Bulletin of the *Transilvania* University of Braşov • Vol. 8 (57) No. 2 - 2015
Series I: Engineering Sciences

ROBOT PROGRAMMING STRATEGY BY REPRODUCTION OF VIRTUAL ROBOT MODEL' GESTURES

Aurel FRATU[1]

Abstract: *In this paper the robot programming is accomplished by combining off-line and on-line programming techniques. The method consists in using a programming platform that allows us to expediently compose robot programs. On the programming platform there is carried out the virtual prototype of the physical robotic arm to be programmed and the real working space wherein it is intended to work. The gestures for the each robotic task are computed for virtual robot prototype and are transferred online, with a central coordination, to corresponding physical robot, which must imitate her virtual "homonymous".*

Key words: *virtual robots, learning by imitation, motion programming, gestures reproduction.*

1. Introduction

Virtual prototyping is an aspect of information technology that is considered to be an effective technique to create the conceptual models that become visible by means of simulation and visualization. Virtual prototypes enable end- users to experience first hand whether the innovative concepts fulfill their needs. With this conceptual model we attempt to make possible new robots programming philosophy.

Prototypes are an important element of the design thinking process. Different prototypes can provide different representations of design and specific characteristics of each prototype have a major impact on the way designers understand and react on the inventive design ideas. Additionally, virtual prototypes can help to establish a common understanding between design thinkers and end users as well as between design thinkers themselves. Virtual prototyping typically aims at creating concrete representations of design ideas. Depending on how well end-users can experience and perceive a prototype, they are able to judge and evaluate the design idea or even the rationale behind it.

Current virtual prototyping practices are based on development software systems focused on the graphical user interfaces (GUI). However, these prototypes usually represent an individual and isolated end-user's perspective on the system to create. Therefore, such prototypes are not suitable to obtain feedback about the underlying concepts of how activities are executed or how an innovative software solution could support them.

---

[1] Dept. of Automatics and Technology of Information, *Transilvania* University of Braşov.

Thus, while the end-users can judge whether the system is built right, i.e., usable for them, they can hardly answer whether the right system is built, i.e., suitable for the task it was designed for, since only a holistic view of all the perspectives could answer this question.

2. Physical Robot Programming Based on Virtual Model

Robot Programming by Demonstration (PbD) covers methods by which a robot learns new skills through human guidance. Also referred to as learning by imitation, lead-through teaching, tutelage or apprenticeship learning, PbD takes inspiration from the way humans learn new skills by imitation to develop methods by which new tasks can be transmitted to a robot [1], [10].

Learning control strategies for numerous degrees of freedom platforms that interact in complex and variable environments is faced with two key challenges: first, the complexity of the tasks to be learned is such that pure trial and error learning would be too slow. PbD thus appears as a way to speed up learning by reducing the search space, while still allowing the robot to refine its model of the demonstration through trial and error [3].

Second, there should be a continuum between learning and control, so that control strategies can adapt in real time to perturbations, such as changes of position and orientation of objects. The present work addresses both challenges in investigating and comparing methods by which PbD is used to learn the dynamics of demonstrated movements, and, hence, provides the robot with a generic and adaptive model of control.

We present a description of the theoretical aspects of the physical robot motion programming using a virtual model. The advantages of such approach as an alternative to the classical methods (e.g. vision guided trajectory imitation [5]) are on-line adaptation to the motion of the virtual prototype. A solution to the above problem is to construct a virtual prototype model and to transfer the virtual trajectory by interacting with the physical robot model.

This approach also estimates the response of each action through a predictive motion virtual model to more accurately predict theirs consequences. Pre-computed trajectories sets come from the virtual prototype and are used to autonomously guide the robot. Designing a virtual model would be an option; however, the behavior of the robots is very difficult to model. Moreover, the use of system knowledge is contrary to our research aim. Therefore we focus on creating a virtual prototype model from experimental data obtained from the physical robot model.

