Preprint version of Special Issue on Advances in Developmental Robotics,
Paladyn. Journal of Behavioral Robotics, 2013

A Predictive Network Architecture for a Robust and Smooth Robot Docking Behavior

J. Zhong, C. Weber, S. Wermter

April 24, 2013

Abstract

Robots and living beings exhibit latencies in their sensorimotor processing due to mechanical and electronic or neural processing delays. A reaction typically occurs to input stimuli of the past. This is critical not only when the environment changes (e.g. moving objects) but also when the agent itself moves. An agent that does not predict while moving may need to remain static between sensory input acquisition and output response to guarantee that the response is appropriate to the percept. We propose a biologically-inspired learning model of predictive sensorimotor integration to compensate for this latency. In this model, an Elman network is developed for sensory prediction and sensory filtering; a Continuous Actor-Critic Learning Automaton (CACLA) is trained for continuous action generation. For a robot docking experiment, this architecture improves the smoothness of the robot's sensory input and therefore results in a faster and more accurate continuous approach behavior.

1 Predictive Visual Processing

In the sensorimotor cycle of a robot, there usually exists a temporal delay mainly contributed by the processing time of sensors, transmission time of signals and mechanical latency. For example, because few object recognition programs can recognize the identity of human faces from visual inputs in real time, the running speed of human following behavior of robots based on object recognition should be slower than a normal walking speed of human beings, if it keeps searching a person in image sequences in a short time-scale. A simple predictive mechanism,

The original publication is available at www.springerlink.com

such as a Kalman Filter, can solve this problem by predicting the movement of a person as soon as he/she has been identified (e.g. [1]). However, since a Kalman Filter is based on a linearity assumption, it does not consider more complicated movements with e.g. non-linear influence from context. Such problems may be tackled by neural networks, which can learn to predict percepts in a general dynamic environment. Sensory prediction is of great benefit to dynamic robot behaviors such as obstacle avoidance, visually guided navigation, reaching, visual search, and rapid decision-making under uncertainty, since these kinds of behavior highly rely on current sensory information. In these scenarios with non-linear dynamics, a developmental sensory prediction is needed to learn to compensate for the latency of the sensorimotor cycle.

A second reason for employing predictive mechanisms is that the sensory inputs are often noisy and inaccurate in a real environment, which may lead to failure of robot behaviors. In that case, a predictive sensory module can compare its prediction based on previous short-term sensory percepts to the current sensory value. A noisy sensor value can thereby be identified and an action adjustment executed. A severe case may be due to sensor failure caused by hardware problems or a change of environment (e.g. lighting conditions in cameras). In such cases, an embodied predictive sensory module can act as a filter to recursively estimate the incoming percepts.

Mechanisms of prediction have also been found in human perception. For example, in visual motion perception humans keep track of a moving object by observing saliency (i.e. the most occurring features) from the visual processing in the thalamus and visual cortex [2]. This object feature selection, on the other hand, also provides a cue in the early visual cortex to predict upcoming sensory data via recurrent and top-down connections based on the prior noisy sensory information [3, 4]. For instance, the flash-lag effect [5] shows that a perceptual prediction may exist based on visual motion cues and it extrapolates the forthcoming visual information [6]. In a similar manner as for motion, it is proposed that specialized cue detectors, such as colors or shapes, also predictively code the evidence of preferred features according to an internal model [7]. Research shows that prediction in the visual cortex is not hard-wired, but rather a process learned by a flexible system whose contingencies are adapted in different environments [8, 9]. These biological findings suggest that an adaptable predictive sensory module is beneficial for faster and more natural robot behaviors.

From a neural modeling point of view, to realize a sensory prediction function and to embody it in a robot, recurrent connectivity is one option due to its capability to store previous dynamical activities and to represent a short-term memory

of previously perceived visual information. Recurrent connections – horizontal excitatory and inhibitory connections in neuroanatomy – are found throughout visual areas [10, 11] and they account the neural activity changes during responses to object movements [12]. Furthermore, functions of de-noising and filtering during the unfolding of a spatiotemporal sequence can be performed by the same recurrent connectivity. The fluctuation caused by the sensory error can be efficiently smoothed by recurrent loops, which is consistent with neuroanatomical findings that inhibitory feedback connections suppress neural population-rate fluctuations [13]. These two properties of recurrent connections – prediction and filtering – fulfill our requirements for building up a sensory module that supports robust robot behavior.

2 Sensory Prediction

Prediction within an autonomous cognitive robot can happen on various levels, but in this paper we only consider predictive sensory models, i.e. a system predicting sensory signals given the current and previous sensory states. Since prediction of the complete raw sensory percept (e.g. all pixels of the camera image) is not desirable and would be very difficult for an autonomous robot, it is advisable to predict only few features extracted from the sensory percepts [14, 15] as human perception does. This can be implemented as a learnt non-linear mapping of sensory representations to predict the forthcoming sensory flow. In previous research, prediction helped to e.g. determine object affordance [16], to make a robot avoid moving humans [17] or to steer a mobile robot [18, 19].

