Proceedings in Manufacturing Systems, Volume 9, Issue 3, 2014, 125–130 ISSN 2067-9238

PREDICTIVE STRATEGY FOR ROBOT BEHAVIORAL CONTROL

Aurel FRATU[1,*], Bernard RIERA[2], Valeriu VRABIE[3]

[1)] Prof., PhD, Dept. of Automatics and Technology of Information, "Transilvania" University of Brasov, Brasov, Romania
[2)] Prof., PhD, CReSTIC Center, University of Reims Champagne-Ardenne, Reims, France
[3)] Assoc. Prof., PhD, CReSTIC Center, University of Reims Champagne-Ardenne, Reims, France

Abstract: *In this paper, based on original idea, the authors propose a new strategy for physical robot behavioral control using predictive control strategy. To program the desired motion sequence for the physical robot, one captures the motion reference paths from the virtual robot model and maps these to the joint settings of the physical robot. Physical robot reproduces the behavior of the virtual prototype. This requires transfer of a dynamical signature of a movement of the virtual robot to the physical robot, i.e. the robots should be able to imitate a particular path as one with a specific velocity and/or an acceleration profile. Furthermore, the virtual robot must cover all possible contexts in which the physical robot will need to generate similar motions in unseen context. The physical robot acts fully independently, communicating with corresponding virtual prototype and imitating its behavior.*

Key words: *virtual robots, path learning by imitation, motion programming, predictive control, behavioral control*

1. INTRODUCTION

Robotic and Automation systems are the core of today achievements towards future developments of Science and Technology. The challenges are still faced in many aspects for a full success of those systems in industrial and non-industrial applications for which fundamental are design and manufacturing issues. Innovation and advances in the fields of Robotics and Automation are nowadays continuously proposed with characters of novelty and update of past solutions.

The concept of robot motion prediction was introduced to clearly understand what the robot must to do when trying to localize visual objects. His suggestion was that the physical robot can predict the situation (position and orientation) using its virtual prototype rather than physical sensory signals. For a system with a demand of reacting as precisely as possible, its past information is not suitable for control planning any more.

We should predict the future behavior, at the time when the control command arrives at the physical robot and is executed.

The ability of predicting of the behavior of robots is important in design; the designers want to know whether the robot will be able to perform a typical task in a given time frame into a space with constraints.

The control engineer cannot risk a valuable piece of equipment by exposing it to untested control strategies. Therefore, a facile strategy for contact detection and collision avoidance, capable of predicting the behavior of robotic manipulators, becomes imperative.

When the robots need to interact with their surrounding, it is important that the computer can simulate the interactions of the participants, with the passive or active changing environment in the graphics field, using virtual prototyping.

For effective involved path navigation, a technique is needed which can exploit the reference trajectory structure to search in the local continuum for actions which minimize path deviation and avoid obstacles. A pursuit planner is adequate in order to permit relaxation of a trajectory (for optimization reasons) by searching a small number of degrees of freedom.

The accepted class of "pursuit" algorithms will be used to generate a path planner in order to accomplish robotic tasks,

In this paper the authors develop a formally analyzis of the concept of robot motion prediction and propose a "pursuit" algorithm to programming the robots' motion.

The actions for the each robotic task are computed for virtual robot and are transferred, with a central coordination to corresponding physical robot which must imitate its virtual homonym.

The objective of the present paper is giving an overview of a *predictive control strategy* for robots, based on the virtual prototypes.

The remainder of this paper is organized as follows. In Section 2 an overview of physical robot motion control, using a simulated virtual model, is presented.

In Section 3 the online behavioral imitation for predictive control is formulated using a new method.

Theoretical considerations concerning the predictive control model are given in Section 4. In Section 5 the impedance model to force control of the physical robot is presented, as basic strategy for predictive motion control in the real environment. Section 6 concludes the ideas which are announced in the previous sections.

[*] Corresponding author: Str. Mihai Viteazu nr. 5, Corp V, et III
Cod. 500174, Jud. Braşov, Romania,
Tel.: +40 268 418 836,
E-mail addresses: *fratu@unitbv.ro* (A. Fratu)

2. PHYSICAL ROBOT MOTION CONTROL BASED ON SIMULATED VIRTUAL MODEL

We present a description of the theoretical aspects of the physical robot motion control using a virtual motion prediction model. The advantages of such approach as an alternative to the classical methods (e.g. vision guided trajectory imitation [15]) 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.

