2014 14th IEEE-RAS International Conference on
Humanoid Robots (Humanoids)
November 18-20, 2014. Madrid, Spain

Adaptive visual pursuit involving eye-head coordination and prediction of the target motion

Lorenzo Vannucci[1], Nino Cauli[1], Egidio Falotico[1], Alexandre Bernardino[2], Cecilia Laschi[1]

Abstract— **Nowadays, increasingly complex robots are being designed. As the complexity of robots increases, traditional methods for robotic control fail, as the problem of finding the appropriate kinematic functions can easily become intractable. For this reason the use of neuro-controllers, controllers based on machine learning methods, has risen at a rapid pace. This kind of controllers are especially useful in the field of humanoid robotics, where it is common for the robot to perform hard tasks in a complex environment. A basic task for a humanoid robot is to visually pursue a target using eye-head coordination. In this work we present an adaptive model based on a neuro-controller for visual pursuit. This model allows the robot to follow a moving target with no delay (zero phase lag) using a predictor of the target motion. The results show that the new controller can reach a target posed at a starting distance of 1.2 meters in less than 100 control steps (1 second) and it can follow a moving target at low to medium frequencies (0.3 to 0.5 Hz) with zero-lag and small position error (less then 4 cm along the main motion axis). The controller also has adaptive capabilities, being able to reach and follow a target even when some joints of the robot are clamped.**

I. INTRODUCTION

Following a moving target with a foveal vision is one of the essential tasks of humans and humanoid robots. Humans, in order to perform this task, use a combination of eye and head movements in conjunction with prediction [1] of the target dynamics in order to align eye and target motion.

The reasons for the occurrence of head motions are a wider visual information, obtained by moving the neck in comparison to visual information obtained by eye movements, and the costant relative positions of the object and the eye in case of the neck motion. During the head-unrestrained tracking of a periodical target, Lanman and colleagues [2] reported that head movements accounted for approximately 75% of the gaze displacements while the eye-in-head remained relatively stationary at the center of the orbit.

These results confirm the central role of the head in pursuit task. From a robotic point of view, several implementations inspecting the effects of the gaze movements on 3D reconstruction ([3], [4]) or replicating the visual pursuit exist. The investigation of the oculomotor behavior from the computational neuroscience standpoint was oftentimes performed making use of simple pan/tilt cameras [5]. Although these systems may consistently represent some specific features of the eye movements, they seem to be somehow inadequate for their inability to incorporate the coordinated motion of the head.

Different considerations may be applied to the eye-head robot WE-3 [6] and its successors [7]. The WE-3 robotic head is anthropomorphic (in terms of geometry, mass and kinematic variables), therefore it offers the major advantage of performing coordinated head-eye motion and pursuit motion in the depth direction but without including prediction. Shibata and colleagues suggested a control circuits to realize three of the most basic oculomotor behaviors [8]: the vestibulo-ocular reflex and optokinetic response (VOR and OKR) for gaze stabilization, smooth pursuit for tracking moving objects, and saccades for overt visual attention. This model, based on the prediction of target dynamics, is capable to execute fast gaze shift, but it did not consider the head motion as part of the oculomotor control in the pursuit task. The same consideration can be applied to the model of smooth pursuit and catch-up saccade [9] implemented on the iCub robot.

Rajruangrabin and Popa [10] proposed a model which tries to replicate a realistic motion coordination for a humanoid's robot neck and eyes while tracking an object (with no prediction of the target motion). This model also takes into consideration that under certain conditions the robot kinematics model might be difficult to obtain. As a result of this they proposed a novel reinforcement learning approach for a model-free implementation. This aspect is really basic if we consider the field of humanoid robotics, where the robot has multiple end effectors (head, hands, etc...) and/or non-rigid links. For these reasons, in the last years the usage of neuro-controllers, controllers based on machine learning methods, has risen at a rapid pace. These models are inspired by human biology and they try to replicate the control found in nature by mimicking the function of some parts of the nervous system. The advantages of using neuro-controllers over classic control methods is that they can easily adapt to any kind of robot, without prior knowledge of the parameters of the kinematic chain linking to the end effector, even if the chain is made up of non-rigid links.

