

Research Article

Neural Behavior Chain Learning of Mobile Robot Actions

Lejla Banjanovic-Mehmedovic,¹ Dzenisan Golic,²
Fahrudin Mehmedovic,³ and Jasna Havic⁴

¹ Faculty of Electrical Engineering, University of Tuzla, 75000 Tuzla, Bosnia and Herzegovina

² Infonet, 75000 Tuzla, Bosnia and Herzegovina

³ ABB, 75000 Tuzla, Bosnia and Herzegovina

⁴ General Secretariat Council of Ministers of B&H, 71000 Sarajevo, Bosnia and Herzegovina

Correspondence should be addressed to Lejla Banjanovic-Mehmedovic, lejla.mehmedovic@untz.ba

Received 25 April 2012; Accepted 24 September 2012

Academic Editor: R. Saravanan

Copyright © 2012 Lejla Banjanovic-Mehmedovic et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a visual/motor behavior learning approach, based on neural networks. We propose Behavior Chain Model (BCM) in order to create a way of behavior learning. Our behavior-based system evolution task is a mobile robot detecting a target and driving/acting towards it. First, the mapping relations between the image feature domain of the object and the robot action domain are derived. Second, a multilayer neural network for offline learning of the mapping relations is used. This learning structure through neural network training process represents a connection between the visual perceptions and motor sequence of actions in order to grip a target. Last, using behavior learning through a noticed action chain, we can predict mobile robot behavior for a variety of similar tasks in similar environment. Prediction results suggest that the methodology is adequate and could be recognized as an idea for designing different mobile robot behaviour assistance.

1. Introduction

The robotics research covers a wide range of application scenarios, from industrial or service robots up to robotic assistance for disabled or elderly people. Robots in industry, mining, agriculture, space exploration, and health sciences are just a few examples of challenging applications where human attributes such as cognition, perception, and intelligence can play an important role. Inducing perception and cognition, and thence the intelligentsia into robotics machines is the main aim in constructing a robot, able to “think” and operate in uncertain and unstructured conditions.

To successfully realize the instruction capability (e.g., object manipulation, haptically guided teleoperation, robot surgery manipulation, etc.), a robot must extract relevant input/output control signals from the manipulation system task in order to learn the control sequences necessary for task execution [1]. The concept of the visual-motor mapping, which describes the relationship between visually perceived features and the motor signals necessary to act, is

very popular in robotics [2]. There are many visual-motor mappings, defined between cameras and a robot. Since the large variation of visual inputs makes it nearly impossible to represent explicitly the sequence of actions, such knowledge must be obtained from a set of machine learning technique examples [3]. A robot fulfils appropriate purposes using its learning and prediction skills.

Predictive strategy in robotics may be implemented in the following ways [4, 5].

- (i) Model-based reinforcement learning. The environment model is learnt, in addition to reinforcement values.
- (ii) Schema mechanism. A model is represented by rules and learnt bottom-up by generating more specialized rules where necessary.
- (iii) The expectancy model. Reinforcement is only propagated once a desired state is generated by a behavioral module and the propagation is accomplished using dynamic programming techniques, applied to the learnt predictive model and sign list.

- (iv) Anticipatory learning classifier systems. Similar to the schema mechanism and expectancy model, they contain an explicit prediction component. The predictive model consists of a set of rules (classifiers) which are endowed with an “effect” part, to predict the next situation the agent will encounter if the action specified by the rules is executed. These systems are able for generalization over the sensory input.
- (v) Artificial neural network (ANN). The agent controller sends outputs to the actuators, based on sensory inputs. Learning to control the agent consists of learning to associate the good set of outputs to any set of inputs that the agent may experience. The most common way to perform such learning is through using the back-propagation algorithm.

The learning trajectory in the context of programming by demonstration through reinforcement learning is presented under [6]. A visual servobehavior for a real mobile robot, learned through a trial and error method and by using reinforcement learning, is demonstrated under [7].

Different forms of visual-based learning are presented in [8], in each of which the visual perception is tightly coupled with actuator effects, so as to learn an adequate behavior. Learning several behaviors, such as obstacle avoidance or target pursuit through motion sketch, are some of the examples.

The paper [9] deals with the visually guided grasping of unmodeled objects for robots which exhibit an experience-based adaptive behavior. The features are computed from the object image and the kinematics of a hand. Real experimental data on a humanoid robot is employed by a classification strategy, based on the k -nearest neighbor estimation rule, to predict the reliability of a grasp configuration.

In [10], results are presented from experiments with a visually guided four wheeled mobile robot carrying out perceptual judgment based on visual-motor anticipation to exhibit the ability to understand a spatial arrangement of obstacles in its behavioral meaning. The robot learns a forward model by moving randomly within arrangements of obstacles and observing the changing visual input. For perceptual judgment, the robot stands still, observes a single image, and internally simulates the changing images, given a sequence of movement commands (wheel speeds), as specified by a certain movement plan. With this simulation, the robot judges the distance to a frontal obstacle and recognizes, in the arrangement of obstacles, either a dead end or passage. Images are predicted using a set of multilayer perceptrons, where each pixel is computed by a three-layer perceptron.

