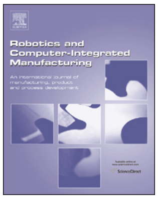
Robotics and Computer-Integrated Manufacturing 39 (2016) 9–21

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Full length Article

Accurate sensorless lead-through programming for lightweight robots in structured environments

Matteo Ragaglia, Andrea Maria Zanchettin, Luca Bascetta, Paolo Rocco

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo Da Vinci 32, Milano, Italy

ARTICLE INFO

Article history:
Received 5 February 2015
Received in revised form
10 August 2015
Accepted 6 November 2015
Available online 11 December 2015

Keywords:
Lead-through
Sensorless
Robot programming
Safety
Robot teaching

ABSTRACT

Nowadays, programming an industrial manipulator is a complex and time-consuming activity, and this prevents industrial robots from being massively used in companies characterized by high production flexibility and rapidly changing products. The introduction of sensor-based lead-through programming approaches (where the operator manually guides the robot to teach new positions), instead, allows to increase the speed and reduce the complexity of the programming phase, yielding an effective solution to enhance flexibility. Nevertheless, some drawbacks arise, like for instance lack of accuracy, need to ensure the human operator safety, and need for force/torque sensors (the standard devices adopted for lead-through programming) that are expensive, fragile and difficult to integrate in the robot controller.

This paper presents a novel approach to lead-through robot programming. The proposed strategy does not rely on dedicated hardware since torques due to operator's forces are estimated using a model-based observer fed with joint position, joint velocity and motor current measures. On the basis of this information, the external forces applied to the manipulator are reconstructed. A voting system identifies the largest Cartesian component of the force/torque applied to the manipulator in order to obtain accurate lead-through programming via admittance control. Finally an optimization stage is introduced in order to track the joint position displacements computed by the admittance filter as much as possible, while enforcing obstacle avoidance constraints, actuation bounds and Tool Centre Point (TCP) operational space velocity limits. The proposed approach has been implemented and experimentally tested on an ABB dual-arm concept robot FRIDA.

1. Introduction

Nowadays industrial robots are able to offer fast and accurate task execution in various industrial fields, but they are inherently characterized by a low degree of adaptability to rapidly changing task specifications. The fact that programming an industrial manipulator is a complex and time-consuming activity represents one of the main weaknesses of today's industrial robotic systems, since it is preventing industrial manipulators from being massively used in SMEs, where small size production and rapidly changing product features request the highest production flexibility.

The introduction of lead-through programming (from now on “LTP”) approaches [1,2] has definitely helped increasing the speed and reducing the complexity of the programming phase, by allowing the human operator to manually guide the robot in order to teach new positions. Nevertheless, these solutions present some drawbacks as well. Not only lead-through programming cannot guarantee the same level of accuracy achievable with a teach pendant-based programming, but it can be also potentially unsafe since it requires physical Human–Robot Interaction (pHRI).

Moreover, it is difficult to customize LTP in terms of both selecting directions of motion and/or enforcing speed limits. Although the possibility of constraining the movement of the robot during LTP has been addressed in the field of surgical robotics [3–5], the proposed approaches consist in simply defining a safety envelope from which the manipulator end-effector cannot exit, while it is guided by the human. Only at a later stage [6], the formalization of more generic constraints for lead-through programming of surgical robots was introduced.

Furthermore LTP relies on dedicated hardware, i.e. force/torques sensors. Although various manipulators have been designed by introducing force/torques sensors and/or compliant joints [7–11], the great majority of industrial manipulators is not inherently equipped with hardware that enables LTP. Adding a force/torque sensor to a standard industrial robot is a rather expensive and difficult operation.

E-mail addresses: matteo.ragaglia@polimi.it (M. Ragaglia), andreamaria.zanchettin@polimi.it (A.M. Zanchettin), luca.bascetta@polimi.it (L. Bascetta), paolo.rocco@polimi.it (P. Rocco).

http://dx.doi.org/10.1016/j.rcim.2015.11.002

In order to overcome this particular limitation, several approaches to the problem of sensorless detection of human–robot physical contact have been proposed in the literature, mainly based on fault detection and isolation algorithms fed with motor torque (or alternatively motor current) measurements [12]. As an example, the ABB RobotWare Software running on the IRC5 industrial controller includes this kind of feature [13], while Geravand et al. [14] present a safe collision detection and reaction strategy developed on an industrial manipulator with a closed control architecture.

Nevertheless, sensorless collision detection is not sufficient to completely overcome the need for dedicated hardware, since real-time estimation of the values achieved by interaction forces applied by the human operator to the manipulator (and not just their occurrence) is obviously necessary to perform sensorless LTP.

The problem of sensorless real-time estimation of external forces (or torques due to these forces) has been addressed in the literature, as well. The most relevant proposed solutions (like for instance [15–20]) rely on the calculation of residuals based on the manipulator generalized momentum. Typically estimation of external torques is used as input for impedance and/or admittance control strategies, like for instance in [21,22]. Other strategies for contact force estimation based on both friction estimation and detuning of the low-level joint control loops, have been proposed in [23–25].

