



Robot End Effector Tracking Using Predictive Multisensory Integration

Lakshitha P. Wijesinghe^{1*}, Jochen Triesch² and Bertram E. Shi¹

¹ Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, ² Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany

We propose a biologically inspired model that enables a humanoid robot to learn how to track its end effector by integrating visual and proprioceptive cues as it interacts with the environment. A key novel feature of this model is the incorporation of sensorimotor prediction, where the robot predicts the sensory consequences of its current body motion as measured by proprioceptive feedback. The robot develops the ability to perform smooth pursuit-like eye movements to track its hand, both in the presence and absence of visual input, and to track exteroceptive visual motions. Our framework makes a number of advances over past work. First, our model does not require a fiducial marker to indicate the robot hand explicitly. Second, it does not require the forward kinematics of the robot arm to be known. Third, it does not depend upon pre-defined visual feature descriptors. These are learned during interaction with the environment. We demonstrate that the use of prediction in multisensory integration enables the agent to incorporate the information from proprioceptive and visual cues better. The proposed model has properties that are qualitatively similar to the characteristics of human eye-hand coordination.

Keywords: active efficient coding, developmental robotics, sensorimotor prediction, generative adaptive subspace self-organizing map, reinforcement learning

OPEN ACCESS

Edited by:

Pierre-Yves Oudeyer,
Institut National de Recherche en
Informatique et en Automatique
(INRIA), France

Reviewed by:

Matej Hoffmann,
Faculty of Electrical Engineering,
Czech Technical University in Prague,
Czechia
Kazutaka Takahashi,
University of Chicago, United States
Bruno Lara,
Universidad Autónoma del Estado de
Morelos, Mexico

*Correspondence:

Lakshitha P. Wijesinghe
lpwijesinghe@ust.hk

Received: 31 May 2018

Accepted: 20 September 2018

Published: 16 October 2018

Citation:

Wijesinghe LP, Triesch J and Shi BE
(2018) Robot End Effector Tracking
Using Predictive Multisensory
Integration. *Front. Neurobot.* 12:66.
doi: 10.3389/fnbot.2018.00066

INTRODUCTION

To perform complex manipulation tasks, conventional robotic systems require precise calibration, which must be repeated when their physical configuration changes. In contrast, humans learn manipulation skills autonomously, and automatically recalibrate in response to physical configuration changes, e.g., due to growth and injury. Eye-hand coordination is a key skill required for these tasks. It requires the integration of multiple sensory modalities, such as vision and proprioception. Human infants appear to learn to develop a sense of themselves through observing the temporal contingency and spatial congruency of the sensory (e.g., visual, auditory, and proprioceptive) feedback received during self-produced motion, such as motor babbling (Rochat, 1998). One goal of cognitive developmental robots is to endow robots with this capability so that they will not require any manual calibration before acting in a new environment (Asada et al., 2009).

The mismatch between the motion of objects in the environment and the eye's rotational velocity creates retinal slip. During tracking motions, such as smooth pursuit, the brain attempts to minimize this slip by adapting the eye rotational velocity. Motion in the environment is generated by either self-motion (e.g., of the hand) or exteroceptive motion. When the hand moves, its motion can be sensed via two sensory modalities: retinal slip caused by the relative motion between the hand and eye and proprioceptive sensing of the position and movement of the arm. In contrast, an external object moving in the environment only generates a retinal slip. Moreover, hand motion

in total darkness only provides proprioceptive information to the brain. In such conditions, the human brain has the ability to generate eye movements to follow the hand or an external target using smooth pursuit like eye movements.

In this paper, we propose a novel predictive model for learning robotic visuomotor control. The proposed system model is inspired by recent findings that neurons in the primary visual cortex (area V1) are driven not only by visual but also by the motor input. Activity in V1 was long believed to be driven only by visual inputs. However, recent findings on visual perception in awake mice have suggested that this is not true. For example, the responses in V1 depend on behavioral state (Niell and Stryker, 2011). Experiments conducted in darkness revealed that motor activity alone could trigger responses in V1 neurons (Saleem et al., 2013). The development of V1 depends upon visuomotor coupling (Attinger et al., 2017). Most relevant to this work is the discovery of cells that respond to the mismatch between the actual and predicted visual flow (Keller et al., 2012; Zmarz and Keller, 2016). This suggests that visual areas predict the sensory consequences of motor actions.

Our model takes in visual input from a camera and proprioceptive inputs from the encoders of the robot arm, and produces eye motor actions to track the moving robot hand. The model is based upon the hypothesis that the brain utilizes proprioceptive inputs to predict the visual consequences of motor actions. In line with other work in predictive coding, we use the term “predict” to refer to the process of generating an estimate of one sensory input from the values of other inputs, which may occur at the same time, rather than a more strict definition where future values are estimated from past and present values. The prediction is often used to generate a mismatch signal by comparison with the actual input. For example, Srinivasan et al. (1982) explain center-surround antagonism in the retina using predictive coding, where the predicted intensity at the center based on the surround is subtracted from the actual center signal. Rao and Ballard (1999) predict lower level cortical outputs from higher level cortical outputs. Zmarz and Keller (2016) find mismatch neurons that respond to the difference between the actual visual flow and the prediction of visual flow from self-motion. Our model is most similar to the latter work, where prediction is across sensory modalities.

There are several important novel attributes of the learning methodology compared to similar work in the literature. First, the learning does not depend on any fiducial visual marker to identify the end effector of the robot. Second, the model does not require the forward kinematics of the arm to be known. Third, pre-defined visual feature descriptors are not required, but rather are learned. Finally, our experimental results with this model suggest that the use of prediction enables the model to better integrate proprioception and vision.

The rest of the paper is organized as follows. In section Related Work, we place our work into the context of past work. Section Materials and Methods describes the model, experimental setup and learning algorithms. Then in section Results, we present experimental results comparing the tracking performance. We also compare our model characteristics with human psychophysical data. Finally, in section Discussion,

we further discuss the results presenting the corresponding conclusions.

RELATED WORK

The problem of learning end effector tracking is a part of the larger problem of autonomous learning of the body schema. The body schema is a sensorimotor representation of the body that can be used to direct motion and actions. It integrates multiple cues, including proprioception, vision, audition, vestibular cues, tactile cues, and motor cues, to represent the relations between the spatial positions of the body parts. Knowledge of the body schema can be used in a number of different tasks, e.g., end effector tracking, reaching, posture control and locomotion.

The review by Hoffmann et al. (2010) classifies body schema representations used in robotics into two classes: explicit and implicit. Both have been used to address the problem of end effector tracking. In the explicit approach (e.g., Bennett et al., 1991; Hollerbach and Wampler, 1996; Gatla et al., 2007), transformations between sensory and motor coordinates are broken down into a chain of closed form transformations where each link corresponds explicitly to part of the robot structure. The work we present here falls into the class of implicit models, where an implicit representation (e.g., a look up table or neural network) is used.

Past work has often used a point representation of the end effector, where artificial markers (e.g., color blobs) have been used to enable easy identification of the end effector (Hersch et al., 2008; Sturm et al., 2009). For example, a biologically inspired model to learn visuomotor coordination for the robot Nao was proposed in Schillaci et al. (2014). Learning occurred during motor babbling, which is similar to how infants may learn early eye-hand coordination skills. The proposed method used two Dynamic Self Organizing Maps (DSOMs) to represent the arm and neck position of the robot. The connections between the DSOMs were strengthened if the robot was looking at a fiducial marker positioned on the end effector. After learning, the robot had the ability to track the end effector by controlling the neck joints. One advantage of this model is that the method has no assumption that the forward arm kinematics of the robot is known. However, one limitation of the approach is that it required a fiducial marker.

Subsequent work has relaxed the assumption that the end effector is a point and removed the requirements for explicit markers. However, it has still required hard-coded visual feature descriptors. For example, an algorithm to learn the mapping from arm joint space to the corresponding region in image space containing the end effector was proposed in Zhou and Shi (2016), based on a measure of visual consistency defined using SIFT features (Lowe, 2004). This algorithm did not require prior knowledge of the arm model, and was robust to changes in the appearance of the end effector. Other marker-less approaches have relied upon knowledge of a 3D CAD model of the end effector (Vicente et al., 2016; Fantacci et al., 2017). Vicente et al. (2016) eliminated calibration errors using a particle filter. The likelihood associated with each particle was evaluated by

comparing the outputs of Canny edge detectors applied to both the real and simulated camera images. Fantacci et al. (2017) extended this particle filter and 3D CAD model based approach to estimate the end effector pose. The likelihood was evaluated using a Histogram of Oriented Gradient (HOG) (Dalal and Triggs, 2005) based transformation to compare the two images. The approach to bootstrap a kinematic model of a robot arm proposed in Broun et al. (2014) does not require a priori knowledge of a CAD model, as it constructs a model of the end-effector on the fly from Kinect point cloud data. However, it still requires a hard-coded optical flow extraction stage to identify the arm in the image through visuomotor correlation.

Some of the limitations in the aforementioned research (e.g., the requirement for a marker and/or hard-coded image features) were addressed in our prior work (Wijesinghe et al., 2017), which proposed a multisensory neural network that combined visual and proprioceptive modalities to track a robot arm. Retinal slip during the motion was represented by encoding two temporally consecutive image frames using a sparse coding algorithm where the basis vectors were learned online (Zhang et al., 2014). The sparse coefficients were combined with proprioceptive input to control the eye to track the arm. This paper extends our previous idea by introducing a new model following the hypothesis that the brain generates internal predictions for consequences of actions.

MATERIALS AND METHODS

Our approach is based on the Active Efficient Coding (AEC) framework (Zhao et al., 2012; Teulière et al., 2015), a generalization of the efficient coding hypothesis to active perception. Under the efficient coding hypothesis, the sensory data is encoded efficiently by exploiting redundancies in the statistics of the sensory input signals. In AEC, movements of the sensory organs are also learned so that the inputs can be coded efficiently. In the proposed model, visual, and proprioceptive stimuli are jointly encoded. This perceptual representation is used to generate eye movements for tracking the robot arm. Simulation of the model is performed using the iCub humanoid robot simulator, an open source robot simulator for the iCub robot (Tikhanoff et al., 2008). We provide more detail in the following subsections.

Model Architecture

Figure 1 illustrates the architecture of the proposed model, which evolves in discrete time. We assume that each iteration corresponds to 40 ms.

At each iteration, the right eye of the iCub captures an image with 320×240 pixel resolution. Two foveal subwindows are extracted from the center of this image: a smaller 55×55 pixel fine scale image, $I_{\text{fine}}(t)$, and a larger 110×110 pixels coarse scale subwindow, $I_{\text{coarse}}(t)$, which is subsampled horizontally and vertically by a factor of two. These subwindows cover 11° and 22° , respectively, horizontally and vertically.

The visual stimuli are encoded using Generative Adaptive Subspace Self Organizing Maps (GASSOMs) (Chandrapala and Shi, 2015). Proprioceptive inputs include the arm position,

velocity, and acceleration ($\theta_a(t) \in \mathbb{R}^4$, $\dot{\theta}_a(t) \in \mathbb{R}^4$, $\ddot{\theta}_a(t) \in \mathbb{R}^4$) and the eye position and eye velocity ($\theta_e(t) \in \mathbb{R}^2$, $\dot{\theta}_e(t) \in \mathbb{R}^2$) as reported by the motor encoders. The prediction module predicts the sensory consequences of the arm and eye motions based on proprioceptive inputs. This enables the model to reduce the correlation between the visual features and the proprioceptive inputs during self-motion. Finally, the visual and proprioceptive inputs are integrated using an Artificial Neural Network (ANN) to generate pan and tilt eye acceleration commands, $\ddot{\theta}_e(t) \in \mathbb{R}^2$, enabling the right eye to track the robot arm. The “ \wedge ” symbol is added to distinguish motor commands from proprioceptive information.

