Learning open-loop saccadic control of a 3D biomimetic eye using the actor-critic algorithm

Henrique Granado[1], Reza Javanmard Alitappeh[2], Akhil John[1], A. John van Opstal [3], and Alexandre Bernardino[1]

***Abstract*— The application of reinforcement learning algorithms to robotics has increased over the last decade, especially for the control of robots with non-linear dynamics and a redundant number of degrees of freedom using classic control techniques. Here we study the control of a biomimetic robotic eye with three extraocular muscle pairs as a prime example. Using an actor-critic algorithm, this paper aims to link reinforcement learning to this control problem, and create a framework that will learn the open-loop control of saccadic movements of the robotic eye. The basis for the implemented control is inspired by the primate physiological pulsed control signal, which is generated, integrated and sent to the appropriate muscles to perform the saccade. The metric that evaluates the saccadic output is also inspired by the primate oculomotor system and is used to shape the reward function. This methodology was applied to a simplified 3D physical model of the human eye as a proof of concept. The algorithm managed to learn a saccadic control strategy in 3D. The trajectories obtained, have similar non-linear dynamics as those recorded in humans and their 3D rotational kinematics are constrained by Listing's law.**

I. Introduction

The past decades have yielded significant technological developments in the field of robotics, which may to a large extent be attributed to the use of novel machine-learning techniques. As a result, modern humanoid robots can perform complex tasks in ways that more and more resemble those of humans. However, because of the high complexity of biological systems, mimicking human behaviors in a humanoid robotic system with traditional control techniques is still far from trivial.

One example of a complex subsystem of the human body is the eye. Although the human eye has three rotational degrees of freedom, the brain controls ocular movements in highly stereotyped ways. In this paper, we focus on a particular type of eye movement called "saccade". Saccades are fast eye movements that change the gaze direction in a ballistic fashion (open loop). Human saccadic behavior has been studied extensively in neuroscience, and several methods have been proposed to model the saccadic system and develop controllers for robotic eyes. In a previous work [1], we adopted a model-based approach, based on a physical 3D model of the eye plant (eyeball, six extraocular muscles, and surrounding tissues). It used feedforward optimal-control principles to replicate human saccadic behavior. However, as a model-based approach, it required complex modeling of the eye-muscle system that is hard to generalize to other systems, or to variations of the same system. In the current work, we adopt a model-free stance to the problem and evaluate the ability of the actor-critic algorithm to learn control policies without relying on accurate knowledge of a physical model of the system.

So far, most studies of robotic eyes have predominantly focused on the mechanics or controllers that constrain by design the artificial eye movements to be similar to biological ones [2]–[7]. For example, in [5] the authors implemented the constraints of the eye's 3D rotational kinematics on a purely mechanical basis.

To our knowledge, only two studies have applied reinforcement learning to control a robotic eye with muscle-based actuation [8], [9]. In [8], a 2 degree-of-freedom (DOF) robotic eye with four contractile muscles was controlled using deep deterministic policy gradient (DDPG) to smoothly track or fixate a target, while maximizing reward based on tracking error and control effort. The muscles were not capable of producing fast movements. In [9], the authors developed a nonlinear neuromuscular model of the six muscles for both eyes in a simulation environment. They too used DDPG to perform fast eye movements to target positions by maximizing a reward based on tracking errors and coordination of the left and right eye. However, both papers' control strategies were not designed to produce the stereotypical 3D kinematics and dynamics of human saccades, or realistic agonist/antagonist neural commands. Also, they ignored the effects of cyclotorsion of the eye (rotation about gaze direction). To the best of our knowledge, there is no study of the 3D oculomotor system that applies model-free reinforcement learning control techniques that result in the emergence of human-like saccades.

The present paper develops a novel methodology to study how the eye moves. We will follow a machine-learning approach (reinforcement learning) for the control of the eye, in which the algorithm will have to learn by trial and error how to generate an optimal movement trajectory from an initial to a desired eye orientation. Here, we trained an agent to perform saccades with an actor-critic algorithm adapted from [10].

*Funded by EU Horizon 2020 ERC Advanced Grant 2016 project "Orient", nr. 693400

[1]These authors are with ISR, Instituto Superior Técnico Lisboa, Portugal. {henrique.granado, akhil.john, alexandre.bernardino}@tecnico.ulisboa.pt

[2]Reza Javanmard Alitappeh is with Univ. of Science and Technology of Mazandaran, Iran. rezajavanmard64@gmail.com

[3]A. John van Opstal is with the Neurophysics Section, DCN, Radboud University, Netherlands. john.vanopstal@donders.ru.nl

Briefly, the algorithm maximizes the expected rewards over time. The agent is composed of (i) an actor network that learns the command to drive the eye from the initial to the desired orientation, and (ii) a critic network that learns to predict the reward for that command. Both networks interact in the learning process, as the actor learns to maximize the output of the critic. We validated this approach in a computational simulation of a robotic eye performing fully unconstrained 3D eye movements, using biologically inspired pulse-step inputs. Results show that the pulse-step parameters that lead to saccadic behaviors resemble human performance and can be learned with high precision in a few tens of thousands of iterations.

Our paper is organized as follows: In Section II we provide some background information on the neurophysiology of eye movements to provide the context of our approach. In Section III we explain how the reinforcement learning framework fits with the training of a controller for a 3D human eye model, and in Section IV we formulate the key components of the algorithm that allow it to produce human-like saccades. The results of the reinforcement learning are presented in Section V along with a comparison with human data. Our results are discussed in Section VI and we propose improvements for future work in Section VII.

