

Towards Autonomous Robotic Valve Turning

A. Carrera *, *S. R. Ahmadzadeh***, *A. Ajoudani***, *P. Kormushev***,
*M. Carreras**, *D. G. Caldwell***

* *Computer Vision and Robotics Group (VICOROB), University of Girona, 17071 Girona, Spain*

** *Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy*

*Emails: ACarrera@eia.udg.edu {Reza.Ahmadzadeh |Arash.Ajoudani | Petar.Kormushev}@iit.it
Marc.Carreras@udg.edu Darwin.Caldwell@iit.it*

Abstract: *In this paper an autonomous intervention robotic task to learn the skill of grasping and turning a valve is described. To resolve this challenge a set of different techniques are proposed, each one realizing a specific task and sending information to the others in a Hardware-In-Loop (HIL) simulation. To improve the estimation of the valve position, an Extended Kalman Filter is designed. Also to learn the trajectory to follow with the robotic arm, Imitation Learning approach is used. In addition, to perform safely the task a fuzzy system is developed which generates appropriate decisions. Although the achievement of this task will be used in an Autonomous Underwater Vehicle, for the first step this idea has been tested in a laboratory environment with an available robot and a sensor.*

Keywords: *Autonomous Underwater Vehicle (AUV), Imitation Learning, Fuzzy System, Extended Kalman Filter (EKF), Valve Turning.*

1. Introduction

Nowadays, Autonomous Underwater Vehicle (AUV) robots are used in different applications like seabed survey, mine cleaning, cable or pipeline tracking [1], deep ocean exploration with visual mapping [2] or water quality observation [3]. For intervention tasks, still Remotely Operated underwater Vehicles (ROV) are used

due to the complexity and uncertainty of the work. ROVs are operated by one or two persons, usually one to keep the robot stable and another to control the manipulators [4]. Recently, some positive results have been achieved in the task of recovering objects from the seabed ones using a robotic arm [5].

The main goal in the European project “Persistent Autonomy through learning, adaptation, Observation and Re-planning (PANDORA)”[6] is to develop and evaluate new computational methods to make human-built robots Persistently Autonomous. The goal is to reduce the frequency of assistance requests significantly and the key to this aim is the ability to recognize failure and respond to it autonomously. The PANDORA project is focused on three underwater tasks, one of them consisting in autonomous grasping and turning a valve with a free floating AUV.

The AUV uses a manipulator to grasp the correct valve on a panel and open or close it. Since the vehicle does not dock, it needs to hover by swimming when counteracting reaction forces from the turning and from the sea currents and even minor turbulence from the manifold. Also it must ensure that the gripper position and orientation of the gripper after grasping does not cause significant shear forces in the valve handle (T-bar shape), and break it off. To overcome the difficulties in this task, imitation learning techniques [7] can offer a robust solution and an easy way to teach the robot trajectory using the data from a set of demonstrations. This kind of a learning method includes the most important desirable properties of movement planning which are: ease of representing and learning, compactness of the representation, robustness against perturbations and changes in a dynamic environment, ease of reuse for related tasks and easy modification for new tasks, and ease of categorization for movement recognition. However, no standard approach of movement planning exists that accomplishes all these goals [8-11].

This paper presents the preliminary work about autonomous valve turning done with a real manipulator in lab conditions, not underwater. Several sensors have been used to estimate the distance between the gripper and the valve. Also, an Extended Kalman Filter (EKF) [12] has been applied to improve the estimations and to avoid gaps in the data. Moreover, according to the instant dynamics of the valve, the robot has to decide if it can approach to the valve or not. To make this decision, a fuzzy system has been used.

This paper has the following format. In Section 2 the general task and the proposed experimental environment are explained. In Section 3 the different methods and their combination to solve the task are described. The obtained results are showed in Section 4. And finally, conclusions are exposed in Section 5.

2. Problem description and environment

In PANDORA project, the task of grasping and turning the valve requires a set of actions that have to be done successfully before the final step. It must be noticed that in this paper the task of grasping and turning the valve is investigated, when the vehicle is in the work area in front of the desired valve.

This task will be accomplished by Girona500 [13], which is a compact and lightweight AUV, with hovering capabilities and can fulfil the particular needs of any application by means of mission-specific payloads and a reconfigurable propulsion system. In this case the AUV needs to be equipped with a robotic arm.

Before attempting the valve turning task underwater, an approach system has been built in the facilities of Istituto Italiano di Tecnologia (IIT), using a light weight robotic arm (KUKA/DLR) and an Optitrack system.