We have argued previously that any cortical area should compensate for its own processing delay via prediction [20]. Such local predictions may be easier to implement than prediction of the entire system response, and enable the mixed hierarchical and parallel processing in the (visual) cortex, including shortcut connections, while keeping the representations in all areas temporally aligned. Furthermore, prediction of restricted sensory percepts (instead of prediction of the next action) may be generalizable to many contexts and actions, to allow for latent learning, in which knowledge for specific tasks is acquired while other tasks are being performed.

A well-known method for tracking and prediction is the Kalman filter [21]. It is built on linear operators and models a Markov chain with Gaussian noise with the assumption of a linear dynamic system. Other improved/adaptive Kalman filters have been proposed [22, 23]. Neural networks can learn universal function

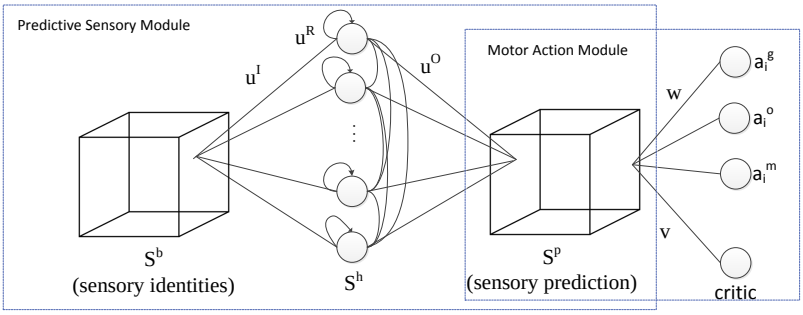


Figure 1: Overall architecture combining sensory prediction and sensorimotor action

approximation and thereby optimally predict non-linear data [24, 25, 26, 27]. For example, a simple feed-forward network approximates a time step occurring in a hidden Markov model (HMM) to estimate an agent's position along a motion sequence [28]. Furthermore, a neural network with recurrent connections is able to predict the movement trajectory recursively, since it also represents the recent data in its short-term memory, it exhibits a smooth and stable neural dynamics. For instance, in a computational model of the visual system, recurrent connections within a "where" stream encode movement and predict position, to complement a separate "what" stream [20]. This work shows that motion perception in early visual processing can be efficiently modeled by recurrent neural connections, and therefore it should be also practical to apply them in our predictive sensory module.

3 System Design

Visual prediction in a robot involves the design of an internal model representation of the world to estimate the future sensory information of the percept. In our proposed design, we also emphasize that the motor action of the robot should be executed based on the predicted sensory percepts, reacting to the forthcoming sensory data, and thus faster and more smooth robot behaviors are produced. Prediction of motion generally includes both active and passive movement, i.e. movement of sensory events generated by action of the agent itself and those generated by the movement of the observed objects [29]. In our experiments, we concentrate on the prediction of active movement, inferred from the perceived visual motion of a fixed landmark. However, since this prediction does not use

additional input from any behaviors or motor commands, this architecture can predict sensory percepts no matter whether they are caused by active or passive movement. In statistical notation, we would regard the prediction process as a hidden Markov model (HMM) rather than a partially observable Markov decision process (POMDP), since the action is unknown.

In previous work [30], we have realized a goal-directed behavior based on reinforcement learning by a NAO humanoid robot [31]. The resulting behavior, however, did not look natural compared to the walking patterns of humans. In this paper therefore, we make several new contributions. First, instead of a discrete representation of state- and action spaces, we use continuous representations in both, which results in more sophisticated walking behaviors. This is made possible by the Continuous Actor-Critic Learning Automaton (CACLA) [32]. The continuous action space facilitates generalization of learned actions to unlearned regions in the state space, which could help speed up learning and optimize the action selection. This reinforcement learning algorithm is similar to motor adaptation according to error-prediction in biology [33, 34].

Another property of our previous work was the robot's slowness. Instead of using prediction, the visual percept was retrieved after a short waiting time in which the robot stood still to obtain a clear camera image. A behavior that now takes little more than a second with our proposed model (see results below) took roughly half a minute due to the still standing periods [30].

Corresponding to the integration of predictive visual information and smooth motor action, two modules, the sensory and motor modules, are incorporated in this architecture (Fig. 1). Within the sensory module, an Elman network is applied to predict the upcoming sensory information, while the motor-action module uses a network trained by CACLA [32]. Integration of these two modules drives a NAO robot to approach a trained position with smooth and continuous behavior in an autonomous manner.

The NAO robot hardware and control commands allow continuous walking without stopping, so that the robot can change the parameters of walking, i.e. speed, walking direction and torso orientation, at any given time while walking. Furthermore, the light weight of the robot guarantees that it reacts fairly fast. However, as a disadvantage also found in other robots, a sensorimotor cycle latency also exists in NAOs. To quantify this latency, an experiment was conducted by capturing two time-stamps between issuing a movement command and perceiving this movement from the robot camera. Based on ten recordings in each case, the delay of NAO's sensorimotor cycle is around $0.8s$ with a wireless connection and $0.36s$ with a LAN cable between the control PC and the robot (Tab. 1).