II. Background

In this section we provide some relevant information on the neurophysiology of the human oculomotor control system that inspired and constrained our design choices in developing and evaluating the proposed approach to the problem.

A. Neural control of saccades

The eye-plant is typically modelled by a second-order, linear and over-damped system with a long time constant of about T ~200 ms. The fatty tissues around the eye and the drag from the optic nerve are the major causes for this overdamped property, making the plant's step response far exceed typical saccade durations, which range between 20-100 ms [11], [12]. As a consequence, the brain employs a non-linear pulse-step control strategy that overcomes the overdamped nature of the system and allows for fast and accurate saccades. Robinson [13] proposed a saccadic eye-movement model (Fig. 1) that provides a solid conceptual framework to study and understand oculomotor control. Central to this concept is a nonlinear pulse generator that sends a high-frequency pulse, $\dot{e}$, to the motor neurons to overcome the eye's frictional forces. In parallel, the pulse is time-integrated to produce the required elastic forces that keep the eye at the desired orientation after the saccade. For accurate saccades without over- or undershoots, pulse and step have to match precisely (e.g., $k_1 = k_3 = 1$, and $k_2 = T$).

In this work, we extend this approach to 3D and to a nonlinear physics-based model of the ocular plant, while employing similar pulse-step parameterizations for the motor-primitives to drive saccades.

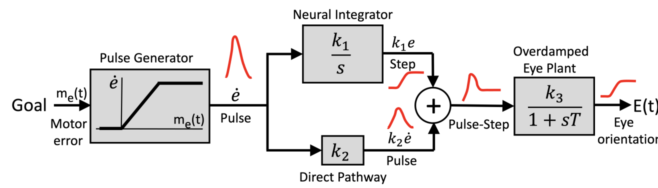

Fig. 1. Robinson's model for horizontal saccades [13]. $m_e(t)$ represents the current motor error, and $\dot{e}$ is the fast eye velocity-related signal from the nonlinear pulse generator, which overcomes the high viscosity of the plant. The neural integrator provides the step input, $e(t)$, to the oculomotor neurons at the summing junction. k_1, k_2, k_3 are gains, and $T \sim 200$ ms is the plant's time constant.

B. Properties of human saccades.

Normal saccades have stereotyped, skewed, velocity profiles that do not scale with the saccade amplitude (Fig. 2A). They obey a so-called 'main sequence' [14], which describes the nonlinear relationship between the saccade amplitude and its peak velocity (Fig. 2B). This latter property is due to the nonlinear saturation in the pulse generator of Robinson's model (Fig. 1). Oblique saccade trajectories are approximately straight, which means that the horizontal and vertical components of their velocity profiles are scaled versions of each other (Fig. 2C). As a consequence, the peak velocities of the components are not constant but have a nearly cosine (horizontal) or sine (vertical) dependence on the saccade direction. This effect is known as 'component stretching' (panel B, gray curves) [15].

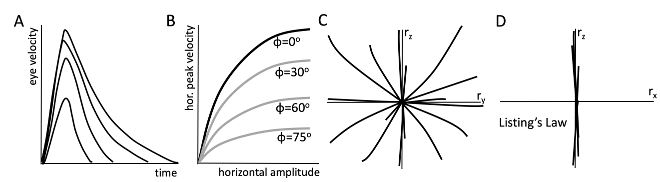

Fig. 2. Main properties of saccades. (A) Skewed saccade velocity profiles. (B) Main sequence for the horizontal components of oblique saccades. Φ is the saccade direction, with $\Phi = 0$ rightward, $\Phi = 90$ upward, etc. Note that peak horizontal velocity depends on saccade direction. Similar relations hold for the vertical components. (C) Oblique saccade trajectories are approximately straight. (D) Listing's law specifies that the amount of cyclo-torsion for 3D eye orientations (r_x) is zero. [11].

Finally, when analyzing saccade trajectories in 3D, the instantaneous orientation of the eye is constrained to two degrees of freedom, which is known as 'Listing's Law' (Fig. 2C,D). Mathematically, one can describe any eye orientation as a virtual rotation from the primary position about a fixed axis, parameterized by the Euler-Rodrigues rotation vector, $\mathbf{r} = (r_x, r_y, r_z)$. Here, r_x represents the eye's cyclo-torsion (rotation about the line of sight), r_y is the horizontal axis for vertical eye orientations, and r_z is the vertical axis for horizontal eye orientations. Listing's Law then holds that all rotation axes lie in the (r_y, r_z) plane, and that, therefore, $r_x = 0$. Current ideas hold that the properties shown in Fig. 2 are the result of a neural optimal control strategy, which aims to generate saccades with the smallest possible errors, shortest durations, and the least amount of effort [1], [16].

Here, we will use the main sequence, component cross-coupling, and Listing's law properties of human saccades to evaluate the quality of our methods.

III. METHODOLOGY

The aim of this work is to use Reinforcement Learning to control a 3D biomimetic eye. In this framework, the agent learns by receiving rewards for the results of its actions and adjusts its behavior accordingly. Its ultimate goal is to find the optimal policy that maximizes the cumulative reward over time. Figure 3 illustrates the general concept of the algorithm.