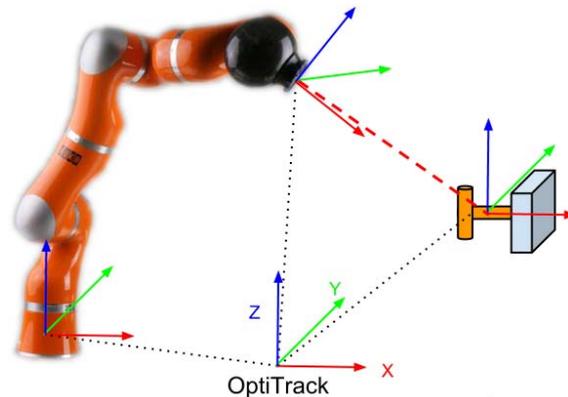


Fig. 1. Schematic diagram for the real scenarios

The KUKA/DLR robotic arm will be used under Cartesian impedance control mode [14]. The robot's position, orientation and the fixed joint or Cartesian stiffness commands will be sent to the KUKA controller using the DLR's Fast Research (FR) Interface libraries [15].

Although the KUKA/DLR is a different kind of manipulator than the one that is going to be attached to the Girona500 robot, the goal of this paper is to learn the attained trajectory during the experiment, not the specific kinematics of the manipulator. The available sensor in the AUV will be simulated using the Optitrack system, which lets you get the 3D position and orientation of a rigid body using a set of cameras and markers, see Fig. 1. This system gives the position and orientation with good precision and high frequency. In this problem we will focus our interest on the distance between the valve and the end effector; therefore we will mark these two elements to be tracked by the Optitrack system.

The real AUVs can acquire data from one camera, but the precision and sampling frequency of the Optitrack's data are much better. To improve the camera information, the robot has more sensors, in this case the gyro-enhanced Attitude and Heading Reference System (AHRS) will be used. The camera and the AHRS will be combined using an EKF to improve the estimation. Both sensors are simulated using the Optitrack's data.

Finally, the last difference between the two environments is the fact that underwater the valve will be fixed and the robot will move, but in the lab the situation is the opposite.

3. Experimental set-up and results

In this section the experimental set-up details, the process, and the results are described step by step. The diagram of the set-up is shown in Fig. 2.

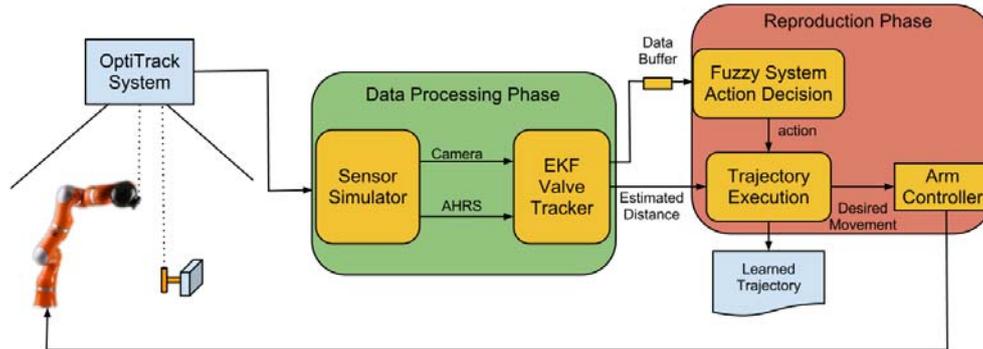


Fig. 2. Flowchart of the system: This diagram represents how the data flow from the Optitrack system to the different phases of experiment to finally convert in commands to the robot. The learning phase is not included because it is done offline with the recorded data

3.1. Data processing phase

In this phase, an EKF is designed to estimate the position. The EKF is an extension of Kalman Filter (KF) [12], able to work with non-linear functions. The KF uses measurements with noise and an approximated model of the process studied to produce measures of the state, which are more accurate than those based on single measurements. Previous works have proved the advantages of using an EKF to track objects [16, 17].

In this task the movement of the end effector towards the valve is represented as a model with a constant acceleration, and the measurements of the system are the position of the valve obtained using a camera, and the acceleration of the vehicle obtained using an AHRS. The orientation of the vehicle is also considered as a control signal.

