

Torque Control with Joints Position and Velocity Limits Avoidance

Venus Pasandi

Artificial and Mechanical Intelligence
Italian Institute of Technology
Genoa, Italy
venus.pasandi@iit.it

Daniele Pucci

Artificial and Mechanical Intelligence
Italian Institute of Technology
Genoa, Italy
daniele.pucci@iit.it

Abstract—The design of a control architecture for providing the desired motion along with the realization of the joint limitation of a robotic system is still an open challenge in control and robotics. This paper presents a torque control architecture for fully actuated manipulators for tracking the desired time-varying trajectory while ensuring the joints position and velocity limits. The presented architecture stems from the parametrization of the feasible joints position and velocity space by exogenous states. The proposed parametrization transforms the control problem with constrained states to an un-constrained one by replacing the joints position and velocity with the exogenous states. With the help of Lyapunov-based arguments, we prove that the proposed control architecture ensures the stability and convergence of the desired joint trajectory along with the joints position and velocity limits avoidance. We validate the performance of proposed architecture through various simulations on a simple two-degree-of-freedom manipulator and the humanoid robot iCub.

Index Terms—torque control, position and velocity limit avoidance, state parametrization

I. INTRODUCTION

Nonlinear feedback control of unconstrained fully actuated manipulators is not new to the control community. A large variety of position, velocity, and torque control algorithms have been developed through feedback linearization, backstepping, robust control, and adaptive tools for steering these nonlinear systems towards the desired quantities [1]. Although, the control algorithm is applicable/safe for a robotic manipulator when its physical constraints, such as the motion and actuation limits, are satisfied. In this paper, we propose a control architecture to ensure the joints position and velocity limits avoidance for torque-controlled manipulators.

The joint position and velocity limits avoidance for the reference trajectory/path planning has been widely studied through neural networks [2], repulsive potentials [3], optimisation [4], and parametrization [5]. However, ensuring the physical limits for the reference trajectory does not imply that the controller's reaction to the initial conditions and disturbances will not lead to violating these limits.

Reducing the joint velocity/acceleration of the robot considering its position and velocity limits is a simple approach for ensuring these limits. For example, the joints acceleration can be constrained with a function of the distance between the joints position and velocity values and the corresponding

limits [6]. However, these approaches rely mostly on hand-tuned and/or offline heuristics.

For handling the joint position and velocity limits of robotic systems, the control objectives can be represented as an optimization problem with inequality constraints corresponding to the joint limits. This technique is widely used in the humanoid whole-body control and human-robot interaction control [7]. This technique, however, still lacks the theoretical guarantee of the stability and convergence properties associated with the evolution of the system.

The barrier Lyapunov function is recently used for handling control problems with joint/state constraints [8], [9]. This function yields a value that grows to infinity when the joints approach their limits. This method is usually complicated and needs to deal with the model inaccuracies and the conflicts between the tracking objectives and the joint limits [10], [11].

Parametrization is another technique used for handling the limitations of the system states and inputs [12], [13]. In this technique, the feasible space of the states/inputs is parametrized by an exogenous state/input. Thus, the constrained control problem is transformed into a non-constrained one where any control tool available in the literature can be employed. To the best of our knowledge, the control problem via parametrization technique has not been investigated for handling both the joint position and velocity limits.

This paper proposes a torque control architecture for processing the joints position and velocity limits for fully actuated manipulators. The proposed control provides the asymptotic stability of the desired trajectory while preserving the constant joints position and velocity limits. Comparing to the existing methods for implementing joints position and velocity limits in torque-controlled manipulators, our proposed architecture, with the help of Lyapunov arguments, ensures that the desired trajectory is asymptotically stable and the time evolution of the joints position and velocity always remains within the associated limits provided that the desired trajectory satisfies these limits. For this purpose, the feasible joint position and velocity space is parametrized by exogenous states. The proposed parametrization introduces a one-to-one map between the joints position and velocity of the robot and the exogenous states. Using the exogenous states, the problem of control

design with constrained states (i.e. constrained joints position and velocity) is transformed to an un-constrained control problem with exogenous states. After that, a control policy is proposed for ensuring the stability properties of the exogenous states which leads us to ensuring the stability properties and joints limitation avoidance of the robot. We have investigated the performance and limitations of the proposed control architecture through simulations on a simple two-degree-of-freedom manipulator, and also humanoid robot iCub [14].

This paper is organized as follows. Section II expresses the notation and the definitions used in the paper. Section III introduces the problem statements. In section IV, a control architecture is proposed for stabilizing a desired joint trajectory and ensuring the joints position and velocity limits avoidance. In section V, the simulation results carried out to validate the performance of the proposed control architecture are illustrated. Finally, section VI concludes the paper by remarks and perspectives.

II. NOTATION

The following notation is used throughout the paper.