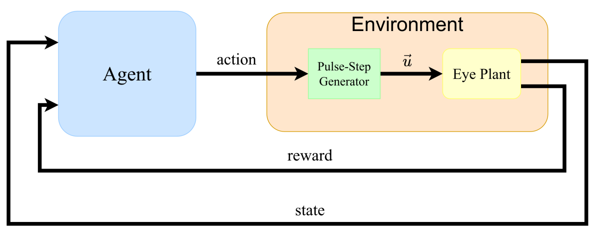

Fig. 3. Diagram illustrating the general idea behind the reinforcement learning algorithm, by highlighting the key elements that are relevant for this paper. The agent's action is sent to the Environment. The latter consists of the 3D pulse-step generators that provide the motor commands, $\mathbf{u}$, for the 3D eye plant. The resulting state of the eye (its 3D orientation) and the associated reward are provided to the agent, which then updates its policy.

Briefly, the agent receives input about the state of the eye (its 3D orientation) and the goal (desired gaze direction). Its output is an action that shapes the 3D pulse-step generator to produce motor commands ($\mathbf{u}$) that control the eye plant. The resulting eye movement is evaluated, leading to a reward that is used to improve the policy of the agent. In the next sections, we describe the key components of this architecture in more detail.

A. Environment

1) Eye Plant: We developed a simulator of our biomimetic robotic eye, using the nonlinear Newton-Euler equations for a rotating rigid body to calculate its rotational movements in response to three motor commands, $\mathbf{u}(t) = (u_H, u_R, u_L)^T$. For details, see [1]. Briefly, actuation was supplied by three independent rotatory motors, each connected to a rod with two elastic strings at its tips that connected to anatomically fixed points on a freely rotating sphere with moment of inertia tensor, I. Each pair of strings represented an antagonistic pair of eye muscles (indicated by H, R, and L, respectively, see Fig. 4) that applied a torque on the eye. The changes in eye orientation are fully determined by the total torque exerted on the eye by the three antagonistic pairs. Nonlinearities arise because of (i) the trigonometric relations between motor rotations and forces exerted by the strings, (ii) the orientation dependence of I and the strings' pulling directions, and (iii) the non-commutativity of finite rotations.

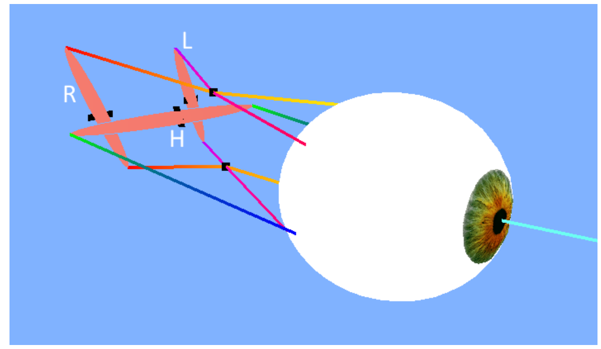

Fig. 4. Visual representation of the simulator. Motor H connects to the medial-and lateral rectus strings for horizontal eye rotations; L to superior rectus/inferior oblique, and R to the superior oblique/inferior rectus strings. Note that L and R generate joint vertical-torsional rotations. Adapted from [1].

2) Pulse-Step Generator: The three motors receive their input control signals from three independent pulse-step controllers that are configured according to Robinson's proposal in Fig. 1. The pulse-step generators (with integrator gains k_H, k_R, k_L) each receive a pulse, $p_H(t), p_R(t), p_L(t)$, as actions from the agent, each parameterized by their amplitude A_m and duration, B_m^{-1} (see below, for details). The agent programs its action on the basis of the initial state of the motors, $\mathbf{u}^0$, the initial state of the eye (rotation vector, $\mathbf{r}^0$), the desired 2D goal for the saccade, $\mathbf{r}^D$, and the reward, R (see Fig. 5).

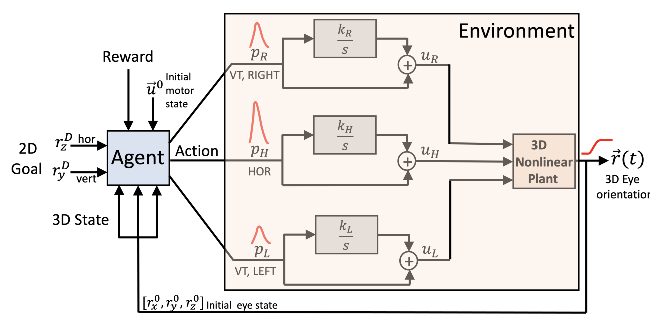

Fig. 5. Representation of the 3D environment, comprising the pulse-step generators and 3D nonlinear eye plant model. The agent sends three pulses, p_m to the environment, which are transformed into three motor commands, u_m, that drive the plant. The initial motor and eye orientations (the 'state', $\mathbf{u}^0, \mathbf{r}^0$) are fed back to the agent, and are used to determine the reward associated with the action, which relies on $\mathbf{r}(t)$. The goal of the system is to maximize the total reward.

B. Agent

The agent specifies the actions to apply to the environment. We implemented an actor-critic algorithm [17] adapted from [10] to the open-loop eye-movement control system described above. The algorithm used two distinct neural networks to learn the optimal actions, or policy. The Q-Network is the critic, which estimates the reward, R, for a given action on the environment, as illustrated in Figure 6. As in [10], this is implemented with two parallel Q-Networks from which the one issuing the lowest reward estimate is used at each step to update the agent's policy. This implementation facilitates the convergence of the method [10].

