

Robot programming by demonstration

Citation for published version (APA):

Denasi, A., Verhaar, B. T., Kostic, D., Bruijnen, D. J. H., Nijmeijer, H., & Warmerdam, T. P. H. (2009). Robot programming by demonstration. In *Proceedings of the 10th Philips Conference on Applications of Control Technology, 3-4 February 2009, Hilvarenbeek, The Netherlands* (pp. 149-153).

Document status and date:

Published: 01/01/2009

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Robot Programming by Demonstration

A. Denasi^{1,2}, B. T. Verhaar¹, D. Kostić², D.J.H. Bruijnen¹, H. Nijmeijer², T.P.H. Warmerdam¹

¹ Philips Applied Technologies, Mechatronics Dep. {alper.denasi, boudewijn.verhaar, dennis.bruijnen, t.p.h.warmerdam}@philips.com

² Dynamics and Control Group, Dep. Mech. Eng., Technische Universiteit Eindhoven a.denasi@student.tue.nl, {d.kostic, h.nijmeijer}@tue.nl

Abstract: The presented poster shows a pilot development of a robot 'task programming method'. In this method, the user programs the robot task by demonstrating it (Programming by Demonstration, PbD). PbD is applied on a robotic arm with 2 degrees-of-freedom shown in Figure 1 for programming a constrained motion task.

Introduction

In our daily lives, many time-consuming and tedious household tasks are partially or entirely done by machines, such as the laundry, dishes, vacuuming, etc. Nevertheless, there are still many time-consuming household tasks that are (partially) done by humans. Examples of such tasks are laundry ironing and folding, loading the dish washer, cooking, etc. An important limitation to the application of robots to such tasks is the combinatorial explosion of number of situations the robot may enter, due to large number of sub-tasks, states, possible environments in which the robot must be able to operate, and possible exceptions that can occur during execution.

To execute such tasks, a certain level of autonomy is required. In theory it is possible to develop software that takes care of the execution, using a 'regular' programming language such as C++. Unfortunately, this is a tedious job, since each of many different situations mentioned above must be detected, and for each of these a functional strategy must be defined. Furthermore, the developed software must be general enough to program robot operation in many different environments, covering numerous and diverse sub-tasks.

Alternatively, programming of the robot tasks can be left to the end-user. In this case, there is less uncertainty about the environment, and the set of possible situations is likely to be smaller. A drawback is that the end-user needs knowledge on the robot-specific technologies, such as programming, coordinate systems, mechanics, control design and implementation, etc.

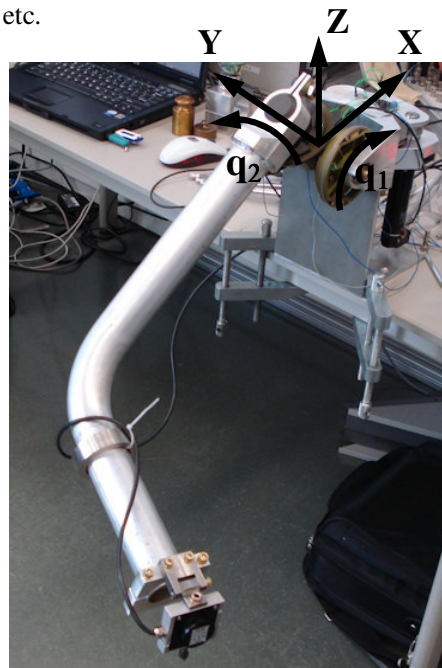


Figure 1: 2 d.o.f. robot with base frame

Programming by demonstration can be an efficient and intuitive method for unskilled end-users to teach skills to robots. The objective is that after a human demonstrates a task, the robot reproduces the essential aspects of this task correctly. The essential aspects might be for example, bringing the robot into contact, applying force on a surface, etc. This poster focuses on PbD of 'blind' tasks, i.e. tasks that can be re-played by making use of only position and force sensors, and without vision equipment (i.e. cameras). It is assumed that easy programming can be achieved by imitation learning, where the user demonstrates a new task by means of e.g. a haptic interface, and the robot "learns" how to repeat it. This poster illustrates an implementation of imitation learning based on "segmented replay". The idea is to use the measured position and force data in order to characterize atomic tasks in different segments and transitions between these tasks (or segments). The robot imitation consists of a replay of these segments and the associated transitions. We can distinguish two phases in this problem: in the first phase, the necessary data is collected during the task demonstration; in the second phase, this data is processed offline so that the essential aspects of the task can be executed correctly by the robot.

