# Autonomous acquisition of pushing actions to support object grasping with a humanoid robot

Damir Omrčen, Christian Böge, Tamim Asfour, Aleš Ude, and Rüdiger Dillmann

*Jožef Stefan Institute, Slovenia, damir.omrcen@ijs.si, ales.ude@ijs.si*

*University of Karlsruhe, Germany, asfour@ira.uka.de, c.boege@gmx.de, dillmann@ira.uka.de*

***Abstract*— There are many situations in which an object that needs to be grasped is not graspable, but could be grasped if it was situated at a different location. By applying nonprehensile manipulation actions such as poking, the object can be moved to a new location without first being grasped. We consider these issues in the context of an artificial cognitive system. The goal of the paper is twofold; firstly, we study how the robot can acquire nonprehensile manipulation knowledge by observing the outcomes of exploratory movements on objects. We propose a learning process that enables the robot to acquire a general pushing rule describing the relationship between the direction of poke and the observed object motion for a class of objects. In this way the robot acquires new action knowledge without having any specialized prior model about the action. Secondly, we investigate how the acquired action knowledge can be used to realize grasping in complex situations where the robot could not grasp the object without moving it to a new location. Here the learned poking behavior serves as a support action for robot grasping. The proposed approach has been implemented and tested on a humanoid robot Armar III.**

## I. Introduction

Autonomous robots should explore their environment actively, persistently, and systematically as most animals and humans do [16]. To study and develop intelligent humanoid robots, we therefore pursue a developmental approach. Cognitive abilities should develop by refining the knowledge acquired in previous stages of the development [8]. In the context of this paper, the first stage involves learning to distinguish the robot body from the rest of the world. Afterwards, the robot can move on to the second stage, that is interaction with external objects. The next stage involves interpreting object-object interactions. In this paper we focus on the second stage – interaction with external objects.

In a cognitive system objects and actions cannot be separated because objects can induce actions (cup → drink), while actions can redefine objects. Objects and actions are inseparably intertwined and higher-level categories are therefore determined (and also limited) by the action an agent can perform and by the attributes of the world it can perceive; the resulting, so-called Object-Action Complexes (OACs) are the entities on which cognition develops (action-centered cognition) [17]. While this paper is concerned with OACs at the level of early perception-action events, our research strives to provide a continuous path from such events to complex cognitive processes, where OACs are used as basic building blocks.

One approach towards the acquisition of new object-action knowledge is that such information can be obtained by performing exploratory primitive actions on a number of different objects. By observing the changes in the environment caused by the applied actions, the robot can associate the applied actions with the resulting object behavior and thus gain understanding of causes and effects. As a representative example we study the acquisition of nonprehensile manipulation knowledge, i.e. object manipulation without a grasp. This kind of manipulation is used when it is difficult or impossible to grasp an object, e.g. when an object is too wide, too large or too heavy. As an example of nonprehensile manipulation, we focus on poking, which is defined as a short term pushing action. Poking can also be used in order to identify and segment the objects from the background [4].

Our aim is to obtain a general **pushing rule** rather than an object specific one. Humans are good at generalization, especially when their experiences are very diverse. The same generalization capabilities need to be achieved in autonomous robots. To achieve a reasonably high level of generalization, some authors use recurrent neural networks with parametric bias [10][14], where static images of objects are linked to dynamic features of objects. In this paper we achieve generalization of the pushing rule by using object images as input to the system, which provides an appropriate data to extrapolate the pushing rule.

When poking an object, the object motion depends on the object's shape, weight distribution and on the support friction forces. A lot of work has already been done in the field of mechanics on the controllability and planning for poking [7] [6]. Obviously, poking could easily be implemented by assuming a proper representation for the physics of the task, but such an approach relies on a priori knowledge about the action and therefore does not solve the complete learning problem. Additionally, it is sometimes difficult to obtain the model parameters using available sensors (e.g. it is very difficult to obtain friction between the object and the pusher using vision). If the physical model of the object and the action is not available like in our work, the robot has to experiment with different poking actions on the object. In this way the robot acquires new knowledge from exploration and human demonstration in the same way as infants learn their actions – performing actions on objects means playing with toys.

