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

Erik Zanolli and Andrea Del Prete

***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

The control of robot manipulators is a well-understood and mostly solved problem. However, it is still challenging when these systems must operate in proximity of their joint limits. This is the case every time a robot moves at maximum speed, or reaches an object at the boundary of its workspace. These hard joint limits, encompassing position, velocity and acceleration, introduce non-trivial constraints in the control problem.

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 [1], considering bounded additive disturbances on the system inputs. The non-robust version of this problem has been already investigated in the literature [2], [3], [4]. Our main contribution is to extend the algorithm of [3], introducing robustness to bounded additive disturbances on the control inputs.

This work was supported by the PRIN Project DOCEAT under Grant CUP E63C22000410001. The authors are with the Industrial Engineering Department of the University of Trento, Via Sommarive 11, 38123 Trento, Italy. E-mail: andrea.delprete@unitn.it, erikzanolli@gmail.com



DOI 10.1109/IROS55552.2023.10341667

### 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.
- $\delta t$ 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 $t$.
- $q_i \triangleq q(i\delta t), \quad \dot{q}_i \triangleq \dot{q}(i\delta t), \quad \ddot{q}_i \triangleq \ddot{q}(i\delta t)$.
- $q^{min}, q^{max}$ are the joint position boundaries.
- $\dot{q}^{max}, \ddot{q}^{max}$ are the maximum velocity and acceleration.[1]

### B. State of the art

Assuming a double-integrator dynamics, the next position $q^+$ and velocity $\dot{q}^+$ depend on the current acceleration $\ddot{q}$:

$$\begin{aligned} q^+ &= q + \delta t\, \dot{q} + \frac{1}{2}\delta t^2 \ddot{q} \\ \dot{q}^+ &= \dot{q} + \delta t\, \ddot{q} \end{aligned} \tag{1}$$

A basic way to handle joint limits is to bound the acceleration $\ddot{q}$ so that $q^+$ and $\dot{q}^+$ remain feasible:

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

This simple approach is unsatisfactory because the resulting upper bound may be smaller than the lower bound, leading to an unfeasible problem [5]. Several improvements have been proposed.

A common approach [5], [6] is to use (2) with a larger value of $\delta t$; this mitigates the problem by decreasing $\ddot{q}$ when getting close to a limit, but does not suffice to guarantee constraint compatibility. Alternatively, one could bound the velocity using a linear function of the joint position [7]. 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 [10], [11]. However, they fail to address the key issue of constraint conflicts, and are therefore ineffective for the problem at hand [9].

A related (but more challenging) problem is the one of *collision avoidance*. Self-collision avoidance has been

[1] Velocity and acceleration limits are often approximated as constant even though they are actually state dependent [5], [6], [7], [8], [9], [2], [4]. Our discussion can be easily extended to asymmetric velocity/acceleration bounds, but for the sake of clarity we focus on symmetric bounds.

tackled with Vector Field Inequalities [12], which, similarly to control barrier functions, do not deal with constraint conflicts. [13] proposed a method to avoid collision ensuring strict bounds between the solution of the inverse kinematic problem and a reference collision-free joint trajectory. Collision avoidance for physical human-robot interaction has also been tackled using Nonlinear Model Predictive Control [14], together with a low-level feedback controller that guarantees joint limits [3].

Decre et al. [2] have been the first ones trying to provide formal guarantees of constraint satisfaction. Their method has two main issues. First, the assumption of constant velocity during the time step, so they do not bound the acceleration, but a pseudo-acceleration, defined as $(\dot{q}^+ - \dot{q})/\delta t$. Second, the potential conflicts between velocity and acceleration limits when approaching position limits. The latter problem was then addressed by Rubrecht et al. [9], [4], but at the price of a conservative solution.