Given the eye acceleration command, the eye velocity is obtained by:

$$\hat{\theta}_e(t+1) = \hat{\theta}_e(t) + \ddot{\theta}_e(t). \quad (1)$$

Equation (1) is similar to the model for the maintenance of pursuit described in Lisberger (2010), where an efference copy of the eye velocity command is fed back in order to determine the current command for eye velocity in the immediate future. This enables eye velocity to be maintained automatically. In our model, both image motion and arm motion can drive changes in eye velocity through the eye acceleration command.

The model presented here uses only afferent information to determine the eye acceleration. In biological systems, both afferent and efferent signals from the arm are involved in arm-eye coordination control. For deafferented monkeys, smooth pursuit eye movements disappeared while tracking a target moved by active arm movements in darkness (Gauthier and Mussa Ivaldi, 1988). Steinbach (1969) found differences in ocular tracking of active and passive hand motions, which suggest that efference commands also play a crucial role. Gauthier and Mussa Ivaldi (1988) and Gauthier et al. (1988) suggested that efferent signals serve to synchronize the onsets of arm and eye motions, whereas proprioceptive signals serve to couple the eye and hand motor signals once movement has started. Subsequent experiments have provided additional support for this hypothesis (Vercher et al., 1996). Since we use only afferent information, the model may provide an account for differences in performance once movement has started. We leave the integration of an efference copy to future extensions of the model.

During training, we control four degrees of freedom (DoF) among the seven DoF in the iCub robot arm. The three joints in the wrist are fixed, and the remaining four joints (the shoulder pitch, shoulder roll, shoulder yaw, and the elbow joint) are controlled. We fix the wrist angles so that the palm of the iCub robot remains approximately parallel to the image plane. As described below, one assumption of our approach is that the retinal flow is uniform across both fine and coarse foveal image regions. Keeping the wrist angle fixed ensures that the image of the palm covers the foveal images when the gaze vector intersects the center of the palm. Modifying the algorithm so that the image region used to generate motor commands vary in size automatically may enable the algorithm to allow all DoF to vary. During testing, we allow the wrist to move (see section Qualitative Evaluation of Performance).

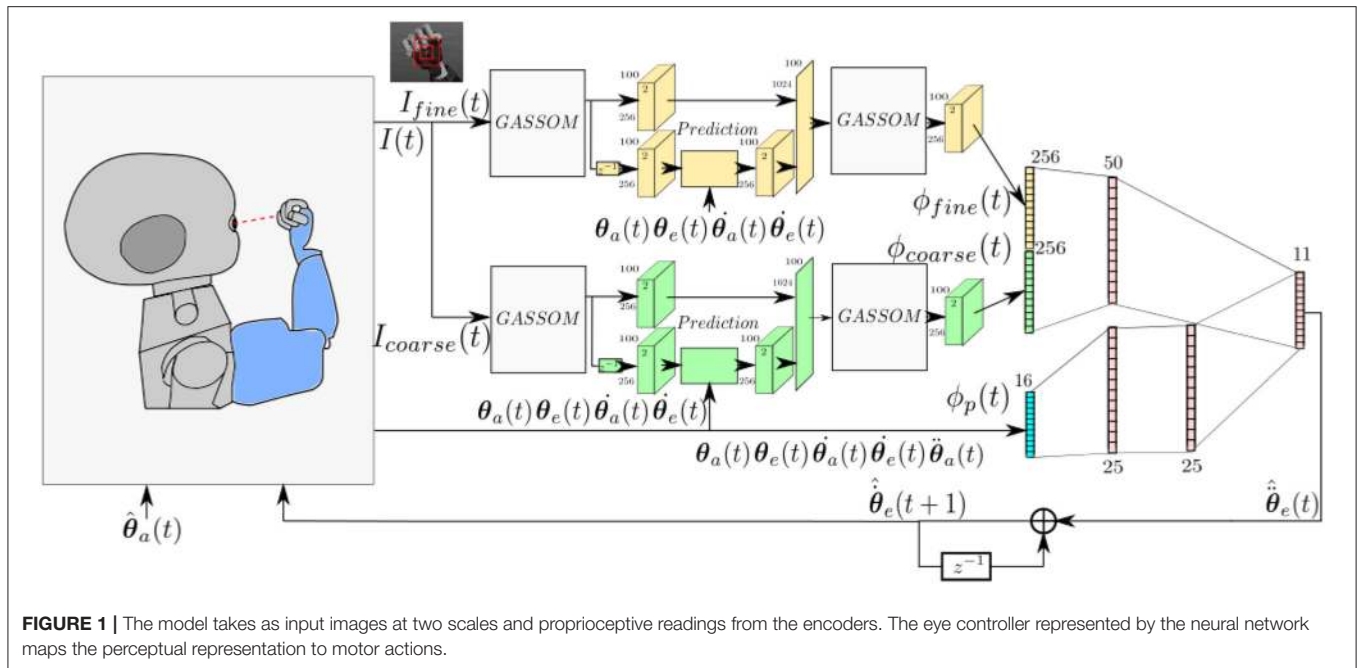


FIGURE 1 | The model takes as input images at two scales and proprioceptive readings from the encoders. The eye controller represented by the neural network maps the perceptual representation to motor actions.

In the following, we describe the model components in more detail.

Visual and Proprioceptive Features

The model encodes the visual stimuli in two stages. In the first stage, each foveal image is divided into a 10×10 array of 10×10 pixel overlapping patches, $\mathbf{x}_{1,s,i}(t) \in \mathbb{R}^{100}$, with a stride of 5 pixels, where s indexes scale and $i \in \{1, 2, \dots, P\}$ indexes the patch ($P = 100$). The subscript “1” indicates the stage.

Patches are encoded using GASSOMs (Chandrapala and Shi, 2015). The GASSOM is a probabilistic generative extension of the Adaptive Subspace Self Organizing Map (ASSOM) (Kohonen, 1996). It assumes that each input vector $\mathbf{x}_{1,s,i}(t)$ is generated by one of $N = 256$ nodes. The generating nodes evolve according to a first order Markov process. Each node is associated with a two dimensional subspace spanned by basis vectors specified by the orthogonal columns of the matrix $\mathbf{B}_{1,s,n} \in \mathbb{R}^{100 \times 2}$. The input is generated by the node by choosing a Gaussian distributed vector lying on the subspace plus a small noise vector lying in the orthogonal complement of the subspace. Using the algorithm described in Chandrapala and Shi (2015), both the transition matrix of the Markov process and the node subspaces are learned so as to maximize the likelihood of the input sensory data. The learned transition matrices have high self-transition probability, which implies that the node generating input $t - 1$ is likely to generate the input at time t , a property we refer to as “slowness.”

The output of the GASSOM is the set of projections of the input vector onto the N subspaces.

$$\mathbf{p}_{1,s,n,i}(t) = \mathbf{B}_{1,s,n}^T \mathbf{x}_{1,s,i}(t). \tag{2}$$

When the eye is viewing the end effector, these projections change in a regular manner, which depends upon the movement of the arm and the eye.

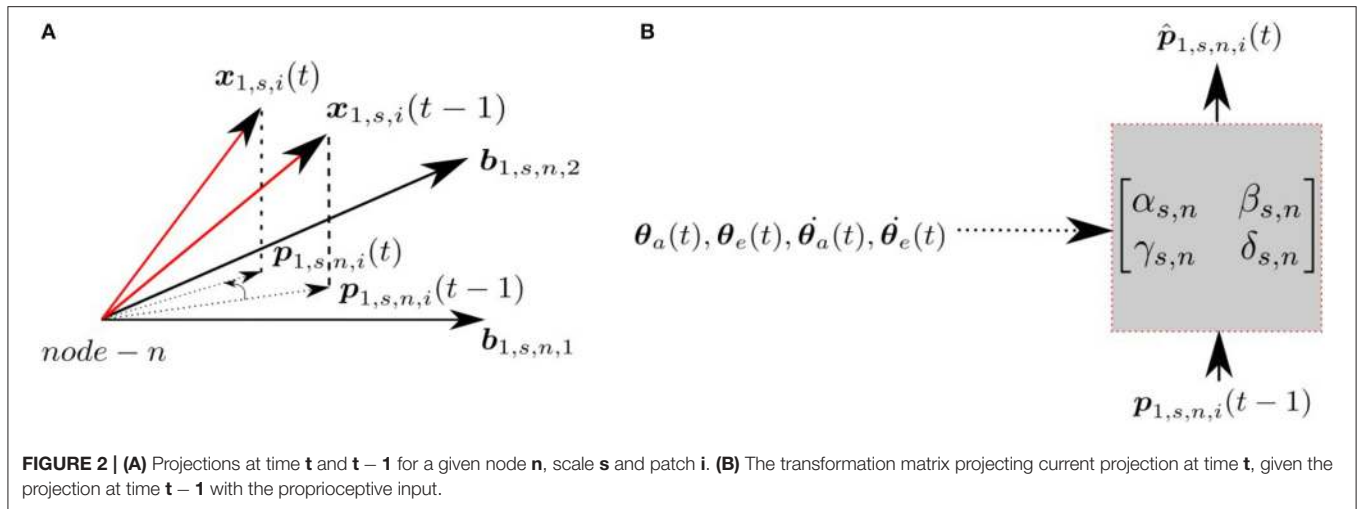
Each node at each scale has an associated prediction module, which predicts the projection of the input at time t , $\mathbf{p}_{1,s,n,i}(t)$, given the input at time $t - 1$, $\mathbf{p}_{1,s,n,i}(t - 1)$ and the proprioceptive signals encoding the arm/eye position/velocity ($\theta_a(t), \dot{\theta}_a(t), \theta_e(t), \dot{\theta}_e(t)$). **Figure 2** indicates the projections of the patch $\mathbf{x}_{1,s,i}$ at times t and $t - 1$ and the corresponding transformation in the subspace. The prediction module assumes the transformation can be modeled as a linear mapping, where the predicted projection at time t is given by;

$$\hat{\mathbf{p}}_{1,s,n,i}(t) = \begin{bmatrix} \alpha_{s,n} & \beta_{s,n} \\ \gamma_{s,n} & \delta_{s,n} \end{bmatrix} \mathbf{p}_{1,s,n,i}(t - 1), \tag{3}$$

where the α , β , γ and δ parameters for each scale s and node n depend upon the arm/eye position/velocity. These parameters are computed using a neural network with four inputs ($\theta_a(t), \dot{\theta}_a(t), \theta_e(t), \dot{\theta}_e(t)$), one hidden layer containing 25 hidden units with tanh activations, and four linear output neurons. Since all patches share the same α , β , γ , and δ parameters, we are assuming that the retinal flow is uniform across the foveal images. By performing the prediction in the projected subspace, rather than the original high dimensional pixel space, we simplify the task of prediction.

The parameters of the 512 (2 scales \times 256 nodes) neural networks are learned online using stochastic gradient descent, where the weights are updated every iteration. Since each foveal image contains 100 patches, we average the gradients of the prediction error across the 100 patches, and update the weights of each neural network with the average gradient.

The second GASSOM encodes the concatenated vectors $\mathbf{p}_{1,s,n,i}(t)$ and $\hat{\mathbf{p}}_{1,s,n,i}(t)$ corresponding to all the nodes $n \in \{1, 2, \dots, 256\}$ in the first GASSOM. Hence, the input vector to the second GASSOM is $\mathbf{x}_{2,s,i}(t) \in \mathbb{R}^{1024}$ for a given scale



s and patch i . The second GASSOM also contains $M = 256$ nodes, each with an associated 2-dimensional subspace spanned by the columns of the matrix $\mathbf{B}_{2,s,m} \in \mathbb{R}^{1024 \times 2}$ where $m \in \{1, 2, \dots, 256\}$ indexes the node. As in Equation (2), the projections are computed by;

$$\mathbf{p}_{2,s,m,i}(t) = \mathbf{B}_{2,s,m}^T \mathbf{x}_{2,s,i}(t). \quad (4)$$

Both the transition matrix and the node subspaces are learned online as the iCub behaves in the environment.