While poking has been used to study cognitive processes before [5], our work focuses on different issue, which is learning generalized pushing rules with the application to

grasping. After learning, the robot can use the newly acquired knowledge to push an object towards a specified location. On the other hand, Fitzpatrick et al. [5] were primarily concerned with using poking actions to extract the associated object properties. Pushing has often been used as an example when learning affordances [13], but this type of research normally focuses on higher-level knowledge such as "this ball affords pushing". Instead, our work focuses on acquiring fine-grained controller that can be used to move a range of different objects in any specified direction.

Only objects in proximal space are graspable and only objects of a certain size and shape can be grasped. There are many studies on graspability of objects [18] and how to generate grasp hypothesis. Even if the object is graspable in free space, it might not be possible to grasp it when other objects, e.g a surface on which it lies, are taken into consideration. In this paper we study how such problems can be resolved by pushing.

## II. Scenario description

The experiment has been designed as follows (see Fig. 1). The robot stands at a table and needs to grasp an object that lies on the table. After detecting that the object is not graspable at a given configuration because it is too wide, the robot has to bring the object to the table boundary, where the object becomes graspable due to its flatness. The robot uses its hand to push the object. The part of the hand that has been used for pushing will be termed as a *pusher*. Once the object is pushed to the table boundary, it becomes graspable and the robot grasps it (see also the attached video).

To learn a general pushing rule, the robot starts by experimenting with different primitive poking actions[1] applied to different objects and at different locations on the objects' boundaries. With this process the robot builds a knowledge base, which describes the relationship between the point and angle of push on one side and the actual object movement on the other side. Based on this data, a neural network is learned, which maps the performed pushing action and shape of the pushed object to the resulting object movement.

Afterwards the robot can use the acquired knowledge to find the right poking action in order to move the object as desired (i.e. the robot should push an object to a graspable position). The objects used in the experiments are planar polygonal objects such as shown in Fig. 7.

For object pose estimation, a stereo-based approach presented in [3] has been applied. Textured object are recognized and localized using 2-D feature point correspondences between the current object snapshots and the off-line learned views, which are stored as part of the object representation in an object database. The pose is computed on the basis of triangulated subpixel-accurate stereo correspondences within the estimated 2-D area of the object, yielding 3-D to 3-D point correspondences with a training view. On the other hand, recognition and localization of single-colored objects combines model-based view generation with stereo-based position estimation. Orientation information is retrieved from the matched views and an accurate pose is calculated by a pose correction procedure, as presented in [3].

The proposed techniques were implemented on a humanoid robot Armar III [1] (see Fig. 1) and industrial arm Mitsubishi Pa-10. Grasping has been implemented only on the humanoid robot using visual servoing techniques as presented in [15]. Given the object position, a collision-free motion is generated to reach a pre-grasp position of the arm using inverse kinematics. To estimate the hand position, an artificial marker is attached to the wrist of the robot and tracked visually while moving the hand to the grasp pose. The required orientation of the hand is computed using the forward kinematics.

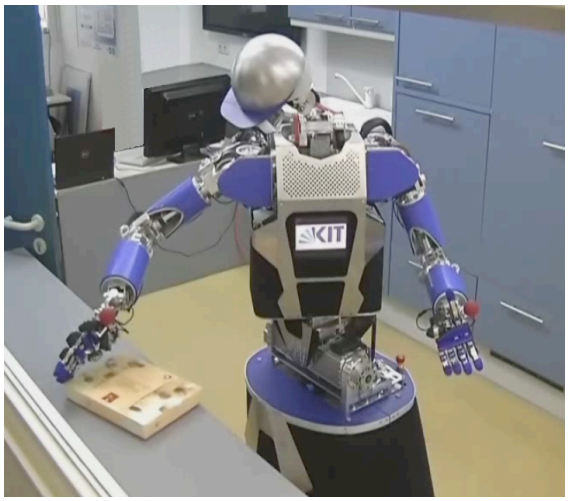


Fig. 1: Humanoid robot Armar III during pushing action

## III. Learning of the pushing rule

The pushing rule is learned using an exploratory process introduced in the previous section. The task of the robot is to learn the relationship between the point and angle of push on the object's boundary and the actual object movement after the pushing action is performed (see Fig. 2). We call the problem of learning the movement of the object being pushed the **direct** pushing problem.