Users interact with the simulation environment through the visualization. This includes, but not limited to, computer screen. Optimization of the real robots behavior is performed in the low dimensional virtual space using the virtual robot prototypes. In the virtual space one simulate even the intersecting of the virtual robot and its environment. The intersecting of two virtual objects is possible in the virtual world, where the virtual objects can be even intersected and there is no risk to be destroyed [4].

The visualization provides an interface to develop interactive implementations based on simulated behavior of the model. In our work we assume that learning of the deterministic part for description motion dynamics should be sufficient to design the corresponding robot control.

We particularly refer to the ability of the system to react to changes in the environment that are reflected by motion parameters, such as a desired target position and motion duration. Therefore, the system is able to manage with uncertainties in the position of a manipulated object, duration of motion, and structure limitation (e.g., joint velocity and torque limits) [3].

The proposed method aims at adapting to spatial and temporal perturbations which are externally-generated. This aspect will be investigated in our future works.

It is easy to recuperate kinematic information from virtual robot motion, using for example motion capture [1]. Imitating the motion with stable robot dynamics is a challenging research problem [8].

In this paper, we propose a predictive control structure for physical robots that uses capture data from their virtual prototypes and transfer them - via communication wireless - to track the motion in the real space.

We will demonstrate the tracking ability of the proposed controller with dynamics simulation that takes into account joint velocity and torque limits. We apply the controller to tracking motion capture clip to preserve the original behavior of virtual robot.

First, a motion capture system transforms Cartesian position of virtual robot structure to virtual joint angles based on kinematic model. Then, the joint angles are converted in binary words and transferred to real robot using a wireless communications.

We employ the control loops structure to establish relationships between the virtual and real robot control systems.

We present results demonstrating that the proposed approach allows a real robot to learn how to move based exclusively on virtual robot motion capture, viewed as predictive control strategy.

3. Online Imitation Method of the Virtual Dynamical Systems

In robotics, one of the most frequent methods to represent movement strategy is by means of the learning from imitation. Imitation learning is simply an application of supervised learning. One goal of imitation of the dynamical systems is to use the ability of coupling phenomena to description for complex behavior [6].

In this paper, we propose a generic modeling approach to generate virtual robot prototype behavior in experimental scenery. The actions for the each task are computed for virtual robot prototype and are transferred online, with a central coordination, to corresponding physical robot, which must imitate her virtual "homonymous". Notice the similarity between moves of the virtual robot prototype in the virtual work space and the "homonymous" moves in the real work space of the physical robot. We assume to use the virtual robot prototypes and the motion capture systems to obtain the reference motion data, which typically consist of a set of trajectories in the operational space.

Our method consists in using a programming platform on which there is carried out the virtual prototype of the physical robotic arm to be programmed and the real working space wherein it is intended to work. The method combines off-line and on-line programming techniques.

In the robot program there is written a source code intended to generate the motion paths of the virtual robotic arm prototype. The numerical values of the prototype articulation variables are sent to the data register of a port of the information system which, via a numerical interface, is on-line transferred into the data registers of the controllers of the actuator of the physical robotic arm.

Finally, there are obtained tracking structures due to which the moving paths of the virtual robotic arm joints are reproduced by the physical robotic arm joints, thereby generating motion within the real working space.

Imitation learning from our strategy demonstrates how to obtain dynamical virtual models with CAD systems. Those online adjusted virtual models are among the most important properties offered by a dynamical systems approach, and these properties cannot easily be replicated without the feed-back from physical robot of our proposed structure.

The objective of a movement is to generate a reaching movement from any start state to a goal state [8]**.** The proposed structure uses a virtual demonstrator for planning the movements of the physical robot.

The imitation strategy consists in a proportional real-time mapping between each virtual joint and the corresponding physical joint.

The system requires the programmer to perform an initial calibration routine to identify the range of motion for each movement. Each range is divided into as many intervals as the number of feasible discrete configurations of the corresponding joint.

3.1. Predicting Robot Control Latency

The imitation method supposes a delay between the action of the virtual prototype and physical robot. The time elapsed between making an action decision and perceiving the consequences of that action in the environment is called the control delay.

