Real-Time Control of Robot Arms for Safe Human-Robot Coexistence*

Kelly Merckaert[1,2], Bryan Convens[1,3], Marco M. Nicotra[4], and Bram Vanderborght[1,3]

Abstract— **In the past years, manufacturers have been moving from automated mass production to automated mass customization, where robots share a workspace with humans in constantly changing scenarios. To avoid collisions with humans and the environment while ensuring efficient task realization, we introduce a computationally efficient control scheme that relies on the Explicit Reference Governor (ERG) formalism to enforce input and state constraints in real-time. The resulting constrained control method can steer the robot arm to the desired end-effector pose (or a steady-state admissible approximation thereof) in the presence of actuator saturation, limited joint ranges, speed limits, static obstacles, and humans. The effectiveness of the proposed solution is shown with experimental validations on the Franka Emika Panda robot arm.**

I. INTRODUCTION

In recent years, manufacturing companies are adopting mass customization strategies. The required increased flexibility in the production environment can be obtained by combining the complementary qualities of humans and robots [1], [2]. Although Human-Robot Collaboration (HRC) can bring the production line to a new level of flexibility and efficiency, the safety issue is of major importance [3].

To obtain safe motions of robot arms with kinematics constraints, sampling-based motion planning algorithms, as Rapidly exploring Random Trees (RRT), are used, e.g. [4]. However, these type of path planning algorithms do not take into account robot dynamics. Path planning approaches that consider both kinematic and dynamic constraints, i.e. kinodynamic path planners, are computationally demanding and therefore hard to implement in real-time, e.g. [5].

In contrast to RRT motion planners, kinetostatic safety fields are reactive and computationally efficient, i.e. [6]. They are based on the cumulative danger field and on the repulsive potential field approach, that are applied in [7] and [8], respectively. Yet, these approaches do not take into account actuator input constraints.

The Saturation in the Null Space algorithm generates velocity commands that allows real-time control of robots for a large number of hard limits [9]. Although it guarantees an optimal solution, it also does not take into account actuator input constraints.

Due to recent advances in computational performances, Model Predictive Control (MPC) can be implemented on robot arms to handle both state and input constraints in real-time, e.g. [10], [11]. However, the application possibilities are limited due to the typically non-negligible computational cost.

For robot control in dynamic environments, strategies based on the Speed and Separation Monitoring (SSM) and the Power and Force Limiting (PFL) criterion modulate the velocity of the robot based on the distance between the human operator and the robot, i.e. [12], [13]. However, the strategies have usually no rigorous proof of convergence in finite time and often slow down the robot excessively.

Another way to tackle the HRC safety problem is by generating reachable occupancies which account for all human movement, i.e. [14]. Although safe, this method results in a more conservative robot behavior.

We base ourselves on the Explicit Reference Governor (ERG), a closed-form feedback control scheme that can enforce both state and input constraints of nonlinear systems without having to solve an online optimization problem [15]. In [16] the idea was explored on a 2DOF robotic manipulator, here we analyze the methodology on the Franka Emika Panda robot arm.

II. TRAJECTORY-BASED EXPLICIT REFERENCE GOVERNOR

Consider the joint-space dynamic model of a robotic manipulator with n joints,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}\,, \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of joint variables, $\boldsymbol{M}(\boldsymbol{q}) > 0$ is the mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ accounts for the Coriolis and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q})$ is the influence of the gravity on the manipulator, and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the torque control input vector.

The system is subject to a variety of input and state constraints. Specifically, we consider the following classes of constraints:

- *Actuator Saturation:* the torque applied to the joints is limited, $\tau_{\text{min},i} \leq \tau_i \leq \tau_{\text{max},i}$, $\forall i = \{1,\ldots,n\}$.
- *Operating Region:* the robot arm has a limited operating range, $q_{\text{min},i} \leq q_i \leq q_{\text{max},i}$, $\forall i = \{1,\ldots,n\}$.
- *Speed Limitation:* to allow human-robot collaboration, we have to take into account the cobot's inherent joint velocity limits, $\dot{q}_{\text{min},i} \leq \dot{q}_i \leq \dot{q}_{\text{max},i}$, $\forall i = \{1,\ldots,n\}$, and its Cartesian end-effector velocity limits, $\dot{\boldsymbol{x}}_{\text{min},e} \leq \dot{\boldsymbol{x}}_e \leq \dot{\boldsymbol{x}}_{\text{max},e}$.