The perception-action scheme for visually guided manipulation that includes mechanisms for visual predictions and detecting unexpected events by comparisons between the anticipated feedback and incoming feedback is proposed under [11]. Anticipated visual perceptions are based on motor commands and the associated proprioception of the robotic manipulator. If the system prediction is correct, full processing of the sensory input is not needed at this stage. Only when expected perceptions do not match

the incoming sensory data, a full perceptual processing is activated.

Artificial neural networks (ANN), as universal approximators, are capable of modeling complex mappings between the inputs and outputs of a system up to an arbitrary precision. The ALVINN example illustrates the power of standard feed-forward networks, as well as their limitations. The control network solved a difficult pattern recognition task, which required complex image preprocessing, the use of line extraction algorithms, and so forth, if programmed by a human designer. However, due to its use of a feed-forward network, the ALVINN is a reactive system. This means that it has no notion of the temporal aspects of its task and will always react to its visual input in the same fashion, regardless of the current context [12].

The situation, however, changes fundamentally, as soon as the artificial neural networks are used as robot controllers; that is, the network could, by means of the robot body (sensors and effectors), interact with the physical objects in its environment, independent of an observer’s interpretation or mediation. In [13], the recurrent control networks (RNN) have analyzed and shown that they utilize their internal state (i.e., the hidden unit activation values) to carry out behavioral sequences corresponding to particular motion strategies instead of merely reacting to the current input. The RNNs play a central role in such approaches to the study of cognitive representation. This is because they account for the (long-term) representation of learning experience in the connection weights, as well as the (short-term) representation of the controlled agent’s current context or immediate past in the form of internal feedback. But, every task solved by a higher-order RNN could also be solved by some first-order net [14].

The various forms of the neurologically inspired RNN networks were referred to in literature in recent years. For example, a continuous recurrent neural network (CTRNN) was implemented in a humanoid robot for object-manipulation tasks [15]. The proposed network is designed to learn and to regenerate trained visual-proprioceptive sensation sequences, which we assume to correspond to a similar activity in the parietal cortex. Its feature is learning spatial-temporal structures in a continuous time and space domain. The latest biological observations of the brain served as an inspiration for developing Multiple Timescales Recurrent Neural Networks (MTRNN) [16]. The initial testing of the MTRNN model is on iCub humanoid robot, which is able to replicate sequences of manipulating the object.

The most important factor of robot assistance by the behavior sequence learning is the design of interface between neural network and sensors/actuators. Although an ANN could theoretically adapt to different representations of sensor/actuator interfaces, it was necessary to find an interface with low cognitive complexity for the ANN [17].

This paper presents the behavior description, which emphasizes the repetition of numbering in a sequence of actions, noticed as Behavior Chain Learning. In our research, using the characteristics of neural networks, the system learns the necessary set of actions for movement of a mobile

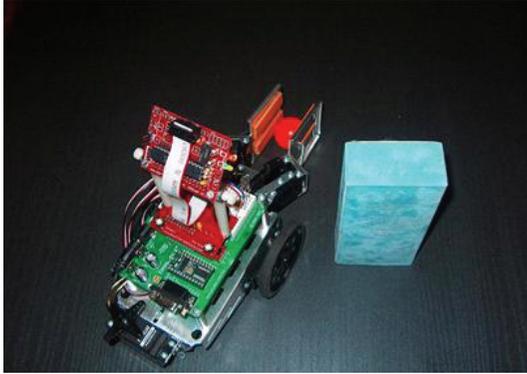


FIGURE 1: Experimental mobile robot with gripped target.

robot in order to access the object in space of observation. On the basis of such trained and tested network, the prediction set of robotic actions for new scenarios of object recognition is constructed. The learned motions can be applied in similar circumstances. Our approach is easily scalable for other applications.

2. Robot Behavior Setting

Our approach focuses on a behavioral system that learns to correlate visual information with motor commands in order to guide a robot towards a target. We chose this task setting, because this approach can be useful for any form of visual/motor coordination, so the task specification can be reformulated as a variety of behavioral responses.

Figure 1 shows an experimental mobile robot platform Boe-Bot by Parallax, with CMUcam1 AppMod vision system for tracking color task.

This camera can detect stationary and moving objects. CMUcam1 is an SX28 microcontroller, interfaced with OV6620 Omnivision CMOS camera on a chip. The mobile robot has a gripper, whose length is 12 cm, and it serves him to grip the ball. The gripper length specifies that it must stop at a distance of 9–13 cm in front of the target.