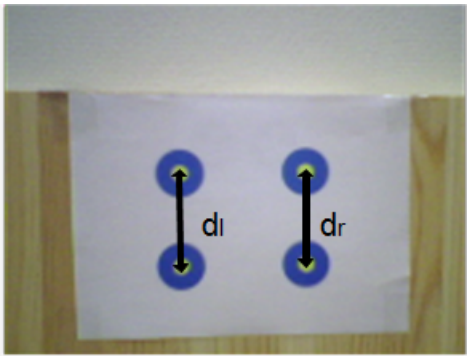


Figure 2: Landmark from the view of NAO camera

This long delay was observed despite the fact that the used QVGA resolution allowed up to 30 frames per second in LAN the modus. This further motivates to realize sensory prediction to compensate for this delay.

3.1 Landmark based Approaching

The NAO robot comes with circular NAOMarks landmarks and a built-in detection routine. However, this closed-source software leads to over-estimation of the landmark size if the images are blurry, which leads to wrongly estimated poses. Therefore, we designed an own landmark to identify the position of the approach target. It consists of four circles, each including a large blue ring with a smaller yellow circle inside (Fig. 2). We detect the positions of the circle centers in the robot's visual field using 2D Gaussian Fourier filters in RGB channels. The color combination of the landmark is different from the wooden docking station to be easily distinguishable. Then, Hough circle detection finds circles within a certain radius range [35]. Our routine is faster than the previous NAOMarks, the total pre-processing and searching time of the landmark data is less than $0.01s$.

Then the position and orientation of the robot with respect to the landmark can

Connection	Mean	Variance
LAN	366.0	84.5
WIFI	797.7	187.1

Table 1: Time delay in the camera-arm cycle of NAO (in milliseconds)

be defined by the following three values:

$$I_1 = \frac{d_l + d_r}{2}\,, \quad I_2 = \frac{d_l - d_r}{I_1}\,, \quad I_3 = \sum_x pix_x \tag{1}$$

where the measurements d_l and d_r are defined according to Fig. 2. pix_x is the summation of the x coordinate of all the four circle centers within the robot visual field. Referring to the overall installation of the shelf in Fig. 3, the first identity I_1 correlates with the proximity between the robot and the landmark. The second identity I_2 correlates with the angle of the robot's position w.r.t. the landmark, while the third identity I_3 informs about the robot's orientation w.r.t. the direction to the landmark.

The three values of Eq. 1 contain all position/orientation information relevant for the robot's approach behaviour. Neurons in the input layer are arranged as a three-dimensional cube, in which they are activated with a Gaussian activation blob that is centered around (s_1^c, s_2^c, s_3^c) defined as:

$$s_n^c = \frac{I_n - I_n^{min}}{I_n^{max} - I_n^{min}} \times N_n \tag{2}$$

where n $(n = 1, 2, 3)$ is the n-th dimension of the cube, corresponding to the n-th identity of Eq. 1. I_n^{min} and I_n^{max} are the minimum and maximum value of the identities data, respectively. N_n is the number of neurons in the n-th dimension. The activation of neighboring neurons $s^b(x_1, x_2, x_3)$ is distributed according to a Gaussian:

$$s_i^b(x_1, x_2, x_3) \sim \mathcal{N}(s_n^c, \delta_m) \tag{3}$$

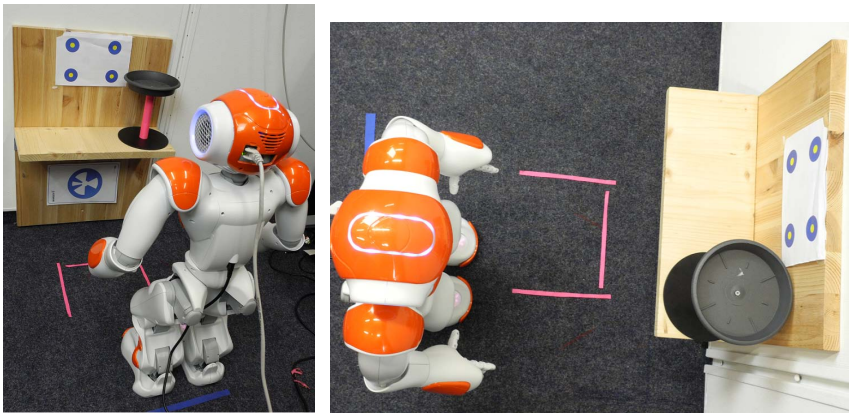
These values define the activation on the sensory input layer.

3.2 Visual Prediction via Lateral Connections

The predictive module consists of a three-layer Elman network. Inputs to this network are the three observed identities of the landmark from the perceived images $\{s^b\}$. Outputs are the one-step ahead predictions $\{s^p\}$ (cf. Fig. 1).

$s_i^b(t)$ denotes the state of input neuron i at the t-th time-step. The activations of the hidden units y_j at time t are defined as

$$y_j(t) = \sum_i^{N_1 \times N_2 \times N_3} s_i^b(t) u_{ji}^I + \sum_i^{N_1 \times N_2 \times N_3} s_i^b(t-1) \bar{u}_{ji}^I + \sum_{j'}^{N_h} s_{j'}^h(t-1) u_{jj'}^R \tag{4}$$



(a) Installation of the shelf (b) Top view of the shelf

Figure 3: Shelf installation / docking station. After docking, the robot could grasp some object from the shelf.

where u^I_{ji} represents the weight matrix between input layer and hidden layer, $\bar{u}^I_{ji}$ represents the weight matrix between the time-delayed input and the hidden layer and $u^R_{jj'}$ indicates the recurrent weight matrix within the hidden layer. The above equation shows that the hidden layer is connected to the weighted stimuli of the current input and delayed input, while there are additional lateral connections.