A predictor approximates a negative delay. It has access to the delayed robot state as well as to the un-delayed controller actions and is trained to output the un-delayed robot state. The predictor contains a forward model of the robot and provides instantaneous feedback about the consequences of action commands to the controller. If the behavior of the robot is predictable, this strategy can simplify controller design and improve control performance.

All physical feedback control loops have a certain delay, depending on the system itself, on the input and output speed and, of course, on the speed at which the system processes and transfer the information from virtual environment to physical environment.

It explains how we solved the task by predicting the movement of the robots, using a virtual prototype model. Just virtual robot positions and orientations as well as the most recent motion commands sent to the physical robot are used as input for the prediction.

In order to control the behavior of the robot in a way that is appropriate for the situation of her virtual homonym on the virtual scene we need the exact positions of them at every moment.

In order to correct this immanent error we have developed an informatics network which processes the positions and the orientations of each joint. We use recorded preprocessed data of moving virtual robot to train the network. It predicts the actual positions of the robots. These predictions are used as a basis for control. The action commands will be sent to the physical robots during the real time.

Using the concept of motor prediction one predicts the joints' position of the robot arm, rather than sensing it by sensorial system.

In this model the predictions are based on a copy of the motor commands acting on the virtual joints. In effect, the virtual joints' position of the virtual robot arm is made available before physical sensory signals become available.

These predictions are used in an inner control loop to generate sequences of actions that guide the robot towards a target state. Since this cannot account for disturbances, the robot predictions are delayed by the estimated dead time and compared to the sensed robot state. The deviations reflect disturbances that are feedback into the controller via an outer loop. The fast internal loop is functionally equivalent to an inverse-dynamic model that controls a robot without feedback.

There are many possibilities to counter the adverse effects of the control delay. The easiest way would be to reduce precision requirements, but this would lead to unnecessary collisions and uncontrolled behavior. Control researchers have made many attempts to overcome the effects of delays.

3.2. System Architecture

Action commands are sent via a wireless link from virtual robot to the physical robot that contains only minimal local intelligence. Our imitation experimental system is illustrated in Figure 1.

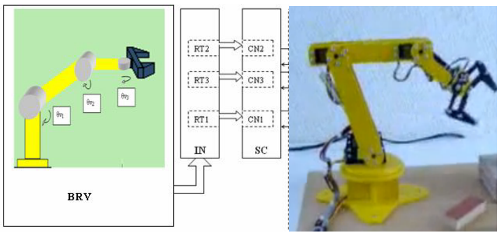

Fig. 1. *Experienced robotic manipulator*

For behavior control of the physical robot is use one Video camera. It looks at the field from above and produces an output video stream, which is forwarded to the central PC. Images are captured by a frame memory and given to the vision module.

The global computer vision module analyzes the images, finds the paths of the physical robots and produces as output the positions and orientations of the robot's joints, as well as the position of the arm. It is described in detail in [3].

Based on the gathered information from virtual robot and the physical robot, the behavior control module then produces the commands for the physical robot: desired rotational velocity, driving speed and direction, as well as the activation of the logical device. The hierarchical reactive behavior control system of the team virtual robot - physical robot is described in [11].

For each robot joint is needed a microcontroller for omni-directional motion control. It receives the commands and controls the movement of the robot using PID controllers (see [4]). Feedback about the speed of the joints is provided by the pulse generators which are integrated in each servo-motor

4. Reproduction by Imitation of Virtual Robot Model' Gestures

Such approaches are advantageous in that the system does not depend on an explicit time variable and can be modulated to produce trajectories with similar dynamics in areas of the workspace, to ensure precise reproduction of the task.

While motion execution is in progress, the real robot joints are activates into the real work space. Each actuator was connected by a sensor in the closed-loop. Each time, a skill primitive is executed by the robot control system SC; it changing the robot joints position. As no time limit for the motion is specified, the physical robot imitates the behavior of the virtual robot.