From the projections $\mathbf{p}_{2,s,m,i}(t) \in \mathbb{R}^2$ at each node, we compute a feature vector $\phi_s(t) \in \mathbb{R}^{256}$ for each scale of the foveal image by computing the average squared length of the projections over all the patches.

$$\phi_s(t) = \begin{bmatrix} \frac{1}{P} \sum_{i=1}^P \|\mathbf{p}_{2,s,1,i}(t)\|^2 \\ \frac{1}{P} \sum_{i=1}^P \|\mathbf{p}_{2,s,2,i}(t)\|^2 \\ \dots \\ \frac{1}{P} \sum_{i=1}^P \|\mathbf{p}_{2,s,256,i}(t)\|^2 \end{bmatrix}. \quad (5)$$

The final feature representation of the visual stimuli is the concatenation of the feature vectors at the two scales.

$$\phi_v(t) = \begin{bmatrix} \phi_{\text{fine}}(t) \\ \phi_{\text{coarse}}(t) \end{bmatrix}. \quad (6)$$

The proprioceptive feature vector $\phi_p(t) \in \mathbb{R}^{16}$ concatenates the arm position, velocity and acceleration measurements from the encoders, $\theta_a(t), \dot{\theta}_a(t), \ddot{\theta}_a(t) \in \mathbb{R}^4$ and the eye position and velocity $\theta_e(t), \dot{\theta}_e(t) \in \mathbb{R}^2$. Each proprioceptive input is normalized by subtracting the mean and dividing by the standard deviation computed over the training data set.

Eye Motor Controller

The eye controller maps the visual $\phi_v(t) \in \mathbb{R}^{512}$ and proprioceptive $\phi_p(t) \in \mathbb{R}^{16}$ feature vectors to an eye acceleration command $\ddot{\theta}_e(t)$ using the artificial neural network shown in **Figure 1**. Only one network is shown, corresponding to the

generation of the acceleration command for one axis (pan or tilt). The other axis is controlled by a network with the same structure, but different weights.

Each neural network has 11 output neurons, corresponding to 11 possible acceleration actions, $a_i \in \mathbf{A}$ for $i \in \{0, 1, \dots, 11\}$, where;

$$\mathbf{A} = \{-1.6, -0.8, -0.4, -0.2, -0.1, 0, 0.1, 0.2, 0.4, 0.8, 1.6\} \text{ deg/sample}^2. \quad (7)$$

The outputs of each neural network, $\pi_{k,i}(t)$ where $k \in \{\text{pan, tilt}\}$ and $i \in \{0, 1, \dots, 11\}$ encode the probabilities that the actions are chosen at each time t . Mathematically;

$$P[\ddot{\theta}_{e,k}(t) = a_i] = \pi_{k,i}(t), \quad (8)$$

where $\ddot{\theta}_e(t) = [\ddot{\theta}_{e,\text{pan}}(t) \ \ddot{\theta}_{e,\text{tilt}}(t)]^T$. The eye acceleration command is generated by sampling from this probability distribution.

Within the neural network, the visual input first passes through a single fully connected 50 neuron hidden layer and the proprioceptive input first passes through two fully connected 25 neuron hidden layers with tanh activations before the two pathways are combined at the output layer, which is fully connected with a softmax output non-linearity. Mathematically;

$$\pi_{k,i}(t) = \frac{\exp(z_{k,i}(t)/\tau)}{\sum_{j=1}^{11} \exp(z_{k,j}(t)/\tau)}, \quad (9)$$

where $\tau = 1$ is a temperature parameter. The vector, $\mathbf{z}_k(t) \in \mathbb{R}^{11}$, is given by;

$$\mathbf{z}_k(t) = \mathbf{W}_{k,2}^T \tanh(\mathbf{W}_{k,1}^T \phi_v(t)) + \mathbf{W}_{k,5}^T \tanh(\mathbf{W}_{k,4}^T \tanh(\mathbf{W}_{k,3}^T \phi_p(t))), \quad (10)$$

where $\mathbf{W}_{k,1} \in \mathbb{R}^{512 \times 50}$, $\mathbf{W}_{k,2} \in \mathbb{R}^{50 \times 11}$, $\mathbf{W}_{k,3} \in \mathbb{R}^{16 \times 25}$, $\mathbf{W}_{k,4} \in \mathbb{R}^{25 \times 25}$ and $\mathbf{W}_{k,5} \in \mathbb{R}^{25 \times 11}$ are weight matrices. Our implementation includes constant bias terms at all layers, which we have not shown explicitly in the notation to avoid clutter.

The weights of the eye motor controller are learned online as the iCub behaves in the environment, using the natural actor-critic reinforcement learning algorithm (Bhatnagar et al., 2009). The network generating the probabilities described above is the actor (policy) network. The actor-critic algorithm also requires a second network to approximate the value function, which depends upon both $\phi_v(t)$ and $\phi_p(t)$. We use a single layer linear network with 528 inputs feeding into a single linear output neuron for the critic network.

The instantaneous reward is given by;

$$r(t) = -\frac{1}{2} (e_{\text{fine}}(t) + e_{\text{coarse}}(t)), \quad (11)$$

where;

$$e_s(t) = \frac{1}{P} \sum_{i=1}^P \max_n \left\| \mathbf{p}_{1,s,n,i}(t) - \mathbf{p}_{1,s,n,i}(t-1) \right\|^2, \quad (12)$$

for $s \in \{\text{fine, coarse}\}$. This reward penalizes changes in the patch projections, which should be constant if the image of the end effector is stabilized.

Training and Testing Environment

Training runs for a total of 400,000 iterations. It interleaves two different types of sessions (Figure 3), each lasting 5,000 iterations at a time. For both session types, the environment contains a planar object mapped with a natural image texture chosen at random from the (Olmos and Kingdom, 2004) database of size 2142×1422 pixels located at 0.4–0.8 m distance in front of the iCub. When placed directly in front of the robot, the plane subtends 136° of visual angle horizontally and 118° of visual angle vertically. The texture is changed every 500 iterations. In session type 1, the arm remains stationary in a position where it is outside the field of view of the robot. Only the textured plane, which moves randomly horizontally and vertically, is visible. In session type 2, the robot arm babbles so that the end effector moves randomly in front of the iCub. Depending on the eye and arm position, the center of the eye gaze may fall on the robot arm or on the resting plane. If the eye gaze falls on the center of the palm of the iCub, the hand fills both the coarse and fine scale foveal windows, but this is not always the case. This training setup is intended to mimic a general environment, where an agent is exposed to both self-generated and exteroceptive motion in the visual environment.

In session type 2, we use motor babbling to generate the visual and proprioceptive data for the learning algorithm. The arm babbles around a home pose $\theta_a^h = [-82^\circ \ 22^\circ \ 40^\circ \ 90^\circ]$, which is chosen so that the center of the iCub's hand falls on the image center of the right eye when its pan and tilt angles are zero. The arm moves through a randomly generated trajectory ($\hat{\theta}_a(t)$ in Figure 1) in the arm joint space. The babbling trajectory is generated by feeding a set of via points sampled from a uniform distribution $[\theta_{a,i}^h - 12^\circ, \theta_{a,i}^h + 12^\circ]$ for $i \in \{1, 2, 3, 4\}$ into the "mstraj" function of the Robotics Toolbox (Corke, 2017) to generate a trajectory consisting of linear segments connected by polynomial blends.

In session type 1, the planar object follows a trajectory created by first generating an arm trajectory as described above, and then moving the plane so that its center point follows the same angular trajectory as the center of the iCub's hand. This ensures that the statistics of the visual motion induced by the plane are similar to the statistics of those induced by the hand.

The eye rotational angles are restricted to $\pm 40^\circ$ and $\pm 30^\circ$ in pan and tilt, respectively. The rotational velocity of the eye is also limited to ± 3 deg/sample in both pan and tilt. During training, we reset the eye position to a "home" position $\theta_c^h = [0^\circ \ 0^\circ]$ and the velocity to zero every 500 iterations. This ensures the eye orientation does not drift off so far that the eye never sees the hand. For each trajectory during testing, we initialize the eye velocity to zero and the eye position so that the gaze vector intersects the center of the palm.

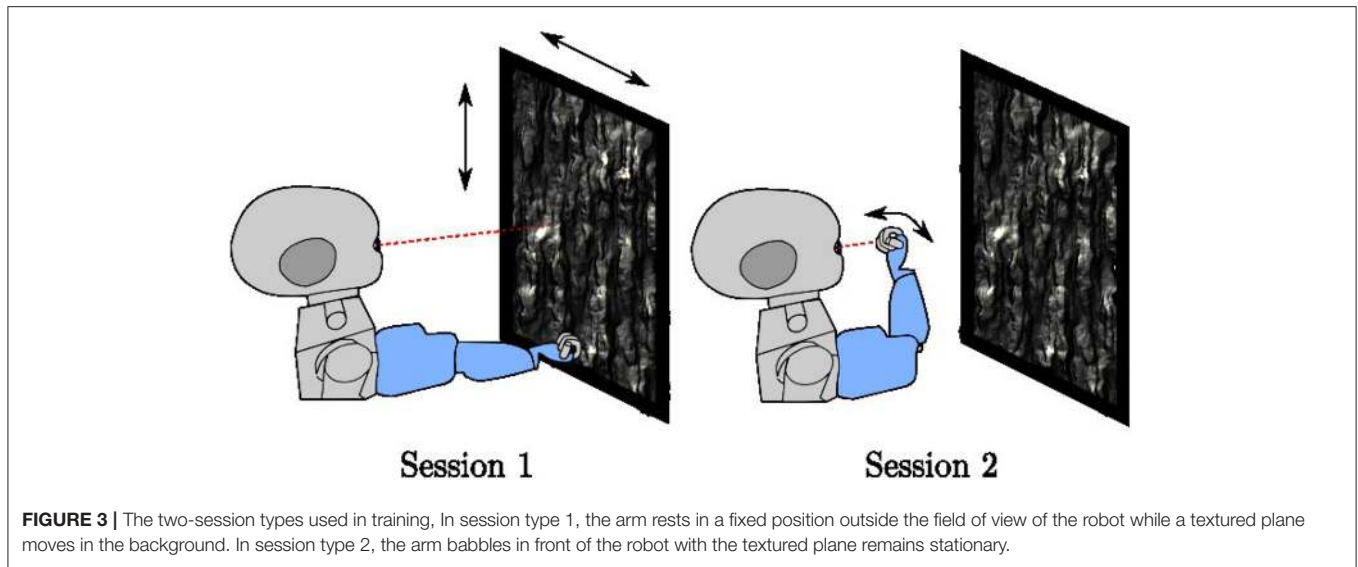
We chose this method of random babbling and trajectory generation for its simplicity. There are a number of ways we can make the motion more biologically realistic, e.g., through the use of dynamic movement primitives (Schaal, 2006) for trajectory generation, or through the use of goal babbling (von Hofsten, 2004) to choose the via points. The use of dynamic movement primitives would alter the statistics of the image motion induced by the hand, which might change the smooth pursuit performance. The use of goal babbling might improve the speed of learning (Baranes and Oudeyer, 2013). These would be interesting extensions of the model to investigate. However, we do not expect their incorporation to change the main qualitative findings we report here.

For the sake of simplicity in our simulations, we use the iCub robot simulator to take into account the kinematics of the iCub robot as well as to model the geometry and appearance of the visual environment. We do not take into account the dynamics of the robot, nor do we incorporate a biologically realistic model for the eye movement dynamics. Rather, in each iteration, we move the robot arm to the configuration determined by the random babbling arm motion via position control. We assume that the eye velocity command is executed perfectly, by determining the location of the eye in the next iteration as the sum of the encoder measurement of the current position plus the velocity command in Equation (1), and move the eye there via position control. The images taken by the iCub in the new arm/eye positions determine the next visual input to the model. The proprioceptive input is determined from the motor encoders and their first and second differences in time. We believe that incorporating more realistic models of arm and eye dynamics are a natural next step. If these models are more biologically realistic, the model may give a better quantitative account of the performance of human subjects.