In our previous work [11], the robot learned the response of only one object after the push has been performed. Thus for each new object, the robot had to learn everything from scratch. There was no prediction and no generalization to other objects. Here we propose an approach that can generalize the pushing rule to objects that differ in shape and size, including objects whose response to pushes was not observed during learning. The set of objects used in the experiment is shown in Fig. 7.

In the learning phase, the robot experiments with a number of different poking actions. The robot has to poke an object from different sides and at various angles (see Fig. 2). Additionally, it has to experiment with different objects to achieve generalization. In the beginning of the process the robot has no knowledge about how objects respond to the primitive poking actions, thus initially the robot experiments with different poking actions randomly.

[1] We implemented the primitive poking actions are straight line movements of a pusher in a given direction. Regarding pushing, this is the only prior knowledge available to the system.

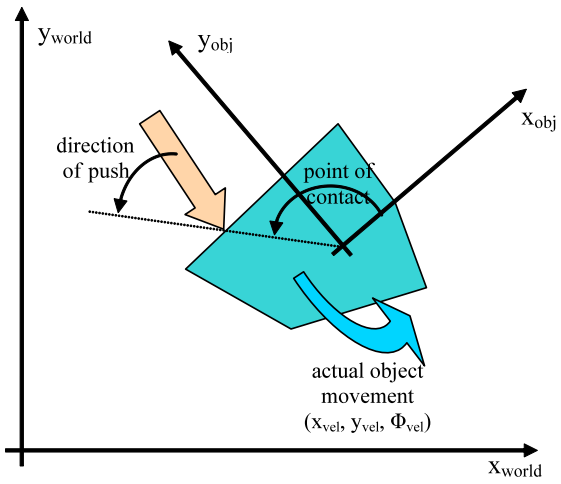


Fig. 2: Schematics of a poking action

After a poking action has been applied to an object, the object accelerates and changes its position and orientation. Since the objects are relatively light and the friction between the object and the table is relatively high, we can neglect the dynamic properties of motion. Typical response of the object is shown in Fig. 3. The object velocity settles in less that 200 ms. The object velocity estimated by vision is noisy, but since learning takes place after the push is completed, the data can be filtered and processed well before the use.

Since the object settles its motion in a very short time, the response of the object to the poking action is determined with sufficient accuracy by simply observing the displacements of the pusher and the object. The displacement of the pusher is expressed by two parameters: the point and the angle of contact on the object boundary, which define the direction of push. The primitive poking actions keep the velocity of the pusher constant while performing the action. The point of contact is expressed as the angle between the line segment connecting the point of contact and the centre of the object and the x-axis of the object's coordinate system. Similarly, the angle of a contact is expressed as the angle between the pushing direction and the tangent at the point of contact (see Fig. 2).

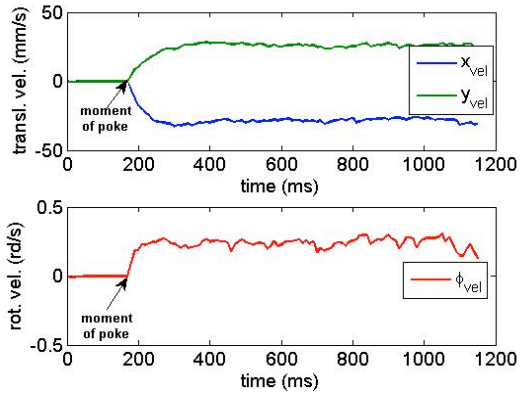


Fig. 3: Typical response (velocity) of an object after applying a poking action

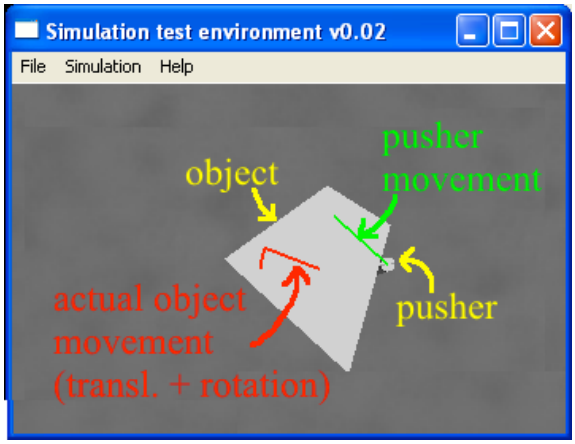


Fig. 4: Agent view of a scene during learning

The response of an object is represented by three parameters, i.e. the planar velocity of the object centre and the rotational velocity about the centre point on the object. The abstract robot's view of the experiment is shown in Fig. 4.