In our laboratory currently we are developing Cartesian control architecture able to interpret the physical robot commands in the above given form. The basis of our implementation is a flexible and modular system for robot programming by imitation.

In our experimental configuration in order to prove the correctness of the robot programming by imitation we have chosen an robotic manipulator equipped with electrical actuators, mounted on the physical robot's joints.

The robot's control unit is connected via TCP/IP to a PC equipped with the interface card; the PC is running the simulation and control process. The robot control system receives and executes each 16 ms, an elementary move operation.

Due to the kinematics limitations of physical robot, the resolution of the joints is quite limited. After the calibration phase, physical robot starts imitating the gestures of the virtual robot demonstrator. The user can decide to activate all the three degrees of freedom concurrently or with a restricted subset by deactivating some of the sensors attached to each joint of the physical robot.

Our system requires an essential step in that one converts the position errors into motor commands by means of the PD controller [7].

Figure 2 shows the structure of the system used for joint motion imitation. One can see the interest hardware components their interconnections for closed-loop control, associated of robot joint number one. The sensor TP1 is in close proximity with the actuator ACT1 and is interconnected with hardware components specific for a closed-loop.

We aim at developing controllers for learning by imitation with a single joint robot demonstrator. For this purpose, we assume a simple control system where the position of the 1 DOF discrete dynamical system, drives the time evolution of joint variable θ_{r1}, which can be interpreted as the position controlled by a proportional-derivative controller. All other DOFs of the arm demonstrate the same behavior.

Tracking of the desired velocity $\dot{\theta}_d$ and desired position θ_d is then ensured by the proportional-derivative controller. The acceleration command is determined by:

$$\ddot{\theta} = k_v(\dot{\theta}_d - \dot{\theta}) + k_p(\theta_d - \theta) = \ddot{\theta}^v + \ddot{\theta}^p, \quad (1)$$

where k_v and k_p are gain parameters similar to damping and stiffness factors. In the above equation, $\ddot{\theta}^v$ allows the robot to follow the demonstrated velocity profile. $\ddot{\theta}^p$ prevents the robot from departing from a known situation, and forces it to come back to the subspace of demonstrations, if a perturbation occurs.

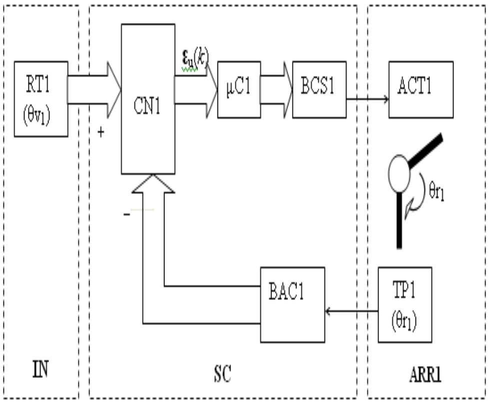

Fig. 2. *Closed-loop motion control for robot joint*

The non-linear dynamics of the movement is thus approximated by a mixture of linear systems, where the influence of the different linear models is estimated through a non-linear process. Eq. (1) can be formulated as a mixture of linear systems, see [4] for details. Note that the tracking term in Eq. (1) may distort the original estimate of the dynamics (oscillations around the original demonstrations).

Avoiding such oscillations and minimizing the distortions depends on choosing carefully the gains parameters. In practice, for the experiments reported here (and for well chosen gains), the behavior of the system followed the desired dynamics. Analysis and solutions to the problem of stabilizing the first order system can be found in [5]. One made reproduction path with the stabilizer, where the robot smoothly comes back to the demonstrated movement when starting from a different initial situation.

Using constant gains in Eq. (1) may distort the demonstrated dynamics of the movement in-between two consecutive virtual postures. While this solution may be acceptable for some tasks, we suggest here the use of adaptive gains.

By setting a proportional gain that decreases when the system is close to the virtual demonstrated trajectories, the system reproduces not only the demonstrated virtual path, but also follows the dynamics of the movement while following this virtual path, see [7] for details. Parts of the movement where the variations across the demonstrations are high indicate that the position does not need to be tracked very precisely.