RESULTS

Learned Visual Representation

The basis vectors in the first stage GASSOM are analogous to the receptive fields of orientation-tuned simple cells in the human primary visual cortex (Chandrapala and Shi, 2015). As shown in Figures 4A,B, the basis vectors of the first GASSOM are tuned to specific orientations and spatial frequencies. The basis vectors corresponding to fine and coarse scales have similar



characteristics. The corresponding basis vectors associated with a node in the first GASSOM have a phase difference close to 90° to represent the two orthogonal basis vectors. We initialize these basis vectors with the basis vectors learned on natural images. In addition, the parameters corresponding to the first GASSOM are fixed during the visual representation learning.

The prediction module predicts the projections at time t of the first stage GASSOM given the projections at time $t - 1$ and the proprioceptive inputs. We evaluate the accuracy of the prediction module using the average cosine similarity between $\hat{\mathbf{p}}_{1,s,m,i}(t)$ and $\mathbf{p}_{1,s,m,i}(t)$. For this test, the eye and the arm are moved independently of each other. The eye velocities are sampled from Gaussian distributions fitted to the distribution of the eye velocity during training in both pan and tilt directions. The standard deviations of the fits are $\sigma_{\hat{\theta}_{e,\text{pan}}} = 0.7337$ deg/sample ($r^2 = 0.982$) and $\sigma_{\hat{\theta}_{e,\text{tilt}}} = 0.6387$ deg/sample ($r^2 = 0.989$). Arm trajectories are generated in a similar way to the training. In order to maintain the gaze on the robot hand, we execute a saccade to bring the eye gaze back to the center of the hand once the eye gaze drifts outside of the arm region. We evaluate the predictors over 10,000 iterations. The basis vectors from the subspaces with the highest and lowest average cosine similarity for two scales are outlined in green and blue in **Figures 4A,B**.

We fit a two-dimensional Gabor function to each basis vector to identify the factors influencing the prediction accuracy. **Figures 4C,D** show that the average cosine similarity of a predictor is related to the spatial frequency ($\frac{1}{\lambda}$; where λ is the spatial wavelength) of the basis vector. Higher spatial frequencies have a lower cosine similarity for predictors in both fine and coarse scales. Intuitively, basis vectors with higher spatial frequencies are more sensitive to retinal motion than those with low spatial frequencies. The transformations associated with higher spatial frequency basis vectors are more difficult to predict. The cluster of data points close to $\frac{1}{\lambda} = 0$ in **Figures 4C,D** are

basis vectors whose fitted Gabor functions had very long spatial wavelengths. These typically corresponded to basis vectors with main support near the edges of the patch.

The second stage GASSOMs jointly encode the actual and predicted projections onto the subspaces of the first GASSOM. If the predictions from knowledge of the proprioception are accurate, there should be little difference between the actual and predicted projections. Differences between the two arise due to exteroceptive motion, which cannot be predicted from proprioception, as well as inaccuracy in the predictor. The basis vectors in the second GASSOM encode these differences and inherit orientation and spatial frequency tuning from the first stage. Encoding only the residual motion after prediction helps to reduce correlation between the visual and proprioceptive cues.

We examine the tuning of the basis vectors in the second GASSOM using drifting two-dimensional cosine gratings in 10×10 pixel image patches. For this test, we fix the proprioceptive input to zero self-motion. We record the responses from the subspaces of the second GASSOM to all combinations of motion, spatial frequency and orientation, where spatial wavelengths varied from 3 to 20 pixels, motion from -2 to 2 deg/sample, orientation from 0° to 180° . For each subspace, we determine the preferred tuning from the combination that resulted in the maximum magnitude response.

The tuning characteristics for the fine scale basis vectors are provided in **Figures 5A,B**. Here, we also present the tuning statistics corresponding to a model without the prediction module shown in **Figures 5C,D**. The majority of the basis vectors are tuned to zero velocity for both architectures as shown in **Figures 5A,C**. The tuning velocities of the architecture with, without prediction have a variance of $0.3660, 0.4286$ (deg/sample)². The tuning orientations are distributed close to a uniform distribution as shown in **Figures 5B,D**. The KL divergence with the uniform distribution of orientations for the architecture with, without prediction is $0.0113, 0.0397$. Hence, the two architectures prefer zero retinal slip for all the tuning

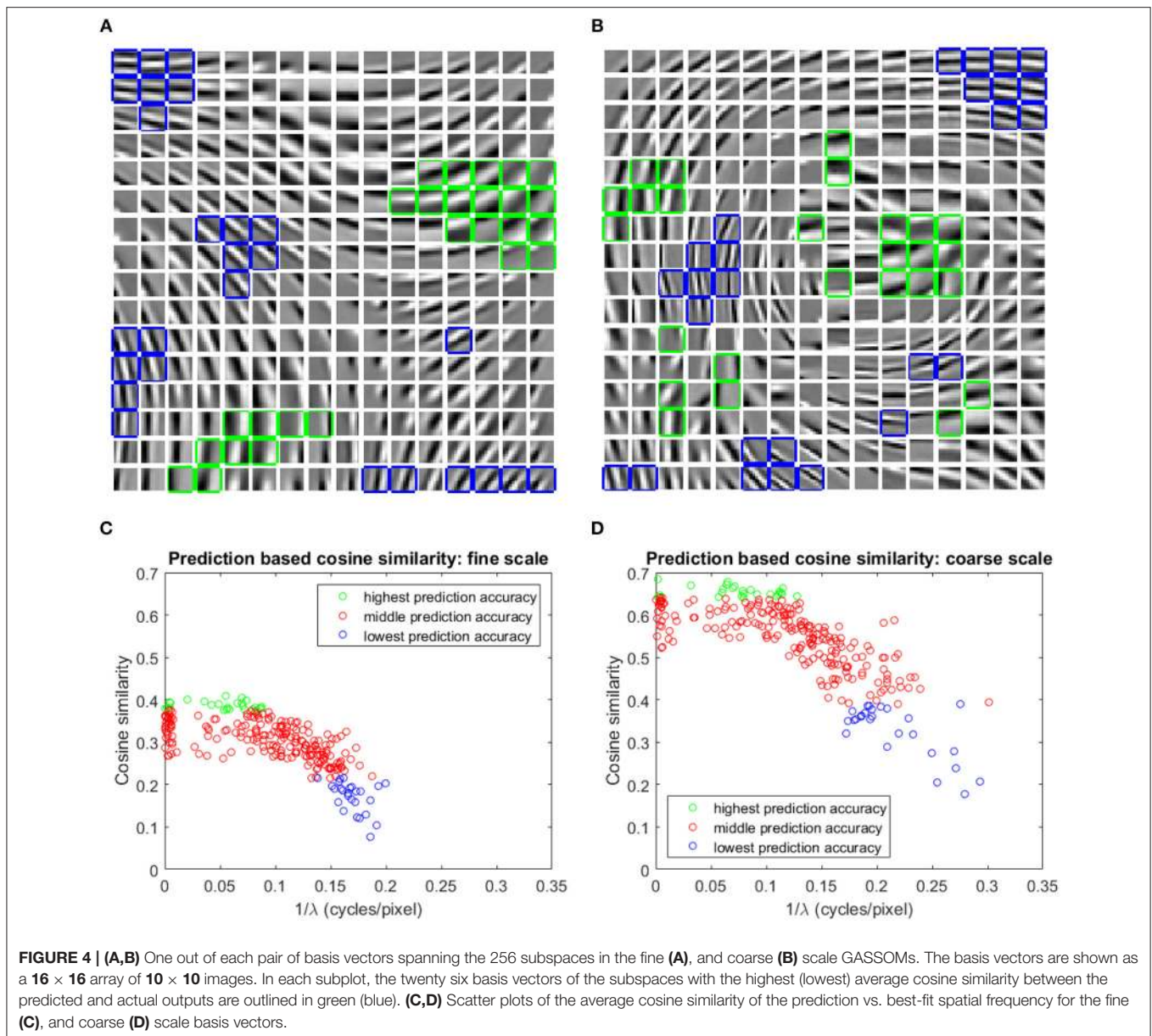


FIGURE 4 | (A,B) One out of each pair of basis vectors spanning the 256 subspaces in the fine (A), and coarse (B) scale GASSOMs. The basis vectors are shown as a 16×16 array of 10×10 images. In each subplot, the twenty six basis vectors of the subspaces with the highest (lowest) average cosine similarity between the predicted and actual outputs are outlined in green (blue). **(C,D)** Scatter plots of the average cosine similarity of the prediction vs. best-fit spatial frequency for the fine (C), and coarse (D) scale basis vectors.

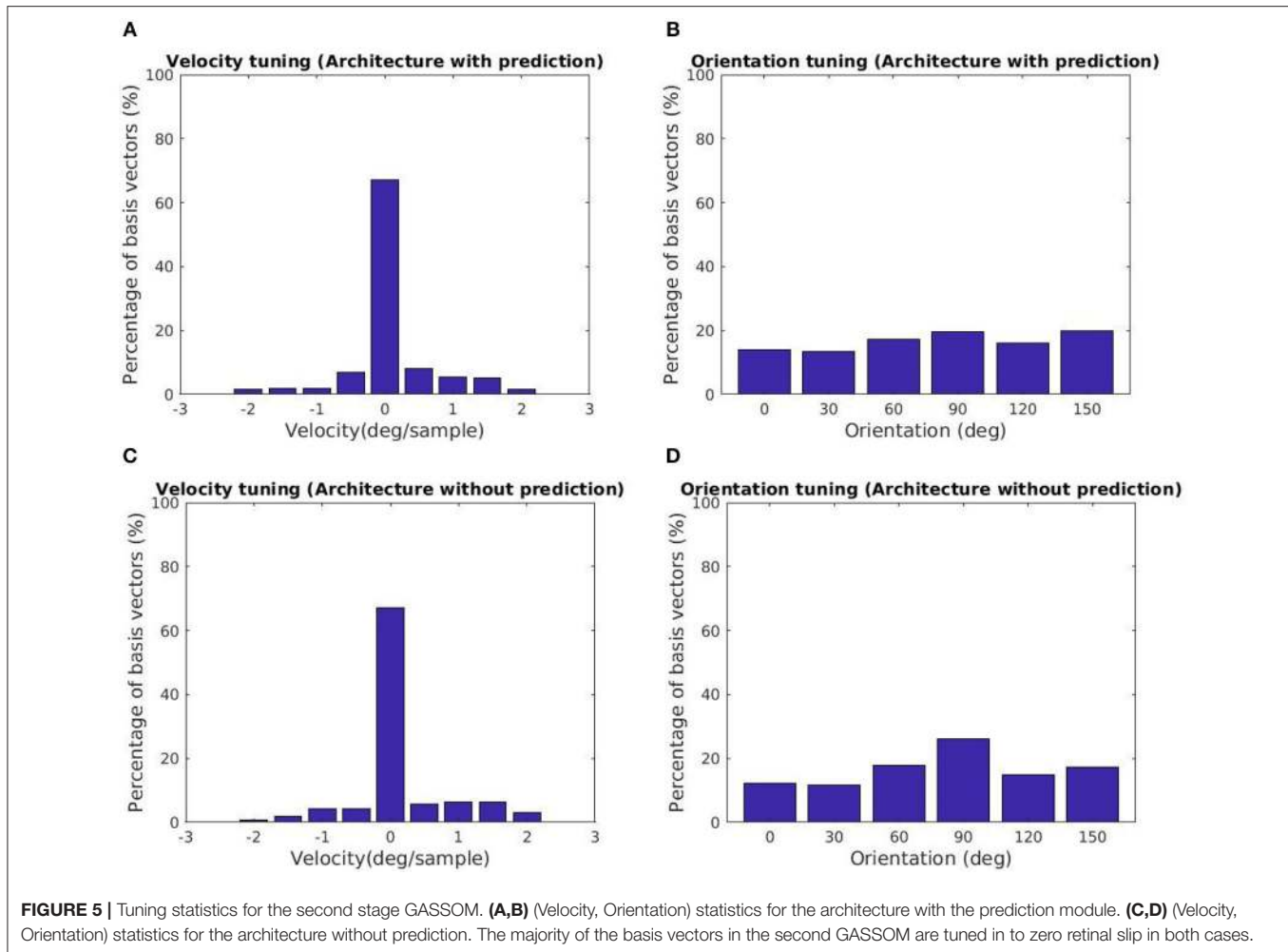
orientations. For the architecture with prediction, the variance of the tuning velocities is lower compared to the other architecture.

