Control Parameters Tuning Based on Bayesian Optimization for Robot Trajectory Tracking

Loris Roveda $^{1,*}$, Marco Forgione $^{1,*}$ and Dario Piga $^{1,*}$

Abstract—Industrial manipulators are increasingly required to be capable of autonomously tuning their behaviour, adapting to evolving tasks. However, due to the high variability of working conditions (such as payload, motors temperature, etc.), such an adaptable behavior definition is not trivial. The aim of the paper is, therefore, to propose an approach to autonomously tune the manipulator controller on the basis of the faced task. A Bayesian optimization based algorithm is proposed to tune the controller parameters online. By the definition of a reward function (based on the required task performance), the algorithm is capable to adapt the control parameters on the basis of a trial-and-error procedure. The proposed approach has been evaluated in simulation for a trajectory tracking control problem on a 7 degrees of freedom manipulator (a Franka EMIKA robot has been considered) with uncertain dynamic model, autonomously tuning the feedback linearization control parameters and the PID trajectory control parameters (i.e., 25 optimization variables). Preliminary experimental tests have been also executed on a real Franka EMIKA robot.

I. INTRODUCTION

A. Context

Industrial manipulators have to execute highly dynamic tasks, requiring reconfigurability, adaptability and flexibility [1]. Such working conditions require, therefore, the manipulator to be adaptable to different scenarios, updating their behaviors to face the target task. In order to increase productivity and flexibility in the industrial environments, the manipulator has to autonomously adapt its behavior, in order to avoid its manual configuration by the operator, that is time-consuming. Some works faced the issue of self-tuned controllers, proposing different methodologies and target performance. [2] proposed an iterative reinforcement learning approach for the friction and control parameters tuning in interaction tasks. [3] investigates a neural network approach to self-tune the robot control in an object balancing task. [4] proposes a sensor-based strategy to self-tuning the robot impedance control for interaction tasks. [5] defines a fuzzy logic approach to self-tuning the PID controller of a flexible single link manipulator. In general, the approaches available in the literature are mostly dealing with few degrees-of-freedom systems, simulations studies or require complex implementation/experimental procedures.

B. Paper Contribution

This paper aims to apply Bayesian optimization for self-tuning control parameters, in order to optimize the manipulator behavior on the basis of the required task performance, considering a trajectory tracking control problem. In such a way, a simple and fast procedure can be developed and applied to industrial manipulators in real working scenarios. A PID trajectory control loop with feedback linearization is implemented as a control strategy. The aim of the Bayesian optimization algorithm is twofold: (i) tuning of the PID control gains; and (ii) tuning of the equivalent link-mass parameters used by the feedback linearization control loop. As an optimization function, position and velocity errors (considering the average trajectory errors and the peak errors at both position and velocity levels for all the robot joints) have been considered in order to guide the control gains learning. The proposed approach has been evaluated in simulation on a 7 degrees of freedom manipulator (a Franka EMIKA robot has been considered) with uncertain dynamic model. The proposed procedure is capable to autonomously tuning in the same time the feedback linearization control parameters and the PID trajectory control parameters (i.e., 25 optimization variables). The simulation has been implemented in C++ exploiting KDL libraries for the robot modeling [6] and the limbo library for Bayesian optimization [7]. The algorithm converges in less then 50 iterations. Preliminary experimental results have been executed on a real Franka EMIKA robot exploiting the same programming framework as for the simulation part, splitting the optimization of the feedback linearization controller (i.e., 4 optimization variables) and the PID controller (i.e., 21 optimization variables) for safety reasons. The proposed approach shows promising performance (convergence of proposed results within 50 iterations for each the optimization loops, with a maximum position error in the trajectory tracking of 0.3\degree).

II. PROBLEM FORMULATION

The aim of the paper is to achieve a self-tuning of the control parameters in order to perform a trajectory tracking application. Figure 1 shows the control scheme with online optimization based on the Bayesian optimization approach. In particular, the PID trajectory tracking control parameters together with the equivalent robot mass parameters used in the feedback linearization controller are online optimized in

$*$ First author

The work has been developed within the project DAEDALUS - Distributed control and simulation platform to support an ecosystem of digital automation developers, funded by the European Commission under the grant agreement H2020-723248.

1 Loris Roveda, Marco Forgione and Dario Piga are with Istituto Dalle Molle di studi sull’Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera italiana (USI), via Cantonale - 6928, Manno, Switzerland loris.roveda@idsia.ch
order to achieve target performance in the trajectory tracking application. Such performance measurement, used in the Bayesian optimization approach as the objective function, takes into account position and velocity errors (considering the average trajectory errors and the peak errors at both position and velocity levels for all the robot joints).