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) [15], with most work dealing with linear *discrete-time* dynamical systems, subject to linear constrains [16], [17]. 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 [18], 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 velocity limits, the set of feasible states for a single joint is:

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

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 an additive disturbances on the inputs $w$, which is bounded by its maximum value $\bar{w}$, i.e. $w(t) \in [-\bar{w}, \bar{w}]$. 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:

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

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

*1) Mapping disturbance sources:* The disturbances $w$ could be used to account for different disturbance sources, such as sensor noises, modelling errors, and communication delays. For instance, assuming bounded noise on the joint position-velocity measurements/estimations:

$$q^{meas} = q + w_q, \qquad \dot{q}^{meas} = \dot{q} + w_v \tag{5}$$

with $|w_q| \leq \bar{w}_q$ and $|w_v| \leq \bar{w}_v$. Assuming a PD control law:

$$\begin{aligned} \ddot{q} &= k_p(q^d - q^{meas}) + k_d(\dot{q}^d - \dot{q}^{meas}) \\ &= k_p(q^d - q) + k_d(\dot{q}^d - \dot{q}) - k_p w_q - k_d w_v \end{aligned} \tag{6}$$

The errors due to $w_q/w_v$ can be mapped to errors in $w$ with bound $\bar{w} = k_p \bar{w}_q + k_d \bar{w}_v$. In case of a nonlinear control law, such as an inverse dynamics controller, the mapping from $w_q/w_v$ to $w$ is nonlinear, and cannot be computed exactly in general. However, conservative bounds can be computed using interval arithmetic, affine arithmetic [19] or more recent and accurate methods [20]. Similarly, known error bounds on the parameters of the robot model (e.g., link lengths, link masses) and the communication delays could be mapped to (possibly conservative) bounds on $w$.

### C. Problem formulation

Finding the maximum (or minimum) acceleration such that the state constraints can be satisfied in the future can be formulated as an infinite-horizon optimal control problem:

$$\begin{aligned} \ddot{q}_0^{max} = \underset{\ddot{q}_0, \ddot{q}_1, \dots}{\text{maximize}} \quad & \ddot{q}_0 \\ \text{subject to} \quad & q(i\delta t + t) = q_i + t\,\dot{q}_i + \frac{t^2}{2}(\ddot{q}_i + w_i) \\ & \dot{q}(i\delta t + t) = \dot{q}_i + t\,(\ddot{q}_i + w_i) \\ & (q(i\delta t + t), \dot{q}(i\delta t + t)) \in \mathscr{X} \\ & |\ddot{q}_i| \leq \ddot{q}^{max} \\ & (q(0), \dot{q}(0)) \quad \text{fixed} \\ & \forall i \geq 0, \forall t \in [0, \delta t], \forall w_i \in [-\bar{w}, \bar{w}] \end{aligned} \tag{7}$$

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 (7). A state is viable if, starting from that state, it is possible for the system to satisfy the constraints in the future. Formally, a state $(q(0), \dot{q}(0))$ belongs to the robust viability kernel $\mathscr{V}$ if and only if, using that state as initial conditions for problem (7), the problem admits a solution. The interest in the introduction of the robust viability kernel $\mathscr{V}$ is that it allows us to simplify the problem of ensuring the future satisfaction of the joint limits into the simpler problem of ensuring that the next state be viable. In the following, we derive a definition of $\mathscr{V}$ that allows us to check membership easily.

### 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 $\mathscr{V}^C$ that is a superset of the previous one: $\mathscr{V} \subset \mathscr{V}^C$. Assuming an initial position $q_0$, the maximum initial velocity $\dot{q}_M^{\mathscr{V}}$ that allows us to satisfy the position limits in the future can be found as:

$$\begin{aligned} \dot{q}_M^{\mathscr{V}}(q_0) = \underset{\dot{q}_0, \ddot{q}(t)}{\text{maximize}} \quad & \dot{q}_0 \\ \text{subject to} \quad & \frac{\mathrm{d}q(t)}{\mathrm{d}t} = \dot{q}(t) \\ & \frac{\mathrm{d}\dot{q}(t)}{\mathrm{d}t} = \ddot{q}(t) + w(t) \\ & (q(t), \dot{q}(t)) \in \mathscr{X} \\ & |\ddot{q}(t)| \leq \ddot{q}^{max} \\ & q(0) = q_0, \quad \dot{q}(0) = \dot{q}_0 \\ & \forall t \geq 0, \forall w(t) \in [-\bar{w}, \bar{w}] \end{aligned} \tag{8}$$

In the case without disturbance, the maximum initial velocity is such that, by applying maximum deceleration, we reach $q^{max}$ with null 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 when the velocity is null: $t^0 = \dot{q}_0 / (\ddot{q}^{max} - \bar{w})$,
3) compute the initial velocity that results in: $q(t^0) = q^{max}$.