The reference frame has the center in the valve position. In this way we avoid the differences between the two environments and solve the problem of changing the position of the valve. In this case, the mobile part in the coordinate system will be the end-effector and the base of the robot, all movements of the valve will be reflected in these two elements. The next equation shows the homogeneous transformation Matrix done to convert the data from the Optitrack frame to the valve frame, in this case a composed matrix is used with translation and standard rotation with RPY Matrix:

$$(1) \quad \text{optitrack } K_{\text{valve}} = \begin{pmatrix} R(\alpha_{\text{valve}}, \beta_{\text{valve}}, \gamma_{\text{valve}}) & P_{x_{\text{valve}}} \\ & P_{y_{\text{valve}}} \\ & P_{z_{\text{valve}}} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The two inputs of EKF are generated using different samples rates from the Optitrack's data. The data selected to generate one sensor will never be used to generate the data of the other, making both sensors independents. Moreover, appropriate noise is added to the signals, using a precision similar to the sensors, for the camera it is ± 0.005 m, and for the AHRS it is ± 0.5 m/s².

3.2. Learning phase and trajectory execution

A robot should be able to acquire new skills using various forms of learning and when direct physical contact to the robot is possible, kinaesthetic teaching offers a user-friendly and intuitive method to demonstrate new skills to a robot by manually guiding the robot's arm through the motion.

Briefly, using several demonstrations of a similar task, the robot creates a compact model of the skill by taking into account the variations and correlations observed along the movement. The positional constraints of the demonstrated skill are represented as a matrix of dynamical systems encoding robustly position trajectories. The Dynamic Movement Primitives (DMP) framework, which is used in this paper, was originally proposed by Ijspeert et al. [7] and after that in [18-20].

M examples of a skill are demonstrated to the robot with different initial positions. Each demonstration $m \in \{1, \dots, M\}$ consists of a set of T_m positions x , velocities \dot{x} and accelerations \ddot{x} of the end-effector in Cartesian space, where each position has three dimensions. Using the datasets from demonstrations, a mixture of K proportional-derivative systems is created as a model of the skill [20].

In this approach a decay term defined by a canonical system $\dot{s} = -\alpha s$ is used to create an implicit time dependency property $t = -\ln(s)/\alpha$, in which the initial value for s is 1 and converges to zero. Also for the backward movement, which is used in retracting mode, a complementary equation is used to generate time starting from the final time to the initial time.

A set of K Gaussians is defined in time space, with centres μ_i^T equally distributed in time, and variance parameters set to a constant value inversely proportional to the number of states. By determining the weights $h_i(t)$ through the decay term s , the system will sequentially converge to the set of attractors in Cartesian space defined by centres μ_i^X and stiffness matrices k_i^P , which are learned from the observed data, either incrementally or in a batch mode.

The desired acceleration to generate the trajectory is computed using the next equation, where x and \dot{x} are the current position and velocity and k^v defines the damping factor:

$$(2) \quad \hat{\ddot{x}} = \sum_{i=1}^K h_i(t) \left[k_i^P (\mu_i^X - x) - k^v \dot{x} \right].$$

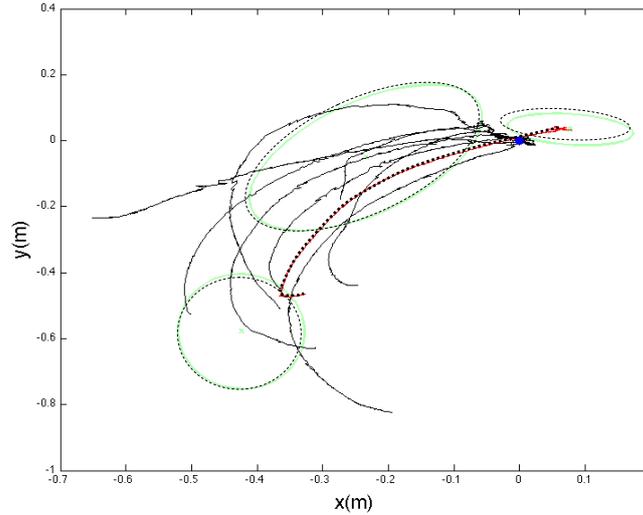


Fig. 3. In dense line you can see all the trajectories in 2D, given by demonstration. The broken line ellipsoids are the states learned by the algorithm to represent the trajectory and in dotted line you can see one trajectory produced using the learning part

It can be seen that for parts of the movement where the variations across the different demonstrations are large, the reference trajectory does not need to be tracked precisely. By using this information, the controller can focus on the other constraints of the task, such as collision avoidance. On the other hand, for other parts of the movement, exhibiting strong invariance across the demonstrations should be tracked precisely.

