Strategies for modelling open-loop saccade control of a cable-driven biomimetic robot eye

Reza Javanmard Alitappeh1, Akhil John1, Bernardo Dias1, A. John van Opstal2, and Alexandre Bernardino1, Senior Member, IEEE

1Institute for Systems and Robotics, Instituto Superior Técnico, ISR, Lisbon, Portugal
2Dept. Biophysics, Donders Centre for Neuroscience, Radboud University, Nijmegen, The Netherlands

In human-robot interactions, eye movements play an important role in non-verbal communication. However, controlling the motions of a robotic eye that display similar performance as the human oculomotor system is still a major challenge. In this paper, we study how to control a realistic model of the human eye, with a cable-driven actuation system that mimicks the 6 extra-ocular muscles. We have built a robotic prototype and developed a non-linear simulation model, for which we compared different techniques to control its gaze behavior to match the main characteristics of saccade eye movements. In the first approach, we linearized the six degrees of freedom nonlinear model, using a local derivative technique, and designed linear-quadratic optimal controllers to optimize a cost function that accounts for accuracy, energy and duration. The second method learns a dynamic neural-network that matches the system dynamics, trained from sample trajectories of the system, and a non-linear trajectory optimization solver optimized a similar cost function. We focused on the generation of rapid saccadic eye movements with fully unconstrained kinematics, and the generation of control signals for the six cables that simultaneously satisfied several dynamic optimization criteria. The model faithfully mimicked the three-dimensional rotational kinematics and dynamics observed for human saccades. Our experimental results indicate that while the linear model provides a more accurate eye movement, the nonlinear model simulate eye dynamic properties in a better way faithful approximation to the properties of the human saccadic system than the linearized model, at the cost of larger training and optimization time.

Index Terms—human oculomotor system; gaze control; recurrent neural networks; degrees of freedom; main sequence kinematics; Listing's law; optimal control.

I. INTRODUCTION

With increasing utilization of robots in our daily lives, effective human-robot interaction (HRI) and communication is becoming an important challenge. Gaze plays an important role in social interactions, not only by signalling one’s attention to external events, but also to reflect attitudes, affects, or emotions [1]. This has spawned numerous studies of gaze behavior in human-robot interaction (see [2], [3], [4], [5], for a review), not only to enable robots to correctly respond to human gaze signals, but also (and more relevant to the present study) to equip robots with legible gaze behaviors when communicating with a human interlocutor.

Thus, an important objective for effective HRI systems is to develop biomimetic robotic eyes that can perform natural gaze behaviors [6] [7]. In addition, while the appropriate control of a biomimetic robotic eye can play a vital role in HRI systems, it can also be highly useful to better understand the human oculomotor system, including the vast variety of oculomotor disorders [8] [9], the principles that guide active vision when scanning the environment [10], [11], or other applications that aim to model and understand sensorimotor behaviors that are similar to that of humans.

Most existing approaches to display realistic eye movements in robots either try to replicate pre-recorded psychophysical data from real human gaze shifts [5], [12], [13] or hard-code the biological control rules into the robot control system [14], [15]. Although these may seem viable approaches to design a biomimetic robotic eye, they are limited in several ways. First, the underlying principles that rule the emergence of the gaze kinematics are left uncovered. Thus, if the robot's mechanical structure does not perfectly match that of the human eye, the resulting behavior may not be “optimal”. Furthermore, if new movement tasks have to be implemented, new behavioral data will be needed, since it is not clear how to generalize the movement control principles from one task to another. Second, most existing models have so far addressed relatively simple settings, but have so far neglected the planning and execution of the full dynamics that underlie the eye-movement trajectories. Moreover, typical robotic eyes are equipped with only 2 degrees-of-freedom (DOF), usually pan-tilt serial kinematics to control the azimuth and elevation of gaze, and cannot independently control ocular cyclotorsion.

To address these limitations, in our recent work, we considered the generation of saccades (rapid eye movements) in a 3 DOF eye model [16]. The bio-inspired eye had three independent motors that were each coupled to an agonist-antagonistic cable pair. The feedforward optimal control proposed could reproduce the dynamics and kinematics of human saccadic eye movements in all directions, when the total cost comprising accuracy, energy expenditure, saccade duration, and total fixation force was minimized. In the present paper, we extend this work by making the the eye model more biologically accurate and abandoning the restriction of coupled agonist-antagonist cables. For this we designed a new scaled up biomimetic cable-driven eye prototype with six independent actuators to mimic the six extraocular muscles. A simulation of the prototype was used to investigate different
techniques to find the optimal control policy. The proposed approaches would possibly lead to energy-efficient and more durable robotic systems, with more flexibility in replicating the complex repertoire of oculomotor behaviors exhibited by humans.