In this way we get:

$$\dot{q}_M^{\mathscr{V}}(q) = \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q)} \tag{9}$$

Following the same steps, we can define also the minimum velocity (i.e. maximum negative velocity) to ensure viability:

$$\dot{q}_m^{\mathscr{V}}(q) = -\sqrt{2(\ddot{q}^{max} - \bar{w})(q - q^{min})} \tag{10}$$

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

$$\mathscr{V}^c = \{(q, \dot{q}) : (q, \dot{q}) \in X, \dot{q}_m^{\mathscr{V}}(q) \leq \dot{q} \leq \dot{q}_M^{\mathscr{V}}(q)\} \tag{11}$$

With this definition of $\mathscr{V}^c$ we can easily check the viability of a state by just evaluating three inequalities. Fig. 1 shows how the viability kernel varies with the presence of disturbances.

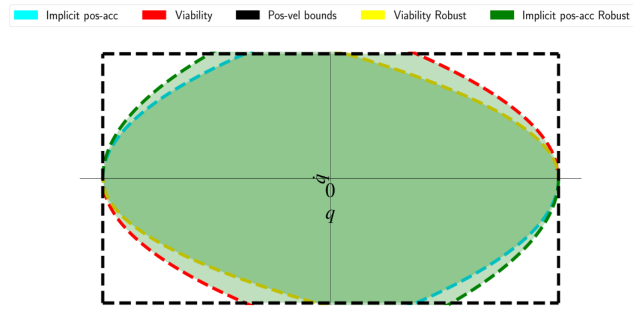


Fig. 1: Example of robust viability kernel with $\bar{w} = 0$ and with $\bar{w} = \frac{1}{3}\ddot{q}^{max}$.

### 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. In the following we derive the conditions under which the discrete-time viability robust viability kernel is equivalent to its continuous-time version.

To do so, let us analyze the worst case, which is when the system reaches a position bound $(q_0, \dot{q}_0) = (q^{max}, 0)$ 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:

$$\begin{cases} q_1 = q^{max} - 0.5\delta t^2(\ddot{q}^{max} + \bar{w}) \\ \dot{q}_1 = -\delta t(\ddot{q}^{max} + \bar{w}) \end{cases} \tag{12}$$

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

$$-\dot{q}^{max} \leq -\delta t(\ddot{q}^{max} + \bar{w}) \tag{13}$$

We can see this as a bound on $\bar{w}$, which gives us a first condition for equivalence of discrete and continuous time viability:

$$\bar{w} \leq \frac{\dot{q}^{max} - \delta t \ddot{q}^{max}}{\delta t} \tag{14}$$

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.

$$\begin{cases} q_2 = q^{max} - \delta t^2(\ddot{q}^{max} + 2\bar{w}) \\ \dot{q}_2 = -2\delta t \bar{w} \end{cases} \tag{15}$$

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:

$$\dot{q}_3 = \delta t(\ddot{q}^{max} - 3\bar{w}) \tag{16}$$

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 $\bar{w}$:

$$\delta t(\ddot{q}^{max} - 3\bar{w}) \geq 0 \quad \Rightarrow \quad \bar{w} \leq \frac{1}{3}\ddot{q}^{max} \tag{17}$$

The assumption of a positive $\dot{q}$ after 3 time steps implies that we assumed 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.

$$\begin{gathered} \dot{q}_3(t) = \dot{q}_2 + t(\ddot{q}^{max} - \bar{w}) = 0 \\ t = \frac{-\dot{q}_2}{\ddot{q}^{max} - \bar{w}} \quad \Rightarrow \quad t = \frac{2\delta t\bar{w}}{\ddot{q}^{max} - \bar{w}} \triangleq t_0 \end{gathered} \tag{18}$$

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.