The transfer function of the hidden layer is the logistic function,

$$s^h_j(t) = \frac{1}{1 + exp(-\beta y_j(t))} \tag{5}$$

The k-th output for sensory prediction $s^p_k(t)$ at time t is

$$s^p_k(t) = \sum_j^{N_h} s^h_j(t) u^O_{kj} \tag{6}$$

where u^O_{kj} are the weights to the output layer (cf. Fig. 1).

3.3 Smooth Action Generation

In the CACLA trained reinforcement learning network, the input layer codes the predicted sensory states s^p and the output layer encodes a critic value and three robot action commands. Two of these represent the moving direction and torso

orientation change, and are activated linearly as

$$a^{m/o/g}(t) = \sum_k s_k^p(t) w_k^a \tag{7}$$

where w_k^a is the weight matrix between the sensory state units $s_k^p(t)$ and the action units $a^{m/o/g}(t)$. The third action unit signals the robot to stop walking (to initiate a possible grasping action) and is activated by a sigmoid function:

$$grasp(t) = \frac{1}{1 + exp(-a^g)} \tag{8}$$

A value of $grasp(t) > 0.5$ in this unit causes the robot to stop, while a value < 0.5 lets it continue the docking behavior.

A critic unit guides reinforcement learning. It is activated as

$$c(t) = \sum_k s_k^p(t) v_k \tag{9}$$

where v_k are the weights between the sensory state units $s_k^p(t)$ and the critic.

The action weights w^a are updated by the following rule:

$$w_j^a(t+1) = w_j^a(t) + \eta_w \delta_a a^{m/o/g} s_j \tag{10}$$

where δ_a is the action output error. According to CACLA [32], this update is only performed while the critic value (Eq. 9) is increasing. The updating of the critic weights is defined by:

$$v_{ij}(t+1) = v_{ij}(t) + \epsilon \delta_p s_j \tag{11}$$

where δ_p is the prediction error. When the reward is achieved, it is defined as

$$\delta_p = r - c(t) \tag{12}$$

while the reward is not yet achieved,

$$\delta_p = r + \gamma c(t+1) - c(t) \tag{13}$$

where γ is the temporal discount factor. An important difference of CACLA and conventional actor-critic learning is that the action weights w^a are only updated when the state value is increased, but not if it is decreased, since the optimum may exist between the current selection and the executed one due to the continuous encoding.

Tab. 2 shows the training parameters used in the two network modules.

Parameters	Parameter description	Value
δ_s	Variance of Gaussian Distribution in Sensory Module	1.0
$N_1 \times N_2 \times N_3$	Size of Input Layer in Sensory Module	$10 \times 10 \times 10$
N_h	Size of Hidden Layer in Sensory Module	1,500
η	Learning Rate in Sensory Module	0.1
β	Slope in Logistic Function	1.0
δ_m	Variance of Gaussian Distribution in Motor Action Module	1.0
η_w	Learning Rate in Motor Action Module	0.4
γ	Discount Factor of Reinforcement Learning	0.8
ϵ	Decay Rate of Reinforcement Learning	0.5

Table 2: Network parameters

4 Case Study

Previously we have studied, based on reinforcement learning, autonomous robot approaching [30], which can serve as a basis for different kinds of robot behaviors, such as grasping, human-robot-interaction, re-charging, etc. We test our predictive sensory model in our home lab, where a shelf with a landmark is installed (Fig. 3a). The goal of autonomous docking is to approach a narrow area which allows the robot to execute the grasping behavior afterwards. Grasping will be controlled by a self-organized map with supervised control output (yet unpublished), which ensures robustness and tolerance towards the position and pose reached by the docking: as long as the object (a plastic goblet) is visible in a certain area of the visual field, the robot is able to grasp it.

In our scenario, we define an area of 220mm by 130mm in front of the shelf as the optimal stopping area within which both feet of the robot must halt after approaching (red square in Fig. 3b) based on the kinematics and dimensions of the NAO robot for the grasping behavior. The largest distance is limited by the robot's arm length, while a too short distance to the shelf leads to the robot arm being blocked by the shelf when raised. A larger area for the start of the approach is constrained by the requirement that the landmark must be visible within the robot's visual field. A long distance navigation routine [36] is able to navigate the robot into the starting area, but its lack of precision necessitates the docking routine described in this paper.

4.1 Training Scheme

First, training sequences of the robot were collected through manual control in real-world experiments, which leads to supervised learning of the forward model and a form of supervised reinforcement learning for the action model [30]. We avoided the use of a robot simulator since it is very difficult to configure it to reflect the exact physical parameters of the real-world, such as the camera optics or friction on various carpets. With real-world training these factors are learnt without explicit model. The Gaussian activations (Eq. 3) speed up training and a large training set can be avoided [30].