II. BACKGROUND

The eye is enclosed inside a conical cavity, where fats and other connective tissues restrict its translation [17], thus, the eye effectively moves with only three degree of freedom, in a rotational motion. The human eye is actuated by six extracocular muscles that control its orientation (see Fig. 1). The muscles on the left and right of the eye, the medial (MR) and lateral (LR) rectus, mostly rotate the eye horizontally. The top and bottom muscles, the superior (SR) and inferior (IR) rectus, along with the sideways pointing muscles, the superior (SO) and inferior (IO) oblique, allow for vertical and cyclo-torsional rotations. It is important to note that the pulling directions of these muscles are not completely independent of each other. As can be seen in Fig. 1, any eye orientation in 3D space can be defined by rotations in $x$, $y$, and $z$ specifying torsional ($\psi$) angle, elevation ($\theta$) and azimuth ($\phi$), correspondingly. Each of these three rotational motions, although not independent, are mainly driven by actuating a pair of extra-ocular motors in an approximate agonist-antagonist configuration, where each motor pulls the eye in a different direction. The eye is enclosed inside conical cavity, where fats and other connective tissues restrict its translation [17], thus, the eye effectively moves with only three degree of freedom, in a rotational motion.

A. Saccade kinematics

Typically, humans make about 3-4 saccades per second to scan the visual environment [18] and these are constrained by Donders’ law, which restricts the movement degrees-of freedom of the eyes from three to two [19]. Donders’ law states that any eye orientation has a unique cyclotorsional angle, regardless of the path followed by the eye to reach that orientation. In other words: $\psi = f(\theta, \phi)$, with $\psi$ the torsional angle around the $x$-axis, $\theta$ the horizontal angle around the $z$-axis, and $\phi$ the vertical angle around the $y$-axis. The precise shape of the function $f(\theta, \phi)$ depends on the motor system and motor task (e.g., vergence, vestibular, etc.). It has been argued that Donders’ law avoids the problems associated with the non-commutativity of rotations in three dimensions, which becomes especially important when planning sequences of eye- and head movements [20]. Listing’s law is a further specification of Donders’ law. It provides an extra restriction on the eye’s cyclotorsional angle for a specific condition: for any eye orientation, with the head upright and still, and the eye looking at infinity: $\psi = 0$. Listing’s law thus constrains all such eye orientations to the $(y, z)$ plane (Fig. 1). Note that Listing’s law not only holds during steady eye fixations, but also during smooth-pursuit eye movements and rapid saccadic eye movements. The law does not hold during eye-head coordination, during vestibular or optokinetic stimulation, or for disjunctive vergence eye movements to fixate nearby targets. For those movements, Donders’ law applies.

B. Saccade dynamics

A further important property of saccadic eye movements is their nonlinear dynamics, described by the so-called main sequence, which specifies the relationship between the saccade amplitude and its duration (an affine relation), and between amplitude and peak eye-velocity (a nonlinear, saturating relation). In addition, saccades reach their peak velocity at an approximately fixed acceleration phase of about 20-25 ms [21], regardless the saccade amplitude. This causes the skewness of the saccade velocity profile to increase from near zero (symmetric) for small saccades, to more positively skewed profiles for larger saccade amplitudes [22]. Finally, behavioral experiments have shown that oblique saccade trajectories are approximately straight [23]. As a consequence, the horizontal and vertical component velocities are scaled versions of each other (i.e., they are synchronized), resulting in a significant stretching of the smaller component when it participates in an oblique saccade. For example, a purely horizontal 10 deg saccade can have a peak velocity of 300 deg/s and a duration of 50 ms. However, when it is part of a 20 deg oblique saccade, made at an angle of 60 deg with the horizontal, the peak velocity of the 10 deg horizontal component will be reduced to about 150 deg/s ($\cos(60) \cdot 300$) and its duration will have increased to about 80 ms (corresponding to the duration of a 20 deg saccade).

