Proceedings in MANUFACTURING SYSTEMS
Proceedings in Manufacturing Systems, Volume 11, Issue 3, 2016, 131–136 ISSN 2067-9238

IMITATION LEARNING BY FOLLOWING THE VIRTUAL PATHS FOR PHYSICAL ROBOTS

Aurel FRATU[1,*] Mariana FRATU[2]

[1)] Prof., PhD, Dept. of Automatics and Technology of Information, "Transilvania" University of Brasov, Brasov, Romania
[2)] Assoc., Prof. PhD, Dept. of Installations, "Transilvania" University of Brasov, Brasov, Romania

Abstract: *This paper addresses the problem of imitation learning for physical robots. Getting inspiration from the way humans learn new skills by imitation, we adopt virtual robot prototype as behavior identification guide for physical robot. Virtual prototyping technique provides a virtual robot homonym. Virtual robot prototype' experience will greatly improve thanks to an always-changing scenario. Combined with a dataset transferring algorithm, we obtain an imitation learning strategy with good performance under the physical constraints. The online dataset transferring, of the mini-segments of virtual trajectories, make opportune the guidance of the physical robot. The virtual trajectories must be able to make early predictions of the physical robot's intentions. Developing a new training policy is the scope of this study. We believe that this new approach outperforms previous approaches on challenging imitation learning problems.*

Key words: *virtual prototyping, virtual trajectory, segmentation of motion data, learning by imitation, guidance by imitation, training policy*.

1. INTRODUCTION

Imitation learning techniques have proven very useful in practice in a variety of applications including robotics. Imitation learning has been shown to be successful in solving many challenging real-world problems as robot guidance by imitation.

Imitation learning covers methods by which a robot learns new skills through training. Learning by imitation supposes control strategies for each degrees of freedom of robot that interact with a complex and variable environments, according a demonstrator model.

The present work addresses both challenges in investigating and comparing methods by which imitation technique is used to learn the dynamics of demonstrated movements, and, hence, provides the robots with a generic and adaptive model of control.

The demonstrations of good behaviour of the virtual robot prototype are perceived as prediction problems. Imitation learning such as sequential prediction problems, takes inspiration from the way humans learn new skills by imitation to develop methods by which new tasks can be transmitted to a imitator, where future actions depend on previous predictions actions.

The proposed solution will make robot training more challenging and entertaining for engineers by providing two robot agents–the physical robot agent and virtual robot prototype agent – against each other – in a certain scenarios.

We present a description of the theoretical aspects to learn the controller of the physical robot using a virtual prototype robot model.

In our system imitation learning is made in two steps: first step by training a virtual robot prototype obtained as original training policy, second step by dataset transferring from virtual environment to physical environment where physical robot must reproduce such demonstrations and can follow an optimal path. The control system can adapt the strategy in real time to avoid obstacles in the work space.

Our approach, based on on-line imitation of the motion of the virtual prototype, is an alternative to the classical methods (e.g. vision guided trajectory imitation) where standard control methods fail.

The rest of this paper is organized as follows: first we discuss issues related work in literature is presented in Sec. 2; then, in Sec. 3, related the iteratively training policy. The proposed solution is introduced and discussed in Sec. 4 and Sec. 5 while Sec. 6 provides details about to prediction and optimization of dataset transferring strategy. Finally, Sec. 7 concludes the paper.

2. RELATED WORKS

In the previous works the imitation process was based on human motion primitives skilled through a motion recognition and control approach, using some controller and predictor modules. We now briefly review previous approaches and their guarantees.

Kadone and Nakamura [1] introduced an incremental algorithm to learn human motion primitives. Their model was able to automatically segment path memorize, and recognize demonstrated motions, using associative neural networks.

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

Another study based on plan recognition by Pardowitz et al. [2] states that human behavior follows stereotypical patterns, which can be expressed as preconditions and effects. However, these constraints must be specified in advance, which is a problem when trying to use them in different domains.

In the advanced works, algorithms were proposed for incremental and autonomous acquisition and learning of human motions from continuous demonstrations (Kulic et al. [3]). However, in the proposed model, abstraction was based on perceptual similarity, and also the sequence of symbols was given to the agent by communication. Like previous works, the obtained symbols were categorized based on perceptual information.