- \mathbb{R} is the set of real numbers.
- I_n is the $n \times n$ identity matrix.
- For a vector $\mathbf{q} \in \mathbb{R}^n$, the i^{th} component of \mathbf{q} is written as q_i .
- The transpose operator is denoted by $(\cdot)^\top$.
- For a vector $\mathbf{q} \in \mathbb{R}^n$, the diagonal matrix of \mathbf{q} is written as $\check{\mathbf{q}}$.
- For a vector $\mathbf{q} \in \mathbb{R}^n$, the Euclidean norm of \mathbf{q} is denoted by $\|\mathbf{q}\|$.
- For a vector $\mathbf{q} \in \mathbb{R}^n$, the absolute value vector of \mathbf{q} is denoted by $|\mathbf{q}|$.
- For a vector $\mathbf{q} \in \mathbb{R}^n$, the function $\tanh(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined as $\tanh(\mathbf{q}) = [\tanh(q_1), \tanh(q_2), \dots, \tanh(q_n)]^\top$.
- For a scalar $q \in \mathbb{R}$, the function $\text{Sat}(q, q_{min}, q_{max}) : \mathbb{R} \rightarrow \mathbb{R}$ is defined as

$$\text{Sat}(q, q_{min}, q_{max}) = \begin{cases} q_{max} & q \geq q_{max} \\ q & q_{min} \leq q \leq q_{max} \\ q_{min} & q \leq q_{min} \end{cases}$$

- For a vector $\mathbf{q} \in \mathbb{R}^n$, the function $\text{Sat}(\mathbf{q}, \mathbf{q}_{min}, \mathbf{q}_{max}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined as $\text{Sat}(\mathbf{q}, \mathbf{q}_{min}, \mathbf{q}_{max}) = [\text{Sat}(q_1, q_{min_1}, q_{max_1}), \dots, \text{Sat}(q_n, q_{min_n}, q_{max_n})]^\top$.

III. PROBLEM STATEMENT

The equations of motion for a fully actuated robotic manipulator with n degrees of freedom can be written in the following form [15]

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \boldsymbol{\tau}, \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^{n \times 1}$ is the vector of generalized coordinates, $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is the vector of the Centrifugal and Coriolis effects, and $G(\mathbf{q}) \in \mathbb{R}^n$ is the vector of the Gravitational effects. $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of the actuator forces/torques.

The control objective is defined as computing $\boldsymbol{\tau}$ such that \mathbf{q} tracks the desired trajectory (e.g. $\mathbf{q}_d(t)$). Though, $\boldsymbol{\tau}$ is feasible if it preserves the physical limits of the robot such as joint position, velocity, and torque limits. In the present paper, we propose a control policy that ensures

- 1) the asymptotic stability of the desired trajectory, and
- 2) the joint position and velocity limit avoidance.

Assume that the feasible region for the generalised coordinates is as

$$\mathcal{Q}_q := \{\mathbf{q} \in \mathbb{R}^n : \mathbf{q}_{min} < \mathbf{q} < \mathbf{q}_{max}\}, \quad (2)$$

where $\mathbf{q}_{min}, \mathbf{q}_{max} \in \mathbb{R}^n$ denote the vectors that define the minimum and maximum values for generalised coordinates. Moreover, assume that the feasible region for the generalised velocities is as

$$\mathcal{Q}_{\dot{\mathbf{q}}} := \{\dot{\mathbf{q}} \in \mathbb{R}^n : \dot{\mathbf{q}}_{min} < \dot{\mathbf{q}} < \dot{\mathbf{q}}_{max}\}, \quad (3)$$

where $\dot{\mathbf{q}}_{min}, \dot{\mathbf{q}}_{max} \in \mathbb{R}^n$ denotes the vector of the minimum and maximum values for generalised velocities.

We propose a control architecture ensuring that \mathbf{q} tracks the desired trajectory $\mathbf{q}_d(t)$ while the evolution of \mathbf{q} and $\dot{\mathbf{q}}$ always remain within \mathcal{Q}_q and $\mathcal{Q}_{\dot{\mathbf{q}}}$, respectively.

Throughout this paper, we assume that:

Assumption 1. *The first and second-order time derivatives of \mathbf{q}_d are well-defined and bounded $\forall t \in \mathbb{R}^+$. Moreover, the desired trajectory \mathbf{q}_d is feasible, i.e. $(\mathbf{q}_d(t), \dot{\mathbf{q}}_d(t)) \in (\mathcal{Q}_q, \mathcal{Q}_{\dot{\mathbf{q}}})$, $\forall t \geq 0$.*

Assumption 2. *Each generalized coordinate possesses a free motion domain different from zero i.e. $q_{max_i} - q_{min_i} > 0$ and $\dot{q}_{max_i} - \dot{q}_{min_i} > 0$, $\forall i = 1 : n$.*

IV. CONTROLLER ARCHITECTURE

We parametrize the space of the generalised coordinates, as well as, the generalised velocities as follows

$$\begin{aligned} \mathbf{q} &= \mathbf{q}_0 + \check{\delta}_q \tanh(\boldsymbol{\zeta}), \\ \dot{\mathbf{q}} &= \dot{\mathbf{q}}_0 + \check{\delta}_{\dot{\mathbf{q}}} \tanh(\boldsymbol{\psi}), \end{aligned} \quad (4)$$

where

$$\begin{aligned} \mathbf{q}_0 &= \frac{\mathbf{q}_{max} + \mathbf{q}_{min}}{2}, \\ \dot{\mathbf{q}}_0 &= \frac{\dot{\mathbf{q}}_{max} + \dot{\mathbf{q}}_{min}}{2}, \\ \delta_q &= \frac{\mathbf{q}_{max} - \mathbf{q}_{min}}{2}, \\ \delta_{\dot{\mathbf{q}}} &= \frac{\dot{\mathbf{q}}_{max} - \dot{\mathbf{q}}_{min}}{2}. \end{aligned}$$