Our approach represents a technique for generating animated navigation offline, by pre-computing layered trajectories for a physical robot. Pre-computed trajectories sets 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 imitate them 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. 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 BEHAVIORAL IMITATION FOR PREDICTIVE CONTROL

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 [12–14]. One goal of imitation of the dynamical systems is to use the ability of coupling phenomena to description for complex behavior [10]. Anything a robot does is called a behavior. Moving, turning, stopping, picking things up, putting them down, delivering a message are all behaviors that a robot can perform.

In this paper, we propose a generic modeling approach to generate robot prototype behavioral model, in virtual environment 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 its 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 that are used to generate feasible reference trajectories for physical robot and which typically consist of a set of trajectories in the Cartesian space.

The paper relates to a method for robot programming by combining off-line and on-line programming techniques. The 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.

In the robot program there is written a source code intended to generate the motion paths of the virtual robotic arm prototype [16]. The numerical values of the prototype joints variables are sent to the data register of a port of the information system which, via a numerical interface, are on-line transferred into the data registers of the controllers of the servo system 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.

4. PREDICTIVE CONTROL MODEL

A virtual trajectory over a virtual environment is able to predict the evolution of the physical robot in the phisical environment under any selected constraints. Once an virtual trajectory has been computed; this is used as predictive model.

For the optimization behavior of virtual models the simulation is repeated starting with the different initial condition and the trajectory optimization techniques are used to generate corrective actions. The initial corrective action is evaluated through the predictive virtual motion model.

By recomputing the trajectory for the current state, the predictive controller generates a feedback action that can effectively compensate the uncertainties and disturbances and reacts such as an obstacle for example, appearing along the maneuver path, may be avoid [2, 9].

In our experiments various synthetic behavior models, close to real robot behavior, have been created. However, the use of realistic models for interaction between the virtual world and physical world requires significant computation time and large amounts of data transfer, and most currently existing virtual prototypes have a limited number of predefined expressions.

From the above there are three main questions for supporting interaction between the virtual world and physical world:

1. How to create believable environments
2. How to create believable virtual robot prototypes
3. How to set in motion the physical robot of appropriate behaviors

In this paper we propose a structure that could contribute to answering these three questions and enhance cooperative interaction within virtual worlds; the Virtual world Maker to create realistic scene-referenced 3D environments, a 3D animation system to display tasks' evolution, and the Analyzer to automatically initiate the motion of physical robot.

The question about how to create believable environments is answered by the automatic creation of identifiable scenes.

The question about the creation of realistic virtual robot prototypes is answered by the development of an automated 3D modeling system for to display of the virtual prototype behavior.

The question of how to set in motion the physical robot of appropriate behaviors is answered by a system for automating transfer the virtual robot' poses to physical robot servo system.

4.1. Platform structure

Initially, a set of virtual postures is created for the virtual robot and the pictures' positions are recorded for each posture, during motion. These recorded pictures' positions provide a set of Cartesian points in the 3D capture volume for each posture.

To obtain the physical robot postures, the virtual pictures' positions are assigned as positional constraints on the physical robot. To obtain the physical joint angles one use standard inverse kinematics (IK) routines. The IK routine then directly generates the physical joint angles on the physical robot for each posture.

We start with a 3 degree-of-freedom (DOF) discrete movement system that models point-to-point attained in a 3D Cartesian space.

Figure 1 shows our experiment involving the imitation learning for a physical robotic arm with 3 degrees-of-freedom (DOFs) for performing the manipulate tasks.

We demonstrated the imitation of elbow, shoulder and wrist movements. Importantly, these tasks required the coordination of 3 DOFs, which was easily accomplished in our approach.

The imitated movement was represented in joint angles of the robot. Indeed, only kinematic variables are observable in imitation learning.

The physical robot was equipped with a controller (a PD controller) that could accurately follow the kinematic strategy (i.e., generate the torques necessary to pursue a particular joint angle trajectory, given in terms of desired positions, velocities, and accelerations) [11].

Figure 1 also displays (left image) the user interface of a virtual robotic manipulator arm, which has been created which a dynamical simulator.

Referring the Figure 1 we comment the following: on programming platform, a robot program is carried out off-line, and one sends into the data registers of a port of the hardware structure, the numerical values of the joint variables of the virtual prototype of the robotic arm (BRV) and displays on a graphical user interface, the evolution of the virtual prototype during the carrying out of the robotic task.

