Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 50-1 (2017) 12095–12100

Iterative Learning of Feasible Time-optimal Trajectories for Robot Manipulators

Armin Steinhauser * **Jan Swevers** *

* *KU Leuven, Department of Mechanical Engineering Celestijnenlaan 300B, 3001 Heverlee, Belgium (e-mail: {armin.steinhauser, jan.swevers}@kuleuven.be)*

Abstract: Time-optimal trajectories describe the minimum execution time motion along a given geometric path while taking system dynamics and constraints into account. By using a model of the real plant, inputs are provided that ought to yield minimal execution time and good tracking performance. In practice however, due to an imperfect model, the computed inputs might be suboptimal, result in poor tracking or even be infeasible in that they exceed given limits. This paper therefore presents a novel two-step iterative learning approach for industrial robots to find time-optimal, yet feasible trajectories and improve the tracking performance by repeatedly updating the nonlinear robot model and solving a time-optimal path tracking problem. The proposed learning algorithm is experimentally validated on a serial robotic manipulator, which shows that the developed approach results in reduced execution time and increased accuracy.

Keywords: Learning control, Iterative improvement, Time-optimal control, Robotic manipulators, Parameter estimation

1. INTRODUCTION

Iterative learning control (ILC) has been intensely researched to improve the performance of repetitive processes for over 30 years with a first mentioning in the English speaking community in Arimoto et al. (1984). Executing the same task repeatedly, the output error of one execution is used to update the input of the next run by implicitly performing a model correction and a model inversion. As summarized in Bristow et al. (2006), wafer stages, batch processes in chemical plants and robotic applications are amongst the mentionable applications. Although there are approaches for applying ILC to nonlinear systems with specific structures, see Xu (2011), the majority of algorithms requires a linear model of the considered system. A generic approach for nonlinear systems was first presented in Volckaert et al. (2010) and elaborated in Volckaert et al. (2011) and Volckaert et al. (2013). It is shown that norm-optimal ILC can be interpreted as a two-step procedure: First computing an explicit model correction and subsequently inverting the corrected system dynamics. Within this framework, both steps are formulated as optimization problems and are efficiently solved.

The research on optimal path tracking for robotic manipulators can be traced back to the early 1970s, such as the contribution of Kahn and Roth (1971). Given a geometrically defined path, the main aim is to find a feasible trajectory that is optimal w.r.t. a desired objective, e.g. minimal execution time or energy. A feasible solution thereby has to meet defined constraints, e.g. on the inputs or outputs, and describes a mapping of the geometric path to a time-dependent trajectory. Various approaches to find such an optimal solution for complex system dynamics have been proposed, reaching from bang-bang acceleration profiles as presented in Bobrow et al. (1985), to specific path-parametrizations and subsequent optimization as in Verscheure et al. (2008). Considering robotic manipulators, it was shown in Debrouwere et al. (2013) that a number of characteristics can be exploited that yield efficient convex-concave optimization problems.

Although both topics attracted wide attention in the past, their combination is rare due to their unlike nature. While optimal path tracking operates in time domain, ILC concerns the so-called iteration domain and assumes identical execution time for every iteration. The spatial-based ILC algorithm introduced in Moore et al. (2007) can be considered the first attempt of combination but restricts the inputs to be either on or off and requires a bang-bang velocity reference. Therefore, no other system limitations or objectives such as the input effort can be taken into account. In Janssens et al. (2013) a more extensive approach including the solution of an optimal path tracking problem is proposed. Due to the simplicity of the studied application, the algorithm needs further elaboration to be used for a robotic manipulator. A recent contribution of Milosavljevic et al. (2016) investigates a different approach where the time-optimal trajectory is fixed while the ILC converges and shows simulation results for a robotic application.

In this paper we propose a novel two-step iterative learning approach for a path tracking problem applied to a robotic manipulator that directly combines ILC and optimal path tracking. We implement an explicit model correction that represents the first step of an ILC algorithm and substitute the second step by a state-of-the-art optimal path tracking algorithm to obtain not only a feasible but also time-optimal trajectory. As a result, the inputs for this trajectory, given the latest model estimate, are obtained in every iteration.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.2123

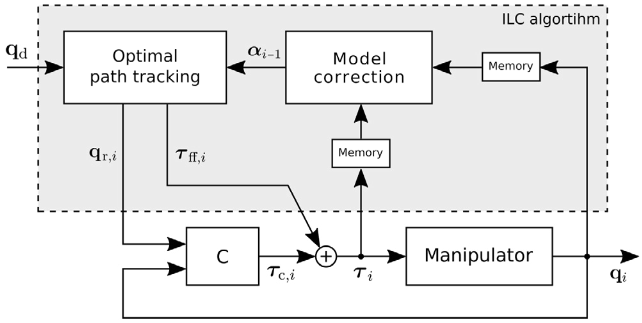

Fig. 1. Configuration of the closed-loop and scheme of the iterative learning algorithm.

The remainder of this paper is organized as follows. Section II introduces the nomenclature and terms that are used. Section III details the algorithm by outlining the proposed procedure and describing the separate steps of model correction and optimal path tracking. Results of an experimental validation are given together with details of the considered setup in Section IV, and Section V concludes this paper.