The robot is in the center of environment and the ball could be at any position in front of robot with respective angle in scope (0–180°). An interaction between the visual perception and motor behavior (a sequence of actions) is obtained through the real-time visual 2D tracking routines.

Figure 2 shows all ball positions, for which the sequences of actions are experimentally determined.

The robot is able to find the ball by turning A_0 from its starting position until it enters robot’s field of view, after which the robot can reliably track red color of the ball while driving toward it. In the environment without obstacles, the robot selects possible actions in sequence due visual tracking: A_0 —turning from its starting position until the object is in robots field of view, A_1 —turning left by 10°, A_2 —turning right by 10°, A_3 —translate straight away, A_4 —stop moving.

The behavior mobile robot scheme consists of the following stages: (1) vision processing involves detecting features, such as color or spatial and temporal intensity gradients;

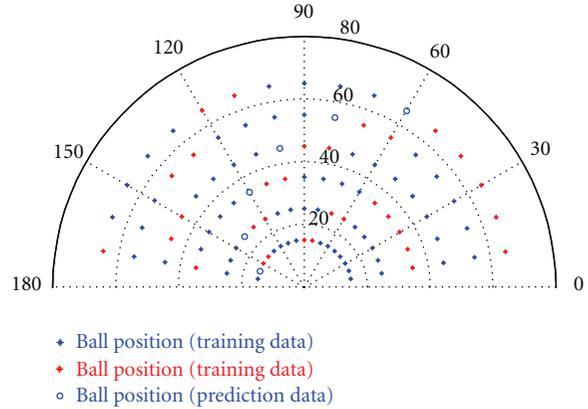


FIGURE 2: Initial target position for experimental setup.

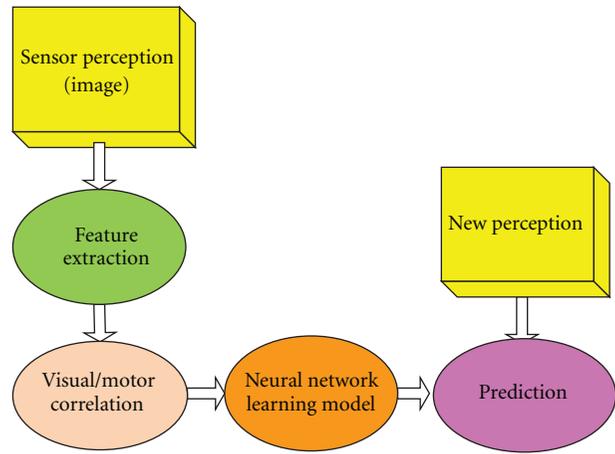


FIGURE 3: Behavior mobile robot scheme.

(2) obtaining the fundamental relationship between visual information and robot motions by correlating visual patterns and motion commands; (3) mobile robot behavior of learning to grip a target; (4) prediction of motor actions for new visual perceptions, Figure 3.

3. Visual Detection of Features

In the first phase, visual detection of features is made on the basis of a specific data set from the camera’s streaming video sent to the mobile robot. Centre of the mass (RCVData(2)), number of pixels within the window (RCVData(8)), and data reliability regarding the color (RCVData(9)) are specific parameters from image.

When the object is positioned in middle of robot’ camera window, the variable called RCVData(2) has value 45. Possible action selections are represented in (Pseudocode 1). For example, if we get RCVData(2) greater than 55 and data reliability regarding the color (RCVData(9)) greater than 25, the object is left of the centre. In that case, the robot needs to turn to the left.

```

Step 1: Start, Move = 0
Step 2: Send command "Track Color" in order to get back
the array of the data
if (RCVData(2) > 55 & RCVData(9) > 25) then
    Move = 1; A1 action (move of the robot to the left)
else if (RCVData(2) < 35 & RCVData(9) > 25) then
    Move = 1; A2 action (move of the robot to the right)
else if (RCVData(8) < 140 & RCVData(9) > 25) then
    Move = 1; A3 action (move of the robot straight ahead)
else if RCVData(8) > 140 then
    Move = 1, A4 action (stop the move)
else if Move = 0 then
    A1 action (move of the robot to the left)
else go to Step 2.

```

PSEUDOCODE 1: Pseudocod of action creating.

4. Behavior Chain Model

The behaviors can be implemented as a Finite State Acceptor (FSA) [18], which describes aggregations and sequences of behaviors. They make explicit the behaviors active at any given time and the transitions between them. FSA is best used to specify complex behavioral control systems where entire sets of primitive behaviors are swapped in and out of execution during the accomplishment of some high-level goal.

We propose Behavior Chain Model (BCM) in order to generalize the form to cope with a variety of similar tasks in similar environment. Each change in action type presents a behavior changing. For example, each time, when the human starts to do something new, it starts to counter (we counter feet in one directions and then change direction or when cooking, we counter spoons, before mixing, etc.). This is inspiration for introducing a formal definition of this behavior model.