$$q_3(t_0) \triangleq q_{discr} = q_2 - \frac{4\delta t\bar{w}}{\ddot{q}^{max} - \bar{w}}\delta t\bar{w} + \frac{4\delta t^2\bar{w}^2}{2(\ddot{q}^{max} - \bar{w})^2}(\ddot{q}^{max} - \bar{w}) \tag{19}$$

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

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

$$q^{max} - q^{min} \geq \delta t^2\left(\ddot{q}^{max} + 2\bar{w} + \frac{2\bar{w}^2}{\ddot{q}^{max} - \bar{w}}\right) \tag{21}$$

As expected, (21) 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 (21), which seems a second-order polynomial of $\bar{w}$, but with some straightforward manipulations can actually become linear inequality in $\bar{w}$:

$$\bar{w} \leq \ddot{q}^{max}\frac{q^r - \delta t^2\ddot{q}^{max}}{q^r + \delta t^2\ddot{q}^{max}}, \tag{22}$$

where $q^r \triangleq q^{max} - q^{min}$. 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 (21) we get:

$$\alpha \leq \frac{q^r - \delta t^2\ddot{q}^{max}}{q^r + \delta t^2\ddot{q}^{max}} \triangleq \alpha_1 \tag{23}$$

Together with this bound, we should consider also the bound on $\alpha$ imposed by (17) and the bound coming from the velocity constraint (14):

$$\alpha \leq \frac{\dot{q}^{max} - \delta t\ddot{q}^{max}}{\delta t\ddot{q}^{max}} \triangleq \alpha_2 \tag{24}$$

Considering all the contributions we can now define the final range for the allowable $\alpha$:

$$\alpha \in \left[0, \min\left(\alpha_1, \alpha_2, \frac{1}{3}\right)\right] \tag{25}$$

If $\alpha$ 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: $\mathscr{V} = \mathscr{V}^C$.

### C. Reformulation in terms of viability

We have now obtained a formulation of $\mathscr{V}$ so we can reformulate problem (7). Starting from the initial state $(q(0), \dot{q}(0)) \in \mathscr{V}$ we want to compute the maximum $\ddot{q}$ so that:

1) the next state is viable: $(q(\delta t), \dot{q}(\delta t)) \in \mathscr{V}$;
2) the trajectory in $[0, \delta t]$ is inside the feasible set $X$.

The first condition alone is not sufficient, similarly to what happens in (18), because the trajectory between two viable states may violate a constraint. We can reformulate (7) as:

$$\begin{aligned} \ddot{q}_0^{max} = \underset{\ddot{q}}{\text{maximize}} \quad & \ddot{q} \\ \text{subject to} \quad & q(0) = q, \quad \dot{q}(0) = \dot{q} \\ & q(t) = q + t\,\dot{q} + \frac{t^2}{2}(\ddot{q} + w) \\ & \dot{q}(t) = \dot{q} + t\,(\ddot{q} + w) \\ & (q(t), \dot{q}(t)) \in \mathscr{X} \\ & \dot{q}_m^{\mathscr{V}}(q(\delta t)) \leq \dot{q}(\delta t) \leq \dot{q}_M^{\mathscr{V}}(q(\delta t)) \\ & |\ddot{q}| \leq \ddot{q}^{max} \\ & \forall t \in [0, \delta t], \forall w \in [-\bar{w}, \bar{w}] \end{aligned} \tag{26}$$

This problem is much simpler than the previous one: instead of having an infinite sequence of decision variables it has only one, and its constraints are limited to the time interval $[0, \delta t]$, rather than $[0, \infty]$. However, the constraints are infinitely many and nonlinear, therefore problem (26) is still hard. In the following, we reformulate the inequalities of problem (26) as simple bounds on $\ddot{q}$.

### D. Position inequalities

The position bounds of (26) are:

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

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:

$$\begin{aligned} q^{min} \leq q + t\dot{q} + \frac{1}{2}t^2(\ddot{q} - \bar{w}) \quad \forall t \in [0, \delta t] \\ q + t\dot{q} + \frac{1}{2}t^2(\ddot{q} + \bar{w}) \leq q^{max} \quad \forall t \in [0, \delta t] \end{aligned} \tag{28}$$

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

$$q + t\dot{q} + \frac{1}{2}t^2\gamma \leq q^{max} \;\forall t \in [0, \delta t] \tag{29}$$

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 $\gamma$, and then convert it to a bound on $\ddot{q}$ by simply subtracting $\bar{w}$ from it. A similar approach can be followed for the lower position limit. The computation is summarized in Alg. 1.