2. PRELIMINARIES

We consider a serial robotic manipulator with n degrees of freedom and define its joint angles as minimal coordinates $\mathbf{q} \in \mathbb{R}^n$. Following Siciliano et al. (2010), one can derive the manipulator's equations of motion

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}_\mathrm{v}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{f}_\mathrm{c}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \qquad (1)$$

by utilizing the Euler-Lagrange formalism, where $\boldsymbol{\tau} \in \mathbb{R}^n$ are the joint torques, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the matrix of Coriolis and centrifugal forces, $\mathbf{f}_\mathrm{v} \in \mathbb{R}^n$ and $\mathbf{f}_\mathrm{c} \in \mathbb{R}^n$ are vectors of viscous and Coulomb friction forces, respectively, and $\mathbf{g} \in \mathbb{R}^n$ is a vector of gravitational forces. Furthermore, we introduce an abbreviated form as

$$\boldsymbol{\tau} = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \qquad (2)$$

and assume that a – potentially bad – initial estimate of the model parameters appearing in the matrices of (1) is known, e.g. from data sheet values. Finally, a controller

$$\mathcal{C}(\mathbf{x}_\mathrm{c}, \mathbf{q}_\mathrm{r}, \mathbf{q}) = \boldsymbol{\tau}_\mathrm{c} \qquad (3)$$

is introduced with the controller's states $\mathbf{x}_\mathrm{c}$, its output $\boldsymbol{\tau}_\mathrm{c}$ and the joint angle reference $\mathbf{q}_\mathrm{r}$ that represents the closed-loop input. Considering additional feed-forward torques $\boldsymbol{\tau}_\mathrm{ff}$, we write the manipulator's input as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_\mathrm{ff} + \boldsymbol{\tau}_\mathrm{c} \qquad (4)$$

and close the loop as shown in Fig. 1.

3. ALGORITHM

This section introduces the overall structure of the proposed algorithm and subsequently explains the two separate steps, namely nonparametric model correction and optimal path tracking. Further details on matters specific to the implementation are given in the corresponding subsections.

3.1 Procedure

Consider the current iteration i, corresponding values denoted with subscript i and measured values denoted with subscript m. After applying the reference $\mathbf{q}_{\mathrm{r},i}$ and possibly feed-forward torques $\boldsymbol{\tau}_{\mathrm{ff},i}$, every cycle of the algorithm consists of the following two steps:

(1) Nonparametric model correction

Using the measured input $\boldsymbol{\tau}_{\mathrm{m},i}$ and output $\mathbf{q}_{\mathrm{m},i}$, the inverse dynamics of the manipulator (1) are updated by nonparametric correction terms $\boldsymbol{\alpha}_i$ to yield the improved model

$$\boldsymbol{\tau} = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \boldsymbol{\alpha}_i. \qquad (5)$$

(2) Optimal path tracking

Taking this correction into account, an optimal path tracking problem is solved to obtain a time-optimal and feasible trajectory. The resulting trajectory in joint space and the corresponding torques represent the next iteration's pair of inputs $\mathbf{q}_{\mathrm{r},i+1}$ and $\boldsymbol{\tau}_{\mathrm{ff},i+1}$, respectively.

Note: Prior to the first full ILC iteration denoted by $i = 1$, one iteration using the uncorrected model, i.e. $\boldsymbol{\alpha}_0 = \mathbf{0}$, is executed to yield the initial trajectory $\mathbf{q}_{\mathrm{r},1}$ and $\boldsymbol{\tau}_{\mathrm{ff},1}$.

3.2 Iterative Model Correction

Through numerical differentiation of the measured joint angles $\mathbf{q}_{\mathrm{m},i}$, estimates of the joint velocities $\dot{\mathbf{q}}_{\mathrm{m},i}$ and accelerations $\ddot{\mathbf{q}}_{\mathrm{m},i}$ are obtained, and used to define the model mismatch:

$$\Delta\boldsymbol{\tau}_i = \boldsymbol{\tau}_{\mathrm{m},i} - \mathcal{T}(\mathbf{q}_{\mathrm{m},i}, \dot{\mathbf{q}}_{\mathrm{m},i}, \ddot{\mathbf{q}}_{\mathrm{m},i}). \qquad (6)$$

Although – given this iteration's measurements – this would be the ideal model correction, it is in general not an advisable choice due to various error sources. Therefore, the task of finding a nonparametric model correction is cast into a rather simple optimization problem that also allows additional regularization terms:

$$\begin{aligned} \underset{\boldsymbol{\alpha}_i}{\text{minimize}} \quad & \|\Delta\boldsymbol{\tau}_i - \boldsymbol{\alpha}_i\|_2 + \gamma_1\|\boldsymbol{\alpha}_i\|_2 + \\ & \quad + \gamma_2\|\Delta_i\boldsymbol{\alpha}\|_2 + \gamma_3\|\Delta_k\boldsymbol{\alpha}\|_2 \\ \text{subject to} \quad & \Delta_i\boldsymbol{\alpha} = \boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1}, \\ & \Delta_k\boldsymbol{\alpha} = \boldsymbol{\alpha}_i(k+1) - \boldsymbol{\alpha}_i(k), \\ & \qquad \text{for } k \in \{0, \ldots, N_i - 1\}, \end{aligned} \qquad (7)$$

where N_i is the number of samples of the current iteration's measurements. While the first term of the objective seeks the solution of the bare model correction problem, the appended regularizations ensure robust convergence in time and iteration domain. Choosing $\gamma_1 > 0$ penalizes the total model correction and can therefore be used to prevent the correction terms from getting too big. The term weighted by $\gamma_2 \geq 0$ regularizes the change of model correction in iteration domain and increases the robustness of the learning by disregarding iteration-varying disturbances. The last term, weighted by $\gamma_3 \geq 0$, regularizes the change of model correction in time domain and therefore increases the robustness in consideration of noise on the measured signals. Remark that, although these regularizations and the thereby increased robustness sounds appealing, there is an inevitable trade-off between