Via numerical interface (IN) the virtual joint dataset, from the data registers of the port of the hardware structure of the programming platform are transferred into the data registers of the numerical comparators of the controllers. These datasets are reference inputs of the pursue loops, resulting a system control (SC).

The reference datasets are obtained using a motion capture channel taking into account the joints motion range.

The easiest way to generate the spatial relations explicitly is the interactively programming of the behavior of the virtual prototype in its virtual environment, in order to specify suitable positions θ_{v1}, θ_{v2}, θ_{v3}.

This kind of specification provides an easy to use interactive graphical tool to define any kind of robot path; the user has to deal only with a limited and manageable amount of spatial information in a very comfortable manner.

The applicable robot tasks are designed and the desired pathways are programmed off-line and stored in the buffer modules RT1, RT2, RT3.

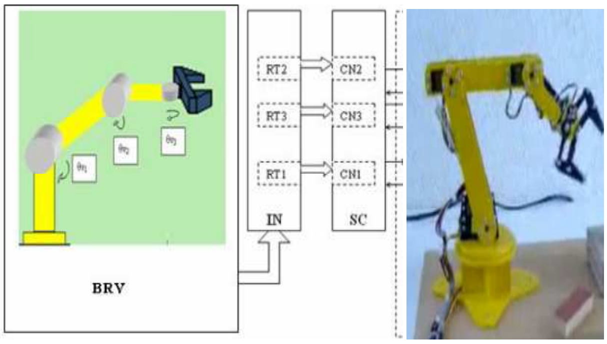

Fig. 1. Imitation software platform structure.

The comparative modules CN1, CN2, CN3 furnish, to the pursuit controllers, the datasets involving the expected state of the virtual robot prototype and the measured state 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.

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 Cartesian space [17].

The data is obtained using a motion capture channel taking into account the joint motion range. Due to the joint limits and the difference between the kinematics of the virtual robot and real physical robot, the joint angle data are pre-processed.

In our pre-processing, we assume that both virtual and physical robots are on the scene at the same time and estimate the correct arms position and orientation. We then compute the inverse kinematics for new posture to obtain the cleaned joint angles and retain the difference from original joint angles.

At each frame during control, we add the difference to the original data to obtain the cleaned reference joint angles. This correction is extremely simple and our controller does not require supplementary cleanup.

4.2. Basic servo system

We aim at developing controllers for learning by imitation with a virtual robot demonstrator. For this purpose, we assume a simple control system where the position and velocity of the 1 DOF discrete dynamical system, drives the time evolution of joint variable, which can be interpreted as the position controlled by a proportional-derivative controller.

In order to command a robotic manipulator, first of all, a servo system is considered and designed. Here, the resolved acceleration controller [5] is picked up in a servo system.

The computed torque control method is used for nonlinear control of robotic manipulator, which is composed of a model base portion and a servo portion. The servo portion is a close loop with respect to the position and velocity.

We aim to obtain the structure of the system used for behavior imitation and the interest hardware components in close proximity with the sensors and their interconnections in closed-loop. Closed-loop servo system are created out of one relatively simple set of equations; based only on the capture of the trajectory of a virtual robot prototype.

Computed torque control. The dynamic model of a robotic manipulator without friction term is generally given by

$$M(q)\ddot{q} + H(q,\dot{q}) + G(q) = \tau , \qquad (1)$$

where, $M(q)$ {6×6} is the inertia term in joint space. $H(q,\dot{q})$ {6×1} and $G(q)$ {6×1} are the Coriolis/ centrifugal term and gravity term in joint space, respectively. q {6×1}, $\dot{q}$ {6×1} and $\ddot{q}$ {6×1} are the position, velocity and acceleration {6×1} dimensional vectors in joint coordinate system, respectively. The vector τ_i {6×1} is the joint driving torque {6×1} dimensional vector. In the case that the resolved acceleration control law is employed in the servo system of a manipulator, desired position, velocity and acceleration vectors in Cartesian coordinate system are respectively given to the references of the servo system.

Desired Trajectory

For instance, in order to apply the computed torque control method to the manipulator, the desired trajectory composed of x_r, $\dot{x}_r$ and $\ddot{x}_r$ in Cartesian coordinate virtual system must be prepared. First of all, the desired trajectory in Cartesian coordinate system is designed, in which the manipulator moves from the initial pose to the final pose.

The desired trajectory x_r, $\dot{x}_r$ and $\ddot{x}_r$ are calculated from virtual joint angle θ_v, virtual joint velocity $\dot{\theta}_v$ and virtual joint acceleration $\ddot{\theta}_v$ respectively, using the robot analytical models.