The above parametrization guarantees that $(\mathbf{q}, \dot{\mathbf{q}}) \in (\mathcal{Q}_q, \mathcal{Q}_{\dot{\mathbf{q}}})$ for bounded $(\boldsymbol{\zeta}, \boldsymbol{\psi})$. This parametrization is a one-to-one nonlinear map between $(\mathbf{q}, \dot{\mathbf{q}})$ and $(\boldsymbol{\zeta}, \boldsymbol{\psi})$. The new states $(\boldsymbol{\zeta}, \boldsymbol{\psi})$ is computed according to $(\mathbf{q}, \dot{\mathbf{q}})$ as

$$\begin{aligned} \boldsymbol{\zeta} &= \tanh^{-1} \left(\check{\delta}_q^{-1} (\mathbf{q} - \mathbf{q}_0) \right), \\ \boldsymbol{\psi} &= \tanh^{-1} \left(\check{\delta}_{\dot{\mathbf{q}}}^{-1} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0) \right). \end{aligned} \quad (5)$$

The second time derivative of the generalized coordinates can be computed with respect to (ζ, ψ) as

$$\ddot{\mathbf{q}} = J_\psi \dot{\psi}, \quad (6)$$

where $J_\psi = \check{\delta}_{\dot{q}} \left(I_n - \tanh^2(\psi) \right)$. Thus, (1) can be written in terms of (ζ, ψ) as

$$\begin{cases} J_\zeta \dot{\zeta} = \check{\delta}_{\dot{q}} \tanh(\psi) + \dot{\mathbf{q}}_0 \\ M(\zeta) J_\psi \dot{\psi} + h_{\zeta\psi}(\zeta, \psi) = \tau, \end{cases} \quad (7)$$

where $J_\zeta = \check{\delta}_{\dot{q}} \left(I_n - \tanh^2(\zeta) \right)$ and $h_{\zeta\psi} = C(\zeta, \psi) \left(\check{\delta}_{\dot{q}} \tanh(\psi) + \dot{\mathbf{q}}_0 \right) + G(\zeta)$.

Remark 1. One can observe the following properties:

R1.1 The matrices J_ζ , J_ψ , \dot{J}_ζ , and \dot{J}_ψ are diagonal,

R1.2 For all ζ , $0 \leq J_\zeta \leq \delta_q$,

R1.3 For all ψ , $0 \leq J_\psi \leq \delta_{\dot{q}}$,

R1.4 The matrix J_ζ is positive definite for bounded ζ (i.e. for $\mathbf{q} \in \mathcal{Q}_q$),

R1.5 The matrix J_ψ is positive definite for bounded ψ (i.e. for $\dot{\mathbf{q}} \in \mathcal{Q}_{\dot{q}}$),

R1.6 The matrix $M J_\psi$ is positive definite for bounded ψ .

If the desired trajectory is feasible i.e. $(\mathbf{q}_d, \dot{\mathbf{q}}_d) \in (\mathcal{Q}_q, \mathcal{Q}_{\dot{q}})$, the desired trajectory of (ζ, ψ) is defined as

$$\begin{aligned} \zeta_d &= \tanh^{-1} \left(\delta_q^{-1} (\mathbf{q}_d - \mathbf{q}_0) \right), \\ \psi_d &= \tanh^{-1} \left(\delta_{\dot{q}}^{-1} (\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_0) \right). \end{aligned} \quad (8)$$

For ensuring asymptotic stability of (ζ_d, ψ_d) , one can use any control technique from the literature. The control policy is required to ensure that (ζ, ψ) is bounded and converge to (ζ_d, ψ_d) . We use the feedback linearisation technique and consider the control policy as

$$\tau = h_{\zeta\psi} + M J_\psi \left(\dot{\psi}_r - k_2^{-1} k_1 \dot{e}_\zeta - k_2 e_\zeta - k_3 e_\psi \right), \quad (9)$$

where

$$e_\zeta = \zeta - \zeta_d,$$

$$e_\psi = \psi - \psi_r,$$

$$\psi_r = \tanh^{-1} \left(\check{\delta}_{\dot{q}}^{-1} \left(J_\zeta J_{\zeta_d}^{-1} \left(\check{\delta}_{\dot{q}} \tanh(\psi_d) + \dot{\mathbf{q}}_0 \right) - \dot{\mathbf{q}}_0 \right) \right).$$

and $k_1, k_2, k_3 \in \mathbb{R}^n$ are constant matrices.

Theorem 1. Considering the equations of motion presented in (7) and the control policy (9), given that

T1.1) k_1, k_2, k_3 are diagonal and positive definite,

T1.2) $k_3 - k_2 k_1^{-1} k_2$ is positive definite, and

T1.3) Assumption 1 and 2 hold,

T1.4) ψ_r is well-defined,

then (ζ_d, ψ_d) is the globally asymptotically stable trajectory of the closed loop system.