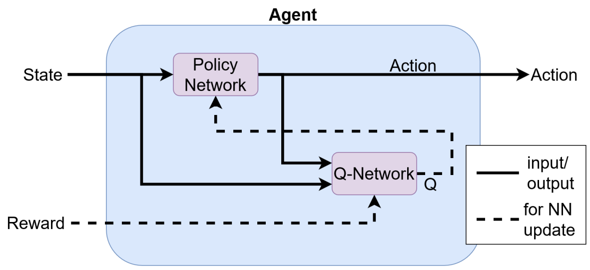

Fig. 6. Structure of the agent within the actor-critic algorithm, which operates within the reinforcement learning framework to optimally act on the environment. The Q-network effectively consists of two parallel networks to mitigate positive bias in the policy improvement step that is known to degrade performance of value-based methods [10]

The actor, or policy π-Network, determines the actions to be applied to the environment. The actor is guided by the reward estimate from the Q-Network to improve its policy.

The π and Q networks were fully connected nets with Rectified Linear Unit activation functions. The π network had 32 neurons in each of its 3 hidden layers. We implemented two Q Networks as in [10] to avoid overestimation of the policy network (see Fig.6). Each had 3 hidden layers with 128-64-32 units, and 128-128-16 units, respectively, which were randomly initialized.

IV. Implementation

We here specify the key components (state, action and reward) for the reinforcement learning algorithm, and how these are linked to the training of the oculomotor model.

A. State

The state, $\mathbf{s}$, that is fed into the policy network is a tuple composed of the initial orientation of the eye, $\mathbf{r}^0$, and its motors, $\mathbf{u}^0$, as well as the desired coordinates of the 2D goal, r_y^D and r_z^D, of the eye:

$$\mathbf{s} = (r_x^0, r_y^0, r_z^0, u_L^0, u_H^0, u_R^0, r_y^D, r_z^D) \tag{1}$$

Note that the amount of cyclo-torsion of the desired eye orientation, r_x^D, is not specified as we what to leave it as unconstrained as possible. For the task of directing the fovea towards the target, only the y and z components are required.

B. Action

The agent's action will change the orientation of the eye (Fig. 5). Since the control is open-loop, the actions, $p_m(t)$, and the resulting motor commands, $u_m(t)$, need to be specified as functions of time. Our proposed actions are inspired by the pulsed control of Robinson's model in Fig. 1. The pulse function $p_m(t)$, then relates to $u_m(t)$ through (see Fig. 5):

$$u_m(t) = p_m(t) + k_m \int_0^t p_m(x)dx + u_m^0 \tag{2}$$

where k_m is the integrator gain of control component m, and u_m^0 is the initial position of motor $m \in [H, R, L]$. As a simple approximation for the pulses, we used the following parameterization:

$$p_m(t) = A_m t^2 e^{-B_m t} \tag{3}$$

where A_m and B_m are parameters that define the amplitude and exponential decay of pulse $p_m(t)$.

Substituting (3) into (2) yields for the motor profile:

$$\begin{aligned} u_m(t) = u_m^0 + 2\frac{A_m}{B_m}\frac{k_m}{B_m^2}(1 - e^{-B_m t}) - \\ 2\frac{A_m}{B_m}\left(\frac{k_m}{B_m}t + \frac{k_m - B_m}{2}t^2\right)e^{-B_m t} \end{aligned} \tag{4}$$

For each motor, u_m, three free parameters needed to be optimized by the actor-critic algorithm: A_m, B_m and k_m. For ease of interpretation, we substituted parameter A_m by $D_m \equiv u_m(\infty)$, the final position of the motor (4). The values of A_m and D_m are related by:

$$A_m = \frac{(D_m - u_m^0)B_m^3}{2k_m} \tag{5}$$

Taken together, the action a is specified by parameters

$$a = (B_L, D_L, k_L, B_H, D_H, k_H, B_R, D_R, k_R) \tag{6}$$

C. Reward

The reward function follows from an optimal control strategy, which simultaneously accounts for the following five costs (penalties): lack of saccade accuracy, C_{acc}, total energy spent by the motors, C_{en}, saccade duration, C_{dur}, and total force applied on the eye at steady fixation, C_{for}, as in our previous work [1]. We here added an additional cost for unwanted overshoots, potentially due to pulse-step mismatch, C_{over}. In reinforcement learning, the reward is calculated as the negative of the total cost ($R_i = -C_i$):

$$R_{tot} = \lambda_a R_{acc} + \lambda_e R_{en} + \lambda_d R_{dur} + \lambda_f R_{for} + \lambda_o R_{over} \tag{7}$$

Weight λ_i specifies the relevance of each reward. Note that the optimal solutions obtained after training will vary with changes of these weights. As no analytical solution can be found, we set the λ_i through a coarse trial and error search over the parameter space, resulting in $\lambda_a = 100$, $\lambda_e = 10^{-6}$, $\lambda_d = 1$, $\lambda_o = 1$ and $\lambda_f = 2$. We tuned these parameters so the saccades resulting from the optimized models would have a natural look in duration and amplitude, but we did not impose any constraints on their properties related to Listing's Law or main sequence.

The accuracy cost is the squared difference between the desired goal and the final (2D) saccade orientation:

$$R_{acc} = -((r_y^d - r_y^f)^2 + (r_z^d - r_z^f)^2) \tag{8}$$

Note that there is no penalty for the torsional component r_x.