Humans can predict the motion of the object to be pushed based on the acquired object images, which are used as input to the neural networks in the brain. The application of retina images as an input to control the robot behavior has already been studied in robotics. For example, Oztop et al. have used retina images and Hopfield networks to realize hand posture imitation [12]. Inspired by these findings, we utilized the binarized object images as an input to the system. To limit the search space that needs to be explored, we normalized the binarized images with respect to the pushing direction in the retina image. Before each primitive poking action is applied, the observed scene is mapped in such a way that the pushing direction is always at the same position on the retina (see Fig. 5). This normalization process ensures that the acquired knowledge is invariant against object position and orientation as long as the pusher is able to apply the primitive poking actions at the given configuration.

The original resolution of the camera images was 640 x 480 pixels. Due to the computational complexity and to achieve faster convergence and better generalization, we reduced the resolution of the input image to 20 x 15 pixels. Translated and rotated retina images of reduced resolution served as input to the neural network, which is used to represent the pushing rule. The network has three outputs to represent the predicted velocity of an object in all three directions.

We applied a two-layer backpropagation network with 300 input neurons, 10 neurons in the hidden layer, and 3 output neurons. Each input neuron corresponds to a pixel of the object's reduced resolution image. The value of the pixel is in the range from 0 to 1, where 0 means that no object is present at the pixel, while 1 means that the pixel is covered by the object. The three output neurons correspond to the object's linear and rotational velocity on the support surface. The output velocities have been normalized to the range from -1 to 1. The hidden layer as well as the output layer use the tan-sigmoid transfer function. To train the network we employed the Levenberg-Marquardt training function.

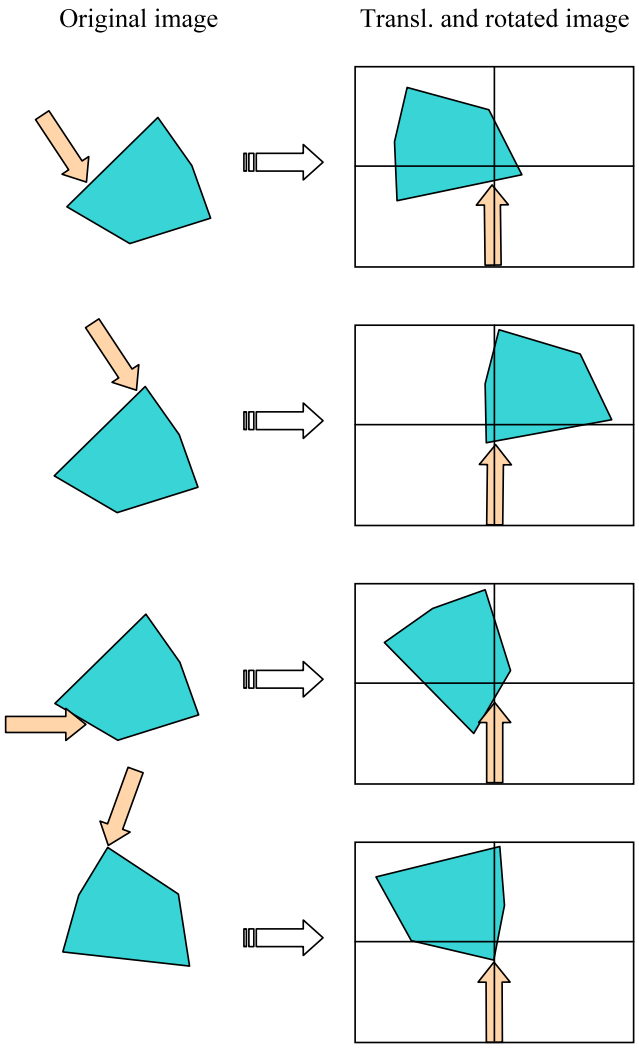


Fig. 5: The binarized object image is rotated and translated to ensure the same pushing position and direction on the retina

## IV. Applying The Pushing Rule

After the learning phase is completed, the robot can generate a poking action to move an object in the desired direction. We call this process the **inverse** pushing problem. The inverse problem deals with where and how the object has to be pushed to achieve motion close to the desired one.

