(\ddot{q} + \bar{w}))} \quad (33)$$

This constraint is clearly nonlinear in \ddot{q} , but we can derive an algorithm to reformulate it as a simple upper bound. First, since \ddot{q} always appears together with \bar{w} , we can introduce a new variable $\gamma \triangleq \ddot{q} + \bar{w}$ and rewrite (33) as:

$$\dot{q} + \delta t\gamma \leq \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\dot{q} - 0.5\delta t^2\gamma)} \quad (34)$$

Since the right-hand-side (RHS) of (34) is a square root, it is always positive, so if the left-hand-side (LHS) is negative (34) is always satisfied:

$$\dot{q} + \delta t\gamma \leq 0 \Leftrightarrow \gamma \leq -\frac{\dot{q}}{\delta t} \triangleq \gamma^1 \quad (35)$$

Instead if the LHS is positive we can square both sides:

$$(\dot{q} + \delta t\gamma)^2 \leq 2(\ddot{q}^{max} - \bar{w}) \left(q^{max} - q - \delta t\dot{q} - \frac{1}{2}\delta t^2\gamma \right) \quad (36)$$

This can be reformulated as $a\gamma^2 + b\gamma + c \leq 0$, with:

$$\begin{cases} a = \delta t^2 \\ b = 2\delta t\dot{q} + \delta t^2(\ddot{q}^{max} - \bar{w}) \\ c = \dot{q}^2 - 2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\dot{q}) \end{cases} \quad (37)$$

If $\Delta = b^2 - 4ac \geq 0$ this parabola is equal to zero in two points γ_2 and γ_3 (which coincide if $\Delta = 0$):

$$\gamma_2 \triangleq \frac{-b - \sqrt{\Delta}}{2a}, \quad \gamma_3 \triangleq \frac{-b + \sqrt{\Delta}}{2a} \quad (38)$$

Since $a > 0$ by definition, the inequality (36) is satisfied for $\gamma_2 \leq \gamma \leq \gamma_3$. If instead $\Delta < 0$ then the parabola is always positive and there exists no value of γ that satisfies (36). Putting it all together we can rewrite (34) as:

$$(\gamma \leq \gamma_1) \vee [(\gamma > \gamma_1) \wedge (\Delta \geq 0) \wedge (\gamma_2 \leq \gamma \leq \gamma_3)] \quad (39)$$

We can now analyze the different scenarios depending on the values of γ_1, γ_2 and γ_3 . Here we have only three possible orders because by definition $\gamma_2 \leq \gamma_3$. The case $\gamma_1 < \gamma_2 < \gamma_3$

Algorithm 1 accBoundsFromPosLimits

Require: $q, \dot{q}, q^{min}, q^{max}, \delta t, \bar{w}$

```

 $\ddot{q}_1^M \leftarrow -\dot{q}/\delta t$ 
 $\ddot{q}_2^M \leftarrow -\dot{q}^2/(2(q^{max} - q))$ 
 $\ddot{q}_3^M \leftarrow 2(q^{max} - q - \delta t\dot{q})/(\delta t^2)$ 
 $\ddot{q}_2^m \leftarrow \dot{q}^2/(2(q - q^{min}))$ 
5:  $\ddot{q}_3^m \leftarrow 2(q^{min} - q - \delta t\dot{q})/(\delta t^2)$ 
if  $\dot{q} \geq 0$  then
   $\ddot{q}^{LB} \leftarrow \ddot{q}_3^m$ 
  if  $\ddot{q}_3^M > \ddot{q}_1^M$  then
     $\ddot{q}^{UB} \leftarrow \ddot{q}_3^M$ 
10: else
   $\ddot{q}^{UB} \leftarrow \min(\ddot{q}_1^M, \ddot{q}_2^M)$ 
else
   $\ddot{q}^{UB} \leftarrow \ddot{q}_3^M$ 
  if  $\ddot{q}_3^m < \ddot{q}_1^M$  then
     $\ddot{q}^{LB} \leftarrow \ddot{q}_3^m$ 
15: else
   $\ddot{q}^{LB} \leftarrow \max(\ddot{q}_1^M, \ddot{q}_2^m)$ 
return  $\{ \ddot{q}^{LB} + \bar{w}, \ddot{q}^{UB} - \bar{w} \}$ 
```