In the lights of the above theorem, if T1.1 to T1.4 are satisfied, then (ζ, ψ) is bounded and converges to (ζ_d, ψ_d) . Considering (4), the boundedness of (ζ, ψ) ensures that $(\mathbf{q}, \dot{\mathbf{q}}) \in (\mathcal{Q}_q, \mathcal{Q}_{\dot{q}})$. On the other hand, considering (5) and

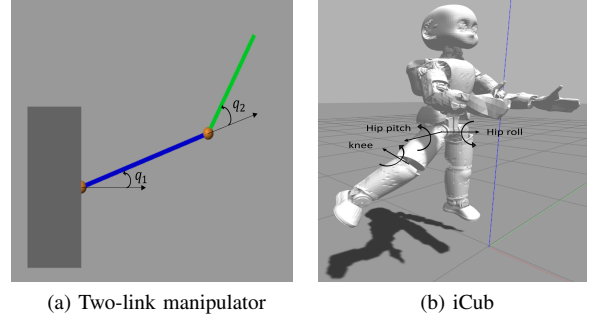


Fig. 1. The robotic manipulators used in the simulation studies

(8), the fact that (ζ, ψ) converges to (ζ_d, ψ_d) implies that $(\mathbf{q}, \dot{\mathbf{q}})$ converges to $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$. Note that T1.4 is satisfied in a neighborhood of the desired trajectory where the error between ζ and ζ_d is sufficiently small. Thus, the proposed control architecture ensures the local asymptotic stability of the desired trajectory if T1.1 to T1.3 are satisfied. Moreover, T1.4 is ensured for the initial conditions satisfying it. However, this assumption could be violated initially due to the initial conditions or during the motion because of disturbances. Remark 5 proposes a trick for dealing with these situations.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed control architecture for trajectory tracking in two simulation studies: a simple two-link manipulator and humanoid robot iCub [14] (see Fig. 1). In the first case, the proposed controller is used for controlling a two-degree-of-freedom manipulator simulated in Simscape/MATLAB [16]. In the second case, the proposed controller is employed on the humanoid robot iCub simulated in Gazebo [17]. In both cases, the performance of the proposed control architecture is compared to the joints limit avoidance torque control (JLATC) [12]. All results are available on GitHub¹.

For implementing the proposed control architecture, we need some considerations:

Remark 2. Theorem 1 assumes that the controller is continuous. In both simulation and real experiments, however, the implementation of the controller is a discrete system. Thus, the controller is required to be implemented by an appropriate time step for achieving the results explained in Theorem 1. The appropriate time step depends on the system dynamics and the desired trajectory. But, a small time step drastically slows down the simulations, and is not always achievable in real applications. To resolve the requirement of small time steps, we modify the control policy as the following

$$\tau = h_{\zeta\psi} + M J_\psi \left(\dot{\psi}_r - k_2^{-1} k_1 \dot{e}_\zeta - k_2 e_\zeta - k_3 J_\psi^{-1} e_\psi \right). \quad (10)$$

Remark 3. In practice, the noise signals can cause the joints position and velocity of the system to go beyond their limits.

¹https://github.com/ami-iit/paper_pasandi_2023_icra-joint-limit-avoidance

In this case, the definition of (ζ, ψ) is not valid. To resolve this issue, we modify (5) as

$$\begin{aligned}\zeta &= \tanh^{-1} \left(\check{\delta}_q^{-1} (Sat(\mathbf{q}, \mathbf{q}_{min}, \mathbf{q}_{max}) - \mathbf{q}_0) \right), \\ \psi &= \tanh^{-1} \left(\check{\delta}_{\dot{q}}^{-1} (Sat(\dot{\mathbf{q}}, \dot{\mathbf{q}}_{min}, \dot{\mathbf{q}}_{max}) - \dot{\mathbf{q}}_0) \right).\end{aligned}\quad (11)$$

Remark 4. In practice, it can happen that the desired joints trajectory does not satisfy the joints position and velocity limits of the system. In this case, the definition of (ζ_d, ψ_d) are not valid. To resolve this issue, we modify the definition of (ζ, ψ) as the following

$$\begin{aligned}\zeta_d &= \tanh^{-1} \left(\check{\delta}_q^{-1} (Sat(\mathbf{q}_d, \mathbf{q}_{min}, \mathbf{q}_{max}) - \mathbf{q}_0) \right), \\ \psi_d &= \tanh^{-1} \left(\check{\delta}_{\dot{q}}^{-1} (Sat(\dot{\mathbf{q}}_d, \dot{\mathbf{q}}_{min}, \dot{\mathbf{q}}_{max}) - \dot{\mathbf{q}}_0) \right).\end{aligned}\quad (12)$$

Remark 5. The definition of ψ_r is well-defined if $\left| \check{\delta}_{\dot{q}}^{-1} \left(J_\zeta J_{\zeta_d}^{-1} \left(\check{\delta}_{\dot{q}} \tanh(\psi_d) + \dot{\mathbf{q}}_0 \right) - \dot{\mathbf{q}}_0 \right) \right| \leq 1$. This condition can be violated because of the error between ζ_d and ζ . Thus, we modify the definition of ψ_r as

$$\psi = \tanh^{-1} \left(\frac{\gamma}{(I_n + |\check{\gamma}|^p)^{1/p}} \right), \quad (13)$$

where $\gamma = \check{\delta}_{\dot{q}}^{-1} \left(J_\zeta J_{\zeta_d}^{-1} \left(\check{\delta}_{\dot{q}} \tanh(\psi_d) + \dot{\mathbf{q}}_0 \right) - \dot{\mathbf{q}}_0 \right)$ and $p \in \mathbb{R}$ is a positive constant.