### E. Velocity inequalities

The velocity evolves linearly in time, so we only need to verify the bounds for $t = \delta t$:

$$|\dot{q} + \delta t(\ddot{q} + w)| \leq \dot{q}^{max} \qquad \forall w \in [-\bar{w}, \bar{w}] \tag{30}$$

Since $\bar{w}$ is the worst-case disturbance for the upper bound, rearranging (30) we obtain:

$$\ddot{q} \leq \frac{\dot{q}^{max} - \dot{q}}{\delta t} - \bar{w} \tag{31}$$

Analogous to 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:

$$\frac{1}{\delta t}(-\dot{q}^{max} - \dot{q}) + \bar{w} \leq \ddot{q} \leq \frac{1}{\delta t}(\dot{q}^{max} - \dot{q}) - \bar{w} \tag{32}$$

### F. Viability inequalities

Let us consider the upper bound of the viability inequality.

$$\dot{q}(\delta t) \leq \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q(\delta t))} \qquad \forall w \in [-\bar{w}, \bar{w}] \tag{33}$$

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

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

Since $\ddot{q}$ always appears together with $\bar{w}$, we can introduce a new variable $\gamma \triangleq \ddot{q} + \bar{w}$ and rewrite (34) as:

$$\dot{q} + \delta t\gamma \leq \sqrt{2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\dot{q} - 0.5\delta t^2\gamma)} \tag{35}$$

At this point the inequality is equivalent to the one for the non-robust case, therefore we can use the algorithm of [3] to find the bounds on $\gamma$, and then map them to bounds on $\ddot{q}$ by simply subtracting $\bar{w}$ from them. The computation is 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 shows the state space divided in regions based on which acceleration upper bound dominates the others.

**Algorithm 1** accBoundsFromPosLimits

**Require:** $q, \dot{q}, q^{min}, q^{max}, \delta t, \bar{w}$
- $\ddot{q}_1^M \leftarrow -\dot{q}/\delta t$
- $\ddot{q}_2^M \leftarrow -\dot{q}^2/(2(q^{max} - q))$
- $\ddot{q}_3^M \leftarrow 2(q^{max} - q - \delta t\,\dot{q})/(\delta t^2)$
- $\ddot{q}_2^m \leftarrow \dot{q}^2/(2(q - q^{min}))$
- $\ddot{q}_3^m \leftarrow 2(q^{min} - q - \delta t\,\dot{q})/(\delta t^2)$
- **if** $\dot{q} \geq 0$ **then**
  - $\ddot{q}^{LB} \leftarrow \ddot{q}_3^m$
  - **if** $\ddot{q}_3^M > \ddot{q}_1^M$ **then**
    - $\ddot{q}^{UB} \leftarrow \ddot{q}_3^M$
  - **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**
    - $\ddot{q}^{LB} \leftarrow \ddot{q}_3^m$
  - **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:** $q, \dot{q}, q^{min}, q^{max}, \ddot{q}^{max}, \delta t, \bar{w}$
- $a \leftarrow \delta t^2$
- $b \leftarrow \delta t(2\dot{q} + (\ddot{q}^{max} - \bar{w})\delta t)$
- $c \leftarrow \dot{q}^2 - 2(\ddot{q}^{max} - \bar{w})(q^{max} - q - \delta t\,\dot{q})$
- $\gamma_1 \leftarrow -\dot{q}/\delta t$
- $\Delta \leftarrow b^2 - 4ac$
- **if** $\Delta \geq 0$ **then**
  - $\gamma^{UB} \leftarrow \max(\gamma_1, (-b + \sqrt{\Delta})/(2a))$
- **else**
  - $\gamma^{UB} \leftarrow \gamma_1$
- $b \leftarrow 2\delta t\,\dot{q} - (\ddot{q}^{max} - \bar{w})\delta t^2$
- $c \leftarrow \dot{q}^2 - 2(\ddot{q}^{max} - \bar{w})(q + \delta t\,\dot{q} - q^{min})$
- $\Delta \leftarrow b^2 - 4ac$
- **if** $\Delta \geq 0$ **then**
  - $\gamma^{LB} \leftarrow \min(\gamma_1, (-b - \sqrt{\Delta})/(2a))$
- **else**
  - $\gamma^{LB} \leftarrow \gamma_1$
- $\{\ddot{q}^{LB}, \ddot{q}^{UB}\} \leftarrow \{\gamma^{LB} + \bar{w}, \gamma^{UB} - \bar{w}\}$
- **return** $\{\ \ddot{q}^{LB}, \ddot{q}^{UB}\ \}$