Algorithm 2 accBoundsFromViability

Require: $q, \dot{q}, q^{min}, q^{max}, \ddot{q}^{max}, \delta t, \bar{w}$

```

 $a \leftarrow \delta t^2$ 
 $b \leftarrow \delta t(2\dot{q} + (\ddot{q}^{max} - \bar{w})\delta t)$ 
 $c \leftarrow \dot{q}^2 - 2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\dot{q})$ 
 $\gamma_1 \leftarrow -\dot{q}/\delta t$ 
5:  $\Delta \leftarrow b^2 - 4ac$ 
if  $\Delta \geq 0$  then
   $\gamma^{UB} \leftarrow \max(\gamma_1, (-b + \sqrt{\Delta})/(2a))$ 
else
   $\gamma^{UB} \leftarrow \gamma_1$ 
10:  $b \leftarrow 2\delta t\dot{q} - (\ddot{q}^{max} - \bar{w})\delta t^2$ 
   $c \leftarrow \dot{q}^2 - 2(\ddot{q}^{max} - \bar{w})(q + \delta t\dot{q} - q^{min})$ 
   $\Delta \leftarrow b^2 - 4ac$ 
if  $\Delta \geq 0$  then
   $\gamma^{LB} \leftarrow \min(\gamma_1, (-b - \sqrt{\Delta})/(2a))$ 
15: else
   $\gamma^{LB} \leftarrow \gamma_1$ 
   $\{\ddot{q}^{LB}, \ddot{q}^{UB}\} \leftarrow \{\gamma^{LB} + \bar{w}, \gamma^{UB} - \bar{w}\}$ 
return  $\{ \ddot{q}^{LB}, \ddot{q}^{UB} \}$ 
```

results in a disconnected set of feasible accelerations, which is impossible because \mathcal{V} is connected. In the other two cases the bound is simply given by the maximum between γ_1 and γ_3 . Finally, we can convert these bounds on γ to bounds on \ddot{q} by simply subtracting \bar{w} from them. Repeating the same analysis for the lower bound, we get the computation summarized by Alg. 2.

G. Final Algorithm

To conclude, we need to take into account all the bounds in a unique algorithm, which is summarized by Alg. 3. Fig. 2

Algorithm 3 Compute Joint Acceleration Bounds

Require: $q, \dot{q}, q^{\min}, q^{\max}, \ddot{q}^{\max}, \ddot{q}^{\min}, \delta t, \bar{w}$
 $\ddot{q}^{UB} \leftarrow [0, 0, 0, \ddot{q}^{\max}]$
 $\ddot{q}^{LB} \leftarrow [0, 0, 0, -\ddot{q}^{\max}]$
 $(\ddot{q}^{LB}[0], \ddot{q}^{UB}[0]) \leftarrow \text{accBoundsFromPosLimits}(\dots)$
 $\ddot{q}^{LB}[1] \leftarrow (-\ddot{q}^{\max} - \dot{q})/\delta t + \bar{w}$
5: $\ddot{q}^{UB}[1] \leftarrow (\ddot{q}^{\max} - \dot{q})/\delta t - \bar{w}$
 $(\ddot{q}^{LB}[2], \ddot{q}^{UB}[2]) \leftarrow \text{accBoundsFromViability}(\dots)$
return $\{\max(\ddot{q}^{LB}), \min(\ddot{q}^{UB})\}$