In this work we present a model based on a neuro-controller which was first introduced by Asuni and colleagues [11]. They proposed an approach based on the motor babbling technique in conjunction with a growing neural gas to achieve the goal of making a robotic head fixing a static target. Despite having some remarkable properties, their model had some limitations mainly in terms of performances (it takes around 250 control steps to reach a static target). We propose an adaptive controller, based on [11] able to accomplish a visual pursuit task involving eye-head coordination and prediction of a moving target.

[1] The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy
[2] Computer and Robot Vision Laboratory, Instituto de Sistemas e Robotica, Instituto Superior Tecnico, Lisbon, Portugal

978-1-4799-7174-9/14/$31.00 ©2014 IEEE

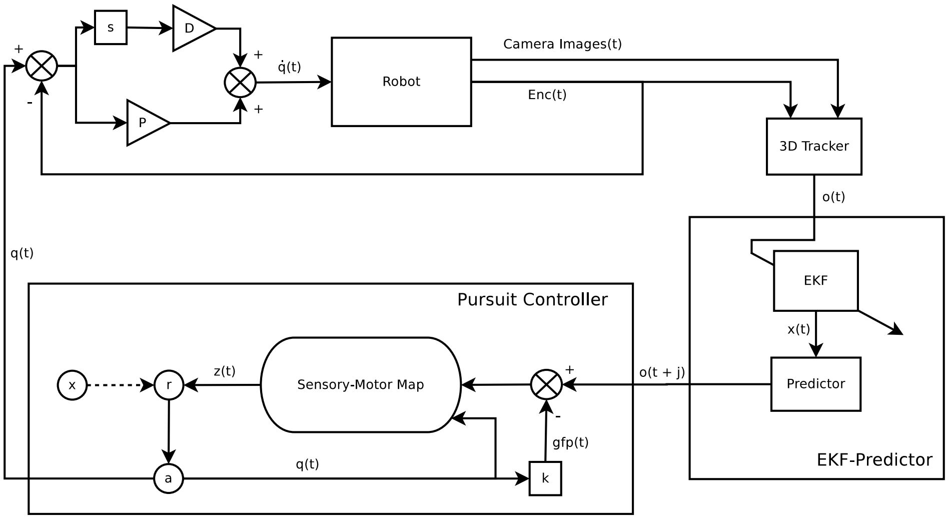

Fig. 1. General model of the adaptive controller for visual pursuit tasks

II. MODEL

In this section we describe the general model of the controller, as depicted in Fig. 1. The control loop starts with a 3D tracker able to calculate the 3D position of the target o in the world reference frame. After that, an Extended Kalman Filter predicts the position o of j steps ahead. The prediction is used as reference of the Pursuit Controller that calculates the position q of the head and eyes joints needed to center the target in the camera image. The difference of joints positions q and encoders Enc is then transformed by a PD controller in joint velocities $\dot{q}$ used as motor commands for the robotic head.

A. EKF Predictor

In order to predict target dynamics an Extended Kalman Filter was used. Because we are dealing with measurements that are noisy and there is time variability in the parameters of our target, our internal models should have the following properties: fast convergence, long term learning and noise robustness. Another important property needed to implement the control system is the capacity to predict the entire time varying trajectory of the target, and not only the final position. A model that iterates a single step learning through time is a smart solution. Moreover, an approximation of the target dynamics is known in advance. The Kalman Filter (KF) fits perfectly with these requirements and has been used successfully in many applications. For these reasons the target internal model was implemented using an Extended Kalman Filter.

In order to predict the target position the filter first updates its state x and then iterates its prediction step j times.

B. Pursuit Controller

The goal of the pursuit controller is to move the current gaze fixation point (gfp) of the robot towards the target point. This controller is an improved version of [11]. The core part of this controller is the Sensory-Motor Map, which is a network able to respond to a stimulus by activating some specific units. This is similar to the functioning of the somatosensory cortex in the human brain: a stimulus reaches the cortex and the section dedicated to respond to that stimulus activates. In this case, the stimulus, or input, is a proprioceptive anticipated feedback (the motor command generated at the previous control step) combined with the current displacement in space of the gaze fixation point with respect to the desired target, calculated as a normalized difference vector:

$$p = \left(\frac{o(t+j) - gfp(t)}{||o(t+j) - gfp(t)||_2}, q(t) \right) \tag{1}$$

This map is represented by a neural model capable of learning how to distinguish between different inputs, the Growing Neural Gas[12] (GNG). The Neural Gas model[13] (NG) is similar to a Kohonen's map[14], but has the benefit of not having a fixed topology. The main difference between the GNG and the NG is that in the first model the network of units grows dynamically during the training phase, instead of being of a fixed size decided beforehand. The output of such network is the activated unit, or winning unit:

$$gng(p) = \arg\min_{s \in A} ||p - w_s||^2 \tag{2}$$

where A is the set of units and w_s is the weight vector of the s-th unit. For each unit s, its topological neighbourhood is defined as

$$N_s = \{u \in A | (s,u) \in E\} \tag{3}$$

where E is a dynamically built subset of $A \times A$.