Solution method

A task will typically consist of a sequence of interactions of the robot with the environment. One example is shown in Figure 2: move to the object (segment I), then grip it (segment II), and finally release it (segment III).

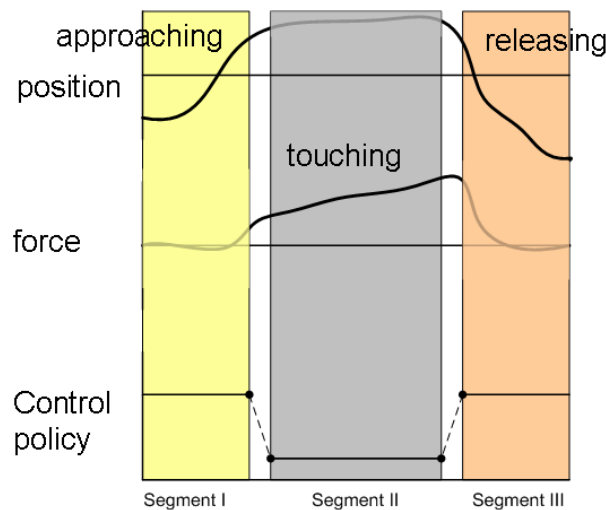


Figure 2: Manipulation sequence example

The desired control strategy within different segments is likely to be different. For example:

- Segment I : move to the object → position control
- Segment II : grip it → force control

To implement this framework, it is important to generate different control policies in different segments and to have the ability of switching between these control policies. At this point, we assume that two important policies need to be implemented: position control and force control.

The challenge of imitation learning consists of distilling from the traced data the sequence of control policy segments, the associated force and/or position setpoints, and the switching conditions for going from one segment to another. Estimating (or perceiving) the mechanical impedance that the environment exhibits during interaction with the manipulator allows intuitive selection of the correct control policies in different segments. The solution method is summarized by a diagram shown in Figure 3.

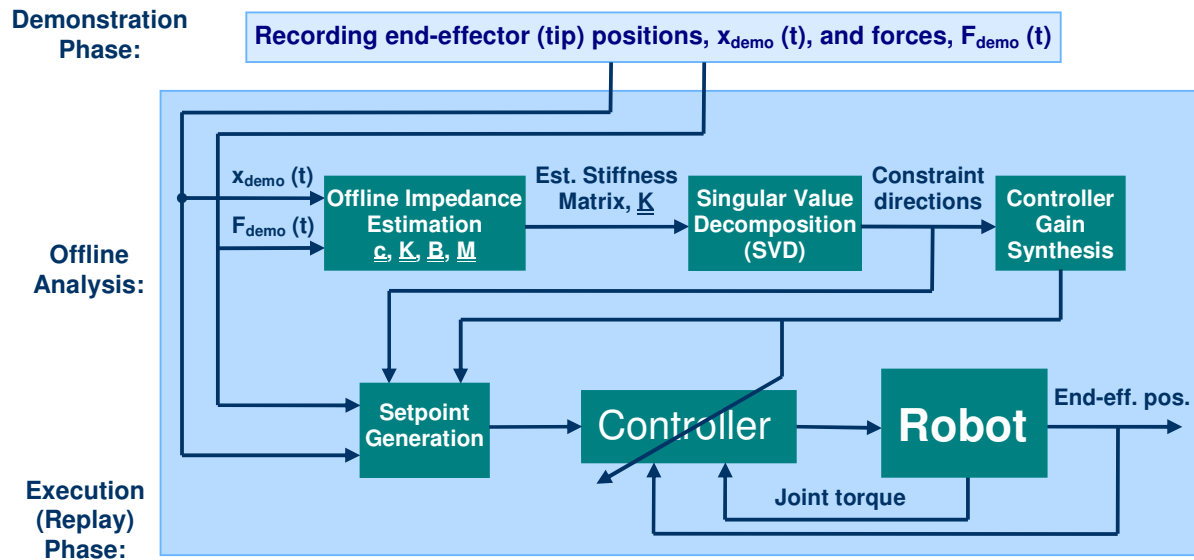


Figure 3: Demonstration—Analysis—Execution block diagram

The offline impedance estimation is done by fitting the following dynamic equation by using weighted least squares