3.3. Reproduction phase

A fuzzy system is used to generate a decision command based on linguistic variables and rules. In a fuzzy system the fuzzifier section maps the crisp inputs into some fuzzy sets. Then the fuzzy inference engine uses fuzzy IF-THEN rules from the defined rule base to reason for the fuzzy output. The generated output in a fuzzy term is converted back to the crisp value by the defuzzifier section [21].

Since we are using EKF to make estimation and fill in the gaps from our sensor data, when we do not receive data for a while the uncertainty becomes bigger and bigger. In addition, the robot or the operator needs to know about the dynamics of the valve and decide if it can be approached, because if the relative movement exceeds the normal range we may miss the valve or break it off. Therefore, the first

input for our fuzzy system is an estimated movement between the valve and the arm. And the second input is receiving the data delay from the sensor. The output of the fuzzy system is a numerical command in the continuous range of grasping, waiting, and retracting actions $[-1, 1]$.

Here, we use Sugeno inference which consists of product inference engine, singleton fuzzifier, and center average defuzzifier. We use Gaussian membership functions in our fuzzy sets, and the designed fuzzy rule base is showed in Table 1.

Table 1. Fuzzy rule base

| | | Relative Movement | | |
|--------------|--------|-------------------|----------|----------|
| | | Small | Medium | Big |
| Sensor Delay | Low | Forward | Stop | Backward |
| | Medium | Forward | Stop | Backward |
| | High | Stop | Backward | Backward |

After defining the rules for the system, let w be the output value of each step and z be the weight for each rule, then the final output of the fuzzy system is [21]:

$$(3) \quad \text{FinalOutput} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i}.$$

Finally, with the input of the distance between the end effector and the valve, and with the decision of the fuzzy system the learning part generates a proportional movement with respect to the position to move the robot. The result is evaluated and if the condition of grasping the valve is completed, an instruction is sent to turn the valve, in the other case the new position is sent to the KUKA robotic arm.

3.4. Complete experiment

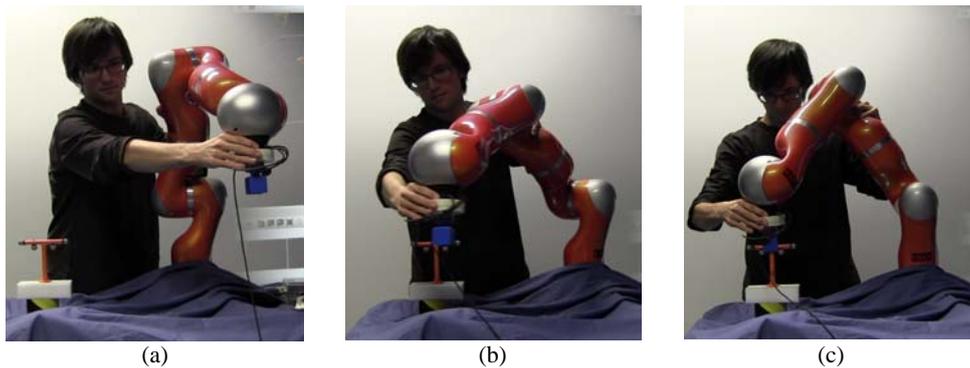


Fig. 4. This set of images represents the process of one demonstration of the task of grasping the valve