After a stimulus is given to the Sensory-Motor Map, this will then respond by generating the motor commands that will move the gaze fixation point towards the desired position. This is done by passing the activation signal through a series of units specifically trained by a motor babbling algorithm to generate these motor commands. The r units represent the agonist and antagonist activations for each motor. Thus, the number of these units is double the number of the actual actuators so that the couple r_{2i}, r_{2i+1} are the agonist and antagonist units for the i-th motor, also called r_i^E, r_i^I, which stand for the excitatory and inhibitory stimuli for the i-th actuator. Their activation is computed as:

$$r_i = x_i + z_{wi} + \frac{\sum_{k \in N_w} \nu z_{ki}}{|N_w|} \tag{4}$$

where w is the index of the winning unit, z_{ki} is a weight coming from the k-th unit of the GNG, ν is a proper constant and x_i is a random activation coming from x units population, only present during the training phase. The a units are responsible for the generation of the actual motor commands that have to be sent to the actuators, starting from the excitatory and inhibitory stimuli coming from the r population. At each control step the new output values are computed in terms of a difference between current and previous step values:

$$\frac{\mathrm{d}a_i}{\mathrm{d}t} = \epsilon(||o(t+j) - gfp(t)||_2) \cdot (r_i^E - r_i^I) \cdot g(r_i^E, r_i^I, a_i(t-1)) \tag{5}$$

where $\epsilon(d)$ is a function of the distance between the target and the current gaze fixation point defined as

$$\epsilon(d) = \begin{cases} v \cdot d & \text{during the execution phase} \\ \epsilon & \text{during the training phase} \end{cases} \tag{6}$$

with an appropriate speed parameters v and ϵ, and

$$g(r^E, r^I, \psi) = \begin{cases} \psi_{max} - \psi & \text{if } (r^E - r^I) \geq 0 \\ \psi & \text{if } (r^E - r^I) < 0 \end{cases} \tag{7}$$

where ψ_{max} is the maximum angle of the joint.

The training for this controller is performed in two phases, both unsupervised:

- in the first phase only the sensory-motor map is trained;
- in the second phase the motor babbling learning is performed, the newly trained map is used in the global architecture to train the z connections.

The z connections are trained by using Grossberg's Outstar Law[15]:

$$\frac{\mathrm{d}z_{ki}}{\mathrm{d}t} = \gamma \cdot c_k \cdot (-\delta z_{ki} + r_i) \tag{8}$$

where γ is the learning rate and δ is a decay parameter. At a given step, only the connections going out from the winner units and its neighbours are updated, otherwise the update would not be related to the recently performed movement. This is done by defining:

$$c_k = \begin{cases} 1 & \text{if } k \text{ is the winner unit } w \text{ or } k \in N_w \\ 0 & \text{else} \end{cases} \tag{9}$$

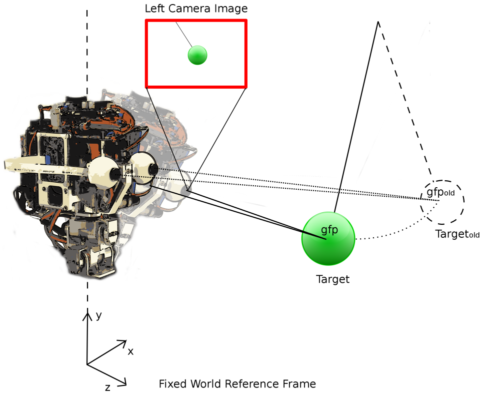

Fig. 2. Task setup. The robotic head pursues with its gfp the moving target keeping it on the center of camera images

The model presented in this section differs from the one proposed in [11] for the normalization of the difference vector given as a input to the GNG, the anticipated proprioceptive feedback and the introduction of the ϵ function, replacing a constant. All these improvements were done to speed-up the controller.