III. RELATED WORK

Even though eye movements have been studied for a long time, they were mostly restricted to 1D simulation studies. The use of robotic eye models for implementation and understanding of humanoid eye movements has been fairly recent [14], [15]. In [14] a neurophysiologically inspired model of
combined saccadic eye-head movements in a robotic eye-head system was implemented, in which each of the two eyes had two degrees-of-freedom (pan and tilt). The system could successfully replicate the velocity profiles of human eye-head gaze shifts, but note that the experiments were performed exclusively along a single dimension, i.e., either horizontal or vertical saccades. In [15] a tendon-driven mechanical eye (the MacEye) was designed to comply with Listing’s law (see section II), by appropriate routing of the cables and precisely calculated insertion points. The orientations of the eye complied with Listing’s law, but the dynamics of the eye’s behavior were not presented. Although a hardwired implementation of Listing’s law seems an interesting engineering solution, it will only be valid for head-restrained eye movements with the head upright and for gazing at far objects. Thus, the MacEye system lacked the true mechanical degrees of freedom that would allow it to generate natural movements of the eyes with three degrees of freedom.

In a related study, Saeb et al. [24] implemented an open-loop neural controller with a local adaptation technique to control an eye-head robotic model. It was shown that the model follows nonlinear behavior of the eye system very well, however in contrast to our proposal, they have a high level control signal for the system without considering the complexity of 6 independent motors to control.

In our recent work we used feedforward open-loop optimal control based on linear approximation of a nonlinear biomimetic robotic eye to reproduce most of the dynamics and kinematics of human saccadic eye movements in the 3 DOF unconstrained robotic eye [16]. The biomimetic eye had three independent motors that were each coupled to an agonist-antagonistic cable pair. Linear approximation for the open-loop controller was done using system identification of the input motor commands to get the optimal trajectory, such that the minimization equation can be written as:

$$\min \sum_{i=0}^{K} C(\mathbf{x}_i, \mathbf{u}_i), \quad s.t. \quad \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i)$$

(1)

Motor angles are limited to upper and lower bounds according to maximum and minimum lengths of the cables.

**Problem 2 (Trajectory optimization):**

In a saccadic system we would find an optimum trajectory $\mathbf{\tau}$ from initial state $\mathbf{x}_0$ to the final state $\mathbf{x}_T$ in a finite interval $i \in [0, T]$.

If we consider $\mathbf{U}$ as a sequence of input motor commands $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_T]$, in this problem, we want to optimize the input motor commands to get the optimal trajectory, such that the minimization equation can be written as:

$$\min \sum_{i=0}^{K} C(\mathbf{x}_i, \mathbf{u}_i), \quad s.t. \quad \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i),$$

where, $C(\cdot)$ indicates the cost of the trajectory, that typically composed of a linear combination of partial cost on the properties of the trajectory, e.g. duration, accuracy, energy. More detail about the cost functions are explained in the next sections.

**V. DESIGNING CABLE-DRIVEN ROBOTIC EYE**

Figure 3 shows a mechanical prototype of our system. Like the human eye, the robotic eye will rotate around its fixed center from the net torque exerted by the six elastic
cables, which represent the extra-ocular muscles. The cables are inserted on the eye ball at similar contact points as on the human eye, to allow it to rotate with 3 DOF. Each cable is controlled by its own motor to rotate at a given speed, thus pulling the cable rapidly around its spindle and exert a torque on the eye. Because the pulling directions of the cables vary with the orientation of the eye, and each muscle can only pull (not push), the total system (as described below in more detail) is nonlinear [16]. To rotate the robotic eye in the same way as the human eye during rapid saccades (i.e., accurate and at high speeds) therefore becomes a highly nontrivial problem.

In order to study the oculomotor system’s properties and behavior, an eye simulator was developed. The eye was designed as a sphere with a fixed center, subject to Newton’s rigid body equation, and Euler’s equation of motion [25] (equation 2) with six cable-driven actuators mimicking the human eye muscle kinematics.

\[ \alpha = \mathbf{I}_e^{-1} (\omega \times \mathbf{r} - \mathbf{w} \times \mathbf{c}) \]  

In the equation above, \( \alpha \) is the acceleration of the eye with respect to the head frame, in eye coordinate frame. \( \mathbf{I}_e \) represents the inertia tensor for the eye model, \( \omega \) is the angular velocity of the eye and \( \tau \) depicts eye torque. This last variable depends on damping and elasticity torques, \( \tau_d \) and \( \tau_k \), respectively.

\[ \tau = \tau_d + \tau_k = \sum_{i=1}^{6} \tau_i - \mathbf{D}_{eye} \omega \]  

where \( \mathbf{D}_{eye} \) quantifies the eye’s damping matrix and

\[ \tau_i = \mathbf{Q}_i \times \mathbf{f}_i \]  