From a robotics viewpoint, Takano et al. [4] proposed an approach for encoding observed trajectories based on hidden Markov model as mimesis models in order to segment and generate humanoid robot motions through imitation. They also provide an algorithm for motion generation.

The closest works to ours is proposed by Takano et al.[4] and Ramirez-Amaro et al. [5]. Takano et al. introduced a bio-inspired model to acquire abstract relational concepts from imitation, using reinforcement learning. Ramirez-Amaro et al., introduced a method that allows to transferring skills to humanoid robots by extracting semantic representations from observations of human activities. However, in all previous works, abstraction and symbolization are based on similarity in perceptual space, and the proposed approaches cannot deal with abstract concepts. A few good surveys of the imitation learning can be found in [6–9].

In our previous study [10, 11], we demonstrated this hierarchical approach using an unsupervised state-of-the art learning algorithm based on independent virtual subspace analysis to extract spatial-temporal features from virtual environment. The results suggested the accuracy of correctly identifying virtual robot prototype behaviors exceeded our expectants which is very good compared with others approaches. Another advantage of our hierarchical approach is that time required for the training process is reduced.

3. ITERATIVELY IMITATION LEARNING USING BINARY-VALUED DATA

Some recent approaches give strong performance guarantees by training the policy iteratively. However, it is important to note that these guarantees depend on how well the policy we found can imitate the demonstrator by transferring the training data. The standard approach is to use supervised learning algorithms and minimize the prototype deficiency with respect to a prediction. However, this method ignores the difference between distributions of states induced by executing the demonstrator's and learner's policy, thus has a deficiency in the task horizon T.

A recent approach called Dataset Aggregation (DAgger) [12] yields a loss linear in the task horizon T by iteratively training the policy in states induced by all previously learned policies. Its theoretical guarantees are relative to performance of the policy that best mimics the demonstrator on the training data.

DAgger algorithm proceeds by collecting a dataset at each iteration under the current policy and trains the next policy under the aggregate of all collected datasets. The intuition behind this algorithm is that over the iterations, we are building up the set of inputs that the learned policy is likely to encounter during its execution based on previous experience (training iterations). This algorithm can be interpreted as a Follow-The-Leader algorithm [13] where during each iteration we decide the best policy between all trajectories of the virtual prototype which minimizes the observed loss in training problem. Similar the DAgger algorithm we propose a training algorithm witch transfer the sequence of data – as dataset – from virtual environment to physical space. With each capturing dataset from virtual environment, we predict a sequence of actions of the physical robot.

4. THE PROPOSED SOLUTION FOR JOINT DATASET INFORMATION

In this section we are going to describe our solution. We first briefly review concepts of online learning and imitation algorithm that will be used for this analysis.

We present a description of the theoretical aspects of the physical robot motion guidance using a virtual robot prototype, which is the problem addressed in this study. This is an interactive algorithm in which the imitator agent receives dataset information from the demonstrator. So, it can form concepts based on functional characteristics of demonstrated behaviours. This requires the analysis of stereotypical and pre-planned motion of the virtual prototype in order to acquire the desired task [10].

Therefore, the major advantage of these approaches is their ability to analyze the details of virtual prototype movements. At rest, this analysis can be performed without using very sophisticated perception systems such motion capturing systems, to identify the position of joints [11]. Another advantage of this method is the ability to generalize the learned models to new situations, mainly since they depend on the correct identification of robot positions, which are difficult to extract from 3D videos.

To design our model we took inspiration from both algorithms "follow the leader" and "datasets aggregation". The virtual robot prototype, which has different behaviours, was determined a priori. To address this issue, we believe that virtual prototype and artificial intelligence techniques can be exploited in order to provide a better user experience.

By using these techniques, it is possible to model every physical robot as an independent agent that may be able to randomly choose its own path while coordinating with others moving objects, to keep path free.

The idea is to use virtual path as a reference trajectory for physical robot. This approach also estimates more accurately the response of each action through a predictive motion virtual model. Pre-computed trajectories come from the virtual prototype and are used to guide the physical robot. The virtual robot prototype needs to be executed *offline* to classify paths of motion for pre-computed trajectories sets.

The segmentation of the virtual path, required by transferring process, must be done in real time. Therefore, in order to solve the issues described above and to demonstrate that our system does not depend on a complex perception system, we propose the use of the simplest computer vision technique to optimizing the robot behavioral. Users interact with the simulation environment through the visualization.