**Algorithm 3** Compute Joint Acceleration Bounds

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

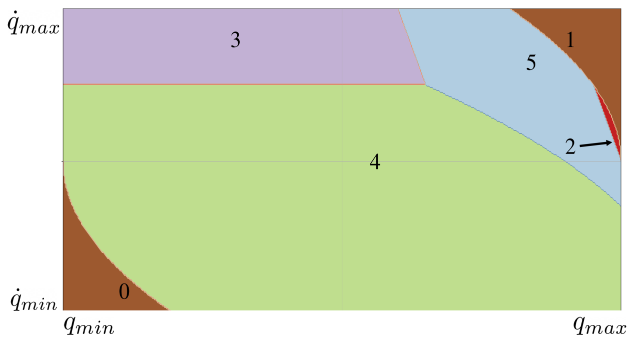


Fig. 2: Feasible state space for $q^{max} = -q^{min} = 0.5$ rad, $\dot{q}^{max} = 2$ rad/s, $\ddot{q}^{max} = 10$ rad/s$^2$, $\delta t = 0.1$ s, and $\bar{w} = \frac{1}{3}\ddot{q}^{max}$. Region 0 represents the unreachable space, while region 1 represents the space that is not viable. In each of the other four regions, a different acceleration upper bound dominates the others. In region 2 it is the one coming from the position inequality (27). In region 3 it is the one coming from the velocity inequality (32). In region 4 it is the acceleration upper bound $\ddot{q}^{max}$. In region 5 it is the one coming from the viability inequality (33).

**Algorithm 4** Dealing with non-viable states

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

## 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 $j$. All our tests use a large value of $\delta t = 0.1$ s to make the constraint violations more easily visible in the plots. 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 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.

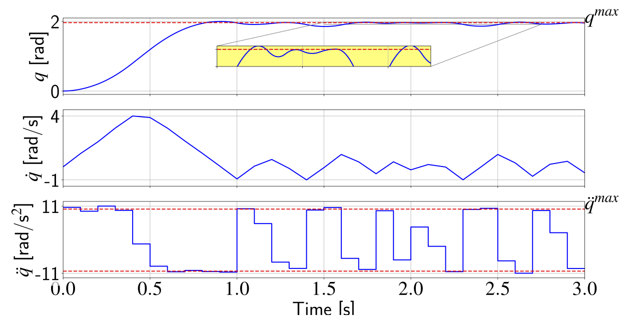


Fig. 3: Single joint, non-robust viability approach, random disturbance.

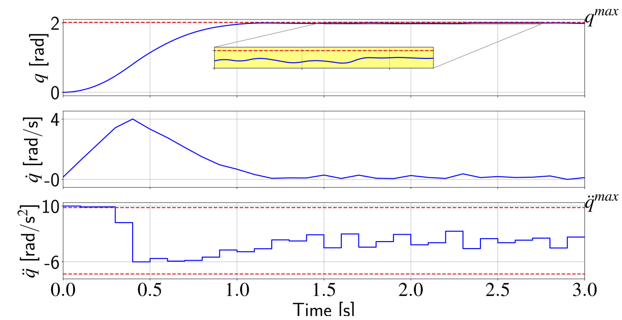


Fig. 4: Single joint, robust viability approach, random disturbance.