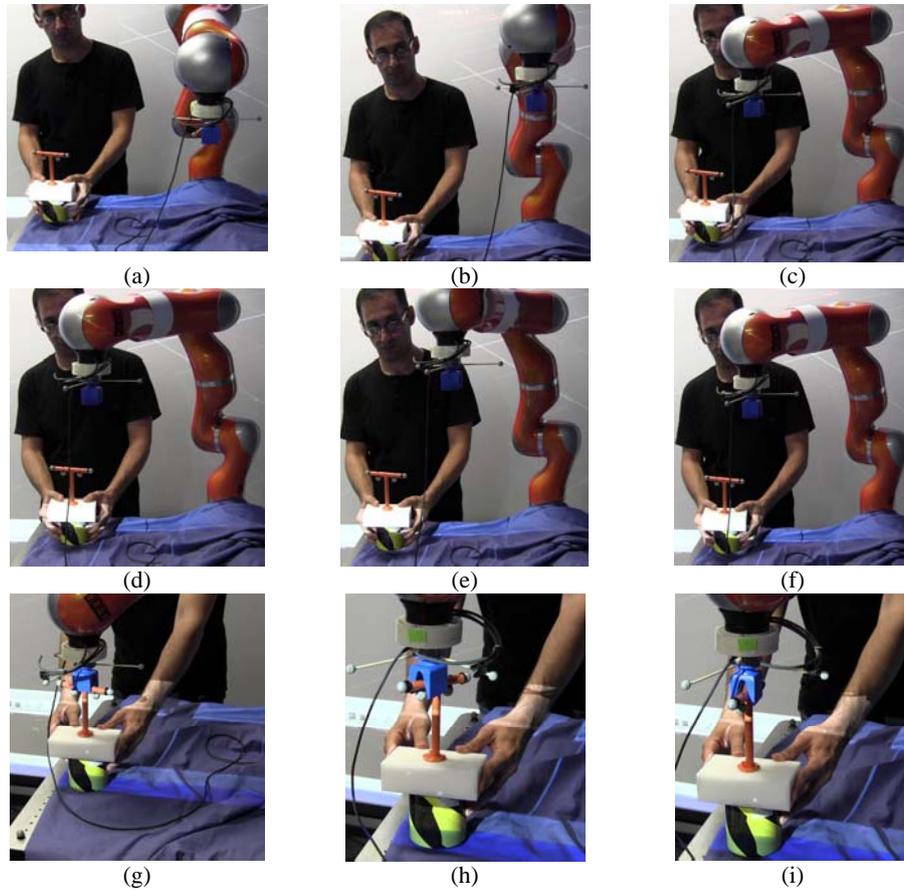


Fig. 5. The set of images show the whole process of reproducing the task with perturbations. Images (a) and (b), the robot reaches the initial common position, from any position. Images (c) and (d), the valve is not stable and the robot moves to a safer position. Image (e). Image (f), the valve is stable in a new position, so the robot moves to grasp it. Image (g), the robot has grasped the valve and finished the trajectory. Images (h) and (i), the robot does the 90° turning

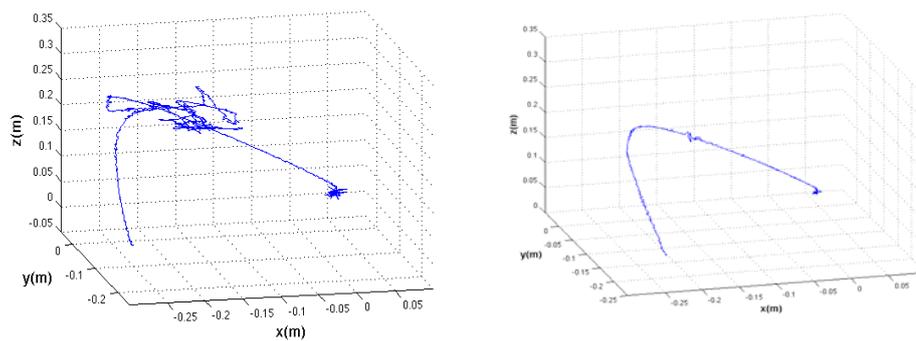


Fig. 6. In this set of images we can see two different trajectories which are done by the end effector to grasp and turn the valve. In the left figure we can see the changes in the trajectory, moving forward or backward, depending on the stability of the valve position. On the other side, the trajectory on the right figure shows a clean trajectory going only forward to the valve, in this case the valve has been stable during the process

In the final experiment all steps of the operation (from learning until valve turning) in a complete loop are accomplished by the robot appropriately and effectively, see Figs 4, 5 and 6. The designed system is capable of providing a smooth trajectory, compensating small perturbations in sensors, analysing the safeties of the situation and moving forward or backward with smooth changes, tracking the trajectory efficiently, and detecting the end of the trajectory and turning the valve. In future works the turning phase will be learned by the robot instead of using pre-programmed commands.

4. Conclusions

The proposed combination of techniques in this paper has allowed a robotic arm to learn the skill to follow a trajectory, grasp a valve and turn it. This experiment has been done as a simulated scenario of an underwater operation with an AUV.

During this experiment the robot has been tracking the distance between the valve and the end effector of the robotic arm, doing a mixture of different kind of measurements using the EKF which generates a smooth movement of the robot and more stability in the control of the position. Moreover, the learning part has provided the ability to extract the important restrictions of a set of trajectories and offer the adaptability and robustness to follow the trajectory. In addition, the method has proved the possibility of utilizing a fuzzy system to study the dynamic behaviour of the valve and choose a proper action.