This includes, but not limited to computer screen. The visualization provides an interface to develop interactive implementations based on simulated behaviour 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). It is easy to recuperate kinematic information from virtual robot motion, using for example motion capture.

We focus on creating a virtual prototype model from experimental data obtained from the physical robot model. A solution to the above problem is to construct a virtual robot prototype, VRP (shorthand for "virtual robot prototype") and to transfer the virtual trajectory by interacting with the physical robot physical robot (PhR) model.

The VRP as virtual homonym of the PhR can interact in real time with the PhR and cooperate together to achieve shared task, such as are cooperative entities, in charge of accomplishing tasks or to collision avoid when they are trying to fulfil tasks quested. In standard implementations, VRP is travelling between pre-defined obstacles on a virtual map.

Due to technical and computational constraints, the designers may be practically sure about not having obstacles collision events. Moreover, this original approach is not very computational intensive and allows for a better scalability. Nevertheless, VRP moving between obstacles will benefit from changing paths when appears obstacles.

The free paths can be easily guessed by experienced scenery and free routes in the work space will be discovered eventually. Optimization of the physical robot behaviour is performed in the low dimensional virtual space by using the virtual robot prototype.

In the virtual environment 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. By combining off-line and on-line programming techniques, our method consists in using a programming platform on which there is carried out the virtual prototype of the real 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 summarize 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, are on-line transferred into the data registers of the controllers of the driving shafts of the real robotic arm; finally, there are obtained tracking structures due to which the moving paths of the virtual robotic arm articulations are tracked by the physical robotic arm articulations, thereby generating motion within the physical working space.

In our training strategy based on the dataset transferring from virtual environment to physical space, we have to sequentially select a subset of data for each instance of transferring process, according to specified accuracy and cost.

To implement a real-time training strategy scheme, three major areas need to be investigated such robot and environment modelling more realistic, dataset transferring and selecting training policy for imitation learning.

5. PHYSICAL ROBOT TRAINING BASED ON VIRTUAL MODEL

We present a description of the theoretical aspects of the physical robot training using a virtual model. The advantages of such approach, as an alternative to vision guided trajectory imitation, 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.

Give a physical robotic agent that move free in work space while being controlled using her virtual homonym. Robots are designed to execute tasks within a defined environment.

The physical entity required to receive real-time digital data while being guided by her virtual prototype homonym.

The physical robot should generate motions based on the current task of its virtual homonym, while considering information related to physical environment. The reference datasets are obtained using a motion capture channel taking into account the joint motion range.

The easiest way to generate the spatial relations explicitly is the interactively programming of the behavior of the virtual prototype in her virtual environment, in order to specify suitable positions.

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.

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.

The dataset 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.

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.

A distributed low-level actuating system handles robot attitude and power monitoring. The main processing unit is currently implemented on a Personal Computer (PC).

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 and displays on a graphical user interface the evolution of the virtual prototype during the carrying out of the robotic task. 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.

The PC runs the operating system with the timeliness support necessary for time-stamping, periodic transmissions and task temporal synchronization provided by a specially developed user-level real time scheduler, the Process Manager.

This approach provides sufficient timeliness support for soft real-time applications, such as multiple robot coordination, and allows profiting from the better development support provided by general purpose operating systems.

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 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.

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

6. IMITATION LEARNING BY DATASET TRANSFERRING

We consider a reduction of imitation learning to online dataset transferring where we treat mini-segments of virtual trajectories under a single policy as a single online-learning example. Our goal is to train the physical robot to do their task based on current virtual image features as input, according to the possible scenario.

Our expert is a near-optimal planning algorithm that has full access to the virtual states and transfer dataset from the virtual demonstrations to physical robot. A transfer action consists of binary variables indicating which subset of virtual path should be reproduced by physical robot joints, at each transferring interval.

However, there are many tasks that are described in terms of possibly conflicting objectives, e.g., a motion control system should minimize latency and maximize the possible quantity of data that can be transmitted between virtual and physical collaborative agents.

We find an optimal control strategy that uses the virtual model of the robot system to obtain an optimal control sequence by minimizing an objective function. At each transferring interval, the virtual model is used to predict the behavior of the physical robot system over a prediction horizon.

Based on these predictions, an objective function is minimized with respect to the future sequence of inputs for each transferring interval.