The trajectory in virtual joint space makes the physical robot follows of the homonym virtual prototype as shown in Fig. 1.

The joint driving torque is calculated from

$$\begin{aligned}\tau = {} & \hat{M}(q)J^{-1}(q)\times[\ddot{x}_r + k_v(\dot{x}_r - \dot{x}) + k_p(x_r - x)] - \\ & \dot{J}(q)\dot{q} + \hat{H}(q,\dot{q}) + \hat{G}(q),\end{aligned} \qquad (2)$$

where, the symbol ˆ denotes the modeled term.

The x {6×1}, $\dot{x}$ {6×1} and $\ddot{x}$ {6×1} are the position / orientation, velocity and acceleration {6×1} dimensional vectors in Cartesian coordinate system, respectively. The x_r {6×1}, $\dot{x}_r$ {6×1} and $\ddot{x}_r$ {6×1} are the desired position / orientation, velocity and $\dot{x}_r$ acceleration {6×1} dimensional vectors, respectively. The diagonal matrix $\boldsymbol{K}_v$ = diag (k_{v1}, . . . , k_{v6}) and $\boldsymbol{K}_p$ = diag (k_{p1}, …, k_{p6}) are the feedback gains of velocity and position, respectively.

The matrix $\boldsymbol{J}(q)$ is the Jacobian matrix which gives the relation $\dot{x} = J\dot{q}$. Note that q , $\dot{q}$, x and $\dot{x}$ in (2) are actual values, i.e., controlled variables.

Figure 2 shows the block diagram of the computed torque control method, in which F_{kine}(„) is the function to obtain the forward kinematics.

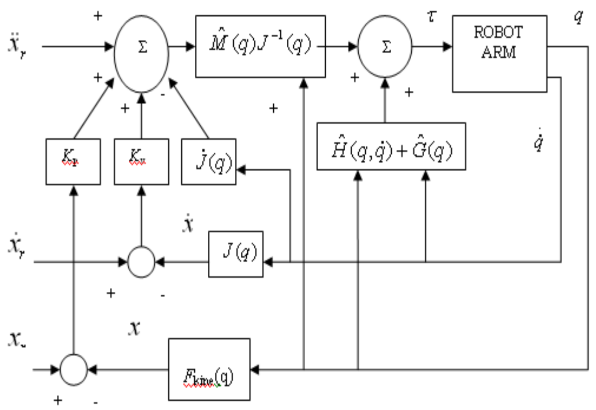

Fig. 2. Block diagram of the resolved acceleration control method, where x_r , $\dot{x}_r$ and $\ddot{x}_r$ are the desired position, velocity and acceleration vectors in Cartesian coordinate system.

The nonlinear compensation terms $\hat{H}(q.\dot{q})$ and $\hat{G}\ (q)$ are calculated to cancel the nonlinearity and are effective to achieve a stable trajectory control.

The tracking controller is responsible for making every joint track the desired trajectory. It solves an optimization problem that respects both joint tracking and desired inputs to the simplified model and obtains the joint torques to be commanded to the real robot.

To obtain satisfactory and safe control performance without falling a singularity, $\boldsymbol{K}_{\mathrm{v}}$ and $\boldsymbol{K}_{\mathrm{p}}$ are approximately tuned in advance with trial and error, considering the combination around critically damped condition. We call this the initial manual tuning process.

Two search ranges for $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ are obtained after the manual tuning process, so that $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ must be further tuned finely within the searched spaces to achieve a desirable motion without large overshoots and oscillations.

In the next section, we propose impedance following force control model which can be applied after the manual tuning process.

5. IMPEDANCE MODEL TO FORCE CONTROL OF THE PHISICAL ROBOT

When a computer is used to control a robotic manipulator, the control law is generally represented by a discrete-time control system. In this section, it is described on how to evaluate the velocity-based discrete-time control system which is implemented as basic strategy for predictive motion control.

Let's consider the impedance model following force control as an example of velocity-based discrete-time control systems. In order to conduct a simulation with a robotic dynamic model, manipulated variables written by velocity commands in discrete-time domain must be transformed into joint driving torques.

Impedance control is one of the effective control strategies for a robotic manipulator to desirably reduce or absorb the external force from an environment [6, 7]. It is characterized by ability which controls the mechanical impedance such as mass, damping and stiffness acting at joints. Impedance control does not have a force control mode or a position control mode but it is a combination of force and velocity.