The aim of the robot in this phase is to perform a set of poking actions in order to bring the object to the desired location. Here, a higher-level motion planner should provide the desired movement of the object, whereas the lower-level controller needs to solve the inverse problem. The agent view of the poking scene is shown in Fig. 6.

Note that the robot cannot always achieve the desired velocity due to the physical limitations of the action (this is still a nonprehensile action). In many cases it happens that an object cannot be moved in the desired direction because the pusher slides from the object boundary or it even moves away from the boundary. Such events cannot happen when the object is firmly grasped.

To solve the inverse pushing problem, i.e. to achieve an optimal pusher motion for the desired pushing direction, the agent needs to optimize a criterion function with respect to the point and angle of push, e. g. the weighted square error between the desired motion and the predicted one. Thus we need to find a global minimum of the following function:

$$e = (\mathbf{W}(X_{des} - X_{pred}))^2, \qquad (1)$$

where $X_{des}$ represents the desired motion in all three DOFs and $X_{pred}$ represents the motion of the object which is predicted by the neural network, respectively. $\mathbf{W}$ is the weight specifying the importance of each direction.

To solve the inverse problem by finding the minimum of Eq. (1), we use classical optimization techniques. For an initially selected point and angle of push on the object boundary, the acquired binarized image is transformed as described in the previous section. Based on the generated object image, we predict the movement of the object using the learned neural network. The predicted velocity is compared to the desired one and a new point and angle of push are determined by the optimization method. The process is repeated until an appropriate point and angle on the object's boundary are found. We believe that this process is similar to how humans visualize their action before doing it.

Pushing is a nonprehensile action and it is therefore difficult to ensure that an object will move exactly in the desired direction, both because of the inaccuracies in the learned model and because the optimization process might not find an optimal solution, e.g. because it is stuck in a local minimum. We therefore realized the "pushing to a desired location" behavior as a feedback process, where the robot repeatedly pushes the object until the final position is reached.

The computational complexity of the optimization process is relatively high. However, the pushing point and angle has to be computed only with a frequency of 1 Hz or even less, thus the computational complexity does not play an important role. Note that only the direction of the object's movement is considered in the optimization. The amplitude of the velocity can be modulated by requiring a stronger (or faster) pushing action.

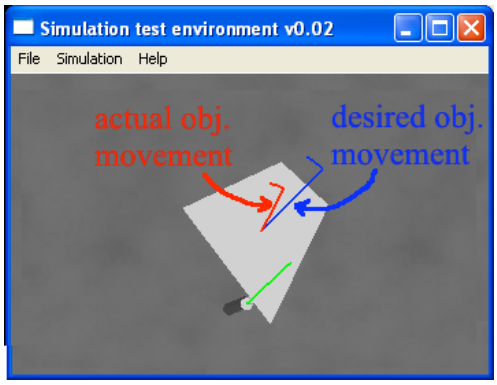


Fig. 6: Agent view of a scene while controlling object movement

## V. PUSHING FOR GRASPING

In our previous work [2], we presented a framework for object grasping and manipulation, which incorporates the described vision system for object localization, a path planner for the generation of collision-free trajectories and an offline grasp analyzer that provides the most feasible grasp configurations for each object. The results provided by the system's components are stored and used by the control system of the robot for the execution of a grasp of a particular object. The central assumption of this framework is the existence of a database with 3-D models of all objects encountered in the robot's workspace and a 3-D model of the robot hand. Grasp hypotheses for each object are generated in simulation using the grasping simulation environment GraspIt! (see [9]) and stored as part of the object representation in the database.