While several force/torque estimation strategies have been developed in order to avoid the use of dedicated sensors, the possibility to apply these strategies in the field of LTP has not been thoroughly explored yet. As a matter of fact, a complete and well-established framework for sensorless, accurate and safe LTP of a typical industrial manipulator in a structured environment is still missing. This possibility to teach an industrial robot by manually driving it without the need of dedicated hardware and taking into account motion accuracy and, above all, operator's safety, definitely represents an interesting enhancement in both the fields of robot programming and pHRI.

In this sense, the main contribution of this work consists in introducing a novel LTP strategy for industrial manipulators. This strategy combines the positive aspects of traditional teach pendant-based techniques (accuracy, high level of customization, safety and no need for additional hardware) with the advantages brought by lead-through techniques (reduced programming time and user friendliness), while mitigating their respective limitations.

More in depth, the key innovative features of the proposed LTP strategy can be summarized as follows:

- *Accuracy*: A voting system and a Finite State Machine (FSM) work together to select the largest Cartesian component of the forces/moments applied by the operator, thus allowing him/her to modify only one operational space degree of freedom at a time;
- *Safety*: The output of the admittance filter is processed by an optimization stage in order to satisfy actuation bounds, safety-related limits on operational space TCP velocity and avoidance of known obstacles. Moreover, a redundancy resolution algorithm ensures that the entire manipulator kinematic chain does not collide with workspace objects.

Finally, from a methodological point of view, the formalization of the "safety constraints", originally presented in [26], is here extended from the case of point-shaped obstacles to the case of arbitrarily-shaped convex obstacles.

The remainder of this work is organized as follows. The formulation of the manipulator dynamic model is described in Section 2. Section 3 presents a detailed explanation of the proposed programming algorithm, whose main building blocks are the model-based estimator of external interaction forces/moments, the voting system, the FSM, the admittance filter, the optimization stage, and the redundancy resolution criterion. Section 4 introduces the experimental setup and presents the validation of the proposed programming strategy performed on the ABB dual-arm concept robot FRIDA (see Fig. 1). Finally, Section 5 discusses the obtained results and presents possible developments.

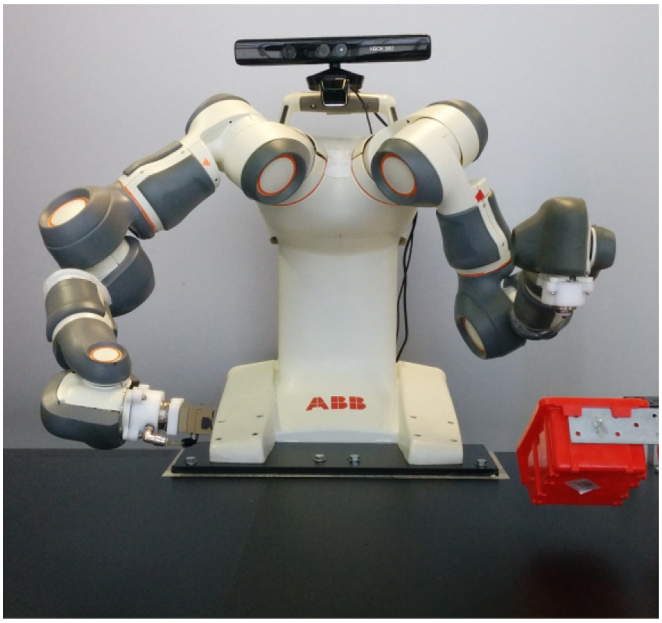


Fig. 1. ABB prototype robot FRIDA. The picture also shows the experimental setup for Experiment #1 and Experiment #2.

2. Manipulator dynamic model

The manipulator dynamic model is expressed, in Euler–Lagrange formulation, by the following equation:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\ \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \qquad (1)$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ represent joint positions, velocities and accelerations, respectively, $\boldsymbol{g}$ is the gravity acceleration vector, $\boldsymbol{B}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ represent the inertia matrix, the Coriolis-centrifugal matrix and the gravitational vector, term respectively, $\boldsymbol{F}(\dot{\boldsymbol{q}})$ is the function modelling friction torques, $\boldsymbol{\tau}$ denotes motor torques. $\boldsymbol{\tau}_{ext}$ represents the external interaction torques due to forces/moments $\boldsymbol{\mu}$ applied to the manipulator, given by the following relation:

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}_C(\boldsymbol{q})^T \boldsymbol{\mu} \qquad (2)$$

where $\boldsymbol{J}_C(\boldsymbol{q})$ is the Jacobian associated to the contact point C.

DH kinematic parameters and masses of the links, gear ratios, centre of gravity positions, inertia tensors, and motor inertias are assumed to be a priori known from manufacturer data-sheets. A very simple, yet effective, friction model has been chosen. The model consists of a viscous friction term plus a static friction component. A linear approximation of the discontinuous static friction is adopted for joint velocities in a range $|\dot{q}_i| \le 0.01$. Identification of friction coefficients is performed via Least Square error minimization with non-negativity constraints. Section 4.1 provides validation results that confirm the accuracy of both the known and the identified parameters for the given case.

Although it is outside the scope of this work, it would be possible to consider more sophisticated friction models in order to