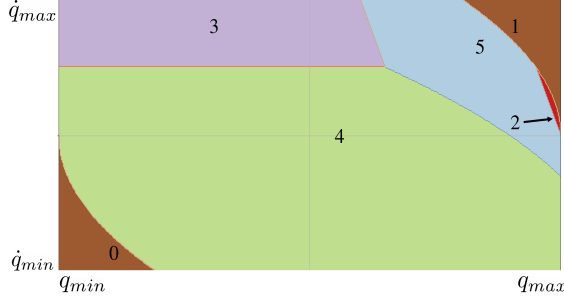


Fig. 2: Feasible state space for $q^{\max} = -q^{\min} = 0.5$ rad, $\dot{q}^{\max} = 2$ rad/s, $\ddot{q}^{\max} = 10$ rad/s², $\delta t = 0.1$ s, and $\bar{w} = \frac{1}{3}\ddot{q}^{\max}$. Region 0 represents the unreachable space, while region 1 represents the space that is not viable. In each of the other four regions, a different acceleration upper bound dominates the others. In region 2 it is the one coming from the position inequality (26). In region 3 it is the one coming from the velocity inequality (31). In region 4 it is the acceleration upper bound \ddot{q}^{\max} . In region 5 it is the one coming from the viability inequality (32).

shows the state space divided in regions based on which acceleration upper bound dominates the others.

IV. TESTS

This section analyzes the behavior of two robotic systems using the proposed algorithm to ensure the robust satisfaction of joint limits. First, we test our method on a single joint, and then on the 6 degree-of-freedom robot Baxter. We represent the maximum disturbance at joint j as $\bar{w}_j = \alpha \ddot{q}_j^{\max}$, where \ddot{q}_j^{\max} is the maximum acceleration at joint j . We compare our method against i) the naive method described in Section I-B and ii) the non-robust viability method from [3]. In all our tests we have verified that the conditions derived in Section III-B, regarding the equivalence of discrete-time and continuous-time viability, were satisfied. The Python implementation of the presented algorithms can be found at <https://github.com/ErikZan/Robust-Joints-bounds-guarantee>.

1) *Dealing with non-viable states:* When the state is outside the viability kernel \mathcal{V} , the robust and non-robust viability algorithms may give inconsistent acceleration limits. Our main concern in these cases is to reach \mathcal{V} , which these algorithm may fail doing. Therefore, when a violation occurs we compute the acceleration bounds using Alg. 4, which is

Algorithm 4 Dealing with non-viable states

Require: $q, \dot{q}, q^{\min}, q^{\max}, \ddot{q}^{\max}, \ddot{q}^{\min}, \delta t, \bar{w}$
if $\text{IsStateViable}(\dots) == \text{False}$ **then**
if $(\dot{q} \geq 0 \wedge q \geq q^{\min}) \vee q \geq q^{\max}$ **then**
 $(\ddot{q}^{LB}, \ddot{q}^{UB}) \leftarrow (-\ddot{q}^{\max}, -\ddot{q}^{\max})$
else
 $(\ddot{q}^{LB}, \ddot{q}^{UB}) \leftarrow (\ddot{q}^{\max}, \ddot{q}^{\max})$
5: **return** $\{\ddot{q}^{LB}, \ddot{q}^{UB}\}$

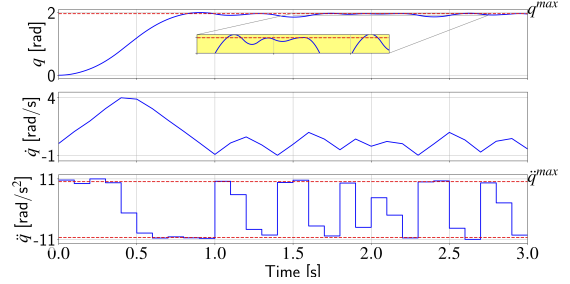


Fig. 3: Single joint, non-robust viability approach, random disturbance.

based on the following reasoning.