The (kinetic) energy cost is quantified by the integrated squared velocities of the motors during the saccade:

$$R_{en} = -\int_0^{t_d} \left(\dot{u}_H^2 + \dot{u}_R^2 + \dot{u}_L^2\right) dt \tag{9}$$

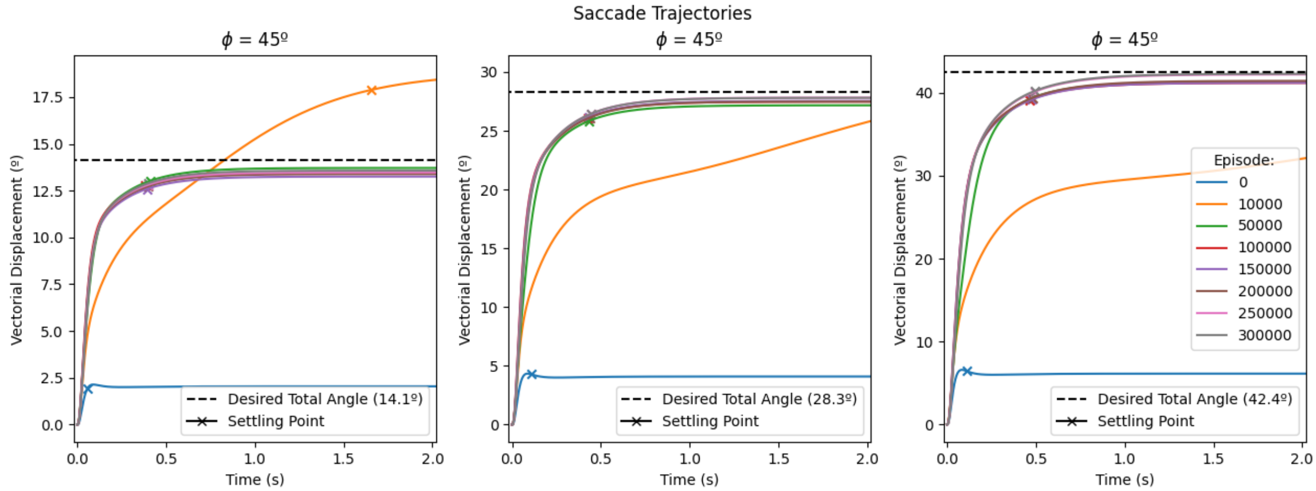

Fig. 7. Amplitude responses for three different goals, $[10^o, 10^o]$, $[20^o, 20^o]$ and $[30^o, 30^o]$, respectively, during the different stages of the training (legend).

The movement duration cost is specified by a hyperbolic discount function [16], [18]:

$$R_{dur} = -\left(1 - \frac{1}{1+\beta t_d}\right) \tag{10}$$

with t_d the saccade duration, which is taken at a settling time of 95%, and $\beta = 0.6\ s^{-1}$ the temporal discount rate.

The force cost penalizes large differences in force applied by each agonist/antagonist pair of muscles at the eye's final orientation. It aims to minimize the amount of muscle co-contraction during fixation at peripheral gaze directions:

$$R_{for} = -\sum_{i \in \{L,H,R\}} (F_i^{ag} - F_i^{ant})^2 \tag{11}$$

with $F^{ag/ant}$ the agonist/antagonist muscle force.

Finally, the overshoot cost was calculated as:

$$R_{over} = -\left(\frac{|\mathbf{r}|^{max}}{|\mathbf{r}|^{f}} - 1\right) \tag{12}$$

where $|\mathbf{r}|^{max/f}$ are the maximum and final vectorial eye displacements from the initial eye orientation.

V. Results

Stable results were obtained after training the agent for 300,000 episodes. Figure 7 illustrates the evolution of the agent's training from the start (blue traces) to episode 300,000 (grey) for three oblique saccade goals (at 14.1, 28.3, and 42.4 deg eccentricity; indicated by the dashed lines). At the start, the responses were small with a slight overshoot, but they rapidly reached near-normal saccade traces after about 150,000 epochs.

To show that the motor controllers produced pulse-step signals to overcome the sluggish plant dynamics, Fig. 8 compares three saccade responses of the model for different

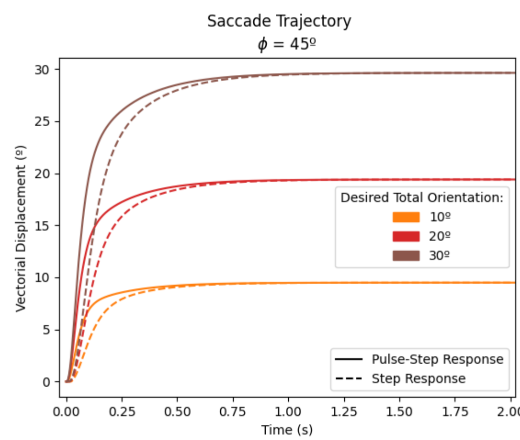

Fig. 8. Amplitude responses of the plant (dashed), and pulse-step saccade responses (solid traces) for three different goals, $[7^o, 7^o]$, $[14.1^o, 14.1^o]$ and $[21.2^o, 21.2^o]$, after training (300k episodes). The step responses were generated by setting the direct-path contribution to zero (see Fig. 1).

amplitudes (solid traces) with the associated step-only responses (dashed). Clearly, the pulse-step controlled responses were much faster than the latter (with durations $\sim$ 500 ms).