Since the NAO robot has many degrees of freedom, for simplicity we keep the robot pose constant except for leg movements to keep the number of action units small. The training sequences are not recorded under continuous walking, but step-wise; in every step, the following data is recorded: the sensory identities from the landmark, the robot movement direction, change of torso orientation and the stop action for grasp. Based on the hardware constraints of the NAO robot, the walking direction is between $-\pi$ and $+\pi$ and the torso orientation change is from -0.1 to $+0.1$ (both in radians). Furthermore, to keep the dimensionality of the continuous action space limited, we try to keep the consistency of the step-walk-distance and the continuous-walk-distance covered by the sampling time, i.e. $\Delta D = 30\text{mm}$.

The approaching can start at any point within the approach area. For the reason of spatially balancing weights representing the walking data in the training sequence, we carefully selected the starting points as four points in the middle, six points in the left half and six points in the right half of the approaching area. We chose more training sequences of walking sideways because they are more challenging.

The recorded training sequences were used to train both modules off-line using the rules described above. The training in the sensory predictive module is identical to the conventional back-propagation through time (BPTT) algorithm. Fig. 4 shows the learning curves representing the output error in both two modules. The curve in Fig. 4a represents the mean squared error between the cube of the predictive sensory identities and the target one. And the blue and green curves in Fig. 4b represent the mean squared error of the walking direction and the change of torso orientation outputs (i.e. a^m and a^o cf. Eq. 7) between the target ones. The figures show that both modules quickly convergence before 50th iteration.

Methods	# of Trials	# of Success	Success Rate	Avg. Duration	Variance
Without Prediction	25	16	0.640	1.57	0.75
With Prediction	25	20	0.800	1.22	0.51

Table 3: Docking trials by reinforcement learning with and without prediction

4.2 Experimental Results

4.2.1 Approaching based on Reinforcement Learning without Prediction

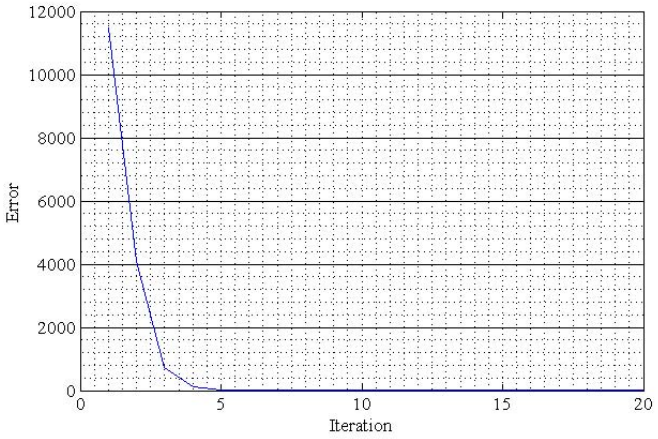
As soon as the first training procedure of CACLA was done, the NAO robot was able to approach the shelf in a continuous way. Connecting the sensory input directly to the motor action module, we conducted twenty-five approach trials without predictive model with LAN connection. Results are shown in Tab. 3. We count a trial successful if the subsequent grasping behavior leads to a successful grasp of the goblet.

From the on-site observations of docking trials, the robot sometimes reacted late when observing the landmark. For instance, it usually stopped too close to the docking station, which can lead to failure of grasping. In some cases the robot performed corrective actions following such responses, which led to longer average docking durations, as evident in Tab. 3.

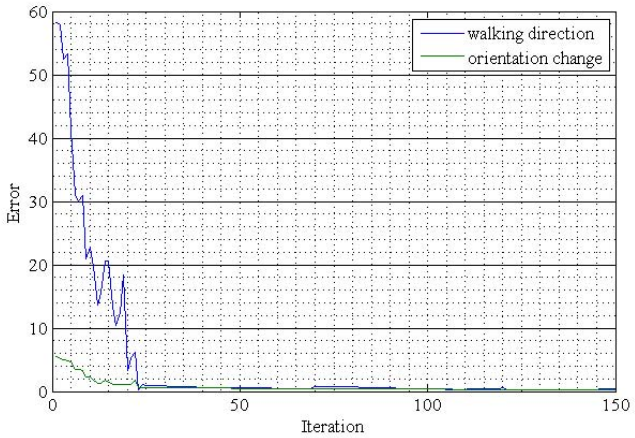
Two trials from both sides are shown in Fig. 5a. To objectively compare the offsets of robot camera, the trajectories were also tracked from a ceiling camera. With $0.2s$ sampling rate, we measured the x-y position of the robot. After synchronizing the ceiling camera and the robot observation data, we observe that a delay happens during the whole docking process, manifested by the observed inferred position being offset from the true position. This latency should be the cause of the observed late NAO reaction, which led to a longer approaching time, when it attempted to produce additional back and forth movements, and even failure of the docking and the following grasping, specifically when NAO elicited the grasping signal when it was already too close to the station (see the final stopping/grasping points were not at $(0\text{mm}, 150\text{mm})$ but beyond it in Fig. 5a).