A. Two-link manipulator

In this study, the two-link manipulator shown in Fig.1a is simulated in Simscape/MATLAB [16]. The two-link manipulator has two degrees of freedom and moves in the vertical plane. The simulation is performed in Simulink/MATLAB with the continuous integrator "ode45" and the time step of 0.01[sec].

The feasible joint position range is equal to $(-45, 90)[deg]$ and $(-90, 90)[deg]$, and the feasible joint velocity range is equal to $(-90, 90)[deg/sec]$ and $(-90, 180)[deg/sec]$ for q_1 and q_2 , respectively .

1) *Constant desired trajectory:* In this scenario, the robot reaches the constant desired joint position $(90 - 2.5, -90 + 2.5)[deg]$ from the initial joint position $(0, 0)[deg]$. The results are shown in Fig. 2. As can be seen, using the JLATC, the joint trajectory respects the predefined position limits but goes beyond the velocity limits. Instead, using the proposed control architecture, the joints trajectory converges to the desired value while respecting the predefined position and velocity limits irrespective of the large control coefficients. The control coefficients for the proposed control architecture are tuned for each joint to provide a convergence rate equal or higher than the one provided by the JLATC. In this simulation, the convergence rate provided for the second joint by the proposed control architecture is the maximum allowable convergence rate considering the corresponding joint velocity limit.

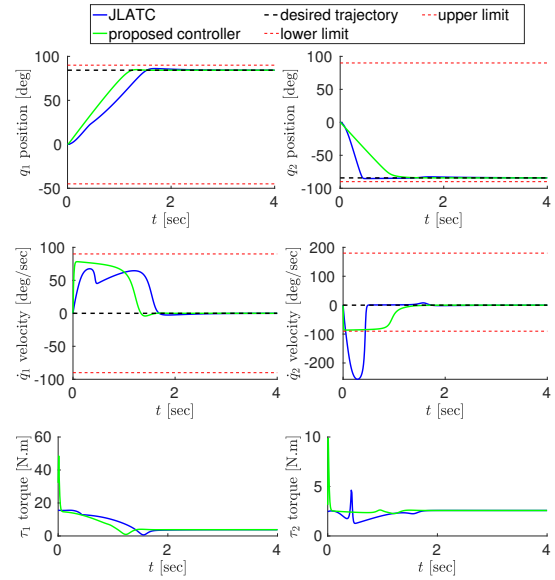


Fig. 2. The time evolution of the joints position, velocity and torque of the two link manipulator in the constant desired trajectory scenario. The coefficients for the proposed controller are $k_1 = \text{diag}(22, 505)$, $k_2 = \text{diag}(20, 50)$, and $k_3 = \text{diag}(10, 5)$. The coefficients for the JLATC are $k_p = \text{diag}(0.4, 0.1)$ and $\text{diag}(0.2, 0.1)$.

2) *Sinusoidal desired trajectory:* In this scenario, the robot tracks the sinusoidal desired trajectory $\mathbf{q} = A \sin(\omega t + \varphi_0) + b$ where $A = (45 - 5, -90 + 5)[deg]$, $\omega = (1.9, 0.9)[rad/sec]$, $\varphi_0 = (0, \pi/2)[rad]$, and $b = (45, 0)[deg]$. The initial position of the robot is $(0, 0)[deg]$. As shown in Fig. 3, using the

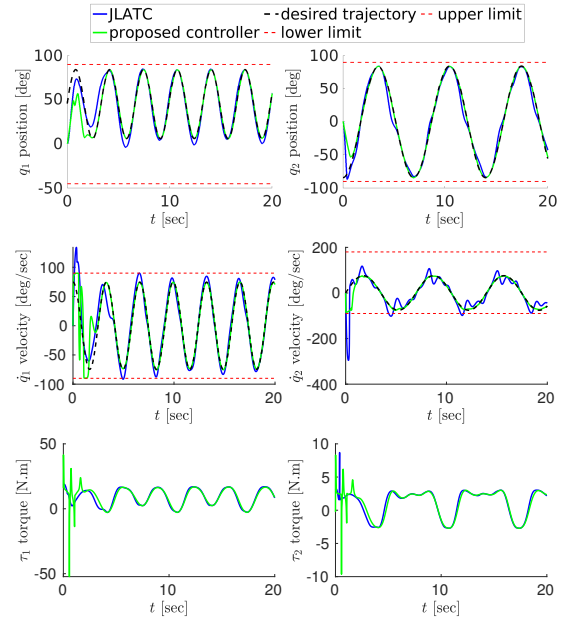


Fig. 3. The joints position, velocity and torque evolution of the two link manipulator in the tracking of the sinusoidal desired trajectory. The coefficients for the proposed controller are $k_1 = \text{diag}(22, 505)$, $k_2 = \text{diag}(20, 50)$, and $k_3 = \text{diag}(10, 5)$. The coefficients for the JLATC are $k_p = \text{diag}(0.8, 0.3)$ and $\text{diag}(0.2, 0.06)$.

proposed control architecture, the joints trajectory tracks the desired trajectory while respecting the predefined joints position and velocity limits. Instead, using the JLATC controller, the joints trajectory respects the joints position limits but not the joint velocity limits.