Setting adaptive gains as in [7] allows the controller to focus on the other constraints of the task, such as following a desired velocity. On the other hand, parts of the movement exhibiting strong invariance across the demonstrations will be tracked more precisely, i.e. the gain controlling the error on position will automatically be increased.

Torque is indirectly controlled via joint dynamics. The internal joint states were unknown and not available to the controller. Only joint variable was available by the joint sensor. The joint kinematics is derived from the CAD model. In the motion imitation additional difficulties arise such as the joints angular velocity and torques limits.

5. Conclusions

This paper is focused on the programming by imitation, transferring of the motion mapping from virtual space in 3-D dimensional real physical space.

We presented and evaluated an approach based on virtual model, and dynamical systems to allow robots to acquire new skills by imitation.

The use of virtual robot model allowed us to get the explicit time dependency that was considered in our previous work [10], by encapsulating precedence information within the virtual representation. In the context of separated learning and reproduction processes, this original formulation was systematically evaluated with respect to our previous approach, *Imitation-Based Motion Programming for Robotic Manipulators* [6], and *Predictive strategy for robot behavioral control* [7]. We finally analyzed applications on different kinds of robots to highlight the flexibility of the proposed approach in different learning by imitation scenarios.

Acknowledgements

The author wish to thank, for cooperation and engagement in research activity, the entire team of Services and Products for Intelligent Environment Laboratory, within the Research & Development Institute ICDT-PRO-DD of the *Transilvania* University of Brașov.

They granted us the opportunities to work in team with them and provided us important experiences to extend this interesting research.

References

1. Alissandrakis, A., Nehaniv, C., et al.: *Correspondence mapping induced state and action metrics for robotic imitation*. In: IEEE Transactions on Systems, Man and Cybernetics, Part B **37** (2007) No. 2, p. 299-307.
2. Argall, B., Chernova, S., et al.: *A survey of robot learning from demonstration.* In: Robot Automation Systems **57** (2009) No. 5, p. 469-483.
3. Argall, B., Browning, B., et al.: *Learning robot motion control with demonstration and advice-operators*. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2008, p. 399-404.
4. Calinon, S., Billard, A.: *Statistical learning by imitation of competing constraints in joint space and task space.* In: Advanced Robotics **23** (2009) No. 15, p. 2059-2076.

5. Ekvall, S., Kragic, D.: *Learning task models from multiple human demonstrations.* In: Proceedings IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Hatfield, UK, September 2006, p. 358-363.
6. Fratu, A., Fratu, M.: *Imitation-Based Motion Programming for Robotic Manipulators*. In: Proceedings of 14th International Conference on Optimization of Electrical and Electronic Equipment OPTIM 2014, May 22-24, 2014, Braşov, Romania, p. 770-775.
7. Fratu, A., Riera, B., et al.: *Predictive strategy for robot behavioral control.* In: Proceeding of the International Conference on Manufacturing Systems - ICMaS 2014, Bucharest, Romania, **9** (2014) Issue 3, p. 125-130.
8. Hersch, M., Guenter, F., et al.: *Dynamical system modulation for robot learning via kinesthetic demonstrations.* In: IEEE Transactions on Robotics **24** (2008) No. 6, p. 1463-1467.
9. Muehlig, M., Gienger, M., et al.: *Task level imitation learning using variance-based movement optimization.* In: Proceedings IEEE International Conference on Robotics and Automation (ICRA), 2009, p. 1635-1642.
10. Pardowitz, M., Zoellner, H., et al.: *Incremental learning of tasks from user demonstrations, past experiences and vocal comments.* In: IEEE Transactions on Systems, Man and Cybernetics, Part B **37** (2007) No. 2, p. 322-332.
11. http://worldwide.espacenet.com/publicationDetails/biblio?DB=worldwide.espacenet.com&II=1&ND=3&adjacent=true&locale=en_EP&FT=D&date=20131230&CC=RO&NR=129121A0&KC=A0.