The graspability of the object depends on the support surface, object properties and the available hand. In order to determine whether an object can be grasped with the available robotic hand, we use the simulator to validate the grasp hypotheses associated with the object. A hypothesis is rejected if its execution (in simulation) causes collision with the surface on which the object is placed. If the system generates some hypotheses but none of them leads to a successful grasp, the object must be first relocated before being grasped. One possible way is to push the object to the rim of the table to make it graspable. For this purpose we select one of the hypotheses and compute the pose of the object at the edge of the table so that the object surface associated with the grasp hypothesis lies beyond the rim of the table.

While the grasping part of the system is implemented in a classic way, we are currently working towards a system that can learn to generate a set of grasp hypotheses based on the general properties of the object and test grasp executions.

## VI. RESULTS

Our humanoid robot uses whole body manipulation to manipulate the object. Here, the mobile platform has three DOFs, the hip has 1 DOF and there are additional 7 DOFs in the arms. Depending on the reachability of the desired point, the robot can use the right or the left arm. Technically, to achieve a pushing action with a cylinder shaped pusher, five DOFs are necessary. Three DOFs are needed to control the position of the pusher and two DOF are needed to control the rotations. One DOF of rotation about the cylinder axis is not important and therefore does not need to be considered in the controller. To control the robot we used a velocity based task controller with null space joint limit avoidance.

We performed the learning process on a set of different planar objects shown in Fig. 7. The real robot generated a hand-guided movement around the object (see Fig. 8, green line), which resulted in several randomly distributed pushes of the object. The experiment took about 2 minutes for each object. After the experimentation the data was filtered, velocities and positions of the object were calculated, and learning instances were generated. Among all the measurement samples we used only those that resulted in significant object movements. For all five objects we collected 867 instances,

Fig. 7: A set of objects that were used for learning

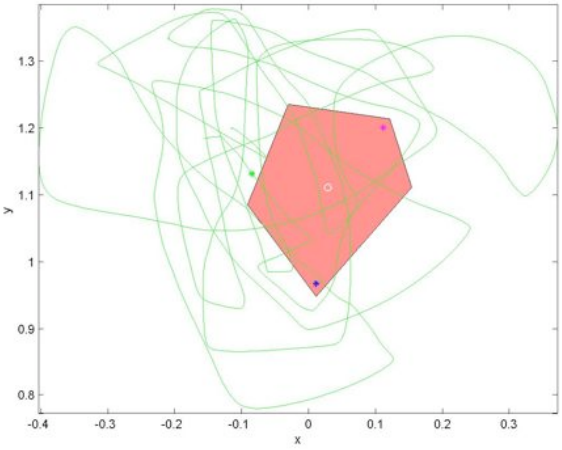


Fig. 8: Green line shows the movement of the pusher during the 2 minute experiment. Red patch represents an object at the initial position.

which were used for training of the neural network. The sample learning instances are shown in Fig. 9.

To validate the proposed approach, we compared the predicted and the actual velocity of the object. We evaluated the differences in the direction of object's movement. Experiments showed that the object movement prediction gets better as the number of instances gets larger. Fig. 10 shows the mean error of all measurements. The x-axis represents the number of instances used for training of the neural network. When the number of instances used for training is very small, e.g. up to 50 instances, the mean error is about 0.5 rad. However, already when using a set of 200 instances, the mean error in prediction gets better and is about 0.2 rad. The errror drops further as more data is collected. Considering unknown friction, low resolution of the image, and the generalization property of the pushing rule, we consider this as a very good result.

The learning process would converge faster and the error would be smaller if we provided more initial knowledge to the system. However, our goal was to develop a system which acquires new knowledge by its own actions, so that a robot could evolve into a more intelligent machine. Therefore, as

little as possible was hardcoded to learn the pushing rule. Note that the primitive poking movements could also be learned.

The acquired pushing rule has been applied to push an object to a graspable position. Fig. 11 (see also the attached video) shows the robot during pushing and grasping. The object was not graspable at its initial location. The robot thus generated a plan to move the object to a position where the robot could grasp it. After a few pushes, the object was brought to a graspable position, where the robot could successfully grasp it.