4.2.2 Approaching based on Reinforcement Learning and Predictive Sensory System

In the following experiment of the integration of both modules, the predictive sensory module should build up an internal mental model to predict the upcoming sensory signal sequence based on the previous sensory experience. We used



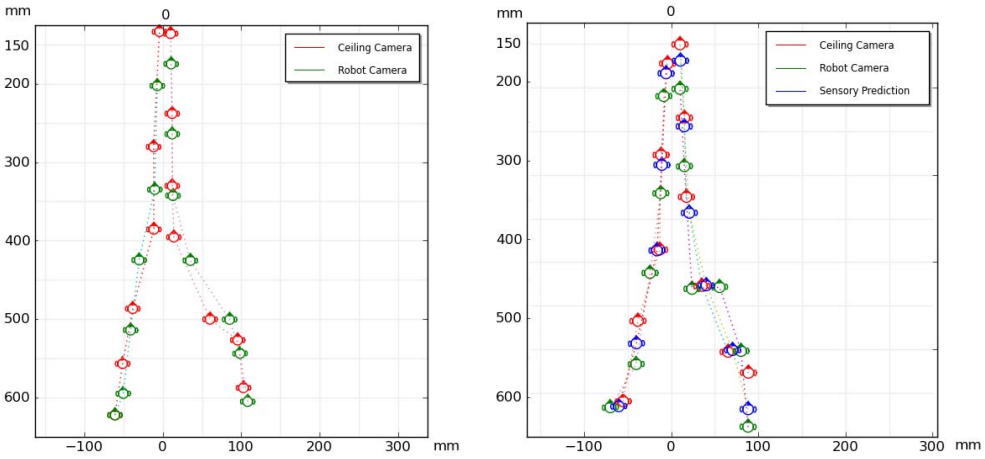
(a) Training curve of the predictive sensory module. The error is calculated by the mean squared of each unit in the output cube.



(b) Training curve of the motor action module. The error is calculated by the mean squared of each unit in the action outputs.

Figure 4: Training curves of two modules

both the predictive sensory value and the real sensory value by averaging them and apply the averaged value as the input of the units a^m and a^o. This method can compensate the sensorimotor cycle latency, filter noisy sensory information and therefore enhance the success-rate and speed while the robot is walking. But the stopping/grasping signal should arrive even earlier due to command delay in mechanics, so only the predictive sensory value is fed as input of the grasping



(a) Robot trajectories without sensory prediction

(b) Robot trajectories with sensory prediction

Figure 5: Trajectories of docking trials observed by a ceiling camera (true positions, red) and inferred from the robot's current sensory perception (green). (a) Without sensory prediction, sensorimotor delay causes the grasp signal to be produced at positions beyond the optimal position at $(0\text{mm}, 150\text{mm})$, which often results in failure of grasping. (b) The sensory prediction (blue) matches the true position better than the current estimate. With prediction, the robot correctly stops and gives the grasping command before the optimal position is inferred from the camera.

command unit a^g to further compensate this delay. These trial results are compared to the former results in shown in Tab. 3.

To visualize the effect of the predictive sensory model, we also synchronized the predictive sensory percepts of x-y distances and the observed ones in Fig. 5b with the ceiling camera data. Since the grasping signal only depends on the predictive sensory information, it solves the problem of the grasping signal coming too late and the robot stopping too close to the docking station, as it occurred in the previous experiment. Besides, in Fig. 5b we can see that the predicted trajectory is smoother than the one obtained from current robot vision, which hints to a denoising function of the predictive sensory module.

Methods	# of Approaches	# of Success	Success Rate	Avg. Time Caused	Variance
LAN	20	16	0.800	1.25	0.59
WIFI	20	15	0.750	1.35	0.58

Table 4: Docking trials in different connections

4.2.3 Docking Trials with Different Connections

To test the model predicting sensory percepts in a general context, an experiment with a different network connection was conducted. Due to different network delays (cf. Tab. 1), the robot received different kinds of sensory percept sequences when we used the slower wireless (WIFI) connection. As mentioned before, the predictive sensory module estimates a HMM process. To train the predictive sensory module for the different connection delay, we adjusted training samples to match the delay time of different connections. Tab. 4 shows that the results of docking trials under different connections are similar, which shows that the predictive sensory architecture adapts its predictions, and hence motor responses will be different, in a different sensory percept context.

5 Discussion

In neuroscience, there are two theories to interpret the integration mechanism of bottom-up sensory and top-down/recurrent processes. The "predictive coding" framework asserts that the error between the bottom-up sensory input and the top-down prediction is propagated from one cortical area to higher areas. The other theory called "biased competition" proposes that top-down feedback enhances the stimulus-driven neural activity that is consistent with the feedback prediction, while depressing inconsistent activity. Both theories can be unified to some extent [37]. The common point of these theories is that the bottom-up and top-down processes in the visual cortex have to be integrated, and the top-down processes act as a correction signal down to the cortical layers [38]. The top-down process is specifically retrieved from internal memory [39] to anticipate the error to the forthcoming sensory input, given the noise and ambiguity of the sensory data.

Furthermore, this integration is performed through the different levels of visual cortex [40]. This hierarchically-organized system enables complex and diverse goal-directed behavior of humans, which may be influenced by top-down goal-directed processes [41]. Cells in various levels of the hierarchy attempt to

unify the sensory inputs and top-down expectations, i.e. minimize the error between sensory percept and prediction. In this light, our model predicts a functional role of the integration of bottom-up sensory information and predicted sensory percepts.