B. Humanoid robot iCub

In this study, the control architecture and iCub are simulated in MATLAB and Gazebo, respectively, and communicate together through Yarp channels. The MATLAB discrete integrator with the time step of $0.001[sec]$ is used as the simulation integrator.

The leg of the iCub is used for this study, forming a 3 degree-of-freedom manipulator with rotational joints at the hip-pitch, hip-roll, and knee (see Fig. 1b). The ankle joint is kept fixed by a position controller. The joint limits of iCub are given in Tab. I.

1) *Constant desired trajectory*: In this scenario, the robot reaches the desired joint position $\mathbf{q}_d = (60, 60, -90)[deg]$ for the hip pitch, hip roll and knee joints respectively, from a given initial position $\mathbf{q}(0) = (0, 0, 0)[deg]$. As can be seen in Fig. 4, using the proposed controller, the joints trajectory converges to the desired constant value and respects the predefined position limits but, in contrast to the simulation results presented in section V-A for the two-link manipulator, in the beginning of the simulation, joints velocity fails to respect its corresponding limits while joints torque noticeably oscillate. Comparing to the JLATC, the proposed control architecture has the advantage that the joints velocity trajectory goes beyond its limits just in some time instances and then goes back to its feasible range. It is worth note that the proposed controller satisfies both the joints position and velocity limits along with smooth joints torque by reducing the simulation time step. For the sake of limited space, the simulation results for the small time step are not presented here, but are available in GitHub². In fact, the proposed controller is sensitive to discretized implementation.

2) *Sinusoidal desired trajectory*: In this scenario, the robot tracks the desired sinusoidal joint position $\mathbf{q} = A \sin(\omega t + \varphi_0) + b$ where $A = (33.5, 50, -36)[deg]$, $\omega = (1.2, 1.6, 1)[rad/sec]$, $\varphi_0 = (0, \pi/3, 2\pi/3)[rad]$, and $b = (45, 36, -60)[deg]$. The initial robot joint position is $(0, 0, 0)[deg]$. The results are illustrated in Fig. 5. As can be seen, using the proposed control architecture, the joint trajectory tracks the desired sinusoidal trajectory while respecting both the joints position and velocity limits. Instead, using the

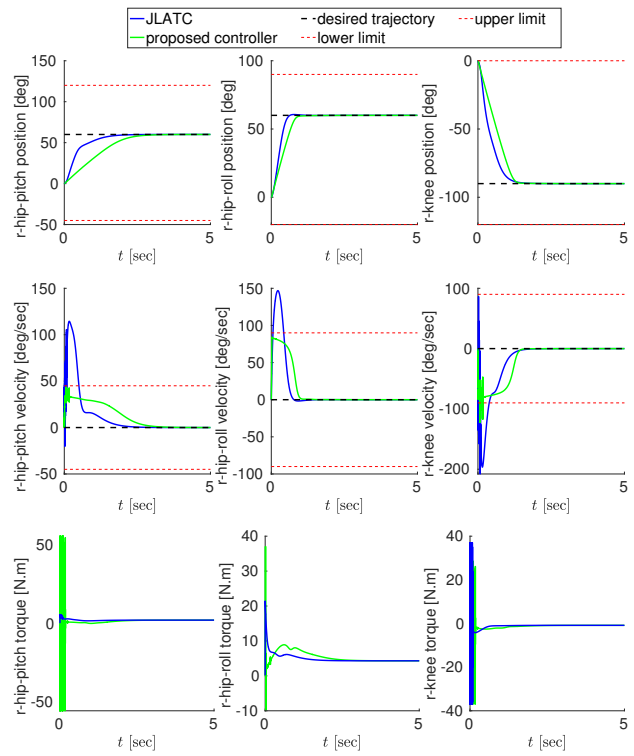


Fig. 4. The joints position, velocity and torque evolution of iCub in the constant desired trajectory scenario. The coefficients for the proposed controller are $k_1 = 2000I_3, k_2 = 310I_3$, and $k_3 = 50I_3$ and for the JLATC are $k_p = \text{diag}(8, 10, 6)$ and $\text{diag}(2, 2, 1)$.

JLATC controller, the joints trajectory goes beyond the joints velocity limits. Note that both the controllers ask high joints torques when the joints trajectory is near its position/velocity limits.

3) *Constant desired trajectory and disturbance*: In this scenario, the robot reaches the constant desired joint position $(60, 60, 90)[deg]$ for the hip pitch, hip roll, and knee joints respectively, from the initial joint position $(0, 0, 0)[deg]$. An external force in the vertical direction is applied to the sole of the robot foot. As can be seen in Fig. 6, the joints trajectory preserves the predefined joints position limits. However, the joint trajectory, as explained before, goes beyond the predefined velocity limits in some time instances at the beginning of the simulation. Later on, the joint trajectory preserves the velocity limits even in the presence of the external force.