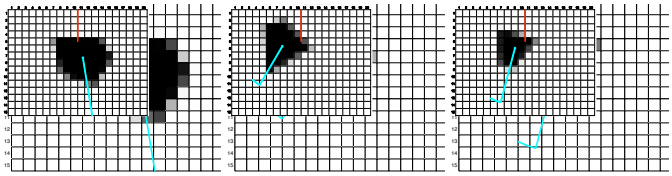

Fig. 9: Sample learning instances. The cyan line indicates the actual object movement, while the red line indicates the pushing direction. The object is shown in black.

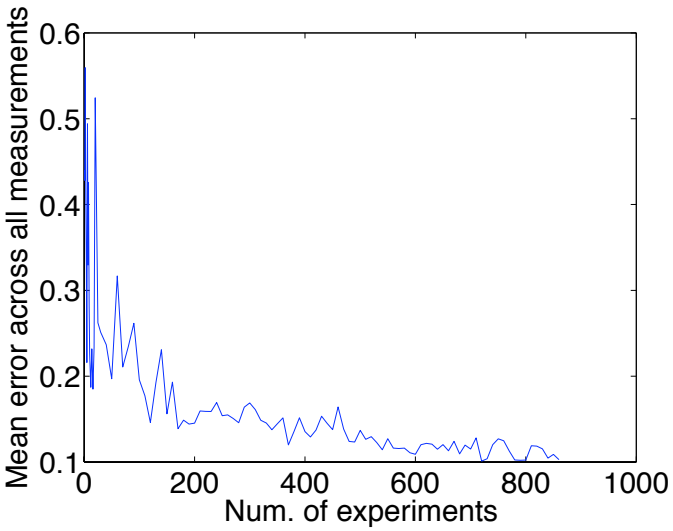


Fig. 10: The error in velocity direction prediction decreases as the number of learning samples increases

## VII. Discussion and Future Work

In summary, we realized the process of associating object-action cause-effects through an explorative, self-emergent process. Such processes are of great importance for the early cognition. No specialized knowledge about the pushing action was provided to the robot. We only provided rules about how to explore the environment and the robot associated the applied actions to object responses independently. We believe that such an explorative learning process, possibly combined with imitation, is one of the keys to natural sensorimotor learning.

While precise learning of pushing actions can take a long time, the agent can learn a rough but reasonable approximation of the behavior already after a few explorative pushes. This initial knowledge can already be used for a rather rough control of the object movement.

While controlling the motion, the robot can update its knowledge base by observing the actual movement of the object. Thus the relationship between the desired and the actual object motion gradually becomes more accurate and the control of the object movement direction improves. Additionally, to make the learning of poking actions more optimal, human instructor can demonstrate the most representative pokes (e.g. perpendicular pokes from a few different sides). Incremental learning combined with imitation is the next important topics of our research.

## References

[1] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An integrated humanoid platform for sensory-motor control.