$$f(t) = \underline{c} + \underline{K} * \underline{p}(t) + \underline{B} * \underline{p}'(t) + \underline{M} * \underline{p}''(t)$$

to forces ($f(t)$) and positions ($p(t)$) that were recorded during demonstrations [1]. The interacted environment can be characterized by stiffness (\underline{K}), damping (\underline{B}), and inertia forces (\underline{M}). The estimated stiffness reveals physical (also called natural) constraints of the environment (in this case, orthogonal contacts). Singular Value Decomposition (SVD) is used to calculate the principal directions of the stiffness matrix that reveal the natural constraints. These constraints can be used for proper selection of the aforementioned control policies.

Controller design

A Cartesian impedance controller [2] and negative joint torque feedback are used for execution of the task, as illustrated by the block diagram shown in Figure 4. The negative joint torque feedback has two important roles: reducing the apparent inertia of the motor (rotor) and reducing the disturbances on the motor dynamics (e.g. friction) [2, 3].

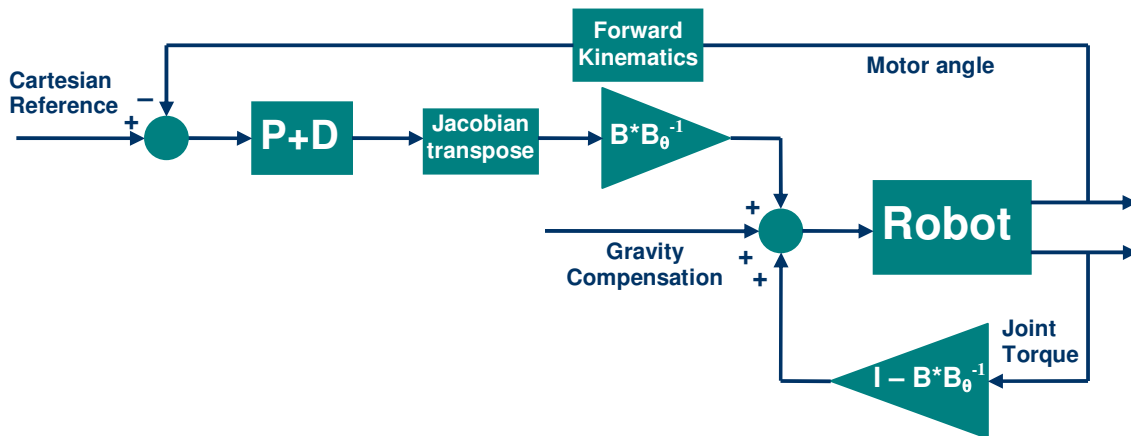


Figure 4: Cartesian Impedance Control block diagram

In this figure, the term $B * B_0^{-1}$ represents the product of motor's apparent inertia (B) and a scaling coefficient (B_0), and the term I represents the identity matrix. For transitions between unconstrained and constrained motions, hysteresis was used based on the measured force signal [4]. High stiffness is selected for the

unconstrained directions (position control) and low stiffness is selected for the constrained directions (force control). In the future, it is intended to use the joint torques for triggering the transitions.

Experimental results

In the experiment carried out on the robotic arm shown in Figure 1, the demonstrated task has consisted of three phases: 1) guiding the arm from its initial posture through free space until its tip makes a contact with a horizontally placed stiff object (which was parallel to the X-Y plane indicated in Figure 1), 2) making a wiping motion with the arm tip over the surface of this object, and, finally, 3) bringing the arm back to its initial posture. During the task demonstration, end-effector forces and positions shown in Figure 5 were recorded.