\( \mathbf{Q}_i \) represents individual muscle’s final contact point on the eye. Force applied by the cables \( (\mathbf{f}_i) \) depends on their individual length. Cable length is determined by the magnitude of vectorial difference between a fixed routing point of the cables (represented as red points in figure 4) and the final contact point on the eye (dark blue points). The length of the cables is variable, depending on the rotation of the motors, and this change defines the force exerted by the cables. The force equation is based on Hooke’s law [26]

\[ \mathbf{f}_i = \frac{k}{l_0} (l_i(u) - l_0) \mathbf{\phi}_i \]  

where \( k \) is the stiffness constant (Young’s modulus) of the cables and \( l_0 \) is the cable length when looking straight ahead for cable \( i \). \( l_i \) is the updated length after a rotation and \( \mathbf{\phi}_i \) is the direction in which the force is applied. Also, the system’s inertia, stiffness and damping parameters were defined to closely replicate the time constants and overdamped characteristics of the human eye.

Note that muscles can only apply pulling force in the human eye, and to emulate said constraint in this system, force was set to zero if it went negative. When this happens, the cables are not applying tension, a phenomenon known as slack. The initial set of motor angles (pre-tension) is chosen such that the cables are taut when the eye is at rest orientation and to mitigate slack even throughout the trajectory.

VI. APPROXIMATION MODELS FOR SYSTEM DYNAMIC

A. Linear Model

From the non-linear system, a variation-based linearization was performed, in which an infinitesimal perturbation is applied to the system around an equilibrium point to obtain a linear approximation [27]. Here, a Lie Group [28] transfer is applied, where a local (linear) state is created from a global equilibrium state and the perturbation is applied to it. Then, through exponential mapping, the resulting orientation and angular velocity in the local state are transformed into the global (nonlinear) state. Consider the following differential equations that represent the system’s nonlinear dynamics.

\[ \dot{w}_e \mathbf{R}_e = w_e \mathbf{R}_e \omega \mathbf{w}_e, \]  

\[ \dot{\omega}_w,e = \mathbf{I}_e^{-1} \left( \tau \omega_e - \omega \omega_e \times \mathbf{I}_e \omega \right) \]  

Consider also the definition of a local state around the equilibrium point \( \mathbf{x} \) as

\[ \dot{\mathbf{x}} = \{ \dot{w}_e \mathbf{R}_e, \dot{\omega}_w,e \} \]  

with

\[ \dot{\mathbf{R}} = \dot{w}_e \mathbf{R}_e \omega \mathbf{w}_e, \]  

\[ \ddot{\omega}_w,e = \mathbf{w}_e \omega_{w,e} - \mathbf{w}_w,e \]  

where \( \mathbf{R} \) is the rotation between equilibrium orientation \( \mathbf{R} \) and the actual orientation \( \mathbf{R} \). Using exponential notation, \( \mathbf{R} \) can be represented by the skew-symmetric of a rotational perturbation \( \eta \).

\[ \dot{\mathbf{R}}(t) = \exp(\eta(t)) \]  

Adapted from [27], an infinitesimal change in body angular velocities can be given as:

\[ \delta \omega(t) = \delta \omega(t) \eta(t) + \dot{\eta}(t) \]  

Fig. 4. The EOM setup where points with dark blue color indicate insertion points on the eye and red points indicate insertion points on the head. It should be noticed that the cable pulling directions are not symmetric about the eye, since both the insertion points on the head are skewed towards one side.
The local state can therefore, be represented in local coordinates and matrix form, as:

$$\dot{\xi} = \frac{d}{dt} \begin{bmatrix} \eta_e \\ \delta \omega_c \end{bmatrix} = \begin{bmatrix} -\omega_e^c \eta_e + \delta \omega_c \\ e \tau_c^{-1} \varepsilon \tau_c - \omega_e \times e \dot{\omega} e \end{bmatrix}$$

(12)

Having all this information allows to build a linear state space model in the following fashion:

$$\dot{\xi} = A \xi + B \delta u$$

(13)

This linear model proved to be faithful to the non-linear system for relative small orientations around the linearization point. Furthermore, the discrepancy between the systems increased when the perturbation was applied to the first state variable, the eye’s orientation. A perturbation applied on the velocity is not as noticeable in the non-linear system.