Learning the Eye Motor Controller

The eye controller is tested by evaluating the tracking performance for a set of 10 different arm trajectories, each lasting 1,000 iterations. The trajectories are generated in the same way as in training. The performance is measured by computing the root mean squared error (RMSE) between the target and actual eye velocities.

We compare the performance of the system in three different scenarios. In all cases, the robot attempts to track the end effector of the robot. In the first case, the model is driven by both visual and proprioceptive stimuli. In the second case, the model is purely driven by vision, with illusory proprioceptive

input being provided that suggests that the arm is fixed at the resting position outside the field of view used in training session type 1. In the third case, the model is purely driven by proprioception, with the visual feature $\phi_v(t)$ replaced by the expected value computed over time $E[\phi_v(t)]$. This approach allows us to compare the three cases by providing the same visual stimuli (the robot hand) in distinct scenarios. **Figure 6** depicts the learning progress recorded at 6 checkpoints occurring every 80,000 iterations. The RMSE for both pan and tilt angular velocities are averaged over 30 different trials comprising 10 trajectories and 3 different training trials. The learning curves in **Figure 6** illustrate that in steady state, using both visual and proprioceptive stimuli is much more accurate than using either stimuli alone. This is typical of multimodal integration.



Comparison With Psychophysical Experiments

In this section, we compare the eye movements generated by the model to that of the human oculomotor system using an experimental protocol similar to that described in Vercher et al. (1993), which is illustrated in Figure 7. Vercher et al. measured the frequency responses of the human oculomotor plant during visual tracking for five subjects (4 males and 1 female) in two different cases. In the first case, the subject was tracking a target moving in a sinusoidal trajectory as shown in Figure 7A. In the second case, the target trajectory was controlled by the subject using his/her arm while tracking the corresponding target with the eyes as shown in Figure 7B. The eye movements in these two cases were compared to understand the role of proprioception in oculomotor control.

In our experiment, with the model to generate comparable data, we use the end effector of the robot arm as the target. The end effector is moved in a sinusoidal trajectory between -6° and $+6^\circ$ in the pan direction with respect to the eye. The frequency of the motion varies from 0.5 to 2 Hz in 0.5 Hz steps. Sample trajectories generated by the model at 1 Hz frequency are shown in Figure 8. We fit a sine function to the eye velocity to

compute the velocity gain and phase difference with reference to the target trajectory. The eye trajectory in Figure 8A shows a higher gain and a lower phase delay compared to the trajectory in Figure 8B. Hence, the addition of proprioceptive information improves the velocity gain in comparison to vision alone. The addition of proprioception also reduces the phase delay. Our system is also able to move the eye solely with the proprioceptive input as illustrated in Figure 8C.

We compare the results of our model with human performance, by extracting the frequency response data from Figures 2, 3 in Vercher et al. (1993), which show the eye velocity gains and phase delays averaged across the five subjects. Figure 9 compares the frequency responses. The model and human data have qualitative similarities. According to the gain plots in Figures 9A,C, both responses have a higher gain in the presence of both vision and proprioception. In addition, proprioception combined with vision has a lower phase delay as shown in Figures 9B,D. For the subjects in this study, both efferent and afferent information is available during tracking of the self-moved target, whereas our model only includes afferent information from proprioception. Vercher et al. (1996) studied the role of proprioception in eye-hand coordination

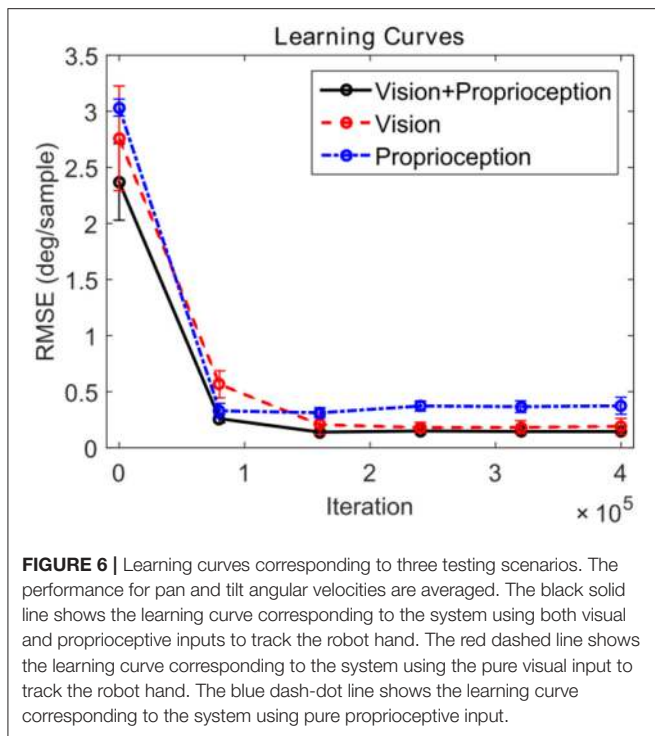


FIGURE 6 | Learning curves corresponding to three testing scenarios. The performance for pan and tilt angular velocities are averaged. The black solid line shows the learning curve corresponding to the system using both visual and proprioceptive inputs to track the robot hand. The red dashed line shows the learning curve corresponding to the system using the pure visual input to track the robot hand. The blue dash-dot line shows the learning curve corresponding to the system using pure proprioceptive input.

by comparing the behavior between deafferented and control subjects. The deafferented subjects showed little difference in performance between tracking a self-moved target and an external target. This highlights the importance of proprioception as a non-visual signal for smooth pursuit control.

Qualitative Evaluation of Performance

To observe the tracking performance qualitatively, we generate a video comparing the eye trajectories as the arm moves randomly for the three different combinations of cues¹. This video demonstrates the performance by projecting the gaze position to an image frame obtained from a fixed camera for three different testing cases. The right eye camera is moving differently in all three testing cases. Relating the gaze to a reference camera frame makes it easier to compare the different cases. From the video, the pure proprioception based tracking underperforms in comparison to the other two cases. This is consistent with the velocity gain and phase delay responses in **Figure 9**.

The performance in the video can be summarized by projecting the gaze vectors to the end effector coordinate system. The origin of this coordinate system is located on the palm of the robot end effector. **Figure 10** shows distributions of the intersections between the eye gaze direction and the plane passing through the origin parallel to the palm surface over the entire tracking trajectory as two-dimensional heat maps, one for each of the testing cases. In comparison to **Figures 10A,B**, **10C** shows larger variability in the gaze position on the end effector.

¹The video is available online at <https://youtu.be/PPb7KwWefBI>

This is consistent with the poorer tracking performance for proprioception alone than in the other two cases. The model does not explicitly define a precise end effector location to be tracked. Since only acceleration commands are generated, the gaze position can drift. Thus, the mean gaze position projected to the palm varies across different training trials for the three different testing cases.

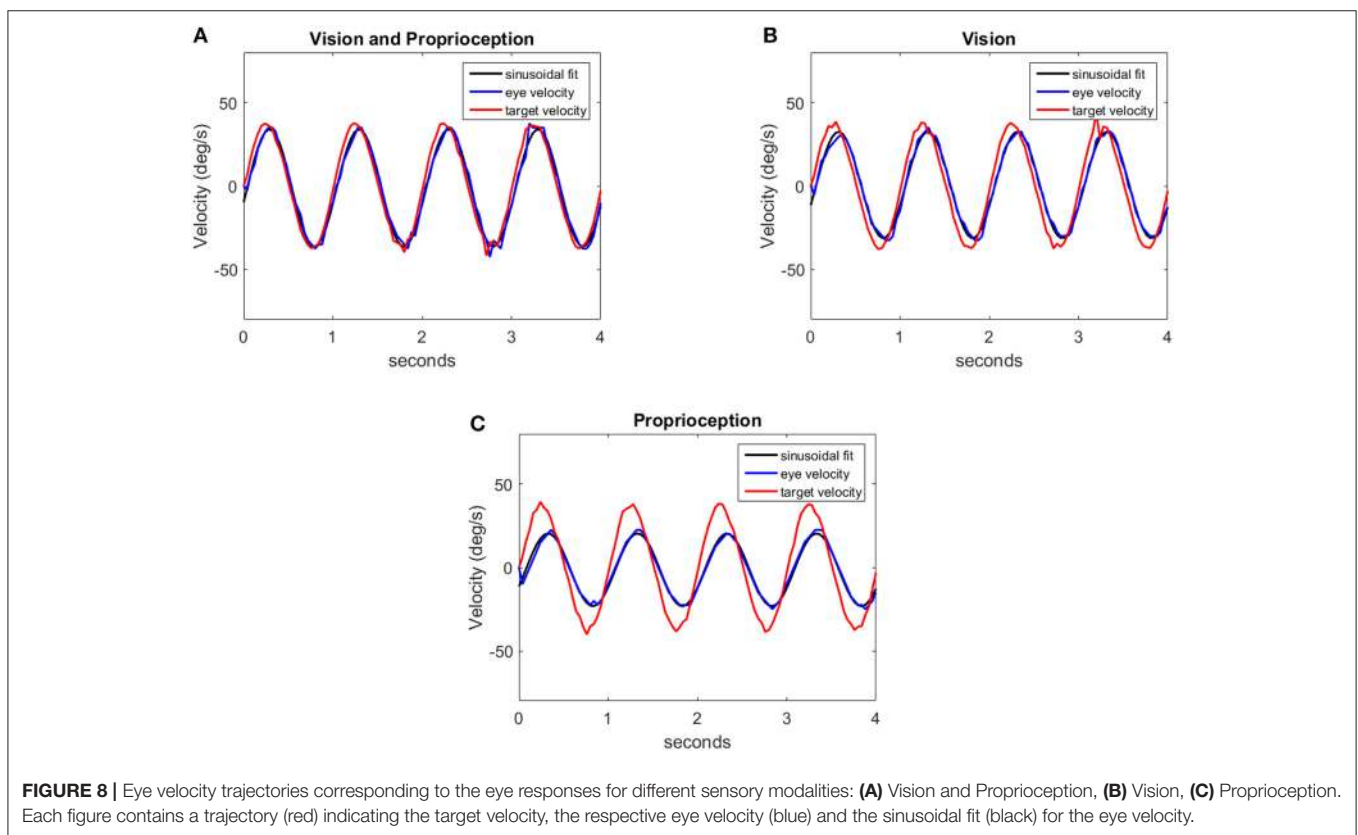
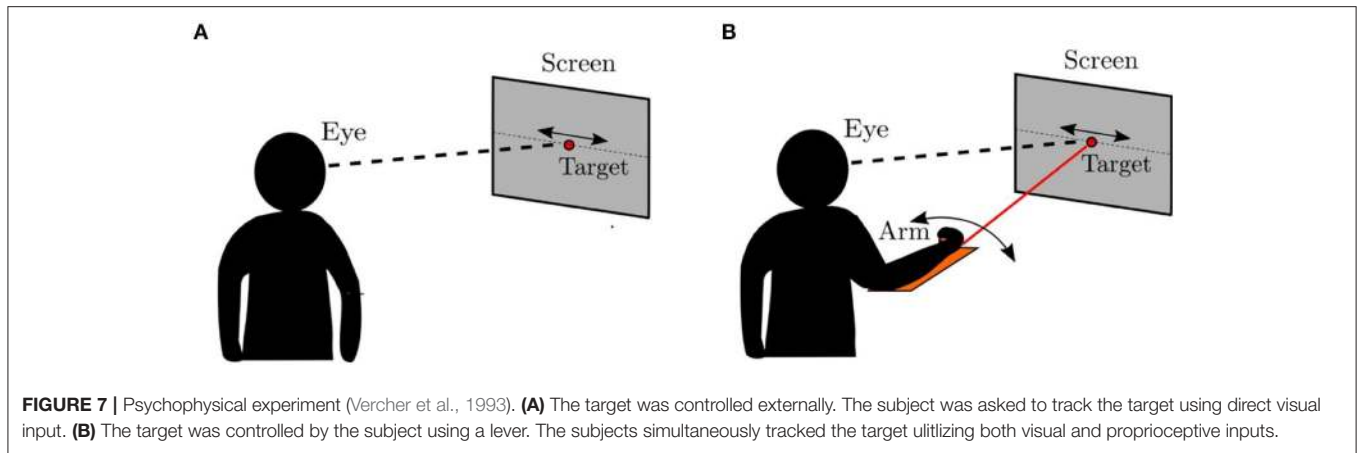
We also illustrate the system's robustness to changes in the visual appearance of the robot end effector qualitatively through an accompanying video². We change the appearance by moving the wrist and fingers of the iCub with sinusoidal joint trajectories. No information about finger and wrist motion is provided to the system. When eye acceleration is driven by vision alone, the eye drifts away faster from the end effector. The changes in the appearance of the end effector introduce visual perturbations, which are challenging to follow in comparison to changes due to translation of the end effector. For the other two cases, the proprioceptive inputs, which are not altered due to the changes in the visual appearance of the robot end effector, enable the robot to maintain the gaze on the palm for a longer time.