VI. CONCLUSIONS

This paper presents a torque control architecture with joints position and velocity limits avoidance for fully actuated manipulators. The presented architecture provides the convergence of the joints position of the robot to a desired trajectory while ensuring that the time evolution of the joints position and velocity remains within the feasible space. The Lyapunov analysis proves the stability and convergence of the tracking error and the limitations of the joints position and velocity. We validated the soundness of the proposed control architecture

²https://github.com/ami-iit/paper_pasandi_2023_icra-joint-limit-avoidance

TABLE I
THE POSITION AND VELOCITY LIMITS OF THE ICUB

| joint | hip-pitch | hip-roll | knee |
|----------------------------|-----------|----------|------|
| minimum position [deg] | -45 | -20 | -120 |
| maximum position [deg] | 120 | 90 | 0 |
| minimum velocity [deg/sec] | -45 | -90 | -90 |
| maximum velocity [deg/sec] | 45 | 90 | 90 |

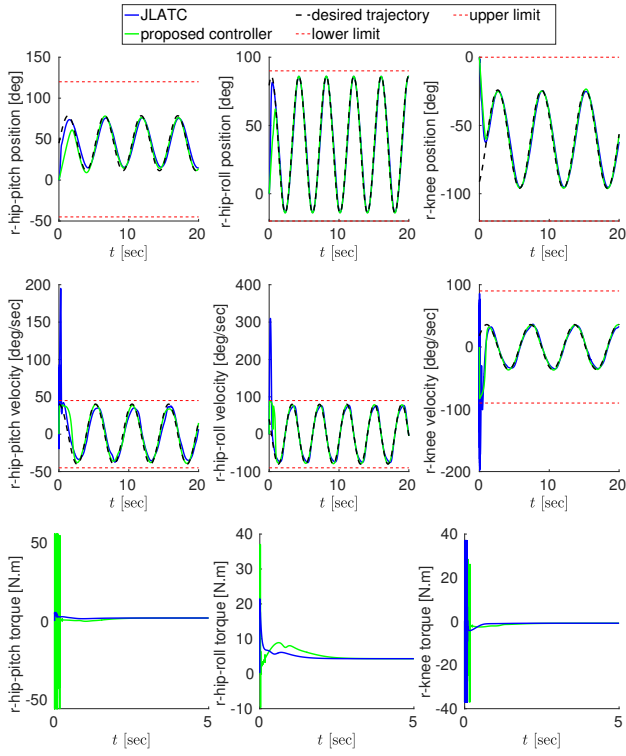


Fig. 5. The joints position, velocity and torque evolution of iCub in tracking a sinusoidal desired trajectory. The coefficients for the proposed controller are $k_1 = 2000I_3, k_2 = 310I_3$, and $k_3 = 50I_3$ and for the JLATC are $k_p = \text{diag}(8, 10, 6)$ and $\text{diag}(2, 2, 1)$.

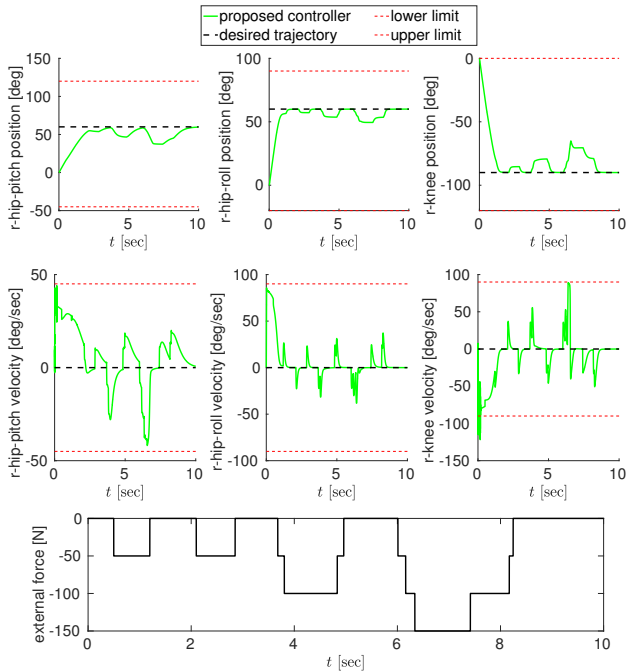


Fig. 6. The joints position, velocity and torque evolution of iCub in presence of an external force. The coefficients for the proposed controller are $k_1 = 2000I_3, k_2 = 310I_3$, and $k_3 = 50I_3$ and for the JLATC are $k_p = \text{diag}(8, 10, 6)$ and $\text{diag}(2, 2, 1)$.

by some simulations on a simple two degrees of freedom manipulator. We also investigated the limitations of the proposed architecture by some simulations on the humanoid robot iCub. We observed that, in discrete implementation, the proposed controller could generate noticeable oscillatory joints torque and fail in joints velocity limit avoidance considering the discrete step time. As future work, it is planned to reform the proposed control architecture for improving its performance in discrete implementation.