III. IMPLEMENTATION

In order to test the model, we decided to perform a pursuit task of a moving target with pendulum motion. A robotic head was fixed on a table and a ball was hanged to a wire in front of it (see Fig. 2). During the experiments the robotic head had to pursue with its gaze fixation point the oscillating pendulum. The goal was to keep the moving target centred on the camera images. The pursuit was performed moving all head motors (neck and eyes).

A. SABIAN head

For our experiments the SABIAN head, which is an iCub head [16] (Fig. 2), was used. This robotic head contains a total of 6 DOFs: 3 for the neck (pan, tilt and swing) and 3 for the eyes (an independent pan for each eye and a common tilt). The visual stereo system consists of 2 dragonfly2 cameras with a maximal resolution of 640X480 pixels. All the joints are actuated by DC servo motors with relative encoders. The processing unit consists of a PC104 with a live Debian distro running on it.

B. Extended Kalman Filter

The EKF is an extension of the Kalman Filter that deals with non-linear dynamics. It is used to estimate the state of a system, in our case the parameters of the pendulum trajectory. The filter keeps an internal state representing the tracked object and its covariance matrix (that represents the reliability of the state). In order to update the internal state the filter uses an a priori model of the object trajectory (in our case a in-plane pendular motion) and an observation model describing the relationship between the measurements and the state (in our case the 3D position of the target extracted from the cameras). For more details about the Kalman filtering see [17].

a) Filter implementation: The model of a 2D pendulum was chosen to describe the target dynamics. The target movement is approximated as an oscillation of a 2D pendulum on a plane A rotated of an angle α around the Y vertical axis. Defined Θ, $\dot{\Theta}$ and $\ddot{\Theta}$ as angular position, velocity and acceleration, g as gravity, L as wire length, d as damping factor and m as ball mass, the 2D pendulum equation is:

$$\ddot{\Theta} + \frac{d}{m}\dot{\Theta} + \frac{g}{L}\sin(\Theta) = 0 \tag{10}$$

Because the observations are expressed in 3D Cartesian coordinates, a non linear observation model is needed. If C is the position of the pendulum pivot in the 3D Cartesian space, then the 3D Cartesian position of the target T is:

$$T(\Theta, L, C, \alpha) = \begin{bmatrix} C_x + L\sin(\Theta)\cos(\alpha) \\ C_y - L\cos(\Theta) \\ C_z - L\sin(\Theta)\sin(\alpha) \end{bmatrix} \tag{11}$$

For these reasons the internal state of the filter needs to store all the 2D pendulum parameters plus the 3D pivot position and the rotation angle of the plane. This results in a eight element state vector:

$$\begin{aligned} x &= \begin{bmatrix} \Theta & \dot{\Theta} & \frac{d}{m} & \frac{g}{L} & C_x & C_y & C_z & \alpha \end{bmatrix} \\ &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{bmatrix} \end{aligned} \tag{12}$$

Using equations 10, 11 and 12 the transition and observation models become respectively:

$$f(x) = x + \Delta t\dot{x} = x + \Delta t \begin{bmatrix} x_2 \\ -x_3x_2 - x_4\sin(x_1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{13}$$

$$h(x) = \begin{bmatrix} x_5 + \frac{g}{x_4}\sin(x_1)\cos(x_8) \\ x_6 - \frac{g}{x_4}\cos(x_1) \\ x_7 - \frac{g}{x_4}\sin(x_1)\sin(x_8) \end{bmatrix} \tag{14}$$

The EKF is not only used to track the target but also to predict its future trajectory. The predicted trajectory is obtained iterating the filter prediction step until needed. First the filter state is saved, then the predicted trajectory is calculated (iterating the prediction step of the filter) and eventually the saved filter state is restored.

C. Pursuit Controller

For the training of the pursuit controller, we started by training the sensory motor-map, namely the growing neural gas. To do this we created a training set of 100000 input patterns which was given in input ten times to the GNG. After the training ended, the resulting network had 3695 units.