- If $\dot{q} > 0$ and $q > q^{\min}$ set $\ddot{q} = -\ddot{q}^{\max}$ to stop the joint as soon as possible.
- If $q > q^{\max}$ set $\ddot{q} = -\ddot{q}^{\max}$ to re-enter the viability kernel as soon as possible.
- If $\dot{q} < 0$ and $q < q^{\max}$ set $\ddot{q} = \ddot{q}^{\max}$ to stop the joint as soon as possible.
- If $q < q^{\min}$ set $\ddot{q} = \ddot{q}^{\max}$ to re-enter the viability kernel as soon as possible.

A. Single joint

This section deals with a single-joint robot. In these tests we try to reach the upper position bound without exceeding it or violating the velocity limit. The values used for this test are: $q^{\max} = 2$ rad, $q^{\min} = -2$ rad, $\dot{q}^{\max} = 5$ rad/s, $\ddot{q}^{\max} = 10$ rad/s², $\delta t = 0.1$ s. In the first test we always apply to the joint the maximum acceleration allowed (as computed by the algorithms). The system is subject to a uniformly-distributed random disturbance w_i . Fig. 3 shows that with the non-robust viability approach the position bound is violated—even though by a small amount—and the joint oscillates close to q^{\max} . Fig. 4 shows instead that with the robust viability approach the joint never reaches q^{\max} , but it comes close to it and then slightly fluctuates.

In the second test, shown in Fig. 5 and Fig. 6, the maximum disturbance \bar{w} is always applied. With the robust method the joint comes closer to the limit than it did in the previous test (Fig. 4), and then it starts oscillating. These oscillations are to be expected due to the discrete-time nature of the controller, and it was also observed in [3]. The non-robust method instead clearly violates the constraint.

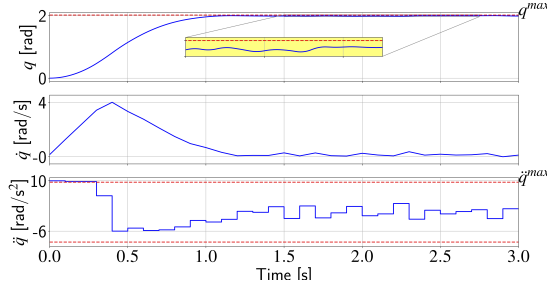


Fig. 4: Single joint, robust viability approach, random disturbance.

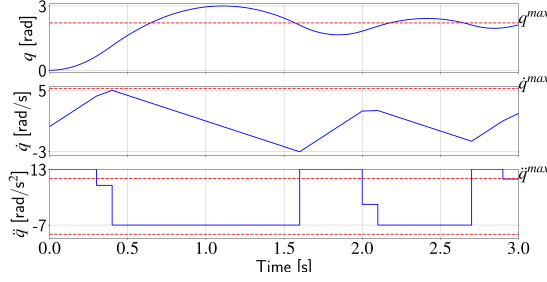


Fig. 5: Single joint, non-robust viability approach, worst-case disturbance \bar{w} .

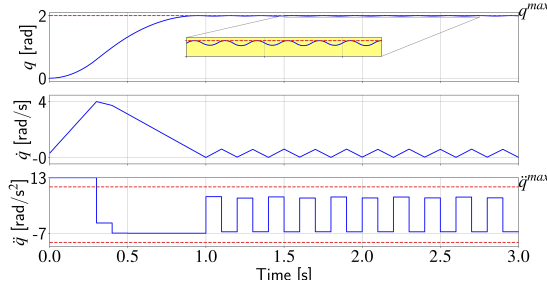


Fig. 6: Single joint, robust viability approach, worst-case disturbance \bar{w} .

B. 6-DoF Baxter robot

For these tests the joints have to reach an unfeasible position: q^d . In particular, we set q^d for the first joint, referred from now on as *joint 0*, 0.5 rad above its upper limit. The desired acceleration is computed with a *PD* control law.

$$\ddot{q}^d = k_P(q^d - q) - k_D\dot{q} \quad (40)$$

where $k_P, k_D \in \mathbb{R}^+$. The resulting acceleration (40) is saturated (if needed) based on the limits computed by the specified algorithm. The values used in this test are $k_P = 1000$, $k_D = 2\sqrt{k_P}$, and $\delta t = 0.05$ s. The applied disturbance w is the maximum for each joint, corresponding to $\bar{w} = \frac{1}{3}\ddot{q}^{max}$.