Figure 9 (left) shows some typical velocity profiles for six oblique saccades with amplitudes between 5 and 30^o, obtained after the training. Note the systematic increase in duration of the saccades with amplitude, as well as the increase in skewness of the profiles, as the peak velocity is reached at a nearly fixed acceleration epoch of about 30 ms. The relatively long exponential tails of these profiles are due to the chosen shape of the agent's policy pulses (Eqn. 3). The right-hand panel shows the main-sequence amplitude/peak velocity relation for saccades in all directions, pooled for many different initial eye orientations. Note the clear saturation of the peak velocity for large saccade amplitudes, and the considerable range of peak velocities for a given saccade amplitude (see also [1]; cf. with Fig. 2A,B). This

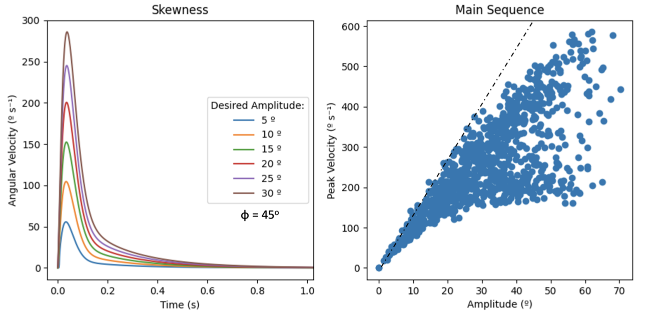

Fig. 9. Track-velocity profiles for six representative oblique saccades ($\Phi = 45^o$), and the amplitude/peak eye-velocity main-sequence relation for 1000 saccades in all directions and from different initial eye orientations. Dashed line: linear relationship, from which the data clearly deviate.

is in accordance with the data reported for human saccades [12].

In Fig. 10 we show the evoked oblique trajectories in 3D for saccades starting from the primary position (at $\mathbf{r} = \mathbf{0}$), recorded during different stages of the training (see legend in right-hand panel). Note that trajectories started extremely curved during the initial stages of the learning phase, but gradually became straighter towards the end of the training. The right-hand panel shows the amount of cyclotorsion during these trajectories. It demonstrates that already in the early stages of the learning the saccades obeyed Listing's Law, as the maximal cyclo-torsion remained well within 2.0 deg (cf. with Fig. 2C,D).

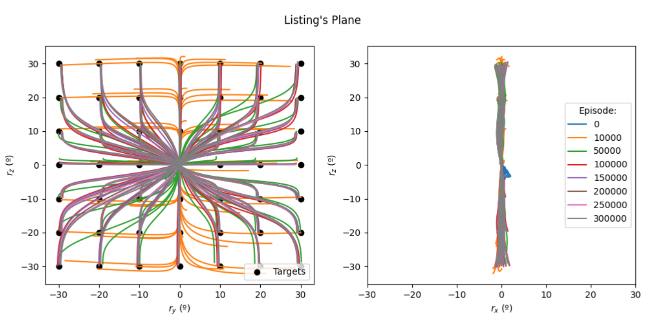

Fig. 10. 3D oblique saccade trajectories starting from the primary position are shown in the r_y, r_z plane (left), and r_x, r_z plane (right) during different phases of the training.

Although the saccade trajecories in Fig. 10 appear less straight than those found in human saccades [15], the critical property underlying human oblique saccades is the presence of cross-coupling between the horizontal and vertical components of the velocity profiles (see Background). Note that, physically, the three motors drive the oculomotor plant independently. Thus, a potential outcome of the learning could be curved trajectories in which the component velocity profiles are independent of saccade direction. This happened indeed at the start of the training (orange traces in Fig. 10). To illustrate cross-coupling in our model, we selected saccades with a fixed leftward horizontal component of 20 degrees, while the vertical component was varied between [-30, +30]^o. The left-hand panel of Fig. 11 shows the horizontal and

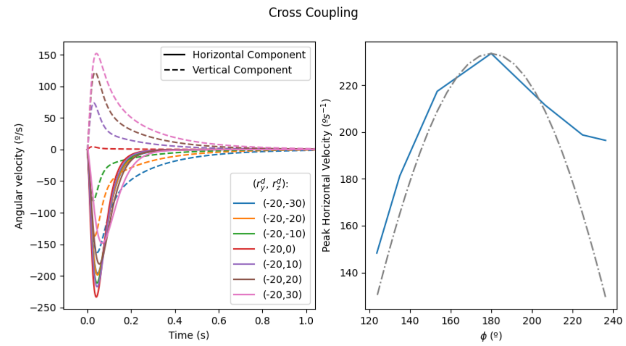

Fig. 11. Cross-coupling of the horizontal and vertical velocity profiles in oblique saccade trajectories. Dotted line shows the cosine of the saccade direction, which would lead to perfectly straight saccades. The actual amount of cross-coupling deviates somewhat from the perfect cosine, but is highly different from zero in which case all peak velocities would be the same.

vertical velocity profiles for these saccades. The right-hand panel of Fig. 11 clearly demonstrates that the peak velocity of the horizontal component strongly depended on the saccade direction in a way that closely resembled the cosine of the angle, as in human saccades. Thus, after training, the amount of cross-coupling was considerable, and of the same order as predicted for straight human saccades (dashed line).