In order to control the contact force acting between the arm tip and environment, we have proposed the impedance model following force control methods that can be easily implemented in robotic manipulators with an open architecture controller. The desired impedance equation in Cartesian space for a robotic manipulator is designed by:

$$\begin{aligned} &M_d(\ddot{x}-\ddot{x}_d)+B_d(\dot{x}-\dot{x}_d)+SK_d(x-x_d)= \\ &SF+(I-S)K_f(F-F_d), \end{aligned} \tag{3}$$

where:

- $x\{6\times1\}$, $\dot{x}\{6\times1\}$ and $\ddot{x}\{6\times1\}$ are the position, velocity and acceleration vectors, respectively;
- $\boldsymbol{M}_d, \boldsymbol{B}_d$ $\{3\times3\}$ and $\boldsymbol{K}_d$ $\{3\times3\}$ are the coefficient matrices of desired mass, damping and stiffness, respectively;
- $\boldsymbol{F}$ is the force vector;
- $\boldsymbol{K}_f$ $\{3\times3\}$ is the force feedback gain matrix;
- $\{6\times1\}$, $\dot{x}_d\{6\times1\}$, $\ddot{x}_d\{6\times1\}$ and $\boldsymbol{F}_d$ are the desired position, velocity, acceleration and force vectors, respectively.

The $\boldsymbol{S}$ is the switch matrix to select force control mode or compliance control mode. If $\boldsymbol{S} = \boldsymbol{0}$, (3) becomes force control mode in all directions; whereas if $\boldsymbol{S} = \boldsymbol{I}$ it becomes compliance control mode in all directions. Here, matrix $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{M}_d$,$\boldsymbol{B}_d$, $\boldsymbol{K}_d$ and $\boldsymbol{K}_f$ are set to positive-definite diagonal matrices.

When force control mode is selected in all directions, i.e., $\boldsymbol{S} = 0$, defining $X = (\dot{x} - \dot{x}_d)$ gives:

$$\dot{X} = -M_d^{-1}B_dX + M_d^{-1}K_f(F-F_d), \tag{4}$$

In general, (4) can be resolved as:

$$\begin{aligned} X &= e^{-M_d^{-1}B_dt}X(0)+ \\ &\int_0^t e^{-M_d^{-1}B_d(t-\tau)}M_d^{-1}K_f(F-F_d)d\tau, \end{aligned} \tag{5}$$

In the following, we consider the form in the discrete time k using a sampling time Δt. If it is assumed that $\boldsymbol{M}_{\mathrm{d}}$, $\boldsymbol{B}_{\mathrm{d}}$, $\boldsymbol{K}_{\mathrm{f}}$, $\boldsymbol{F}$ and $\boldsymbol{F}_{\mathrm{d}}$ are constant at $\Delta t(k-1) \le t < \Delta t_{\mathrm{k}}$, then defining $X(k) = X(t)|t = \Delta tk$ leads to the recursive equation.

Remembering $X = (\dot{x}-\dot{x}_d)$ giving $\dot{x}_d = 0$ in the direction of force control, and adding an integral action, the equation of velocity command in terms of Cartesian space is derived by:

$$\begin{aligned} \dot{x}(k) &= e^{-M_d^{-1}B_d\Delta t}\dot{x}(k-1)- \\ &(e^{-M_d^{-1}B_d\Delta t}-I)M_d^{-1}K_f\{F(k)-F_d\}+ \\ &K_i\sum_{n=1}^{k}\{F(n)-F_d\}, \end{aligned} \tag{6}$$

where $\boldsymbol{K}_i$ $\{3\times3\}$ is the integral gain matrix and is also set to a positive-definite diagonal matrix. The impedance model following force control method is used to control the force which an robotic manipulator gives an environment. As can be seen, the force is regulated by a feedback control loop.

6. CONCLUSIONS

This paper explores the robot motion control based on predictive control model. Prediction in robot behavior control has become an increasingly important subject as the robot work space becomes more crowded and robot tasks objectives require more precision and robustness.

The commands sent to our physical robot are logged for each cycle. So if the robots always executed just as what the command tell them to do, the consequences could be predicted easily. When the physical robot executes commands, its actual position and orientation should be the consequences of the information from a virtual path, transferred from virtual homonym prototype. Our method is just based on this idea.

The problem of the physical robot behavioral control is better analyzed on the virtual prototype model in the virtual environment where one may predict their behavior.