BCM consists of: (1) creating of behavior chain from a sequence of actions and (2) extracting physical variables using behavior transform function. We introduce next definitions:

Definition 1. The behavior of system S , which consists of sequence of N behavior actions

$$S_A = (A_i, A_i, A_j, A_j, \dots, A_N) \quad (1)$$

with k_i repetitions of same action type in continuous sequence of N actions that can be described by Behavior Chain C_B , that is, with chain coefficients $k_i, i = 1, \dots, m$:

$$C_B = (k_1, k_2, \dots, k_m). \quad (2)$$

We introduce a formal definition of the behavior transform function, which give us variables from mathematical description of real problem.

Definition 2. The system behavior transform function F_B

$$F_B : C_B \longrightarrow \Psi \quad (3)$$

transforms chain coefficients $k_i, i = 1, \dots, m$ in physical variables $(\xi_1, \dots, \xi_J), J < m$

$$(\xi_1, \dots, \xi_J) = F_B(k_1, k_2, \dots, k_m), \quad (4)$$

where

$$\begin{aligned}
 \xi_1 &= f_B^1(k_1, k_2, \dots, k_m), \\
 \xi_2 &= f_B^2(k_1, k_2, \dots, k_m) \\
 &\dots \\
 \xi_J &= f_B^J(k_1, k_2, \dots, k_m), \\
 J &< N.
 \end{aligned} \quad (5)$$

For our behavior model, we introduce the coefficients, which counts changing actions:

- (i) k_1 = sum of (numbers of repeated turns) in initial position, before (A_0) and after detecting the ball by camera (A_1 or A_2);
- (ii) k_2 = number of repeated translations straight away from the starting point to new point (A_3 in sequence);
- (iii) k_3 = number of repeated new turns (A_1 or A_2 in sequence);
- (iv) k_4 = number of repeated new translations straight away (A_3 in sequence).

In case of more changing of action repetitions (longer target distances or environment with obstacles), we can introduce more coefficients k_1, k_2, \dots, k_m , with which we can describe a system's behavior.

One example of creating of behavior chain is presented in Figure 4.

Table 1 contains only one part of experimental results, with the coordinates of the ball position, number of turnings from its starting position until target is found in its field of view and a sequence of actions, which the mobile robot must take to target. Consider first example of action's sequence for

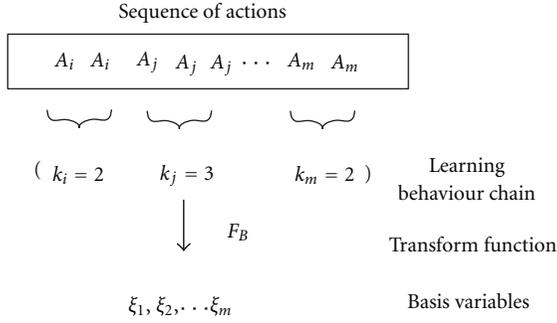


FIGURE 4: Creating of behavior chain.

TABLE 1: Examples of correlating behavior sequence of actions.

Position (ρ, θ)	Turning A_0	Sequence of actions
(55, 170)	6	$A_1 A_3 A_1 A_3 A_3 A_3 A_3 A_3 A_3$
(35, 160)	5	$A_1 A_1 A_3 A_3 A_3$
(15, 150)	3	$A_1 A_1$
(25, 140)	3	$A_1 A_3 A_1 A_3$
(45, 130)	2	$A_1 A_3 A_3 A_3 A_3 A_3$
(45, 120)	1	$A_1 A_1 A_3 A_3 A_3 A_3 A_3 A_1$
(65, 110)	1	$A_1 A_3 A_3 A_3 A_3 A_3 A_3 A_3 A_1 A_3 A_3$
(25, 100)	0	$A_1 A_3 A_3$
(65, 90)	0	$A_3 A_3 A_3 A_3 A_3 A_3 A_3 A_3 A_3$
(55, 80)	0	$A_2 A_3 A_3 A_3 A_3 A_3 A_3$
(35, 70)	0	$A_2 A_2 A_3 A_3 A_3$
(45, 60)	1	$A_2 A_2 A_3 A_3 A_3 A_3$
(65, 50)	3	$A_2 A_3 A_3 A_3 A_3 A_3 A_3 A_2 A_3 A_3$
(25, 40)	2	$A_2 A_2 A_3$
(15, 30)	3	$A_2 A_3$
(45, 20)	6	$A_2 A_2 A_3 A_3 A_3 A_3 A_2 A_3$
(65, 10)	6	$A_2 A_2 A_3 A_3 A_3 A_3 A_3 A_2 A_3 A_3 A_3$

ball position $(55 \text{ cm}, 170^\circ)$ in polar coordinate system. First, the mobile robot made $A_0 = 6$ turning by 10° to the left until it detects the ball, then robot rotates again by 10° to the left, then goes straight away (action A_3), then turns again by 10° left (action A_1) and goes straight away (six repetition of action A_3).