Listing's Law was obeyed with similar accuracy as shown in Fig. 10 when the saccades were generated in arbitrary directions, starting from arbitrary initial eye orientations. This is shown in Fig. 12 for a full data set of 1000 saccade trajectories, in which only the first initial fixation started at $\mathbf{r} = \mathbf{0}$, and every subsequent saccade started from the endpoint of the previous saccade. The data show that there is no accumulation of cyclotorsion in the system.

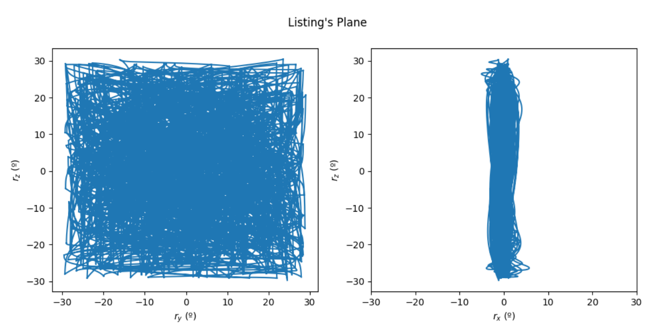

Fig. 12. Listing's Law for 1000 saccade trajectories in randomly selected oblique directions and amplitudes, starting from the end point of the previous saccade. The cyclotorsional component of eye orientation remains within 2º (standard deviation of r_x is 0.6 deg; see Table I).

VI. Discussion

The proposed method generated saccades in 3D with human-like properties that emerged from the optimization of five costs: accuracy, duration, energy consumption, fixation force, and overshoot. Importantly, these results emerged without having to impose constraints that directly relate to the known kinematic and dynamical properties of saccades.

TABLE I

PERFORMANCE COMPARISONS BETWEEN THE ORIGINAL AGENT AND THE AGENTS FOR WHICH A PARTICULAR REWARD FUNCTION IS REMOVED.

Reward Type	Mean Accuracy Error (°)	Average Settling Time of 95% (s)	Mean Distance to Listing's Plane (°)	Standard Deviation of LP (°)	Average Overshoot (%)	Average Energy Spent ($kJ\ kg^{-1}m^{-2}$)
Full	0.64	0.41	-0.08	0.58	0.0	3
$\lambda_f = 0$	0.27	0.37	1.87	0.76	0.3	2
$\lambda_o = 0$	0.51	0.19	0.04	0.68	2.8	3
$\lambda_e = 0$	0.63	0.37	0.01	0.60	0.0	57

The most remarkable emerging properties of the model saccades are: (i) a nonlinear main sequence (Fig. 9), with a strong direction-dependence and initial eye-orientation dependence of the peak track velocity as function of amplitude, with horizontal saccades starting from straight-ahead being fastest, (ii) a considerable amount of cross-coupling between the independent motors, effectively rendering the agent to act as a *vectorial* pulse generator (Fig. 11) , and (iii) Listing's Law (Figs. 10 and 12). The latter property also indicates that during oblique and vertical saccade trajectories the R and L motors jointly canceled each other's cyclotorsional torques. None of these properties was explicitly imposed on the training, but resulted from the joint trade-off of different costs in the reward functional of the reinforcement learning algorithm (7).

It is, therefore, of interest to assess the importance of each cost term in the total reward function with regard to these human-like saccade properties. To that end, we repeated the training with one of the $\lambda_i = 0$ in (7). We thus evaluated the importance of fixation force by setting $\lambda_f = 0$, by training a new agent under the same conditions as described above. We then compared the agent's performance for the same set of 1000 saccades (Figs. 9 (right) and 12) with the default agent that had access to all reward functions. Similarly, in subsequent training sessions, we set $\lambda_o = 0$, or $\lambda_e = 0$. The accuracy and duration rewards were not removed because of their self-evident impact: if accuracy is not accounted for, there is no need for the eye to move, and if the duration is not constrained, saccades will tend to be very slow (not human-like) to spend the least amount of energy possible (9).

The results of the different training sessions without force reward ($\lambda_f = 0$), overshoot reward ($\lambda_o = 0$), or energy reward ($\lambda_e = 0$) are summarized in Table I. As seen in the table, the force reward is important to produce saccades that obey Listing's Law. The mean deviation from Listing's Plane, and the width of the plane, is higher when the agent does not aim to minimize the fixation force. This confirms our earlier finding in [1]. With force minimization there is a slight degradation on the accuracy, caused by the trade-off between the force and accuracy. Note that, as our system is symmetric, the force reward bias attracts the system naturally to the the origin $\mathbf{r} = \mathbf{0}$, for which the force cost will be at a minimum.

When the energy consumption was not included as a cost, no major changes in the saccade properties were observed. This result can be explained by the already low value of $\lambda_e = 10^{-6}$. Thus, the energy budget was not a critical issue to reach suitable solutions.

The absence of the overshoot reward naturally produced a bigger amount of overshoot in the saccades. Interestingly, this overshoot was quite small, as the other cost terms also contributed to prevent large overshoots. As the average overshoots were small and stayed within the 5% range of the desired amplitude, the computed duration of the saccades shortened as well.