Although prediction and optimization are performed over a future horizon; only the values of the inputs for the current transferring interval are used. The same procedure is repeated at the next transferring time. We begin by introducing notation relevant to our situation.

6.1. Time Distribution Multiple Access (TDMA)

According [12] one consider the state *s*, and we denote $C(s, a)$ the expected immediate cost of performing action *a* in state *s* for the task we are considering. One denotes:

$$C_\pi(s) = E_{a \approx d_\pi^t}[C_\pi(s)], \tag{1}$$

the expected immediate cost of policy π in state the *s*. Also we assume *C* is bounded in [0; 1].

One denotes by space Π the class of policies the robot agent imitator is considering and refers to *T* the imitation task horizon.

For any policy π, one let d_π^t indicate the distribution of states at time *t* if the robot learner executed policy π from time step 1 to $t - 1$. In addition, one represents the average distribution of states if a robot agent follows a policy π for *T* steps:

$$d_\pi = (1/T)\sum_{t=1}^{T} d_\pi^t . \tag{2}$$

The total cost of executing policy π for *T*-steps, is denoted by cost function:

$$J(\pi) = \sum_{t=1}^{T} E_{s \approx d_\pi^t}\left[C_\pi(s)\right] = T E_{s \approx d_\pi}\left[C_\pi(s)\right]. \tag{3}$$

In imitation learning, we observe virtual expert' demonstrations and investigate to bound $J(\pi)$ for any cost function based on how well policy *π* mimics the virtual expert's policy π^*.

Indicate *l* – the observed loss function – one minimizes the cost *C*. We are interested in optimizing the learner's ability to predict the actions using a virtual robot prototype expert. A physical robot agent follows a policy $\hat{\pi}$. Our target is to find a policy $\hat{\pi}$ which minimizes the observed loss, under its induced distribution of states, i.e.:

$$\hat{\pi} = \arg\min_{\pi \in \Pi} E_{s \approx d_\pi}\left[l(s, \pi)\right]. \tag{4}$$

As robot dynamics is assumed both unknown and complex, we cannot compute the average distribution of states, d_π and can only sample it by executing policy π in the robotic system. The interaction between policy and the resulting distribution makes optimization complicated.

6.2. Reduction to Online Imitation

Let,

$$l_i(\pi) = E_{s \approx d_{\pi_i}}\left[l(s, \pi, \pi^*(s))\right] \tag{5}$$

The expression $l_i(\pi)$ denote the expected substitute virtual robot loss of executing policy π in states distributed according to $d_{\pi i}$ In an online imitation learning setting, in iteration *i* an algorithm executes policy π_i and observes loss $l_i(\pi_i)$. It then provides a different policy π_{i+1} in the next iteration and observes loss $l_{i+1}(\pi_{i+1})$. A success dataset transferring policy guarantees that, in *N* transferring iterations.

$$\frac{1}{N}\sum_{i=1}^{N} l_i(\pi_i) - \min \frac{1}{N}\sum_{i=1}^{N} l_i(\pi) \le \gamma_N \quad and \quad \lim_{N \to \infty} \gamma_N = 0 . \tag{6}$$

Assuming a strongly convex loss function, our transferring-data algorithm admits a success policy. In each transfer iteration admits the policy that works best so far:

$$\pi_{i+1} = \arg\min \sum_{j=1}^{i} l_j(\pi) . \tag{7}$$

At iteration *i* we choose the policy that has the minimum demonstrator loss on all previous transferred data. Thus, it can be interpreted as virtual trajectories transferred policy is treated as online-programming example. However, it can be hard to find a good policy that has a low training error, since the demonstrator's policy may resides in a space that is not imitable in the learner's policy space [14].

When the optimal action is hard to achieve, we propose to train gradually the imitator with easy-to-learn actions. A virtual demonstrator trains the physical imitator iteratively in a fashion similar to DAgger algorithm.

The training becomes robust by showing more demonstrator actions as the imitator makes progress. Intuitively, this allows the learner to move towards a better action without much effort. Thus our algorithm achieves the best action gradually instead of aiming at an impractical goal from the beginning

6.3. Limitation of the Imitation

A virtual demonstrator can be firm to imitate in two ways. First, the learning policy space is far from the space that the oracle policy lies in, meaning that the learner only has limited learning ability.