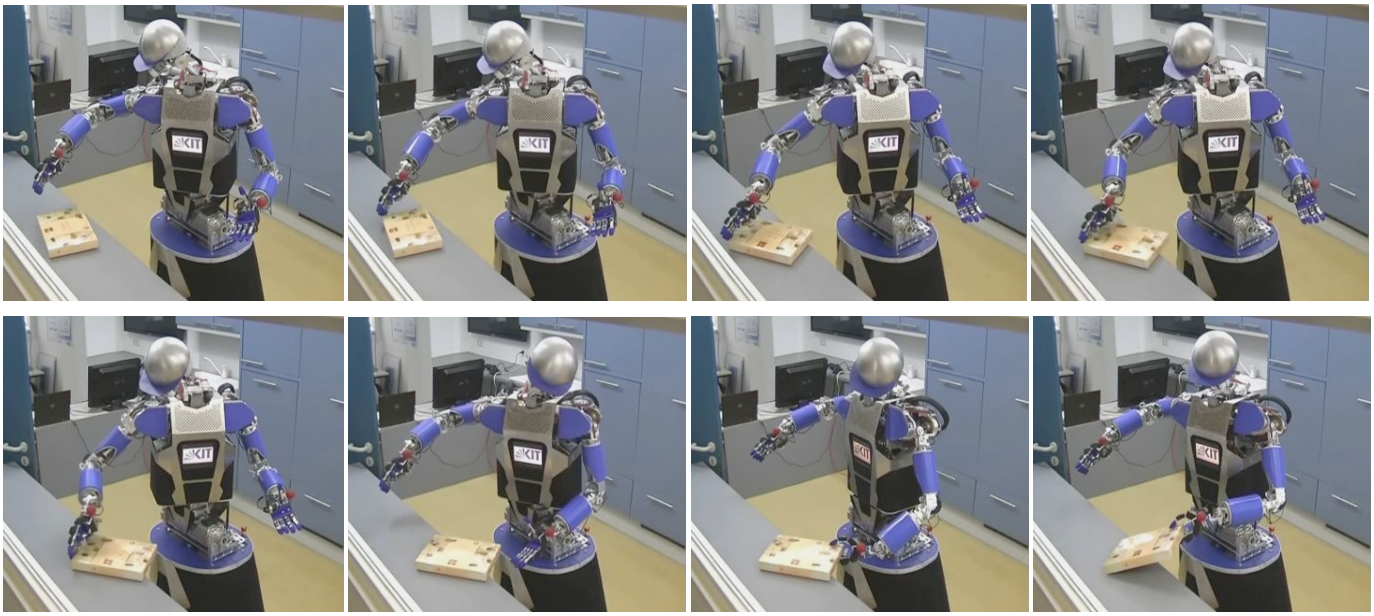

Fig. 11: A sequence of robot pushes that bring the object to a graspable position

In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 169-175, Genoa, Italy, 2006.

[2] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, R. Dillmann, Toward humanoid manipulation in human-centred environments. *Robotics and Autonomous Systems*, 56(1):54-65, 2008.

[3] P. Azad, T. Asfour, and R. Dillmann, Stereo-based 6D object localization for grasping with humanoid robot systems. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 919–924, San Diego, USA, 2007.

[4] P. Fitzpatrick and G. Metta. Grounding vision through experimental manipulation. *Royal Society of London Transactions Series A*, 361(1811):2165–2185, 2003.

[5] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action - initial steps towards artificial cognition. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3140–3145, Taipei, Taiwan, 2003.

[6] Q. Li and S. Payandeh. Manipulation of convex objects via two-agent point-contact push. *The International Journal of Robotics Research*, 26(4):377–403, 2007.

[7] K. M. Lynch and M. T. Mason. Stable pushing: mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[8] G. Metta, G. Sandini, L. Natale, Manzotti R., and F. Panerai. Development in artificial systems. In *Proc. EDEC Symposium at the Int. Conf. on Cognitive Science*, Beijing, China, 2001.

[9] A. T. Miller and P. K. Allen. GraspIt! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine,* 11(4):110-122, 2004.

[10] S. Nishide, T. Ogata, J. Tani, K. Komatani, and H. G. Okuno. Predicting object dynamics from visual images through active sensing experiences. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2501–2506, Rome, Italy, 2007.

[11] D. Omrčen, A. Ude, and A. Kos. Learning primitive actions through object exploration. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 306-311, Daejeon, Korea, 2008.

[12] E. Oztop, T. Chaminade, G. Cheng, and M. Kawato. Imitation bootstrapping: Experiments on a robotic hand. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 189–195, Tsukuba, Japan, 2005.

[13] E. Sahin, M. Cakmak, M. R. Dogar, E. Ugur, and G. Ucoluk. To afford or not to afford: A new formalization of affordances towards affordance-based robot control. *Adaptive Behavior*, 15(4):447-472, 2007.

[14] J. Tani and M. Ito. Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *IEEE Trans. on SMC, Part A*, 33(4):481–488, 2003.

[15] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour and R. Dillmann. Visual servoing for humanoid grasping skills. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots,* pp. 406-412, Daejeon, Korea, 2008.

[16] W. G. Walter. An imitation of life. *Scientific American*, 182(5):42–45, 1950.

[17] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, B. Porr. Cognitive agents — a procedural perspective relying on the predictability of Object-Action-Complexes (OACs), *Robotics and Autonomous Systems*, 57:420-432, 2009.

[18] X. Zhu, H. Ding, and M. Y. Wang, A numerical test for the closure properties of 3-D grasps, *IEEE Trans. Robotics and Automation*, 20(3):543-549, 2004.