Learning approach such as learning by imitation is more flexible and can adapt to environmental change. This method is typically directly applicable due the possibility to transfer the virtual joint trajectories from virtual space to the real space of the physical robots.

As we have known the relationship between the expectation and the actual consequence, we can modify the actual command sent to the robots, making the robots behave just as what we expect.

This new innovative method for behavioral predictive control is attractive for implementation. Not similar to most other methods, our method not only makes a good prediction, but also improves the precision of motion control.

ACKNOWLEDGMENTS: The author wishes 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 Transylvania University of Brasov. We hereby acknowledge the structural funds project PRO-DD (POS-CCE, 0.2.2.1., ID 123, SMIS 2637, ctr. No 11/2009) for providing the infrastructure used in this work.

REFERENCES

[1] V. Zordan and J. Hodgins, *Motion Capture-Driven Simulations that Hit and React*, In Proceedings of ACM SIGGRAPH Symposium on Computer Animation, San Antonio, TX, July 2002, pp. 89–96.

[2] D. Silver, *Cooperative path finding*, In Proceedings of the 1st Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE'05), pp. 23–28, 2005.

[3] J. Pettre, J. Ondrej, A.-H. Olivier, A. Cretual and S. Donikian, *Experiment based modeling, simulation and validation of interactions between virtual walkers*, In Proceedings of Symposium on Computer Animation, ACM, 2009.

[4] F. Nagata, Y. Kusumoto, Y. Fujimoto and K. Watanabe, *Robotic sanding system for new designed furniture with free formed surface,* In Robotics and Computer-Integrated Manufacturing Journal, vol. 23, no. 4, pp. 371–379, 2007.

[5] F. Nagata, K. Kuribayashi, K. Kiguchi and K. Watanabe, *Simulation of fine gain tuning using genetic algorithms for model based robotic servo controllers*, In Proceedings of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation, pp.196–201, 2007.

[6] K. Gold, *An information pipeline model of human-robot interaction*, In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pp. 85- 92, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-404-1. doi:http://doi.acm.org/10.1145/1514095.

[7] A. Powers, S. Kiesler, S. Fussell, and C. Torrey, *Comparing a computer agent with a humanoid robot*, In Proceedings of the ACM/IEEE international conference on Human-robot interaction (HRI '07). ACM, New York, USA, pp. 145-152, 2007.

[8] B. Price and C. Boutilier, *Accelerating reinforcement learning through implicit imitation*, Journal of Artificial Intelligence Research, vol. 19, 2003, pp. 569–629.

[9] A. Fratu, *Collision Prevention Method and Platform for a Dynamic Group of Cooperative Robots Who Communicate Wirelessly*, In RECENT Revue, ISSN 1582 – 0246, Vol. 12 (2011), No. 2 (32), pp. 131−134.

[10] D.Grimes, R. Chalodhorn and R. Rao, *Dynamic imitation in a humanoid robot through non-parametric probabilistic inference*, In Proceedings Robotics: Science and Systems (RSS), Philadelphia, Pennsylvania, USA, August 2006.

[11] F. Nagata, I. Okabayashi, M. Matsuno, et al., *Fine gain tuning for model-based robotic servo controllers using genetic algorithms*, In Proceedings of the 13th Int. Conf. on Advanced Robotics, 2007, pp. 987–992.

[12] M. Nicolescu, M. Mataric, *Natural methods for robot task learning- Instructive demonstrations, generalization and practice,* In Proc. Intl Joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS), Melbourne, Australia, 2003, pp. 241-248.

[13] S. Calinon, F. Guenter and A. Billard, *On learning, representing and generalizing a task in a humanoid robot,* In IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol. 37, No. 2, 2007, pp. 286–298.

[14] M. Pardowitz, R. Zoellner, S. Knoop, and R. Dillmann, *Incremental learning of tasks from user demonstrations, past experiences and vocal comments*, In IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol. 37, No. 2, 2007, pp. 322−332.

[15] J. Ijspeert, J. Nakanishi and S. Schaal, *Movement imitation with nonlinear dynamical systems in humanoid robots*, In Proceedings of the IEEE International Conference on Robotics and Automation, 2002, pp. 1398–1403.

[16] J. Kober, J. Peters, *Learning motor primitives for robotics*, In Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2112–2118, 2009, Piscataway, NJ: IEEE.

[17] A. Fratu, *Method and installation for joints trajectory planning of a physical robot arm*, Proposal patent: A 00482 / 28. 06. 2013, available at: http://worldwide .espacenet.com/?locale=en_E