It can be hard to find a good policy that has a low training error, since the demonstrator's policy may resides in a space that is not imitable in the imitator's policy space. For instance, the task loss function in the imitator's space can be very different from the demonstrator' loss [15, 16].

Second, the environment information known by the virtual demonstrator cannot be sufficiently inferred from the state, meaning that the imitator does not have access to good learning resources.

In the online learning scenery, a too-good demonstrator may result in adversarial varying loss functions over iterations from the learner's perspective. This may cause brutal changes during policy updating. These difficulties result in a substantial gap between the virtual demonstrator's performance and the best performance achievable in the policy space Π.

7. CONCLUSIONS

Virtual prototyping is an aspect of information technology that permits analysts to examine on a computer monitor the behavior of the physical robots using her virtual prototype.

By virtual prototyping one uses a virtual model in lieu of a physical robot model, for test and evaluation of specific characteristics of a candidate design.

There is a trend to use of virtual prototypes during the design analysis process. A virtual prototype is a digital model with a degree of functional realism comparable to a physical model. The value of virtual prototyping is rapidly being recognized for a wide range of engineering applications.

The advantages of such approach, as an alternative to the classical methods, are on-line adaptation to the motion of the virtual prototype obtained as initial training model. These types of trajectory level techniques are very useful for extracting relevant information from activities, as well as for transferring these models to artificial systems. Transferring the models acquired from virtual demonstrations to physical robots is a challenging task for the building of the adaptive and autonomous robots, mainly because it requires the generation of task-specific that should be naturally in a physical environment.

Our major contribution is an online dataset transferring algorithm for imitation learning tasks, where the available space policy is not adequate for imitating of the demonstrator.

ACKNOWLEDGMENTS: The authors 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 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] H. Kadone and Y. Nakamura, *Hierarchical concept formation in associative memory models and its application to memory of motions for humanoid robots*, Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS, 2006, pp. 432–437.

[2] M. Pardowitz et al., *Incremental learning of tasks from user demonstrations, past experiences and vocal comments*, IEEE Transactions on Systems, Man and Cybernetics, Part B 37 (2007), No. 2, pp. 322–332.

[3] D. Kulic, W. Takano, and Y. Nakamura, *Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains*. The International Journal of Robotics Research, vol. 27, no. 7, 2008, pp. 761–784.

[4] W. Takano and Y. Nakamura, *Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation*. In Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS, 2006, pp. 425–431.

[5] K. Ramirez-Amaro et al., *Transferring skills to humanoid robots by extracting semantic representations from observations of human activities*, Artificial Intelligence (2015), http://dx.doi.org/10.1016/j.artint.2015.08.009

[6] M. Muehlig et al., Task level imitation learning using variance based movement optimization, Proceedings IEEE International Conference on Robotics and Automation (ICRA), 2009, pp. 1635–1642.

[7] A. J. Ijspeert, J. Nakanishi, and S. Schaal, *Trajectory formation for imitation with nonlinear dynamical systems*, In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 752–757.

[8] B. Argall et al., *Learning robot motion control with demonstration and advice-operators*, In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2008, pp. 399–404.

[9] B. Argall et al., *A survey of robot learning from demonstration*, In: Robot Automation Systems 57 (2009) No. 5, pp. 469–483.

[10] A. Fratu, M. Fratu, *Imitation-Based Motion Programming for Robotic Manipulators*, Proceedings of 14th International Conference on Optimization of Electrical and Electronic Equipment OPTIM 2014, May 22-24, 2014, Brașov, Romania, pp. 770–775.

[11] A. Fratu, B. Riera et al., *Predictive strategy for robot behavioral control*, Proceedings in Manufacturing Systems, vol. 9, issue 3, pp. 125–130.

[12] St. Ross et al., *A reduction of imitation learning and structured prediction to no-regret online learning*, Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), 2011, pp. 627–635.

[13] T. van Erven et al., *Follow the Leader with Dropout Perturbations*, In Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings, 2014, vol. 35,pp. 1–26.

[14] V. V. V'yugin, *Online Learning in Case of Unbounded Losses Using Follow the Perturbed Leader Algorithm*, Journal of Machine Learning Research, vol. 12, 2011, pp. 241–266.

[15] A. Coates et al., *Learning for control from multiple demonstrations*. In Proceedings of the 25th International Conference on Machine learning, 2008, pp. 144–151.

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