APPENDIX

A. Proof of Theorem 1

Considering the control policy presented in (9), the time derivative of (e_ζ, e_ψ) is

$$\begin{cases} J_\zeta \dot{e}_\zeta = \check{\delta}_q (\tanh(\psi) - \tanh(\psi_r)), \\ \dot{e}_\psi + k_2^{-1} k_1 \dot{e}_\zeta = -k_2 e_\zeta - k_3 e_\psi. \end{cases} \quad (14)$$

To prove the asymptotic stability of $(e_\zeta, e_\psi) = (0, 0)$, we consider the following candidate Lyapunov function

$$V = \frac{1}{2} \begin{pmatrix} e_\zeta^\top & e_\psi^\top \end{pmatrix} P \begin{pmatrix} e_\zeta \\ e_\psi \end{pmatrix} = \frac{1}{2} \begin{pmatrix} e_\zeta^\top & e_\psi^\top \end{pmatrix} \begin{pmatrix} k_1 & k_2 \\ k_2 & k_3 \end{pmatrix} \begin{pmatrix} e_\zeta \\ e_\psi \end{pmatrix}.$$

According to Theorem 1, k_1 and $k_3 - k_2 k_1^{-1} k_2$ are diagonal positive definite matrices, and thus P is a symmetric positive definite matrix based on Schur complement theorem [18]. As a result, V is a positive definite function where $V = 0$ iff $(e_\zeta, e_\psi) = (0, 0)$.

The time derivative of V along (14) is

$$\begin{aligned} \dot{V} &= (e_\zeta^\top k_1 + e_\psi^\top k_2) \dot{e}_\zeta + (e_\zeta^\top k_2 + e_\psi^\top k_3) \\ &\quad (-k_2^{-1} k_1 \dot{e}_\zeta - k_2 e_\zeta - k_3 e_\psi) \\ &= -(\psi - \psi_r)^\top (k_3 - k_2 k_1^{-1} k_2) k_2^{-1} k_1 J_\zeta^{-1} \delta_q \\ &\quad (\tanh(\psi) - \tanh(\psi_r)) - |k_2 e_\zeta + k_3 e_\psi|^2. \end{aligned}$$

Thanks to the diagonal property of $(k_3 - k_2 k_1^{-1} k_2) k_2^{-1} k_1 J_\zeta^{-1} \delta_q$, we observe that \dot{V} is negative definite. Hence, $(e_\zeta, e_\psi) = (0, 0)$ is asymptotically stable based on the Lyapunov stability theorem [19]. In conclusion,

- (e_ζ, e_ψ) is bounded, and thus (ζ, ψ) is bounded if $(\mathbf{q}_d, \dot{\mathbf{q}}) \in (\mathcal{Q}_q, \mathcal{Q}_{\dot{q}})$.
- (e_ζ, e_ψ) converges to $(0, 0)$, and thus (ζ, ψ) converges to (ζ_d, ψ_d) .

Thus, (ζ_d, ψ_d) is an asymptotically stable trajectory of the closed loop system in terms of Lyapunov stability [19]. Finally, since V is radially unbounded, (ζ_d, ψ_d) is globally asymptotically stable.

REFERENCES

- [1] C. Canudas, B. Siciliano, and G. Bastin, "Theory of robot control, communications and control engineering," 1997.
- [2] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE transactions on neural networks*, vol. 14, no. 3, pp. 658–667, 2003.
- [3] A. Dahlin and Y. Karayiannidis, "Adaptive trajectory generation under velocity constraints using dynamical movement primitives," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 438–443, 2019.

- [4] E. Almasri and M. K. Uyguroğlu, "Trajectory optimization in robotic applications, survey of recent developments," 2021.
- [5] V. Pasandi, A. Dinale, M. Keshmiri, and D. Pucci, "A programmable central pattern generator with bounded output," *Robotics and Autonomous Systems*, vol. 125, p. 103423, 2020.
- [6] A. Del Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 281–288, 2017.
- [7] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 120–127.
- [8] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.
- [9] S. Song, Y. She, J. Wang, and H. Su, "Barrier lyapunov function based control of a flexible link co-robot with safety constraints," in *Dynamic Systems and Control Conference*, vol. 51913. American Society of Mechanical Engineers, 2018.
- [10] V. Azimi and S. Hutchinson, "Exponential control lyapunov-barrier function using a filtering-based concurrent learning adaptive approach," *IEEE Transactions on Automatic Control*, 2021.
- [11] M. A. Murtaza, S. Aguilera, V. Azimi, and S. Hutchinson, "Real-time safety and control of robotic manipulators with torque saturation in operational space," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 702–708.
- [12] M. Charbonneau, F. Nori, and D. Pucci, "On-line joint limit avoidance for torque controlled robots by joint space parametrization," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 899–904.
- [13] A. Gazar, G. Nava, F. J. A. Chavez, and D. Pucci, "Jerk control of floating base systems with contact-stable parameterized force feedback," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 1–15, 2020.
- [14] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*, 2008, pp. 50–56.
- [15] B. Siciliano and O. Khatib, *Handbook of robotics*, 2008.
- [16] S. Miller, "Simscape multibody contact forces library," Feb. 2021, [Online; accessed Feb 2021]. [Online]. Available: <https://github.com/mathworks/Simscape-Multibody-Contact-Forces-Library/releases/tag/20.2.5.0>
- [17] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, pp. 2149–2154.
- [18] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.
- [19] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.