A turn to the left has a positive value, while a turn to the right has a negative value in an action matrix. In the this example of ball position $(55, 170^\circ)$, a sequence of actions is presented with the following Behavior Chain:

$$C_B(50, 170) = (k_1, k_2, k_3, k_4) = (7, 1, 1, 6). \quad (6)$$

In the example of ball position $(65, 50^\circ)$, a sequence of actions is presented with the following Behavior Chain:

$$C_B(65, 50) = (k_1, k_2, k_3, k_4) = (-4, 6, -1, 2). \quad (7)$$

The examples of mobile robot's behaviors for some ball positions $((65 \text{ cm}, 50^\circ), (35 \text{ cm}, 70^\circ), \text{ and } (45 \text{ cm}, 120^\circ))$ are presented on Figure 5.

For example, the mobile robot position in polar system is presented with (d, α) . We need transforming process which

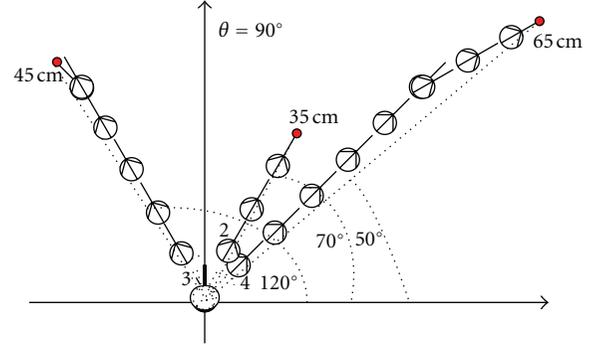


FIGURE 5: Experimental robot scenario.

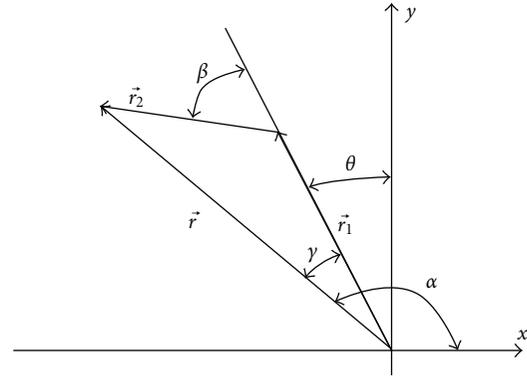


FIGURE 6: Mobile robot positioning.

gives us variables from the mathematical description of a real problem. This transforming process presents second phase of LBCM model.

5. Mathematical Model of Mobile Robot Positioning

In order to calculate the positions (d, α) where a mobile robot comes, we use the mathematical model of mobile robot positioning presented below. In an experiment, we got the sequence of robot motor actions, for given object positions (Figure 6).

In our approach, one turn is 10 degrees $(\theta = 10k_1, \beta = 10k_3)$. For those object positions in environment, which a mobile robot need to recognize visually, the vector distance of the mobile robot can be approximated by a superposition of two vectors \vec{r}_1 and \vec{r}_2 , whose intensities are determined by the expressions $|\vec{r}_1| = 6k_2, |\vec{r}_2| = 6k_4$ (translation step is 6 cm):

$$d = \vec{r} = \sqrt{|\vec{r}_1|^2 + |\vec{r}_2|^2 - 2|\vec{r}_1||\vec{r}_2|\cos(180 - \beta)}, \quad (8)$$

$$d = 6\sqrt{k_2^2 + k_4^2 + 2k_2k_4\cos 10k_3}.$$

Angle α is the sum of the angles of the initial turns, the turns after the initial displacement and 90° , that is,

$$\alpha = 90 + \theta + \gamma. \quad (9)$$

From Figure 5, the following relationship is valid:

$$\frac{|\vec{r}_2|}{\sin \gamma} = \frac{d}{\sin(180 - \beta)} \quad (10)$$

or

$$\sin \gamma = \frac{|\vec{r}_2|}{d} \sin(180 - \beta) = \frac{|\vec{r}_2|}{d} \sin \beta = \frac{6k_4}{d} \sin 10k_3. \quad (11)$$

Finally, angle α can be expressed with:

$$\alpha = 90 + 10k_1 + \arcsin\left(\frac{6k_4}{d} \sin 10k_3\right). \quad (12)$$

Using behavior transform function

$$(\xi_1, \xi_2) = (d, \alpha) = F_B(k_1, k_2, k_3, k_4), \quad J = 2, \quad (13)$$

where

$$d = f_B^1(k_2, k_3, k_4) = 6\sqrt{k_2^2 + k_4^2 + 2k_2k_4 \cos 10k_3},$$

$$\alpha = f_B^2(k_1, k_3, k_4) = 90 + 10k_1 + \arcsin\left(\frac{6k_4}{d} \sin 10k_3\right). \quad (14)$$

Table 2 shows some examples of ball positions, for which the mobile robot positions (d, α) are calculated from above mathematical model.