### Table I

| Perturbation           | Max Magnitude at Origin | Max Magnitude at 30 deg |
|------------------------|-------------------------|-------------------------|
| $\delta R_{\text{eye}}$ | $7.85^\circ$            | $8.89^\circ$            |
| $\delta \omega_{\text{eye}}$ | No max                 | No max                  |

Table I shows the maximum eccentricity for which the accuracy error of a given saccade is below 5%. This means that the approximation is accurate for small orientations around the equilibrium point around which the system is linearized.

### B. Nonlinear NARX Model

A different approach in learning the optimal control in complex robotics applications has been to apply machine-learning techniques, including deep learning and reinforcement learning [30], [31], [32], [33]. For instance, in [34], the authors applied a recurrent neural network (NARX) to learn the forward dynamic model of a soft robot manipulator. Using the learned model, they then designed a trajectory optimization method for predictive control of the manipulator. In a subsequent study, the authors improved their controller by implementing a closed-loop scheme using model-based reinforcement learning [35].Feedforward networks have been improving over the recent decade trying to increase the performance of tasks in classification, detection and recognition. One common type of the networks, in these applications, are based on Convolutional layers which works efficiently on image data. Differently, Recurrent Neural Networks (RNNs) are appropriate for sequence data, such as video, audio and text, where the time is matter. This is done in RNN using a loop mechanism to send the signal from one step to another step, which can produce output in every time step. The non-linear autoregressive network with exogenous inputs (NARX) is one of the RNNs that can be used for the sequence data such as control signals in our saccadic systems.

In our model, we consider current control input, $u_t$ (motor command) and current state, $x_t$, of the system as the inputs, to predict the state in the next time (See Fig. 5). By having motor command as an exogenous input, we designed our Recurrent Neural Networks (RNNs) based on a non-linear autoregressive network with exogenous inputs (NARX) model. This network fits on sequence data which considers dependencies between the time steps of the input data. In our discrete time scheme the NARX model can be written as:

$$x_t = f[x_{t-1:t-n_x}, u_{t-1:t-n_u}],$$

(14)

where, $x_t$ in the network prediction of the state in time $t$, which is depend on control input from time $t$ to $t - n_u$ and network output (model state) from $t - 1$ to $t - n_x$. These two variables $n_u$ and $n_x$ are the time delay corresponding to the input and state of the model (in our model they are the same). In Fig. 5, the architecture of the applied network in our model is shown. $n_h$ is the number of neurons in hidden layer. $u_t$ and $x_t$ are the input of the network. $b_j$ and $b_i$ are the bias in both layers. The output of each layer is computed by applying activation function $f_1$ and $f_2$. Finally, the matrix $w$ contains the weights of the network, where it will be tuned after training the model.

Fig. 5. The architecture of the NARX model with input, hidden and output layers.

The output of the neuron $i$ in hidden layer $H_i(t)$ in time $t$ can be computed using:

$$H_i(t) = f_1 \left[ \sum_{l=0}^{n_r} w_{ir} u_{t-l} + \sum_{l=1}^{n_x} w_{il} x_{t-l} + b_i \right],$$

(15)

$w_{ir}$ is the weight between $i^{th}$ neuron and $u(t - r)$, and $w_{il}$ is the weight between $i^{th}$ neurons and output neuron $y(t - l)$. And, the output of the network is computed using below equation:

$$x_j(t) = f_2 \left[ \sum_{l=0}^{n_h} w_{j} H_i(t) + b_j \right],$$

(16)

To clarify the reader, why we did not use other recently used RNN, such as LSTM (Long Short Time Memory) network, it is because of the amplitude of the input signals in our model. Maximum saccade length in a real eye system is around 300ms, so the problem of short term memory or vanishing problem does not apply for our input signal, which could be a reason to use LSTM.

### VII. Optimal Control

In this section we are going to explain our linear and non-linear control. But first, list of the cost functions that are used in our both controllers is explained.
1. Duration ($J_D$)

Duration term indicates the time of a saccades from initial to final state ($J_D$), which is defined as below:

$$J_D = 1 - \frac{1}{1 + 3K}, \quad (17)$$

$J_D$ forces the optimizer to find the shortest path between initial and final orientation, thus saccades will be penalize in a hyperbolic scheme [36] according to their length $K$. Saccade with shorter length $K$ has better duration cost.

2. Accuracy ($J_A$)

Accuracy cost $J_A$ is defined according to the Euclidean Norm distance between the final orientation reached by the controller and the desired goal. This term is taken with the purpose of penalizing the change in state when the eye reaches the goal. With this cost it is possible to toll the change in orientation and velocity, when compared to the state reached by the optimal control. The function that translates this objective into a mathematical context is:

$$J_A = \sum_{i=1}^{w} (x^K - x^{K+1})^2 \quad (18)$$

where $x^K$ is the final state and $w$ is a time window after the final state.