Importance of the Prediction Module

To identify the role of the prediction module in the proposed model, we compare the performance of three different models. **Figure 11A** shows a general architecture that covers the three models compared in this study. The three models differ only in their visual pathways. The proprioceptive features and the eye controller neural network are identical. **Figure 11B** illustrates the prediction based visual pathway proposed in this paper. The pathway in **Figure 11C** has a very similar structure, except that the prediction module is removed. Comparing these two models enables us to identify the benefit of the prediction module. The pathway shown in **Figure 11D** is the sparse coding based visual representation used in our earlier work (Wijesinghe et al., 2017).

To compare the performance, each model is tested for 20 different testing trials (10 different trajectories for each of the two training scenario types). **Figure 12** shows the RMSEs between the target and actual eye velocities. We perform paired sample *t*-tests to compare the RMSE of the model with prediction with the RMSEs of the other two. The effect size is also computed according to the Cohen's *d* formula. The Differences between the performance of the models in **Figures 11B,C** are not statistically significant for vision and proprioception ($p = 6.27 \times 10^{-1}$, effect size = -0.0876) nor vision alone ($p = 4.40 \times 10^{-1}$, effect size = 0.1619). The model with prediction performs significantly better for proprioception alone ($p = 6.16 \times 10^{-10}$, effect size = 4.2334). We have similar findings for the Sparse Coding based model in **Figure 11D**. Differences in performance are not significant for vision and proprioception ($p = 7.29 \times 10^{-2}$, effect size = 0.6531) nor vision alone ($p = 1.49 \times 10^{-1}$, effect size = -0.4611). The model with prediction performs significantly better with proprioception alone ($p = 5.94 \times 10^{-15}$, effect size = 4.7588).

²The video is available online at <https://youtu.be/RsGTmbb0cf0>



These results demonstrate that the proposed prediction based model exhibits superior performance compared to models without prediction. We attribute this to reduced correlation between the visual and proprioceptive features in the prediction-based model.

DISCUSSION

In this article, we propose a model based on the Active Efficient Coding (AEC) framework that enables a robot to learn to track its end effector using a combination of visual and proprioceptive cues. Rather than simply concatenating the two sets of features,

the proposed model predicts visual consequences of actions, which removes information correlated with proprioception from vision. The model enables a robot to learn to track an object for three cases: using both visual and proprioceptive cues corresponding to the typical case of end effector tracking, using only visual cues corresponding to tracking of an external independently moving object, and using pure proprioception corresponding to tracking of the end effector in darkness.

The incorporation of prediction is motivated by recent studies on neural responses in V1 of mice during locomotion (Niell and Stryker, 2011; Keller et al., 2012; Attinger et al., 2017). These studies suggest that the responses of cells in the V1

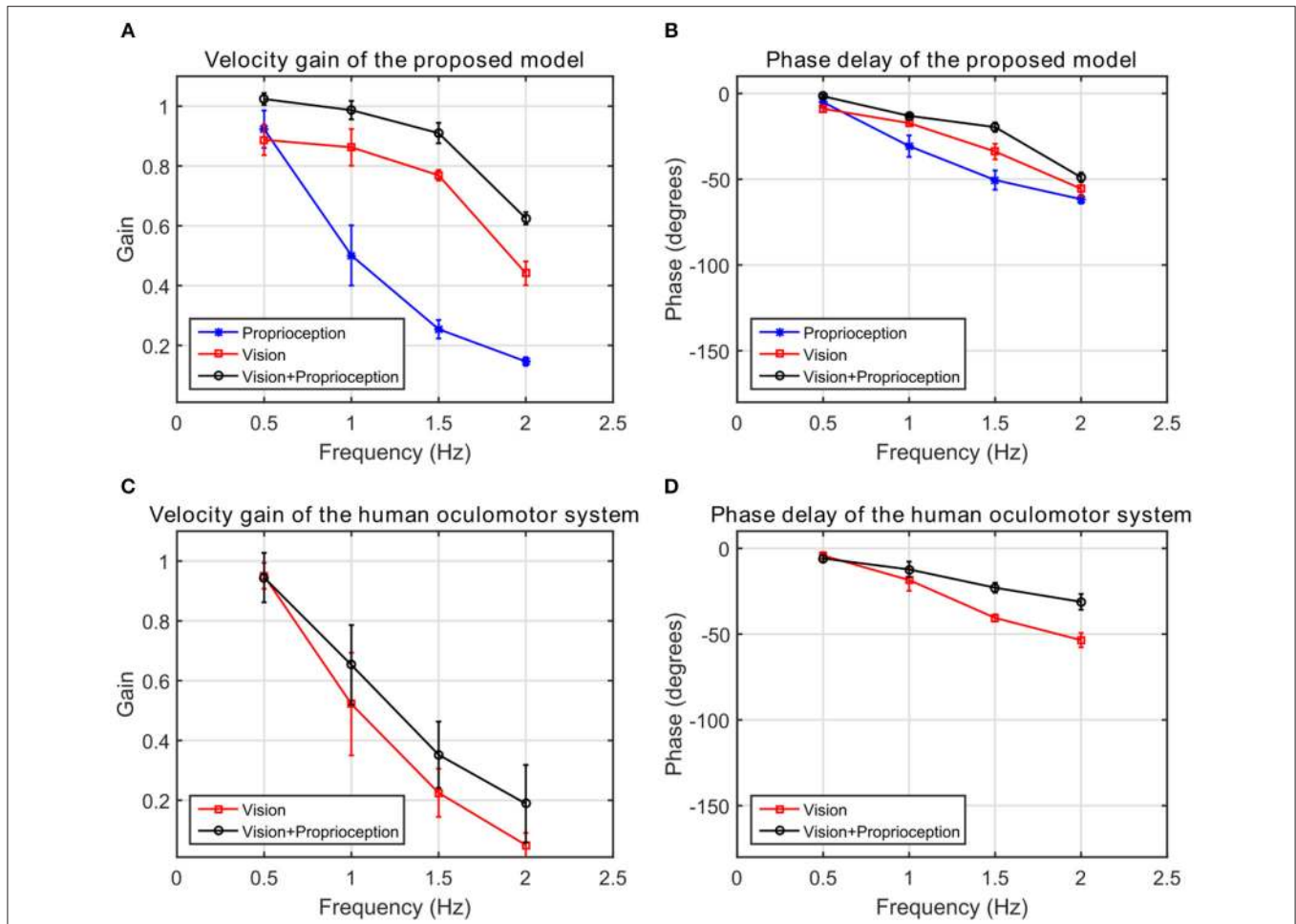


FIGURE 9 | The frequency response of the human oculomotor system compared with the frequency response of the proposed model. **(A)** The velocity gain of the proposed system. **(B)** The phase delay of the proposed system. **(C)** The velocity gain of the human oculomotor system. **(D)** The phase delay of the human oculomotor system.

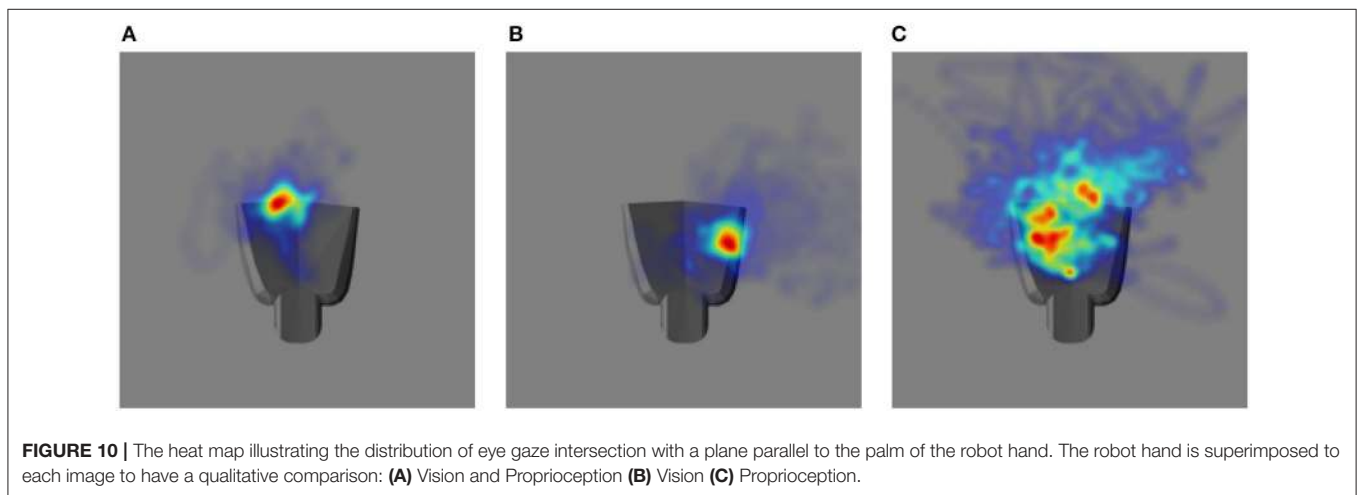


FIGURE 10 | The heat map illustrating the distribution of eye gaze intersection with a plane parallel to the palm of the robot hand. The robot hand is superimposed to each image to have a qualitative comparison: **(A)** Vision and Proprioception **(B)** Vision **(C)** Proprioception.

depend upon predictions of the sensory consequences of motor actions. In fact, locomotion improves the encoding of visual stimuli (Dadarlat and Stryker, 2017). In our model, the visual

representation encodes the residual motion after removing the predicted effects of self-motion from the observed visual flow. We show that the proposed prediction module has the ability