The third phase in our approach is the learning of a sequence of actions, which establishes an appropriate correspondence between the perceived states and actions. The calculated mobile robot positions, based on coefficients extracted from the experimental patterns, will be compared with mobile robot positions (d_{NN}, α_{NN}) , based on a neural learned coefficient, serving for prediction purposes.

6. Robot Behavior Learning

Based on artificial cognition, a robot system can simulate goal-directed human behavior and significantly increase the conformity with human expectations [19]. Our approach stresses the creation of behavior chain from a sequence of actions. To achieve visually guided pointing, our task learns mapping from ball coordinates (ρ, θ) to the mobile robot motor commands, presented with Behavior Chain $C_B = (k_1, k_2, k_3, k_4)$, necessary to achieve these locations. To simplify the dimensionality problems, mobile robot positions are specified as a linear combination of vector primitives, formed from S_A parameters. This form of mapping is exemplified by the neural learning, which leads a robot's prediction ability. After some experiments with different neural network's structures, trained with backpropagation learning algorithm, we used the feed-forward multilayer network with Levenberg-Marquardt (LM) learning algorithm.

A set of input data consists of $N = 102$ target samples (ρ, θ) and a sequence of actions from its starting position to a point from which it is possible to pick up the ball. The collected data are divided into three subsamples, train

TABLE 2: Description examples of behavior sequence of actions.

(ρ, θ)	k_1	k_2	k_3	k_4	d	α
(55, 170)	7	1	1	6	42	169
(35, 160)	7	3	0	0	18	160
(15, 150)	5	0	0	0	0	140
(25, 140)	4	1	1	1	12	135
(45, 130)	4	5	0	0	30	130
(45, 120)	3	5	1	0	30	120
(65, 110)	2	7	1	2	54	112
(25, 100)	1	2	0	0	12	100
(65, 90)	0	9	0	0	54	90
(55, 80)	-1	6	0	0	36	80
(35, 70)	-2	3	0	0	18	70
(45, 60)	-3	4	0	0	24	60
(65, 50)	-4	6	-1	2	47	45
(25, 40)	-5	1	0	0	6	40
(15, 30)	-6	1	0	0	6	30
(45, 20)	-8	4	-1	1	30	8
(65, 10)	-8	5	-1	3	48	6

TABLE 3: Results of error by different neural network training.

L	N_n	M_{ne}	L_r	Error test (MSE)		Error predict (MSE)		RMS
				E_d	E_α	EP_d	EP_α	
1	10	500	0.001	7.74	7.77	3.29	4.55	4.00
1	10	1000	0.001	5.18	8.48	2.45	4.44	3.76
1	20	500	0.01	5.57	10.5	7.13	3.11	4.07
1	30	500	0.001	38.13	13.6	29.7	8.88	7.30
2	30	500	0.001	5.99	8.65	4.20	14.43	3.89
2	20	500	0.01	5.01	7.77	2.43	4.63	3.63
2	20	1000	0.01	7.57	8.07	3.57	4.43	4.02
2	30	500	0.1	8.52	8.93	4.79	4.30	4.25
2	20	1500	0.1	8.86	736	3.04	4.43	3.80
2	30	1500	0.01	17.62	10.6	3.06	4.30	5.40

samples (63%), test samples (31%), and prediction samples (6%). The feed-forward multilayer network is used for training with one and two hidden layers, with tansig or purelin activation functions and with a total number of neurons in the hidden layers (10, 20, or 30). The output layer of neural network has 4 neurons, which present 4 coefficients of S_A used for calculating (d_{NN}, α_{NN}) . A few results are listed in Table 3.

During neural network training, we changed the number of hidden layers (L), the number of neurons in hidden layers (N_n), the maximum number of epochs (M_{ne}), type of activation function in hidden and output layers, and the learning rate (L_r). For each neural network configuration, we calculated MSE (mean square error) between (d, α) and values from neural network learning (d_{NN}, α_{NN}) for both test set and for prediction set, as well as root mean square error (RMS) for the test set. According to the results from Table 3, the best network configuration with a minimum value of root mean square error $RMS = 3.63$ was selected.

TABLE 4: Selected set of mobile robot positions from experiments and predicted values from neural network learning process.