*This work was supported by Fonds Wetenschappelijk Onderzoek (FWO) [grant numbers 37472, 60523, and 62062], the EU H2020 project SOPHIA [grant number 871237], and the Flemish Government under the program *Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen*.

[1] Authors are with the Mechanical Engineering department, Vrije Universiteit Brussel, 1050 Brussels, Belgium. https://www.brubotics.eu/research/renmm kelly.merckaert@vub.be

[2] Kelly Merckaert is affiliated to Flanders Make, Belgium

[3] Bryan Convens and Bram Vanderborght are afilliated to imec, Belgium

[4] Marco M. Nicotra is with the Electrical, Computer, and Energy Engineering department, University of Colorado Boulder, Boulder, CO 80309, USA. https://www.colorado.edu/faculty/nicotra/

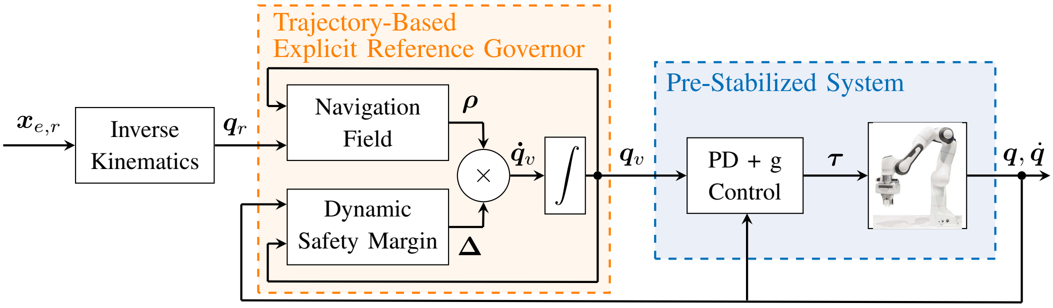

Fig. 1. Constrained Control Architecture The desired end-effector pose $\boldsymbol{x}_{e,r}$ is transformed by means of a *kinematic inversion* into a desired joint reference $\boldsymbol{q}_r$. This desired joint reference $\boldsymbol{q}_r$ is dynamically filtered by the *Trajectory-Based ERG* to the auxiliary joint reference $\boldsymbol{q}_v$ such that the target configuration will be reached while satisfying the system constraints. The trajectory-based ERG consists of a *Navigation Field*, which determines the direction of $\dot{\boldsymbol{q}}_v$, and a *Dynamic Safety Margin*, which regulates the modulus of $\dot{\boldsymbol{q}}_v$. The *PD+g control unit* pre-stabilizes the robot dynamics to the applied reference $\boldsymbol{q}_v$ and sends the computed torques $\boldsymbol{\tau}$ to the robot.

- *Obstacles:* the robot should be able to avoid a collection of possibly moving obstacles. We consider N_w walls, N_s spherical obstacles, and N_c cylindrical obstacles.
  Planar walls can be avoided by enforcing $\boldsymbol{c}_j^\mathrm{w} \cdot \boldsymbol{p}_i \leq d_j^\mathrm{w}$, $\forall i \in \{1, ..., n+1\}$, $\forall j \in \{1, ..., N_\mathrm{w}\}$, where $\boldsymbol{c}_j^\mathrm{w} \in \mathbb{R}^3$ is the unit vector normal to the j-th wall (pointing in the inadmissible direction) and $d_j^\mathrm{w} \in \mathbb{R}$ describing the distance between the plane and the robot base.
  Spherical obstacles can be avoided by enforcing $\|\boldsymbol{p}_{ij}^\mathrm{s} - \boldsymbol{c}_j^\mathrm{s}\| \geq r_j^\mathrm{s}$, $\forall i \in \{1, ..., n\}$, $\forall j \in \{1, ..., N_\mathrm{s}\}$, where $\boldsymbol{c}_j^\mathrm{s}$ and r_j^s are the center and the radius of sphere j, and $\boldsymbol{p}_{ij}^\mathrm{s}$ is the point on link i that is closest to sphere j.
  Cylindrical obstacles can be avoided by enforcing $\|\boldsymbol{p}_{ij}^\mathrm{c} - \boldsymbol{t}_{ij}^\mathrm{c}\| \geq r_j^\mathrm{c}$, $\forall i \in \{1, ..., n\}$, $\forall j \in \{1, ..., N_\mathrm{c}\}$, where r_j^c is the radius of cylinder j, $\boldsymbol{p}_{ij}^\mathrm{c}$ is the point on link i that is closest to cylinder j and $\boldsymbol{t}_{ij}^\mathrm{c}$ is the point on cylinder j that is closest to link i.