From the robotics engineering point of view, a learnt predictive sensory module can assist to estimate the incoming sensory data in a stable way with changing sensory environment (e.g. changing of lighting), or even failure of sensor hardware. Prediction on various processing steps can compensate their latencies, such as of visual information flow within the sensorimotor cycle. Although some engineering methods could fulfill the sensory predictive model requirements, e.g. the potential field method and its augmentation of prediction [42, 43, 17], their limitations include local minima, linearity assumptions or non-adaptability to environmental changes. Biological findings, such as that dragon-flies always estimate their flying positions so as to maintain an interception flight trajectory during prey-pursuit [44], support the importance of predictive action. A plausible sensory model should integrate incoming and predicted data, which coincides with a similar function of filtering to estimate the incoming information from noisy input.

Due to the complexity of a dynamic environment, we cannot build an internal model of the complete raw sensory percepts. Nor do we wish to predict only the action output, since this would lead to a model specialized for one task. Therefore, we propose to learn prediction within individual processing steps, in particular on sensory areas, so as to enable life-long latent learning of a robot. This can be done in an unsupervised manner, for instance, sensory experiences can be recorded during everyday robot activities, from which we can extract docking/grasping-related experiences for further real-time training (e.g. [45]). The robot would then adapt when environmental factors change.

6 Conclusion

In this paper, we presented a predictive sensory architecture that predicts the visually retrieved coordinates. A continuous reinforcement-learnt action strategy was based on these predicted sensory values. In the case study of robot approaching, prediction led to higher robustness and faster approaching behavior. The filtering function of the predictive sensory module provided a smooth sensory signal leading to a smooth and robust behavior. We also showed that the predictive sensory model effectively compensates the latency of the sensorimotor cycle of the robot, which led to less errors being made and to faster executed behavior.

Acknowledgments

This research has been partly supported by the EU projects RobotDoC under 235065 ROBOT-DOC from the 7th Framework Programme (FP7), Marie Curie Action ITN, and KSERA funded from FP7 for Research and Technological Development under grant agreement n°2010-248085. The authors thank Nicolás Navarro-Guerrero, Pascal Schröter, Erik Strahl, Johannes Bauer and Sven Magg for their contributions in technical support, insightful discussions and revision suggestions.

References

[1] G. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1045–1062, 1999.

[2] E. Goldstein, *Sensation and Perception*. Wadsworth Publishing Company, 2010.

[3] M. Bar, "A cortical mechanism for triggering top-down facilitation in visual object recognition," *Journal of Cognitive Neuroscience*, vol. 15, no. 4, pp. 600–609, 2003.

[4] K. Kveraga, A. Ghuman, and M. Bar, "Top-down predictions in the cognitive brain," *Brain and Cognition*, vol. 65, no. 2, p. 145, 2007.

[5] D. MacKay, "Perceptual stability of a stroboscopically lit visual field containing self-luminous objects," *Nature*, 1958.

[6] R. Nijhawan, "Motion extrapolation in catching.," *Nature*, 1994.

[7] R. Nijhawan, "Visual decomposition of colour through motion extrapolation," *Nature*, 1997.

[8] W. Li, V. Piëch, and C. Gilbert, "Perceptual learning and top-down influences in primary visual cortex," *Nature Neuroscience*, vol. 7, no. 6, pp. 651–657, 2004.

[9] L. Trainor, "Predictive information processing is a fundamental learning mechanism present in early development: Evidence from infants," *International Journal of Psychophysiology*, vol. 83, no. 2, p. 256, 2012.

[10] J. Hirsch and C. Gilbert, "Synaptic physiology of horizontal connections in the cat's visual cortex," *The Journal of Neuroscience*, vol. 11, no. 6, pp. 1800–1809, 1991.

[11] Z. Kisvarday, E. Toth, M. Rausch, and U. Eysel, "Orientation-specific relationship between populations of excitatory and inhibitory lateral connections in the visual cortex of the cat.," *Cerebral Cortex*, vol. 7, no. 7, pp. 605–618, 1997.

[12] V. Lamme and P. Roelfsema, "The distinct modes of vision offered by feedforward and recurrent processing," *Trends in Neurosciences*, vol. 23, no. 11, pp. 571–579, 2000.

[13] G. Le Masson, S. Renaud-Le Masson, D. Debay, T. Bal, *et al.*, "Feedback inhibition controls spike transfer in hybrid thalamic circuits," *Nature*, vol. 417, no. 6891, pp. 854–858, 2002.

[14] C. Darrin, G. Christopher, U. Aleš, and G. Cheng, "Learning to act from observation and practice," *International Journal of Humanoid Robotics*, vol. 1, no. 04, pp. 585–611, 2004.

[15] L. Natale, F. Nori, G. Sandini, and G. Metta, "Learning precise 3d reaching in a humanoid robot," in *IEEE 6th International Conference on Development and Learning, ICDL*, pp. 324–329, 2007.

[16] S. Nishide, T. Ogata, J. Tani, K. Komatani, and H. Okuno, "Predicting object dynamics from visual images through active sensing experiences," *Advanced Robotics*, vol. 22, no. 5, pp. 527–546, 2008.