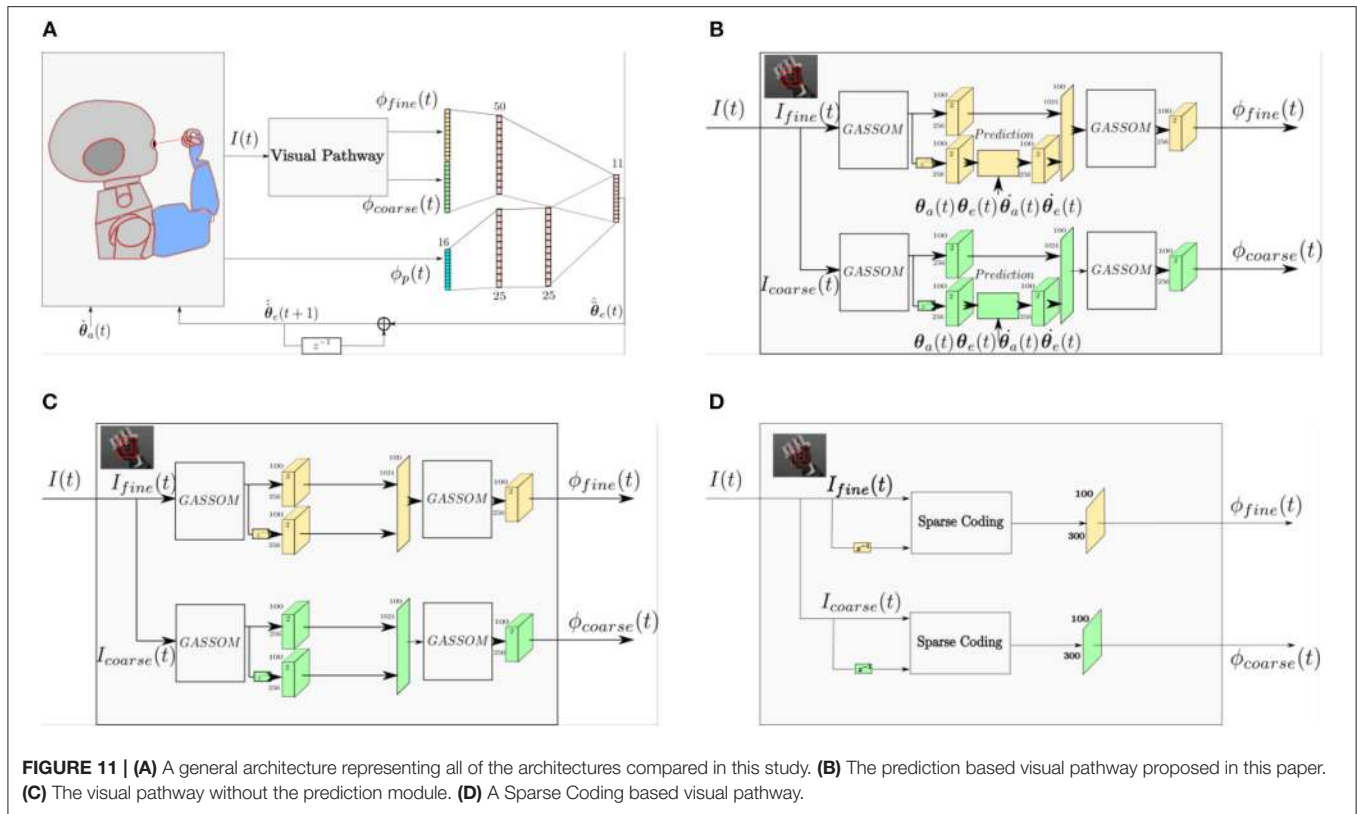


FIGURE 11 | (A) A general architecture representing all of the architectures compared in this study. **(B)** The prediction based visual pathway proposed in this paper. **(C)** The visual pathway without the prediction module. **(D)** A Sparse Coding based visual pathway.

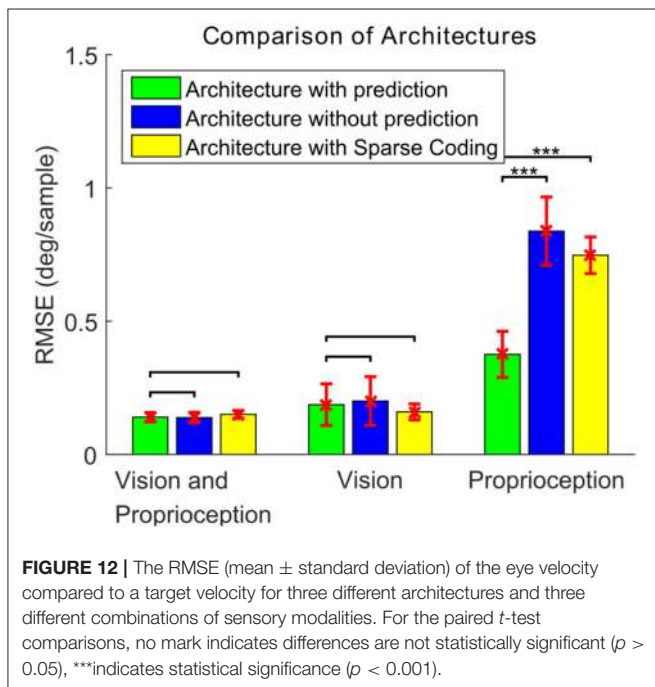


FIGURE 12 | The RMSE (mean \pm standard deviation) of the eye velocity compared to a target velocity for three different architectures and three different combinations of sensory modalities. For the paired t -test comparisons, no mark indicates differences are not statistically significant ($p > 0.05$), ***indicates statistical significance ($p < 0.001$).

to predict the visual sensory consequences of proprioceptive inputs (Figure 4). Specifically, the visual sensory consequences for subspaces with low spatial frequency basis vectors in the first

stage GASSOM are easier to predict compared to higher spatial frequencies.

The incorporation of learning into both the perception and action components of the perception-action loop in this model allows the sensory representation and the action generation network to co-adapt as the agent behaves in the environment. We characterized the performance of the eye controller both quantitatively and qualitatively. Using both visual and proprioceptive sensory stimuli to drive eye motion results in more accurate tracking of the end effector compared to using either sensory stimulus alone. Moreover, the inclusion of proprioception also makes the model more robust to changes in the appearance of the end effector.

We compare predictions of our model with findings from human psychophysical experiments studying the contribution of proprioception to human oculomotor control. Our results in Figure 9B suggest that incorporating proprioception reduces phase delay. This characteristic of the human oculomotor system has been found repeatedly. First, early work analyzing self-moved targets showed that the information about arm motion plays an important role in self-motion tracking (Steinbach and Held, 1968). The eye motion lagged behind target motion less for active arm motions than passive arm motions. Second, active and passive hand motion tracking along with tracking of external visual targets were qualitatively compared in Mather and Lackner (1980). External target tracking used a larger number of saccades per cycle and had larger latency compared

to the oculomotor tracking of the hand. The external target tracking was more challenging since the target motion was unpredictable. Third, experiments conducted by Gauthier et al. (1988) quantitatively compared eye tracking of an external object and of the subject's hand. The average latency when the eye tracked an external object (150 ± 30 ms) was much longer than the average latency when the eye tracked the hand (30 ± 10 ms). This delay was also prominent in the onset of smooth pursuit eye motions (Domann et al., 1989). Finally, Chen et al. (2016) showed that eye precedes a target controlled by the finger in congruent pursuit in comparison to opposite movements. However, our model does not exhibit this property.

Our results qualitatively agree with multiple psychophysical studies showing that proprioception can also change the velocity gain of the human oculomotor plant. During eye tracking of an external object, the eye velocity saturated at very low velocities around 40 deg/s compared to 100 deg/s during eye tracking of the hand (Domann et al., 1989). As shown in **Figure 9A**, velocity gains are larger at higher frequencies when the robot tracks its hand compared to an external object. Our model demonstrates that with the assistance of proprioception, the model is capable of maintaining a higher tracking gain for high frequency stimuli. This property is qualitatively consistent with the human oculomotor system (Vercher et al., 1993).

Human eye motion is mainly driven by visual stimuli. However, non-visual signals can also drive eye motion in certain tasks (e.g., in the darkness). **Figure 8C** illustrates that for our model proprioception alone has the capability to elicit eye movements. Several articles have studied the contribution of non-visual signals to oculomotor control. First, Steinbach (1969) showed that proprioceptive inputs alone were sufficient to generate smooth pursuit eye movements. Second, smooth pursuit movements generated by non-visual stimuli were studied in Berryhill et al. (2006). Tracking a pendulum in the darkness, proprioceptive stimuli had a velocity gain close to 0.3. On the other hand, with direct visibility of the pendulum, the subject had the ability to track the pendulum more accurately with a velocity gain of 0.7. Our model exhibits similar qualitative changes in visual vs. proprioceptive gain.

Although our model exhibits similar qualitative characteristics as human oculomotor tracking, it does not match quantitatively. In particular, the velocity gain of the model is much higher than that of the human oculomotor system (**Figure 9**). This mismatch may arise because we assume in our simulations that the eye velocity command generated by the model is executed perfectly by the eye. A more realistic model would include processing and propagation delays, a more realistic model of the neural control of eye velocity, as well as a dynamical model of the physical oculomotor plant. For example, the cerebellum is a part of the neural circuit for eye-hand coordination for oculomotor control (Miall et al., 2001), but this is not reflected in the proposed model. We anticipate that the incorporation of these elements into future extensions of the model would degrade the velocity gain observed during tracking, bringing

the model simulations into closer quantitative agreement with human performance.

This work can be extended in several directions. First, the model could be extended to include saccadic eye movements. Tracking targets with the eyes typically consists of a combination of pursuit and saccadic eye movements. For example, the majority of the eye movements when humans tracking the arm in darkness are saccades (Dieter et al., 2014).

Second, new sensory modalities might be added to the proposed model. The proposed model is only a first step toward integrating cross-sensory prediction. While a straightforward extension might be to add additional inputs to the predictor network, consideration of the problem of adding new sensory modalities raises a number of intriguing questions. For example, which sensory inputs should be predicted from which others? Here we have considered prediction in only one direction. Another question is how to deal with the different possible combinations of sensory cues that might be available.

Third, the arm trajectory generation could be made more biologically realistic, e.g., through the use of dynamical movement primitives for trajectory generation (Schaal, 2006), or through the use of goal babbling to choose the via points (von Hofsten, 2004). In this work, we used random babbling and trajectory generation for its simplicity. The use of dynamic movement primitives would alter the statistics of the image motion induced by the hand, which might change the smooth pursuit performance, e.g., the final steady state error in **Figure 6** or the shape of the frequency response curves in **Figure 9**. The use of goal babbling might improve the speed of learning (Baranes and Oudeyer, 2013). These would be interesting extensions of the model to investigate. However, we do not expect that their incorporation to change the main qualitative findings we report here, e.g., the ordering of the degradation in performance and proprioceptive or visual cues are removed.

Finally, the model might be integrated into a more comprehensive framework for hand-eye coordination that would include other tasks, such as reaching. In a sense, reaching is the inverse of the problem studied here. Our model generates commands to change eye gaze based on visual information and arm motion. Visually guided reaching involves the generation of an arm motion command to move the end effector to a visual target. The required mappings, e.g., between gaze direction and/or visual target location and end effector motion are often learned using a motor babbling process similar to that used here, where the end effector is tracked as the arm moves (Burger et al., 2018). In other cases, gaze is controlled to bring the target object or end effector to the image center (Huelse et al., 2010; Jamone et al., 2012; Savastano and Nolfi, 2013). In these works, the problem of tracking the end effector/target was simplified by attaching a marker to the end effector/target. Since our model does not require explicit markers, it might be used to relax some of the assumptions made by prior work. Because the location of the eye gaze drifts on the palm area (**Figure 10**), gaze direction does not directly correspond to a specific position on the end effector in trajectories generated by the model. However, the model could be used to generate data to learn an

approximate mapping between gaze direction and end effector position.