ρ	15	25	35	45	55	65
θ	160	140	120	100	80	60
Real values of coefficients k_i						
k_1	6	4	3	1	-1	-3
k_2	0	1	3	5	6	8
k_3	0	1	0	0	0	0
k_4	0	1	0	0	0	0
Real values of (d, α)						
d	0	12	18	30	36	48
α	150	135	120	100	80	60
Values of coefficients k_i gained from neural network learning process						
k_1	6.66	5.09	3.05	1.00	-0.68	-2.70
k_2	-0.06	2.10	3.34	4.68	4.92	5.71
k_3	-0.14	0.09	0.02	0.04	-0.22	-0.32
k_4	0.00	-0.34	-0.07	0.48	1.55	2.14
d_{NN} and α_{NN} gained from neural network learning process						
d_{NN}	0.37	10.57	19.65	30.99	38.83	47.09
α_{NN}	149.95	129.83	119.99	100.04	79.46	59.12

We got a minimum root mean square error (RMS) for the three-layer neural network with 20 neurons in hidden layers, learning coefficient 0.01, the activation functions of neurons in hidden and output layers were tansig and purelin, and the training was conducted through 500 epochs.

7. Results of Predictive Behavior

The data collected during the experiments are comprised of a large amount of information. Several analyses can be carried out over this data, especially those regarding the appropriateness and usefulness of the different features. However, we are more interested in the predictive capabilities that can be inferred from these data and the methods that can make the best use of it.

In the fourth phase of our approach, we present prediction results of selected input data using neural network configuration with minimal value of the error (RMS). For each target position from the prediction set, we calculated (d, α) from experiments using mathematical model and compared them with (d_{NN}, α_{NN}) , gained from the neural network learning process (Table 4).

Using Table 4, we compare graphically mobile robot's position obtained through an experiment, using mathematical model and from neural network learning.

Figures 7 and 8 presents the different sequences of mobile robot actions towards the target. The path (presented with a red line) is a mobile robot path from mathematical model and the path (presented with blue line) is a mobile robot path, gained from parameters from neural network learning. The difference between the distance to the ball ρ and the distance d (where robot came) exists, because the length of

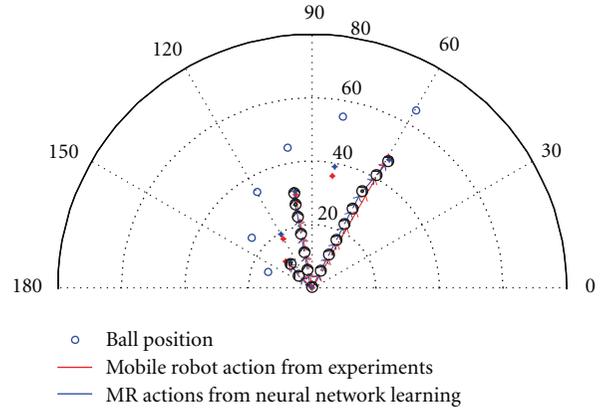


FIGURE 7: Experimental mobile robot with tracked target on position (65, 60), (45, 100), and (25, 140).

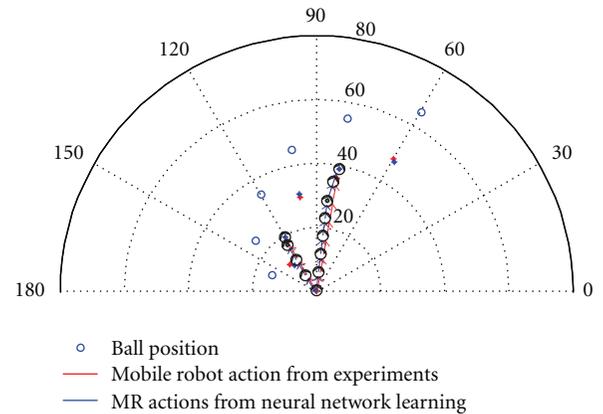


FIGURE 8: Experimental mobile robot with tracked target on position (55, 80) and (35, 120).

gripper specifies that robot must stop at 9–13 cm distance of in front of the target.

For example, the target is located at position (45, 100), Figure 7. From its initial position, the mobile robot turned for angle of 10 degrees to the left, that is, $k_1 = 1$. Then, 4 translation actions were made (each by 6 cm), and 4 cm displacement. After that, a turn of $10k_3$ was made (it is an angle of 0.4 degrees), followed by 2.9 cm displacement, and it came in a position to grip a ball.

For example, the target is located at position (55, 80), Figure 8. From its initial position, the mobile robot turned for angle of 6.8 degrees to the right, that is, $k_1 = -0.68$. Then, 4 translation actions (each by 6 cm) were made, including 5.5 cm displacement. After that, a turn of $10k_3$ was made (it is an angle of 2.2 degrees to the right), then 6 cm and 3.3 cm displacements, and then it came in a position to grip a ball.

8. Conclusion

We proposed a methodology which tries to emulate the human action of vision in a general-conceptual way that includes: primary recognition of object in environment,

visual-based mobile robot behavior learning, and prediction of new situations. This form of robot learning does not need the knowledge about the environment or kinematics/dynamics of the robot itself, because such knowledge is implicitly embodied in structure of the learning process.

This approach is very flexible and can be applied to a wide variety of problems, because behavior description is “elastic” enough to adapt to various situations. In order to apply our approach to any kind of tasks, we have to solve two important problems. One is how to construct the behavior description of actions and other is how to generalize the learned form to cope with a variety of similar tasks in similar environment.