Acknowledgement. This research was sponsored by PANDORA[6] EU FP7-Project under Grant agreement No ICT-288273.

References

1. Gracías, X., S. Negahdaripour. Underwater Mosaic Creation Using Video Sequences from Different Altitudes. – In: OCEANS, 2005. Proceedings of MTS/IEEE, Vol. 2, 17-23 September 2005, 1295-1300.
2. Billar, S. Calinon, R. Dillmann, S. Schaal. Survey, Robot Programming by Demonstration. Handbook of Robotic. Chapter 59. 2008.
3. Guo, S., X. Lin. Development of a Vectored Water-Jet-Based Spherical Underwater Vehicle. – In: N. A. Cruz, Ed. Autonomous Underwater Vehicles. InTech, Janeza Tardin 9, 51000 Rijeka, Croatia, 2011, 3-20.
4. Ryu, J. H. Control of Underwater Manipulators Mounted on an ROV Using Base Force Information. – In: IEEE Int. Conf. on Robotics and Automation (ICRA), Vol. 4, 2001, 3238-3243.
5. Prats, M., D. Ribas, N. Palomeras, J. C. Gacía, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridaó. Reconfigurable AUV for Intervention Missions: A Case Study on Underwater Object Recovery. – Intelligent Service Robotics, Vol. 5, 2012, 19-31.
6. Persistent Autonomy through learning, adaptation, Observation and Re-planning (PANDORA). <http://persistentautonomy.com/>
7. Ijspeert, A. J., J. Nakanishi, S. Schaal. Trajectory Formation for Imitation with Nonlinear Dynamical Systems – In: Proc. IEEE Intl. Conf. on Intelligent Robots and Systems (IROS), Maui, USA, 2001, 752-757.
8. An, C. H., C. G. Atkeson, J. M. Hollerback. Model-Based Control of a Robot Manipulator. MIT Press, 1988.

9. Miyamoto, H., S. Schaal, F. Gandolfo, Y. Koike, R. Osu, E. Nakano, Y. Wada, M. Kawato. A Kendama Learning Robot Based on Bi-Directional Theory. – *Neural Networks*, Vol. **9**, 1996, 1281-1302.
10. Kawato, M. Trajectory Formation in Arm Movements: Minimization Principles and Procedures. – In: H. Z. Zelaznik, Ed. *Advanced in Motor Learning and Control*. Champaign Illinois, Human Kinetics Publisher, 1996, 225-259.
11. Sutton, R., A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
12. Welch, G., G. Bishop. An Introduction to the Kalman Filter. ACM SIGGRAPH 2001, Course 8.
<http://www.cs.unc.edu/~welch/kalman/>
13. Ribas, D., N. Palomeras, P. Ridaio, M. Carreras, A. Mallios. Girona 500 AUV, from Survey to Intervention – *IEEE/ASME Transactions on Mechatronics*, Vol. **17**, 2012, 46-57.
14. Albu-Schäffer, A., C. Ott, G. Hirzinger. A Unified Passivity-Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. – *Int. Journal of Robotics Research*, Vol. **26**, 2007, No 1, 23-39.
15. Schreiber, G., A. Stemmer, R. Bischoff. The Fast Research Interface for the KUKA Lightweight Robot. – In: *IEEE ICRA*, 2010.
16. Funk, N. A Study of the Kalman Filter Applied to Visual Tracking. Technical Report. University of Alberta, 2003.
17. Zhou, Q., J. K. Aggarwal. Object Tracking in an Outdoor Environment Using Fusion of Features and Cameras. – *Image and Vision Computing*, Vol. **24**, 2006, 1244-1255.
18. Schaal, S., P. Mohajerián, A. J. Ijspeert. Dynamics Systems Vs. Optimal Control a Unifying View. – *Progress in Brain Research*, Vol. **165**, 2007, 425-445.
19. Hoffmann, H., P. Pastor, D. H. Park, S. Schaal. Biologically-Inspired Dynamical Systems for Movement Generation: Automatic Real-Time Goal Adaptation and Obstacle Avoidance. – In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, 2587-2592.
20. Kormushev, P., S. Calinon, D. G. Caldwell. Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input. – *Advanced Robotics*, Vol. **25**, 2011, No 5, 581-603.
21. Wang, L. X. *A Course in Fuzzy System and Control*. 2nd Edition. Prentice-Hall International, Inc., 1997. ISBN: 0135408822.