The selection of the hyperparameters for the controller $(\gamma, \delta, \epsilon, \nu)$ is critical and greatly impacts the performance. To appropriately choose such parameters we implemented a model selection procedure during which we measured the performance of the resulting model by computing its mean speed of reaching on a training set D of 20 target points:

$$v = \frac{1}{|D|}\sum_{i \in D}\frac{s_i}{t_i} \tag{15}$$

where s_i the starting distance of the target and t_i the number of control step used to reach it. The model selection was performed on these values for the hyperparameters:

- $\gamma \in \{0.005, 0.01, 0.015, 0.02\}$;
- $\delta \in \{0.0005, 0.001, 0.0015, 0.002\}$;
- $\epsilon \in \{0.1, 0.3, 0.05\}$;
- $\nu \in \{0.1, 0.3, 0.5\}$.

The best results on the validation set were obtained by a model with a mean speed of 0.055 meters/step and the following values for the hyperparameters:

$$\gamma = 0.005 \qquad \delta = 0.002 \qquad \epsilon = 0.1 \qquad \nu = 0.1$$

All the training phase was done in a simulated environment, using the *iCub Simulator*, a software developed alongside the iCub control libraries.

IV. RESULTS

A. Step response

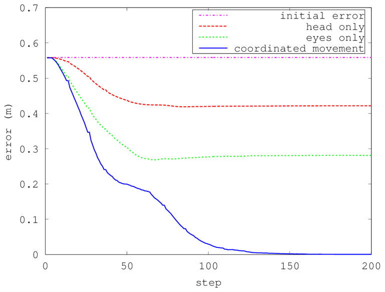

Fig. 3. Distance between the current gaze fixation point and the target during the step response task, with the contribution of the head and the eye joints.

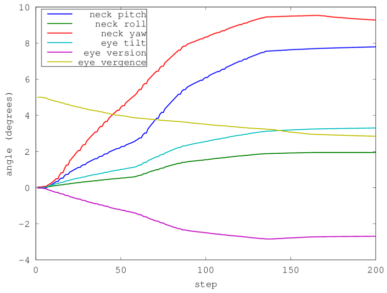

Fig. 4. Joints trajectories during a step response task.

In this experiments, the controller had to reach a static target from a starting position in which all the joints were set to 0 except for the vergence that was set to 5. Fig. 3 shows

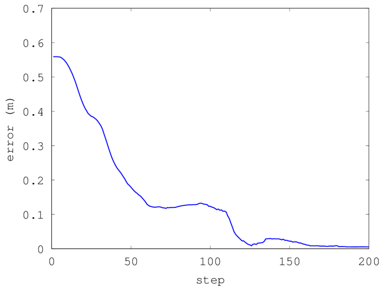

Fig. 5. Distance between the current gaze fixation point and the target during the task with clamped joints.

the behaviour of the distance from the target during such task. Given that the control loop runs at 100Hz frequency, it can be observed that the system takes one seconds to reach a target from a starting distance of more than half metre.

Fig. 3 also shows the contribution to the reduction of the error from the head and eye joints. It can be observed that the sum of both contributes does not nullify the error, thus it is the coordination between head and eye movements that allows the proper reaching.This can also be seen in Fig. 4, where the trajectories of the joints during the task are reported.

In order to properly evaluate the performance we repeated this test 15 times with starting distances ranging from 40cm to 1.2m and we observed that the mean number of steps needed to reach a target with an error under 4cm was 95.73 with a standard deviation of 11.98.

In order to test the adaptivity of the controller we repeated the same 15 tests with some joints of the robot clamped. In particular, we decided to clamp all the head joints. The same model was used, without going through a new learning phase. The result for the same test showed in Fig. 3 is displayed in Fig. 5. In this case, the step response is less smooth and more control steps are necessary in order to reach the target.

The mean execution time steps for the reaching task with clamped joints is 103.73 with a standard deviation of 31.50.

B. Visual pursuit

In this experiments, the robot had to follow a target attached to a pendulum. The performance of the model were tested at various frequencies of the pendulum oscillation, with a starting amplitude of 50cm. All the experiments started after 1 second. This time is necessary to allow the filter to reach convergence. In order to select the proper value for the prediction step j, we estimated the delay of the system as the sum of the visual delay (30ms) and the motor delay (200ms). Thus, we chose $j = 8$, which accounts for a delay of 240ms, given that the predictions are given for 30ms intervals.

The results are shown in Table I, where $\overline{E}_x$ is the mean error alongside the principal motion axis, $\overline{E}$ is the mean distance from the target in metres, $\overline{E}_u$ is the mean error on the camera images in pixels alongside the horizontal axis and $\overline{\dot{u}}$ is the mean horizontal speed of the target on the camera in pixels/step.