Although the neural network could theoretically adapt to different representations of sensor/actuator interfaces, it was necessary to find an interface with low cognitive complexity for neural network, which, in our case, was a simple polar representation of the sensors and intended robot movements through “*k*” parameters. Furthermore, we analyzed the influence of using different sizes and parameters of a multilayer perceptron. While the number of neurons had the smaller effect to the performance, the complete type of representation affected the neural network results.

We have implemented a prediction approach that uses such features to produce reliable output. Feature space data were obtained from real experiments with a mobile robot with camera and gripper. The obtained prediction results are satisfactory enough to suggest that the methodology is adequate and that further progress should be made in this direction. In future work, more involved strategies may be developed by expanding a set of new manipulation tasks, independent learning, adaptation in space, or involving multiagent behaviour learning.

References

- [1] A. M. Howard and C. H. Park, “Haptically guided teleoperation for learning manipulation tasks,” in *Robotics: Science and Systems: Workshop on Robot Manipulation*, Atlanta, Ga, USA, June 2007.
- [2] G. Taylor and L. Kleeman, *Visual Perception and Robotic Manipulation: 3D Object Recognition, Tracking and Hand-Eye Coordination*, Springer, 2006.
- [3] Y. Wu, *Vision and learning for intelligent Human-Computer interaction [Ph.D. thesis]*, University of Illinois, 2001.
- [4] M. V. Butz, O. Sigaud, and P. Gerard, “Internal models and anticipations in adaptive learning systems,” in *Proceedings of the 1st Workshop on Adaptive Behavior in Anticipatory Learning Systems (ABiALS '06)*, 2006.
- [5] A. Barrera, “Anticipatory mechanisms of human sensory-motor coordination inspire control of adaptive robots: a brief review,” in *Robot Learning*, S. Jabin, Ed., InTech, 2010.
- [6] L. Rozo, P. Jimenez, and C. Torras, “Robot learning of container-emptying skills through haptic demonstration,” Tech. Rep. IRI-TR-09-05, Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 2009.
- [7] C. Gaskett, L. Fletcher, and A. Zelinsky, “Reinforcement learning for visual servoing of a mobile robot,” in *Proceedings of the Australian Conference on Robotics and Automation (ACRA '00)*, Melbourne, Australia, August 2000.
- [8] M. Asada, T. Nakamura, and K. Hosoda, “Behavior acquisition via visual-based robot learning,” in *Proceedings of the 7th International Symposium on Robotic Research*, 1996.
- [9] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil, “Experimental prediction of the performance of grasp tasks from visual features,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3423–3428, Las Vegas, Nev, USA, October 2003.
- [10] H. Hoffmann, “Perception through visuomotor anticipation in a mobile robot,” *Neural Networks*, vol. 20, no. 1, pp. 22–33, 2007.
- [11] E. Datteri, G. Teti, C. Laschi, G. Tamburrini, P. Dario, and E. Guglielmelli, “Expected perception: an anticipation-based perception-action scheme in robots,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 934–939, October 2003.
- [12] D. A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Kluwer, Dordrecht, The Netherlands, 1993.
- [13] L. A. Meeden, G. McGraw, and D. Blank, “Emergence of control and planning in an autonomous vehicle,” in *Proceedings of the 50th Annual Meeting of the Cognitive Science Society*, p. 735, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1993.
- [14] T. Ziemke, “Remembering how to behave: recurrent neural networks for adaptive robot behavior,” in *Recurrent Neural Networks, Design and Applications*, L. R. Medsker and L. C. Jain, Eds., CRC Press, 2001.
- [15] J. Tani, R. Nishimoto, J. Namikawa, and M. Ito, “Code-developmental learning between human and humanoid robot using a dynamic neural-network model,” *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 38, no. 1, pp. 43–59, 2008.
- [16] M. Peniak, D. Marocco, J. Taniy, Y. Yamashitay, K. Fischer, and A. Cangelosi, “Multiple time scales recurrent neural network for complex action acquisition,” in *Proceedings of the International Joint Conference on Development and Learning (ICDL) and Epigenetic Robotics (ICDL-EPIROB '11)*, Frankfurt, Germany, August 2011.
- [17] I. Fehervari and W. Elmenreich, “Evolving neural network controllers for a team of self-organizing robots,” *Journal of Robotics Volume*, vol. 2010, Article ID 841286, 10 pages, 2010.
- [18] R. C. Arkin, *Behavior-Based Robotics*, The MIT Press, Cambridge, Mass, USA, 1998.
- [19] M. Mayer, B. Odenthal, and M. Grandt, “Task-oriented process planning for cognitive production systems using MTM,” in *Proceedings of the 2nd International Conference on Applied Human Factors and Ergonomic*, USA Pub, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