AUTHOR CONTRIBUTIONS

All authors participated in designing experiments and writing the paper. LW conducted the experiments.

REFERENCES

- Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., et al. (2009). Cognitive developmental robotics: a survey. *IEEE Trans. Auton. Ment. Dev.* 1, 12–34. doi: 10.1109/TAMD.2009.2021702
- Attinger, A., Wang, B., and Keller, G. B. (2017). Visuomotor coupling shapes the functional development of mouse visual cortex. *Cell* 169, 1291–1302. doi: 10.1016/j.cell.2017.05.023
- Baranes, A., and Oudeyer, P. Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Rob. Auton. Syst.* 61, 49–73. doi: 10.1016/j.robot.2012.05.008
- Bennett, D. J., Geiger, D., and Hollerbach, J. M. (1991). Autonomous robot calibration for hand-eye coordination. *Int. J. Rob. Res.* 10, 550–559. doi: 10.1177/027836499101000510
- Berryhill, M. E., Chiu, T., and Hughes, H. C. (2006). Smooth pursuit of nonvisual motion. *J. Neurophysiol.* 96, 461–465. doi: 10.1152/jn.00152.2006
- Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor-critic algorithms. *Automatica* 45, 2471–2482. doi: 10.1016/j.automatica.2009.07.008
- Broun, A., Beck, C., Pipe, T., Mirmehdi, M., and Melhuish, C. (2014). Bootstrapping a robot's kinematic model. *Rob. Auton. Syst.* 62, 330–339. doi: 10.1016/j.robot.2013.09.011
- Burger, W., Wieser, E., Dean-Leon, E., and Cheng, G. (2018). “A scalable method for multi-stage developmental learning for reaching,” in *IEEE International Conference on Development and Learning and on Epigenetic Robotics* (Lisbon), 60–65. doi: 10.1109/DEVLRN.2017.8329788
- Chandrapala, T. N., and Shi, B. E. (2015). Learning slowness in a sparse model of invariant feature detection. *Neural Comput.* 27, 1496–1529. doi: 10.1162/NECO_a_00743
- Chen, J., Valsecchi, M., and Gegenfurtner, K. R. (2016). Role of motor execution in the ocular tracking of self-generated movements. *J. Neurophysiol.* 116, 2586–2593. doi: 10.1152/jn.00574.2016
- Corke, P. (2017). *Robotics, Vision and Control*. Cham: Springer. doi: 10.1007/978-3-319-54413-7
- Dadarlat, M. C., and Stryker, M. P. (2017). Locomotion enhances neural encoding of visual stimuli in mouse V1. *J. Neurosci.* 37, 3764–3775. doi: 10.1523/JNEUROSCI.2728-16.2017
- Dalal, N., and Triggs, B. (2005). “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition* (San Diego, CA), 886–893. doi: 10.1109/CVPR.2005.177
- Dieter, K. C., Hu, B., Knill, D. C., Blake, R., and Tadin, D. (2014). Kinesthesia can make an invisible hand visible. *Psychol. Sci.* 25, 66–75. doi: 10.1177/0956797613497968
- Domann, R., Bock, O., and Eckmiller, R. (1989). Interaction of visual and non-visual signals in the initiation of smooth pursuit eye movements in primates. *Behav. Brain Res.* 32, 95–99. doi: 10.1016/S0166-4328(89)80077-4
- Fantacci, C., Pattacini, U., Tikhanoff, V., and Natale, L. (2017). “Visual end-effector tracking using a 3D model-aided particle filter for humanoid robot platforms,” in *IEEE International Conference on Intelligent Robots and Systems* (Vancouver, BC), 1411–1418. doi: 10.1109/IROS.2017.8205942
- Gatla, C. S., Lumia, R., Wood, J., and Starr, G. (2007). An automated method to calibrate industrial robots using a virtual closed kinematic chain. *IEEE Trans. Robot.* 23, 1105–1116. doi: 10.1109/TRO.2007.909765
- Gauthier, G. M., and Mussa Ivaldi, F. (1988). Oculo-manual tracking of visual targets in monkey: role of the arm afferent information in the control of arm and eye movements. *Exp. Brain Res.* 73, 138–154. doi: 10.1007/BF00279668
- Gauthier, G. M., Vercher, J. L., Mussa Ivaldi, F., and Marchetti, E. (1988). Oculo-manual tracking of visual targets: control learning, coordination control and coordination model. *Exp. Brain Res.* 73, 127–137. doi: 10.1007/BF00279667
- Hersch, M., Sauter, E., and Billard, A. (2008). Online learning of the body schema. *Int. J. Humanoid Robot.* 5, 161–181. doi: 10.1142/S0219843608001376
- Hoffmann, M., Marques, H., Arieta, A., Sumioka, H., Lungarella, M., and Pfeifer, R. (2010). Body schema in robotics: a review. *IEEE Trans. Auton. Ment. Dev.* 2, 304–324. doi: 10.1109/TAMD.2010.2086454
- Hollerbach, J. M., and Wampler, C. W. (1996). The calibration index and taxonomy for robot kinematic calibration methods. *Int. J. Rob. Res.* 15, 573–591. doi: 10.1177/027836499601500604
- Huelse, M., McBride, S., Law, J., and Lee, M. (2010). Integration of active vision and reaching from a developmental robotics perspective. *IEEE Trans. Auton. Ment. Dev.* 2, 355–367. doi: 10.1109/TAMD.2010.2081667
- Jamone, L., Natale, L., Nori, F., Metta, G., and Sandini, G. (2012). Autonomous online learning of reaching behavior in a humanoid robot. *Int. J. Humanoid Robot.* 09:1250017. doi: 10.1142/S021984361250017X
- Keller, G. B., Bonhoeffer, T., and Hübener, M. (2012). Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron* 74, 809–815. doi: 10.1016/j.neuron.2012.03.040
- Kohonen, T. (1996). Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. *Biol. Cybern.* 75, 281–291. doi: 10.1007/s004220050295
- Lisberger, S. G. (2010). Visual guidance of smooth-pursuit eye movements: sensation, action, and what happens in between. *Neuron* 66, 477–491. doi: 10.1016/j.neuron.2010.03.027
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94
- Mather, J. A., and Lackner, J. R. (1980). Visual tracking of active and passive movements of the hand. *Q. J. Exp. Psychol.* 32, 307–315. doi: 10.1080/14640748008401166
- Miall, R. C., Reckess, G. Z., and Imamizu, H. (2001). The cerebellum coordinates eye and hand tracking movements. *Nat. Neurosci.* 4, 638–644. doi: 10.1038/88465
- Niell, C. M., and Stryker, M. P. (2011). Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron* 65, 472–479. doi: 10.1016/j.neuron.2010.01.033.Modulation
- Olmos, A., and Kingdom, F. A. (2004). A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception* 33, 1463–1473. doi: 10.1068/p5321
- Rao, R. P., and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* 2, 79–87. doi: 10.1038/4580
- Rochat, P. (1998). Self-perception and action in infancy. *Exp. Brain Res.* 123, 102–109. doi: 10.1007/s002210050550
- Saleem, A. B., Ayaz, A., Jeffery, K. J., Harris, K. D., and Carandini, M. (2013). Integration of visual motion and locomotion in mouse visual cortex. *Nat. Neurosci.* 16, 1864–1869. doi: 10.1038/nn.3567
- Savastano, P., and Nolfi, S. (2013). A robotic model of reaching and grasping development. *IEEE Trans. Auton. Ment. Dev.* 5, 326–336. doi: 10.1109/TAMD.2013.2264321
- Schaal, S. (2006). “Dynamic movement primitives - a framework for motor control in humans and humanoid robotics,” in *Adaptive Motion of Animals and Machines*, eds H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witt (Tokyo: Springer-Verlag), 261–280. doi: 10.1007/4-431-31381-8_23
- Schillaci, G., Hafner, V. V., and Lara, B. (2014). “Online learning of visuomotor coordination in a humanoid robot. A biologically inspired model” in

FUNDING

This work was supported in part by the General Research Fund of the Hong Kong Research Grants Council, under grant 16209014 (LW, BS), Johanna Quandt Foundation (JT), and the European Union's Horizon 2020 research and innovation program under grant 713010 (JT).

- IEEE International Conference on Development and Learning and on Epigenetic Robotics* (Genoa), 130–136. doi: 10.1109/DEVLRN.2014.6982967
- Srinivasan, M. V., Laughlin, S. B., and Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. *Proc. R. Soc. B Biol. Sci.* 216, 427–459. doi: 10.1098/rspb.1982.0085
- Steinbach, M. J. (1969). Eye tracking of self-moved targets: the role of efference. *J. Exp. Psychol.* 82, 366–376. doi: 10.1037/h0028115
- Steinbach, M. J., and Held, R. (1968). Eye tracking of observer-generated target movements. *Science* 161, 187–188. doi: 10.1126/science.161.3837.187
- Sturm, J., Plogemann, C., and Burgard, W. (2009). Body schema learning for robotic manipulators from visual self-perception. *J. Physiol. Paris* 103, 220–231. doi: 10.1016/j.jphysparis.2009.08.005
- Teulière, C., Forestier, S., Lonini, L., Zhang, C., Zhao, Y., Shi, B., et al. (2015). Self-calibrating smooth pursuit through active efficient coding. *Rob. Auton. Syst.* 71, 3–12. doi: 10.1016/j.robot.2014.11.006
- Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., and Nori, F. (2008). “An open-source simulator for cognitive robotics research : the prototype of the iCub humanoid robot simulator,” in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (New York, NY), 57–61. doi: 10.1145/1774674.1774684
- Vercher, J. L., Gauthier, G. M., Guedon, O., Blouin, J., Cole, J., and Lamarre, Y. (1996). Self-moved target eye tracking in control and deafferented subjects: roles of arm motor command and proprioception in arm-eye coordination. *J. Neurophysiol.* 76, 1133–1144. doi: 10.1152/jn.1996.76.2.1133
- Vercher, J. L., Volle, M., and Gauthier, G. M. (1993). Dynamic analysis of human visuo-oculo-manual coordination control in target tracking tasks. *Aviat. Sp. Environ. Med.* 64, 500–506.
- Vicente, P., Jamone, L., and Bernardino, A. (2016). Online body schema adaptation based on internal mental simulation and multisensory feedback. *Front. Robot. AI* 3:7. doi: 10.3389/frobt.2016.00007
- von Hofsten, C. (2004). An action perspective on motor development. *Trends Cogn. Sci.* 8, 266–272. doi: 10.1016/j.tics.2004.04.002
- Wijesinghe, L. P., Antonelli, M., Triesch, J., and Shi, B. E. (2017). “Learning multisensory neural controllers for robot arm tracking,” in *IEEE International Joint Conference on Neural Networks* (Anchorage, AK). 4150–4157. doi: 10.1109/IJCNN.2017.7966380
- Zhang, C., Zhao, Y., Triesch, J., and Shi, B. E. (2014). “Intrinsically motivated learning of visual motion perception and smooth pursuit,” in *IEEE International Conference on Robotics and Automation* (Hong Kong), 1902–1908. doi: 10.1109/ICRA.2014.6907110
- Zhao, Y., Rothkopf, C. a., Triesch, J., and Shi, B. E. (2012). “A unified model of the joint development of disparity selectivity and vergence control,” in *IEEE International Conference on Development and Learning and Epigenetic Robotics* (San Diego, CA), 1–6. doi: 10.1109/DevLrn.2012.6400876
- Zhou, T., and Shi, B. E. (2016). Learning visuomotor transformations and end effector appearance by local visual consistency. *IEEE Trans. Cogn. Dev. Syst.* 8, 60–69. doi: 10.1109/TAMD.2015.2502758
- Zmarz, P., and Keller, G. B. (2016). Mismatch receptive fields in mouse visual cortex. *Neuron* 92, 766–772. doi: 10.1016/j.neuron.2016.09.057

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Wijesinghe, Triesch and Shi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.