### 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/s$^2$. 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 $w$. 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

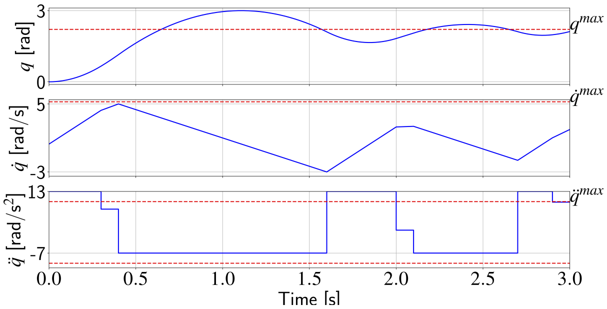


Fig. 5: Single joint, non-robust viability approach, worst-case disturbance $\bar{w}$.

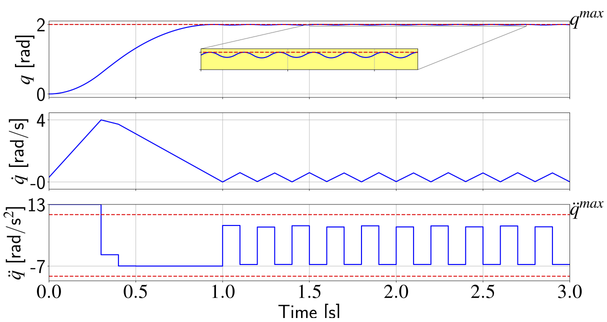


Fig. 6: Single joint, robust viability approach, worst-case disturbance $\bar{w}$.

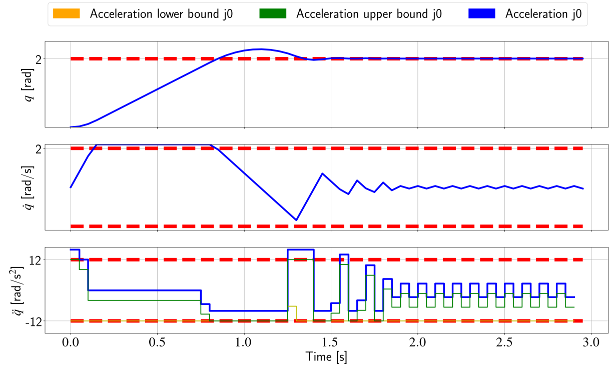


Fig. 7: Baxter robot, naive method.

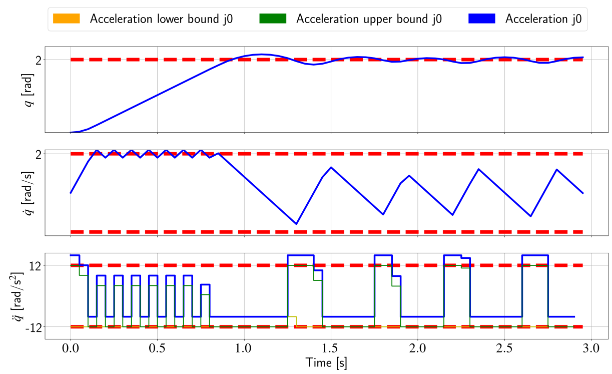


Fig. 8: Baxter robot, viability non-robust method.

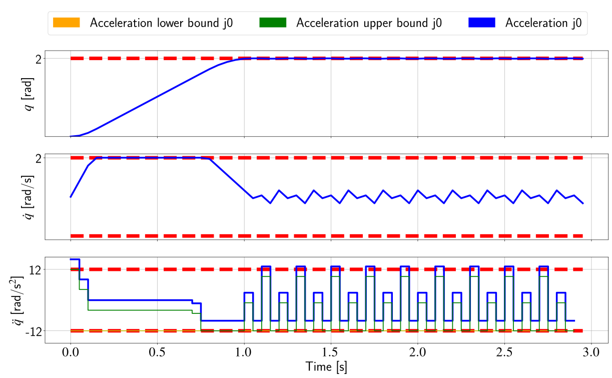


Fig. 9: Baxter robot, viability robust method.

oscillations are to be expected due to the discrete-time nature of the controller and, as also observed in [3], can be mitigated by using a larger value of $\delta t$ in the algorithm. The non-robust method instead clearly violates the constraint.