Given the joint-space dynamic model (1) and the proposed constraints, the control objective is to design a computationally efficient control scheme that can run in real-time without relying on offline pre-generated trajectories that let a robot arm reach the end-effector reference pose $\boldsymbol{x}_{e,r}$, or a steady-state admissible approximation, while satisfying the input and state constraints.
Note that the satisfaction of the obstacle constraints can be guaranteed in the case of static obstacles. For the case of dynamic obstacles, the control law avoids collision if possible and halts the robot otherwise.

Based on the problem statement, we propose a multi-layer control architecture to decouple the control problem into sub-tasks, as depicted in Fig. 1. The *control layer* pre-stabilizes the dynamics of the robot arm to the applied reference $\boldsymbol{q}_v$, which is done by a classic PD control law with gravity compensation that does not account for any system constraints. The *Trajectory-Based Explicit Reference Governor (ERG)* dynamically filters the reference $\boldsymbol{q}_r$ so that all constraints are satisfied and is also responsible for reaching the target configuration $\boldsymbol{q}_r$.
In case a Cartesian end-effector pose reference $\boldsymbol{x}_{e,r}$ is given to the robot arm, it first needs to be transformed into a joint reference $\boldsymbol{q}_r$ by means of a kinematic inversion algorithm.

Given an auxiliary reference $\boldsymbol{q}_v$, the robot arm is pre-stabilized by a classic PD control law with gravity compensation,

$$\boldsymbol{\tau} = \boldsymbol{K}_P(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{K}_D\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\,, \tag{2}$$

which leads to the following closed-loop system dynamics ,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{K}_P(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{K}_D\dot{\boldsymbol{q}}\,. \tag{3}$$

If the applied reference $\boldsymbol{q}_v$ remains constant in (3), the closed-loop equilibrium configuration $\bar{\boldsymbol{q}}_v$ is asymptotically stable. The idea behind the ERG is to generate a reference $\boldsymbol{q}_v(t)$ such that the transient dynamics of the closed-loop system cannot cause a constraint violation and $\lim_{t\to\infty} \boldsymbol{q}_v(t) = \boldsymbol{q}_r$. Rather than pre-computing suitable trajectory $\boldsymbol{q}_v(t)$, the ERG achieves these objectives by manipulating the derivative of the applied reference,

$$\dot{\boldsymbol{q}}_v = \boldsymbol{\rho}(\boldsymbol{q}_v, \boldsymbol{q}_r)\ \Delta(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_v)\ , \tag{4}$$

where $\boldsymbol{\rho}(\boldsymbol{q}_v, \boldsymbol{q}_r)$ is the *Navigation Field* (NF), i.e. a vector field that generates the desired steady-state admissible path, and $\Delta(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_v)$ is the *Dynamic Safety Margin* (DSM), i.e. a scalar that quantifies the distance between the predicted transient dynamics of the pre-stabilized system and the constraint boundaries if the current $\boldsymbol{q}_v$ were to remain constant. In the DSM, the forward system dynamics are simulated by the use of the Simplectic Euler for a finite time horizon $t' \in [t, t+T]$ with the predicted state initializations $\widehat{\dot{\boldsymbol{q}}}_{t'=t} = \dot{\boldsymbol{q}}(t)$ and $\widehat{\boldsymbol{q}}_{t'=t} = \boldsymbol{q}(t)$.

III. RESULTS

We experimentally validated this methodology on the Franka Emika Panda robot arm.

For the experiments, the pre-stabilizing control runs at 1 kHz and the ERG runs in parallel at 100 Hz. The Orocos Kinematics and Dynamics Library [17] is used for kinematic inversion for experiments where an end-effector reference pose is given to the robot. To make the influence of the robot dynamics more significant, we replaced the standard Panda