A. Robot Dynamics

To implement the PID trajectory tracking controller with feedback linearization, the dynamics modeling of the manipulator is required [8]:

\[
B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + h_{f,q}(q) = \tau - J(q)^T h_{ext}
\]  

(1)

where, \(B(q)\) is the robot inertia matrix, \(C(q, \dot{q})\) is the robot Coriolis vector, \(g(q)\) is the robot gravitational vector, \(h_{f,q}(q)\) is the robot joint friction vector, \(q\) is the robot joint position vector, \(J(q)\) is the robot Jacobian matrix, and \(h_{ext}\) is the robot external force/torque vector, \(\tau\) is the robot joint torque vector.

B. Robot Control Design

The PID trajectory tracking controller with feedback linearization is defined as follows:

\[
\tau = \tau_{PID} + \tau_{FL}
\]

(2)

where \(\tau_{PID}\) implements the PID trajectory tracking controller as in (3) and \(\tau_{FL}\) implements the feedback linearization controller as in (4):

\[
\tau_{PID} = B(q) \left( q^d + K_p e_q - K_d \dot{e}_q + K_i \int e_q \right)
\]

(3)

\[
\tau_{FL} = C(q, \dot{q}) + g(q)
\]

(4)

\(e_q = q^d - q\) is the position error (where \(q^d\) is the position reference).

While (3) aims to improve the performance of the trajectory tracking, (4) aims to compensate for the Coriolis and gravity terms of the dynamic equation. In order to achieve the required trajectory tracking performance, it is therefore important to tune the PID control gains \(K_p, K_d, K_i\) in (3) (21 parameters in total) and the link mass parameters \(m_{\text{link}}\) in (4) (4 parameters have been considered: equivalent mass of link 1, equivalent mass of link 2, equivalent mass of link 4, equivalent mass a of link 7).

III. BAYESIAN OPTIMIZATION BASED CONTROL PARAMETERS TUNING

A. Approach Description

The performance achieved by the controller may be quantified in terms of a user-defined performance index \(J\) penalizing, e.g., position and velocity tracking errors norm for all the joints. This quantity depends of course on the control design parameters, i.e., the PID gains and the mass links used for feedback linearization. In this paper, the control parameters are directly optimized performing a sequence of experiments guided by Bayesian optimization on the real system. After running an experiment, the achieved performance index \(J\) is measured and, based on this information, the Bayesian optimization algorithm suggests the next point to be tested according to a criterion that automatically trades off exploitation (trying points where the expected performance index \(J\) is high) and exploration (trying points where the uncertainty is high) [9]. By doing so, Bayesian optimization enables finding a nearly optimal control parameter (with respect to the performance index of interest) with a limited budget in term of (possibly costly) experiments on the robot.

In the Bayesian optimization algorithm, the relation between the control parameters and the performance index \(J\) is modelled as a Gaussian Process (GP) [10] whose \(a\ posteriori\) mean and covariance are iteratively updated throughout the experiments using the previous measurements of \(J\). The advantage of the GP modelling approach is twofold. First, GP it is a very flexible, non-parametric model allowing one to describe a complex relation linking the control parameters to \(J\). Second, it naturally provides an uncertainty measure of this relation (namely, the covariance) that Bayesian optimization explicitly exploits to find the optimal exploitation/exploration trade-off.

B. Objective Function

In order to optimize the control parameters in 2, the following performance index has been selected in order to take into account errors at position and velocity levels:

\[
J = - \sum_{i=1}^{7} \left( \max e_{j,i} + \max e_{\dot{j},i} + \text{mean} e_{j,i} + \text{mean} e_{\dot{j},i} \right)
\]

(5)

where the sign – is because the limbo library that has been used to apply the Bayesian optimization maximizes the optimization function and \(\max e_{j,i} = \max (e_q(t)_i)\),
The aim of such controller is only to avoid unstable behaviors of the manipulator. Second, the PID control parameters have been optimized. Results show the convergence of the approach within 50 iterations for each optimization procedure, with a final maximum error of 0.3° for each robot joint during the execution of the planned Cartesian motion (consistent with simulation results). Such results are comparable with the Franka EMIKA position controller executing the same trajectory.

V. CONCLUSIONS

The paper proposes an approach to autonomously tune the manipulator controller on the basis of the faced task. A Bayesian optimization based algorithm is proposed to tune the controller online. A trajectory tracking control problem has been considered, optimizing both feedback linearization controller and PID trajectory controller. The approach converges quickly (only 50 iterations are required to achieve high tracking performance), making possible to apply it in real tasks. While the feedback linearization controller parameters are equivalent to the link mass (as a physical meaning), such procedure may result in slightly different values from the real link masses. The idea is to online optimize such parameters in order to have higher control performance, therefore do not provide an identification of the real dynamic robot parameters. The proposed approach can be applied to different tasks (e.g., tuning of force control parameters). Such an extension is under investigation.

REFERENCES