[17] N. Pradhan, T. Burg, and S. Birchfield, "Robot crowd navigation using predictive position fields in the potential function framework," in *IEEE American Control Conference, ACC*, pp. 4628–4633, 2011.

[18] T. H. Hong, C. Rasmussen, T. Chang, and M. Shneier, "Road detection and tracking for autonomous mobile robots," in *Proc. of SPIE Aeroscience Conference*, pp. 311–319, 2002.

[19] Y. Matsushita and J. Miura, "On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 274–284, 2011.

[20] J. Zhong, C. Weber, and S. Wermter, "Learning features and predictive transformation encoding based on a horizontal product model," *Artificial Neural Networks and Machine Learning, ICANN*, pp. 539–546, 2012.

[21] S. Chen, "Kalman filter for robot vision: A survey," *IEEE Transactions on Industrial Electronics*, vol. 59, pp. 4409 –4420, nov. 2012.

[22] V. Bonato, E. Marques, and G. Constantinides, "A floating-point extended kalman filter implementation for autonomous mobile robots," *Journal of Signal Processing Systems*, vol. 56, pp. 41–50, 2009.

[23] T. Klein, J. Jeka, T. Kiemel, and M. Lewis, "Navigating sensory conflict in dynamic environments using adaptive state estimation," *Biological Cybernetics*, pp. 1–14, 2012.

[24] A. Schaefer, S. Udluft, and H. Zimmermann, "Learning long-term dependencies with recurrent neural networks," *Neurocomputing*, vol. 71, pp. 2481–2488, 2008.

[25] R. Möller, "A model of ant navigation based on visual prediction," *Journal of Theoretical Biology*, 2012.

[26] J. Hirel, P. Gaussier, and M. Quoy, "Biologically inspired neural networks for spatio-temporal planning in robotic navigation tasks," in *IEEE International Conference on Robotics and Biomimetics, ROBIO*, pp. 1627–1632, 2011.

[27] R. Saegusa, F. Nori, G. Sandini, G. Metta, and S. Sakka, "Sensory prediction for autonomous robots," in *7th IEEE-RAS International Conference on Humanoid Robots*, pp. 102–108, 2007.

[28] S. Thrun, "Bayesian landmark learning for mobile robot localization," *Machine Learning*, vol. 33, no. 1, pp. 41–76, 1998.

[29] K. Cullen, "Sensory signals during active versus passive movement.," *Current Opinion in Neurobiology*, vol. 14, no. 6, p. 698, 2004.

[30] N. Navarro-Guerrero, C. Weber, P. Schroeter, and S. Wermter, "Real-world reinforcement learning for autonomous humanoid robot docking," *Robotics and Autonomous Systems*, 2012.

[31] Aldebaran, "http://www.aldebaran-robotics.com," 2009.

[32] H. van Hasselt and M. Wiering, "Reinforcement learning in continuous action spaces," in *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, ADPRL*, pp. 272–279, 2007.

[33] R. Shadmehr, M. Smith, and J. Krakauer, "Error correction, sensory prediction, and adaptation in motor control," *Annual Review of Neuroscience*, vol. 33, pp. 89–108, 2010.

[34] J. Izawa and R. Shadmehr, "Learning from sensory and reward prediction errors during motor adaptation," *PLoS Computational Biology*, vol. 7, no. 3, p. e1002012, 2011.

[35] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[36] W. Yan, C. Weber, and S. Wermter, "A neural approach for robot navigation based on cognitive map learning," in *International Joint Conference on Neural Networks, IJCNN*, pp. 1–8, 2012.

[37] M. Spratling, "Predictive coding as a model of biased competition in visual attention," *Vision Research*, vol. 48, no. 12, pp. 1391–1408, 2008.

[38] K. Rauss, S. Schwartz, and G. Pourtois, "Top-down effects on early visual processing in humans: A predictive coding framework," *Neuroscience & Biobehavioral Reviews*, vol. 35, no. 5, pp. 1237–1253, 2011.

[39] J. Anderson and L. Schooler, "The adaptive nature of memory.," *The Oxford Handbook of Memory.*, 2000.

[40] A. Alink, C. Schwiedrzik, A. Kohler, W. Singer, and L. Muckli, "Stimulus predictability reduces responses in primary visual cortex," *The Journal of Neuroscience*, vol. 30, no. 8, pp. 2960–2966, 2010.

[41] M. Corbetta, G. Shulman, *et al.*, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews Neuroscience*, vol. 3, no. 3, pp. 215–229, 2002.

[42] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE International Conference on Robotics and Automation.*, vol. 2, pp. 500–505, 1985.

[43] L. Huang, "Velocity planning for a mobile robot to track a moving targeta potential field approach," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 55–63, 2009.

[44] R. Olberg, "Visual control of prey-capture flight in dragonflies," *Current Opinion in Neurobiology*, vol. 22, no. 2, pp. 267 – 271, 2011.

[45] P. Hartono and S. Kakita, "Fast reinforcement learning for simple physical robots," *Memetic Computing*, vol. 1, no. 4, pp. 305–313, 2009.