TABLE I
RESULTS FOR VISUAL PURSUIT.

Freq (Hz)	$\overline{E}_x$	$\overline{E}$	$\overline{E}_u$	$\overline{\dot{u}}$
0.3	0.031	0.091	10.618	0.282
0.4	0.033	0.088	11.687	0.421
0.5	0.032	0.101	12.515	0.485
0.6	0.055	0.143	17.737	0.657

It can be observed that the model is able to pursue a target up to a frequency of 0.5Hz, as the mean error on the x axis, the one on which most of the motion occurred, at 0.6Hz is larger than the radius of the target (4cm). Fig. 6 and 7 show the pursuit of the target, for oscillation frequencies of 0.4 and 0.6, by comparing the target and current gaze horizontal positions in time. It can be observed that, after an initial alignment phase, the robot is able to follow the target at 0.4Hz, while at 0.6Hz the performance deteriorates.

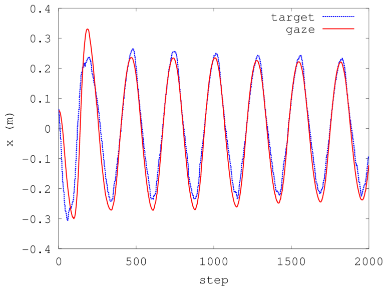

Fig. 6. Target and gaze alongside the x axis during an oscillation frequency of 0.4Hz.

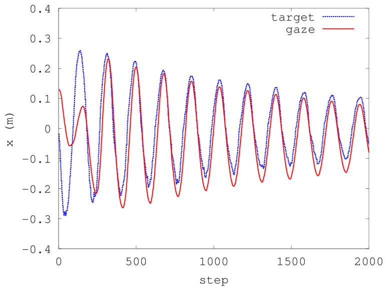

Fig. 7. Target and gaze alongside the x axis during an oscillation frequency of 0.6Hz.

In order to test the adaptivity of the controller we repeated the same experiments with all the head joints clamped. The results for the visual pursuit task are shown in Table II, while Fig. 8 shows the pursuit of the target for an oscillation frequency of 0.4Hz on the horizontal axis. The robot showed capable of following the target but with a considerable undershooting, even at a frequency of 0.4Hz. Thus, we did not investigate higher frequencies. In Fig. 9 is shown the robot and its left camera images during the pursuit experiment at 0.4 Hz oscillations.

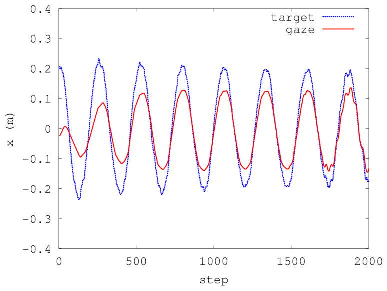

Fig. 8. Target and gaze alongside the x axis during an oscillation frequency of 0.4Hz with clamped joints.

TABLE II
RESULTS FOR VISUAL PURSUIT WITH CLAMPED JOINTS.

Freq (Hz)	$\overline{E}_x$	$\overline{E}$	$\overline{E}_u$	$\bar{u}$
0.3	0.052	0.152	15.113	0.421
0.4	0.054	0.349	28.144	0.685

V. CONCLUSIONS

In this work we proposed an adaptive model for robotic control able to perform visual pursuit with prediction of the target motion. In order to design this controller, we started from a neuro-controller for reaching found in literature that incorporates an unsupervised machine learning model, a growing neural gas, paired with a motor babbling training algorithm. This model was capable of reaching static targets posed at a starting distance of 1.2 meters in roughly 250 control steps. We modified this controller by simplifying the inputs space of the growing neural gas, adding a speed gain based on the target distance and closing the control loop on the motor commands instead of the encoders. This improvements made it work in real-time: the obtained results showed that the new controller could reach a target posed at a starting distance of 1.2 meters in less than 100 control steps (1 second) and it could follow a moving target at low to medium frequencies (0.3 to 0.5Hz) with zero-lag and small position error (less then 4 cm along the main axis of motion). The controller also had adaptive capabilities, being able to reach and follow a target even when some joints of the robot were clamped. In addition, the adaptive property of such a model guarantees the applicability of this approach also to complex robotic platforms. This could fit well in the last trends of humanoid robotics, where complex robots with muscles-like structure are starting to emerge.