A. *Limitations*

We note that our method to compute saccade duration leads to underestimation of duration in case of small ($< 5\%$) overshoots. Additionally, the low decay rate of our pulse function (3) leads to an overestimation of the duration of non-overshooting saccades. As can be seen in Fig. 9 (left), the pulse function created long, relatively slow, exponential decays that lead to extended slow eye movements towards the final orientation, yielding poor assessments of the movement duration and, consequently, the energy consumption. Together, these two effects demand for the penalization of overshoot in the cost function to obtain suitable saccades. We believe that different metrics of computing duration and different pulse parameterizations may allow discarding the use of the overshoot term in the reward function.

We also note that determining the values of λ_i in the cost function is a non-trivial problem. Without a single quantitative metric to evaluate the quality of saccades, it is not possible to optimize these paramenters in a principled way. We have used a trial and error process based on the qualitative evaluation of the generated saccades by the authors.

These imperfections notwithstanding, the overall properties of the agent's saccades were remarkably similar to human performance. We conjecture from this that the precise calibration of the different components in the model (pulse shape, reward weights) was not very critical for obtaining these results.

VII. CONCLUSIONS

In this paper we have demonstrated that human-like saccades can emerge in an artificial system by learning a model-free policy using the Actor-Critic algorithm to optimize basic elementary properties of duration, accuracy, energy, force and overshoot. Complex properties like the main sequence, component cross-coupling, and Listing's Law all emerged from this optimization instead of being imposed by design. The resulting behaviors were obtained despite the limitations in defining and measuring the terms in the reward function and the weights of their contributions.

In future work we will study the application of the method to a more realistic model of the human eye with 6 independent extra-ocular muscles [19]. This model presents additional challenges in the control of muscle pretension and their organization into agonist-antagonist pairs. Also, we will work on the main identified limitations of the current formulation: (i) definition of a better metric to assess saccade duration, a parameterization of the pulse function with a faster decay, and (ii) a more principled way to define the weights of the cost function. For instance, using reinforcement learning from human preferences [20] we could better exploit the expert guidance during the learning process.

REFERENCES

[1] A. John, C. Aleluia, A. J. Van Opstal, and A. Bernardino, "Modelling 3d saccade generation by feedforward optimal control," *PLOS Computational Biology*, vol. 17, no. 5, pp. 1–35, 05 2021.

[2] S. Schulz, S. M. z. Borgsen, and S. Wachsmuth, "See and be seen – rapid and likeable high-definition camera-eye for anthropomorphic robots," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2524–2530.

[3] H. Liu, J. Luo, P. Wu, S. Xie, and H. li, "Symmetric kullback-leibler metric based tracking behaviors for bioinspired robotic eyes," *Applied Bionics and Biomechanics*, vol. 2015, pp. 1–11, 11 2015.

[4] D. Dansereau, D. Wood, S. Montabone, and S. B. Williams, "Exploiting parallax in panoramic capture to construct light fields," in *ICRA 2014*, 2014.

[5] G. Cannata and M. Maggiali, "Models for the design of bioinspired robot eyes," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 27–44, 2008.

[6] X. yin Wang, Y. Zhang, X. jie Fu, and G. shan Xiang, "Design and kinematic analysis of a novel humanoid robot eye using pneumatic artificial muscles," *Journal of Bionic Engineering*, vol. 5, no. 3, pp. 264–270, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1672652908600347

[7] M. Lakzadeh, "A biologically-inspired eye model for testing oculomotor control theories," Master's thesis, University of British Columbia, 2012. [Online]. Available: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0072549

[8] S. K. Rajendran, Q. Wei, and F. Zhang, "Two degree-of-freedom robotic eye: Design, modeling, and learning-based control in foveation and smooth pursuit," *Bioinspiration & biomimetics*, vol. 16, 05 2021.

[9] J. Iskander and M. Hossny, "An ocular biomechanics environment for reinforcement learning," *Journal of Biomechanics*, vol. 133, p. 110943, 2022.

[10] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," *CoRR*, vol. abs/1812.05905, 2018. [Online]. Available: http://arxiv.org/abs/1812.05905

[11] A. J. Van Opstal, *The auditory system and human sound-localization behavior*. Academic Press, 2016, vol. 1.

[12] A. T. Bahill, M. R. Clark, and L. Stark, "The main sequence: a tool for studying human eye movements," *Mathematical Biosciences*, vol. 24, pp. 191–204, 1975.

[13] D. A. Robinson, "Models of the saccadic eye movement control system," *Biologic Cybernetics*, 1973.

[14] S. P. S. Agostino Gibaldi, "The saccade main sequence revised: A fast and repeatable tool for oculomotor analysis," *Behavior Research Methods*, 2021.

[15] A. C. Smit, A. J. Van Opstal, and J. A. M. Van Gisbergen, "Component stretching in fast and slow oblique saccades in the human," *Experimental Brain Research*, vol. 81, pp. 325–334, 1990.

[16] R. Shadmehr and S. Mussa-Ivaldi, *Biological Learning and Control: How the Brain Builds Representations, Predicts Events, and Makes Decisions*. The MIT Press, 2012.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[18] R. Shadmehr, J. Orban, M. Xu-Wilson, and T.-Y. Shih, "Temporal discounting of reward and the cost of time in motor control," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 30, pp. 10 507–16, 08 2010.

[19] R. J. Alitappeh, A. John, B. Dias, A. J. Van Opstal, and A. Bernardino, "Emergence of human oculomotor behavior from optimal control of a cable-driven biomimetic robotic eye," arXiv, 2022. [Online]. Available: https://arxiv.org/abs/2203.00488

[20] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.