Fig. 7 shows that the naive method leads to a violation of both velocity and position constraints for joint 0, as expected. Also, the amplitude of fluctuations around q^{max} is strongly dependent on the values of k_P, k_D .

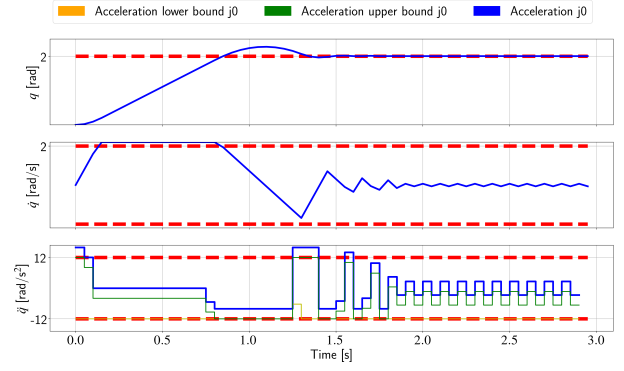


Fig. 7: Baxter robot, naive method.

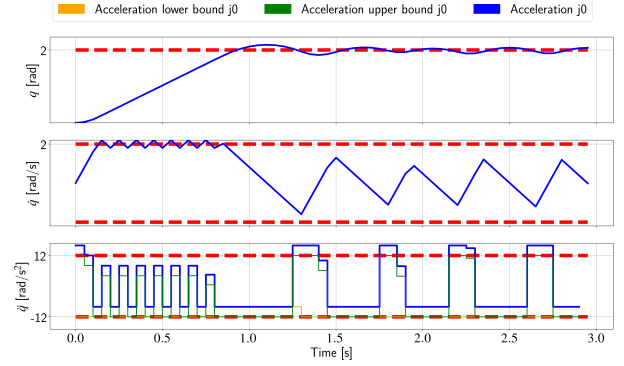


Fig. 8: Baxter robot, viability non-robust method.

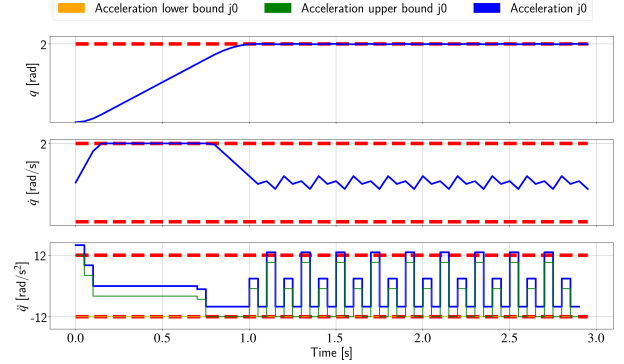


Fig. 9: Baxter robot, viability robust method.

Fig. 8 shows the results obtained with the non-robust viability algorithm. First the joint violates the velocity limit, and so the control chosen with Alg. 4 tries to lead the joint again in the viability kernel. This generates velocity fluctuations in the first part of the plot. Later, the joint violates the position bound, even though the violation is smaller with respect to the naive method.

Fig. 9 shows that with the robust viability algorithm the bounds are not violated, but the oscillations on acceleration in the final part are greater than in the naive or non-robust case. This behavior was expected, as it was documented also for the case without disturbance in the non-robust algorithm [3].