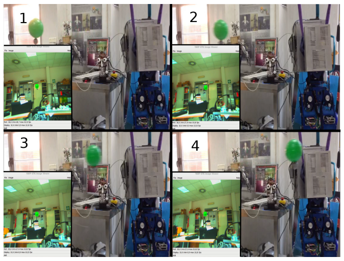

Fig. 9. Four snapshots acquired during the pursuit experiment with 0.4 Hz frequency. In big images it is shown the SABIAN head looking at the target, in small windows it is shown the left eye camera image.

ACKNOWLEDGMENT

The authors would like to thank Italian Ministry of Foreign Affairs and International Cooperation DGSP-UST (Direzione Generale per la Promozione del Sistema Paese - Unità per la Cooperazione Scientifica e Tecnologica Bilaterale e Multilaterale) for the support through Joint Laboratory on Biorobotics Engeneering project.

REFERENCES

[1] G. Barnes and P. Asselman, “The mechanism of prediction in human smooth pursuit eye movements.” *The Journal of Physiology*, vol. 439, no. 1, pp. 439–461, 1991.

[2] J. Lanman, E. Bizzi, and J. Allum, “The coordination of eye and head movement during smooth pursuit,” *Brain research*, vol. 153, no. 1, pp. 39–53, 1978.

[3] M. Antonelli, A. P. del Pobil, and M. Rucci, “Bayesian multimodal integration in a robot replicating human head and eye movements,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’14)*, 2014, pp. 2868–2873.

[4] X. Kuang, M. Gibson, B. E. Shi, and M. Rucci, “Active vision during coordinated head/eye movements in a humanoid robot,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1423–1430, 2012.

[5] A. Bernardino, C. Silva, J. Santos-Victor, and C. Pinto-Ferreira, “Behaviour based oculomotor control architecture for stereo heads,” in *Proc. 3rd International Symposium on Intelligent Robotic Systems, Pisa, Italy*. Citeseer, 1995.

[6] A. Takanishi, T. Matsuno, and I. Kato, “Development of an anthropomorphic head-eye robot with two eyes-coordinated head-eye motion and pursuing motion in the depth direction,” in *Proc. IEEE/RJS International Conference on Intelligent Robots and Systems (IROS’97)*, vol. 2, 1997, pp. 799–804.

[7] H. Miwa, K. Itoh, M. Matsumoto, M. Zecca, H. Takanobu, S. Rocella, M. C. Carrozza, P. Dario, and A. Takanishi, “Effective emotional expressions with expression humanoid robot we-4rii: integration of humanoid robot hand rch-1,” in *Proc. IEEE/RJS International Conference on Intelligent Robots and Systems (IROS’04)*, vol. 3, 2004, pp. 2203–2208.

[8] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal, “Biomimetic oculomotor control,” *Adaptive Behavior*, vol. 9, no. 3-4, pp. 189–207, 2001.

[9] E. Falotico, D. Zambrano, G. G. Muscolo, L. Marazzato, P. Dario, and C. Laschi, “Implementation of a bio-inspired visual tracking model on the icub robot,” in *Proc. 19th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN’10)*, 2010, pp. 564–569.

[10] J. Rajruangrabin and D. O. Popa, “Robot head motion control with an emphasis on realism of neck–eye coordination during object tracking,” *Journal of Intelligent & Robotic Systems*, vol. 63, no. 2, pp. 163–190, 2011.

[11] G. Asuni, G. Teti, C. Laschi, E. Guglielmelli, and P. Dario, “A robotic head neuro-controller based on biologically-inspired neural models,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’05)*, 2005, pp. 2362–2367.

[12] Fritzke, “A growing neural gas network learns topologies,” *Advances in neural information processing systems*, vol. 7, pp. 625–632, 1995.

[13] T. Martinetz, K. Schulten, *et al.*, *A” neural-gas” network learns topologies*. University of Illinois at Urbana-Champaign, 1991.

[14] T. Kohonen, *Self-organizing maps*. Springer, 2001, vol. 30.

[15] S. Grossberg, “Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors,” *Biological cybernetics*, vol. 23, no. 3, pp. 121–134, 1976.

[16] R. Beira, M. Lopes, M. Praga, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarén, “Design of the robot-cub (icub) head,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’06)*. IEEE, 2006, pp. 94–100.

[17] G. Bishop and G. Welch, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, pp. 27 599–3175, 2001.