### 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.

$$\ddot{q}^d = k_P(q^d - q) - k_D\dot{q} \tag{36}$$

where $k_P, k_D \in \mathbb{R}^+$. The resulting acceleration (36) 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. 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] and it can be alleviated by choosing a larger value of $\delta t$ when computing the acceleration bounds [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 [21], [22]. The *stochastic* approach to uncertainty modeling could indeed result in a less conservative behavior, 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 [23]. Finding computationally tractable methods to approximate these sets in the nonlinear setting is still an open problem [24], and an interesting direction for future work.

## REFERENCES


[1] M. H. Yeganegi, M. Khadiv, A. Del Prete, S. A. A. Moosavian, and L. Righetti, "Robust walking based on mpc with viability guarantees," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2389–2404, 2022.

[2] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "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," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 281–288, Jan 2018.

[4] S. Rubrecht, V. Padois, P. Bidaud, M. Broissia, and M. Da Silva Simoes, "Motion safety and constraints compatibility for multibody robots," *Autonomous Robots*, vol. 32, no. 3, pp. 333–349, 2012.

[5] K. C. Park, P. H. Chang, and S. H. Kim, "The Enhanced Compact QP Mathod for Redundant Manipulators Using Practical Inequality Constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[6] L. Saab, O. E. Ramos, N. Mansard, P. Soueres, and J.-y. Fourquet, "Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[7] F. Kanehiro, M. Morisawa, W. Suleiman, K. Kaneko, and E. Yoshida, "Integrating geometric constraints into reactive leg motion generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[8] T. Kröger and F. M. Wahl, "On-Line trajectory generation in robotic systems. Basic concepts for instananeous reactions to unforseen (sensor) events," *IEEE Transaction on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[9] S. Rubrecht, V. Padois, P. Bidaud, and M. De Broissia, "Constraints Compliant Control : constraints compatibility and the displaced configuration approach," in *IEEE/RSJ Intenational Conference on Intelligent Robots and Systems (IROS)*, 2010.

[10] Q. Nguyen and K. Sreenath, "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016.

[11] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained Robot Control Using Control Barrier Functions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[12] J. J. Quiroz-Omana and B. V. Adorno, "Whole-Body Control With (Self) Collision Avoidance Using Vector Field Inequalities," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4048–4053, 2019.

[13] M. Faroni, M. Beschi, and N. Pedrocchi, "Inverse Kinematics of Redundant Manipulators with Dynamic Bounds on Joint Movements," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6435–6442, 2020.

[14] A. Oleinikov, S. Kusdavletov, A. Shintemirov, and M. Rubagotti, "Safety-Aware Nonlinear Model Predictive Control for Physical Human-Robot Interaction," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5665–5672, 2021.

[15] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[16] R. Ghaemi, J. Sun, and I. Kolmanovsky, "Less conservative robust control of constrained linear systems with bounded disturbances," *2008 47th IEEE Conference on Decision and Control*, no. 3, pp. 983–988, 2008.

[17] R. Ghaemi, J. Sun, and I. V. Kolmanovsky, "Robust control of constrained linear systems with bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2683–2688, 2012.

[18] K. Fukuda and A. Prodon, "Double Description Method Revisited," in *Combinatorics and Computer Science*, 1996.

[19] N. S. Nedialkov, V. Kreinovich, and S. A. Starks, "Interval arithmetic, affine arithmetic, Taylor series methods: Why, what next?" *Numerical Algorithms*, vol. 37, no. 1-4 SPEC. ISS., pp. 325–336, 2004.

[20] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, 2019.

[21] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Stochastic and Robust MPC for Bipedal Locomotion : A Comparative Study on Robustness and Performance," in *IEEE-RAS International Conference on Humanoid Robots*, 2021.

[22] 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.

[23] J. M. Bravo, D. Limon, T. Alamo, and E. F. Camacho, "On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach," in *2003 European Control Conference (ECC)*, 2003, pp. 288–293.

[24] A. La Rocca, M. Saveriano, and A. Del Prete, "Vboc: Learning the viability boundary of a robot manipulator using optimal control," 2023.