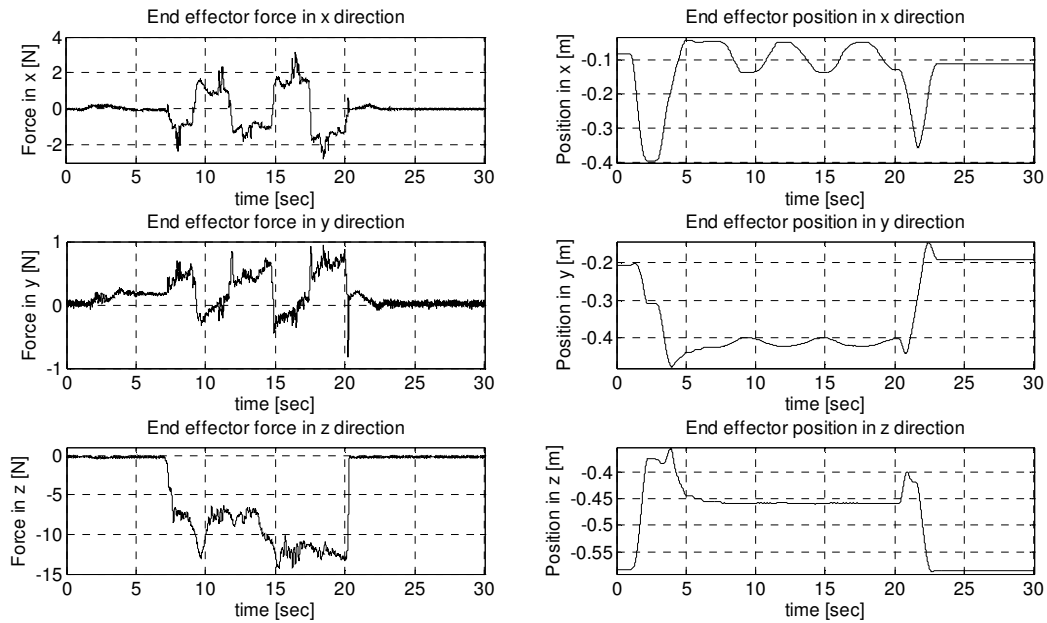


Figure 5: Position and force traces in demonstration phase

From these traces, time-varying stiffness matrix has been estimated as shown in Figure 6.

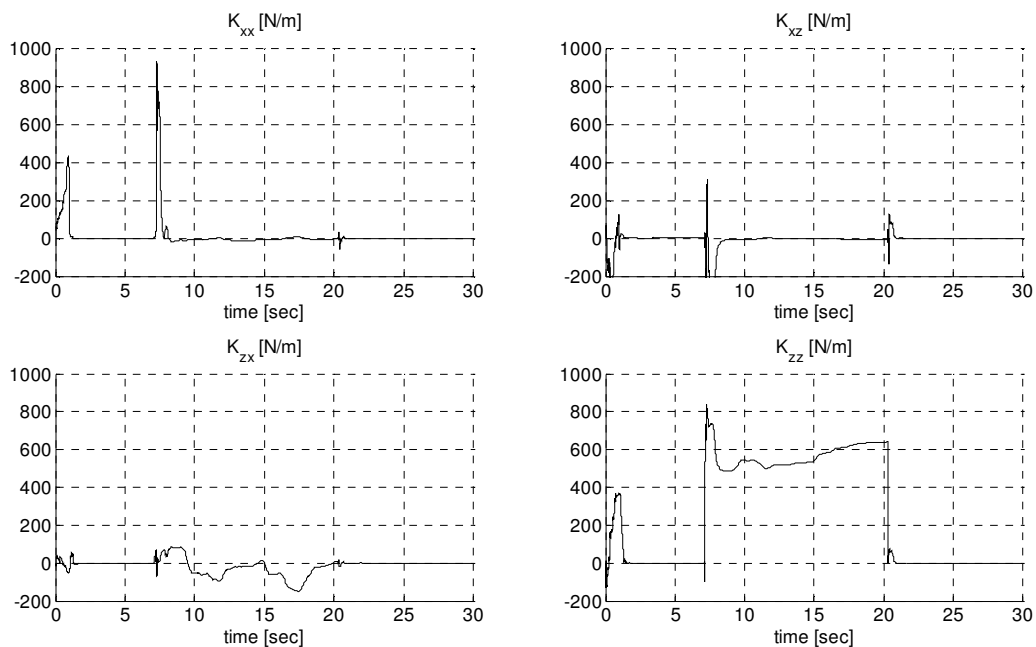


Figure 6: Calculated environment stiffness components for demonstration phase

If we exclude the transient effects, from Figure 6 we can notice that the estimated stiffness in the normal direction (K_{zz}) dominates both the stiffness along the surface (K_{xx}) and the off-diagonal stiffness terms. The end-effector forces and positions recorded during execution of the task by the robot are given in Figure 7. Here, we show only the constrained motion part of this task.