V. CONCLUSIONS AND FUTURE WORK

This paper focused on the robust control of robot manipulators subject to joint position, velocity and acceleration bounds. The problem had already been tackled in the literature [3], but without considering the presence of disturbances. This was a severe limitation because all physical systems are subject to some degree of uncertainty coming from, for instance, modeling errors, sensor noise and communication delays. Therefore, we have developed a new approach that can guarantee the satisfaction of the constraints despite the presence of bounded additive disturbances on the joint accelerations. The results obtained in simulation on a single joint and on the 6-degree-of-freedom Baxter robot arm show better performance with respect to other state-of-the-art methods, with our method being the only one capable of consistently ensuring constraint satisfaction. Moreover, the presented approach has similar computational complexity with respect to [3], making it easily usable for real-world applications. Nonetheless, this challenging problem is still far from being completely solved.

Even though the developed approach is exact (i.e. tight) for the case of bounded additive disturbances, one could prefer to model disturbances using random variables [8]. The *stochastic* approach to uncertainty modeling could indeed result in a less conservative behaviors, and thus improve performance. Moreover, joint acceleration bounds typically depend on the state, because they are induced by motor current bounds. Considering this dependence introduces a nonlinearity in the problem (either in the dynamics or in the constraints), making the computation of the viability kernel extremely challenging, even for the nominal (non-robust) case [1]. Finding computationally tractable methods to approximate these sets in the nonlinear setting is still an open problem, and an interesting direction for future work.

REFERENCES

- [1] J. M. Bravo et al. "On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach". In: *2003 European Control Conference (ECC)*. 2003, pp. 288–293. DOI: 10.23919/ECC.2003.7084969.
- [2] Wilm Decré et al. "Extending iTaSC to support inequality constraints and non-instantaneous task specification". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009.
- [3] A. Del Prete. "Joint Position and Velocity Bounds in Discrete-Time Acceleration/Torque Control of Robot Manipulators". In: *IEEE Robotics and Automation Letters* 3.1 (Jan. 2018), pp. 281–288.
- [4] Komei Fukuda and Alain Prodon. "Double Description Method Revisited". In: *Combinatorics and Computer Science*. 1996.
- [5] Reza Ghaemi, Jing Sun, and Ilya V. Kolmanovsky. "Less conservative robust control of constrained linear systems with bounded disturbances". In: *2008 47th IEEE Conference on Decision and Control* 3 (2008), pp. 983–988.
- [6] Reza Ghaemi, Jing Sun, and Ilya V. Kolmanovsky. "Robust control of constrained linear systems with bounded disturbances". In: *IEEE Transactions on Automatic Control* 57.10 (2012), pp. 2683–2688.
- [7] Fumio Kanehiro et al. "Integrating geometric constraints into reactive leg motion generation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010.
- [8] E. Kofman, J. A. De Doná, and M. M. Seron. "Probabilistic ultimate bounds and invariant sets for LTI systems with Gaussian disturbances". In: *2011 Australian Control Conference*. 2011, pp. 537–542.
- [9] Trosten Kröger and Friedrich M Wahl. "On-Line trajectory generation in robotic systems. Basic concepts for instantaneous reactions to unforeseen (sensor) events". In: *IEEE Transaction on Robotics* 26.1 (2010), pp. 94–111.
- [10] D. Q. Mayne, M. M. Seron, and S. V. Raković. "Robust model predictive control of constrained linear systems with bounded disturbances". In: *Automatica* 41.2 (2005), pp. 219–224.
- [11] Quan Nguyen and Koushil Sreenath. "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints". In: *American Control Conference*. 2016.
- [12] Ki Cheol Park, P. H. Chang, and Seung Ho Kim. "The Enhanced Compact QP Method for Redundant Manipulators Using Practical Inequality Constraints". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1998.
- [13] Manuel Rauscher, Melanie Kimmel, and Sandra Hirche. "Constrained Robot Control Using Control Barrier Functions". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
- [14] Sébastien Rubrecht et al. "Constraints Compliant Control : constraints compatibility and the displaced configuration approach". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010.
- [15] Sébastien Rubrecht et al. "Motion safety and constraints compatibility for multibody robots". In: *Autonomous Robots* 32.3 (2012), pp. 333–349.
- [16] Layale Saab et al. "Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints". In: *IEEE Transactions on Robotics* 29.2 (2013), pp. 346–362.