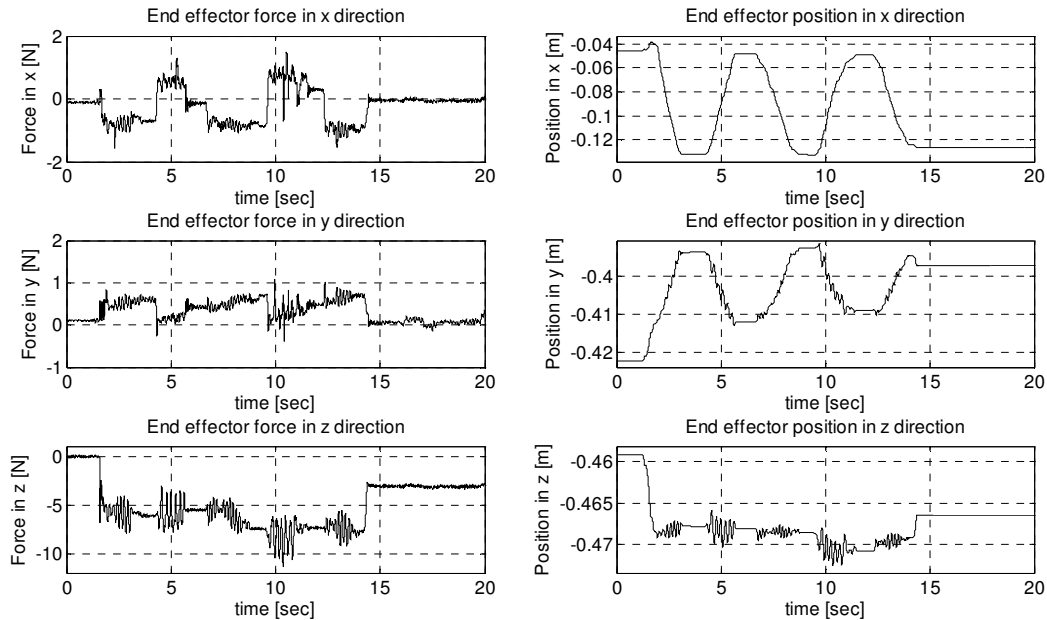


Figure 7: Position and force traces in replay phase

Conclusions

The considered robot ‘task programming method’ can identify essentials of the demonstrated task. The controller does not suffer from transition instabilities that might be encountered during transition between free and constrained motions, especially under rigid contact conditions.

Pilot experiments show that the constraint identification facilitates discriminating between segments of the demonstrated task. In turn, the segmented replay of the demonstrated sub-tasks becomes possible. The transient behavior of the estimation, however, causes some undesirable effects. An example is the initial misestimation of the constraint directions. In the future, we need to improve detection of transitions between the task segments, in order to reduce the undesirable transient effects and determine the triggering events for the transitions. Furthermore, we need to reduce influence of friction on the constraint identification. Finally, the method should be tested in situations that involve more degrees-of-freedom and different types of constraints, such as opening a door, driving a screw, etc.

References

1. R. Kikuuwe, T. Yoshikawa, “Robot Perception of Environment Impedance”, in Proc. of the 2002 IEEE Int. Conf. of Robotics and Automation, 2002, pp. 1661-1666.
2. C. Ott, A. Albu-Schaffer, A. Kugi, S. Stramigioli, G. Hirzinger, “A passivity based Cartesian Impedance Controller for Flexible joint robots – Part I: Torque feedback and gravity compensation”, submitted to ICRA 2004.
3. A. Albu-Schaffer, C. Ott, G. Hirzinger, “A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots”, The International Journal of Robotics Research, 2007.
4. R. Carloni, “Robotic Manipulation: Planning and Control for Dexterous Grasp”, Ph.D. Thesis, University of Bologna, 2007.