3. Energy ($J_E$)

Here, it is aimed to toll the energy expenditure. Since it is assumed that the energy is proportional to the actuator’s rotation (angular velocity), and the timesteps are uniform, this term can be simplified as the difference between angular positions at each timestep. So, it can simply be written as the difference between consecutive $u_i$'s.

$$J_E = \|\Delta u\|^2 \quad (19)$$

where $J_E$ represents the energy cost, $\Delta u$ is the difference between the current vector of motor commands and the previous one. In matrix form, this can be written as Eq. 20.

$$J_E(u) = \left\| \begin{bmatrix} u_0 \\ u_1 - u_0 \\ \vdots \\ u_p - u_{p-1} \end{bmatrix} \right\|^2 \quad (20)$$

A. Linear control

For the task at hand, optimal control is responsible for minimizing a cost function to get the ideal duration of the saccade and control inputs, and therefore, an optimal performance from the system regarding trajectory and effort expenditure. In order to attain said input, for each saccade, the optimal controller is applied to the linear system (local state). This problem is formulated in the following fashion:

$$\min_{u,K} J(u, K) = \sum_{\alpha} \lambda_{\alpha} J_{\alpha}(u, K)$$

subject to

$$\begin{align*}
x_{i+1} &= A x_i + B u_i \\
y_i &= C x_i, \quad i = 0, 1, ..., K \\
u &\geq 0 \quad (21)
\end{align*}$$

where $J_{\alpha}$ represents each cost term and $\lambda_{\alpha}$ is the weight associated with each individual cost term. $K$ is the optimal saccade time. For each $i$, the costs are computed, allowing to find the optimal time that has the minimum total cost for a specific saccade. The solver that adapts best to the needs at hand is quadprog. This is because the problem can be formulated to have quadratic costs, and linear constraints. The output from the solver provides the optimal inputs to the system which then go into the cost function to find the best trajectory for a specific goal.

B. Non-linear Control

In the non-linear control we defined a different representation of trajectories, which is explained in the following. After running the optimizer for several time, for the sake of computational simplification we reduce the number of variables in our optimization problem. To do so, we used Gaussian Mixture Model (GMM), where we approximate the variables in our optimization problem. To do so, we used Gaussian Mixture Model (GMM), where we approximate the input motor command $u$ with multiple Gaussian functions. So, instead of finding values for $\tau$ with multiple Gaussian functions, we reduce the number of variables by creating a trajectory $\tau$ with size $K$, we look for $P$ basis for $\varphi$ where $P < K$.

$$\tau = \varphi.\mu \quad (22)$$

$$\varphi_i = \frac{\exp\left(\frac{(t-c_i)^2}{h}\right)}{\sum_{m=1}^{N} \exp\left(\frac{(t-c_m)^2}{h}\right)} \quad (23)$$

where, in time $t$, $h$ is the reduction rate, $c_i$ is the $i^{th}$ center of the Gaussian function. $\mu$ is the weight matrix which multiplies by $\varphi$ to create the trajectory $\tau$.

Three main steps in our non-linear proposal are: data collection, the forward dynamic and trajectory optimization.

Fig. 6. Random saccades in an eye workspace with length 2M ms. The sphere shows the eyeball.
The forward model has been designed based on a recurrent neural network, which need a dataset of saccadic command to train the eye behavior. In order to create the dataset, continuous saccades are produced with 1ms frequency using a random function in the eye simulator (explained in Sec. V). The total saccade length is 2M ms, which, as can be seen in Fig. 6, cover the entire workspace uniformly. To reduce the size of the dataset sampling with step size 3ms is applied. According to the nature of the motor commands this sampling doesn’t make a major change to the dataset. An example is shown in Fig. 7, where length of the signal is reduced from 3000ms to 1000ms without changing in shape of the motor command signal.

After creating the forward dynamic NARX model and training by using the dataset explained above, the model can reaches to a new state given the current orientation and motor command (explained in Section VI-B). Finally, in the next step we are simulating a saccadic motion starting from an arbitrary command (explained in Section VI-B). Finally, in the next step we are searching for the optimum trajectory leading the eye from an initial orientation to a final destination. To do so, we have set the simulation setup, instead of motor angles we feed GMM parameter \( \mu_0 \) as the initial value to the optimizer, which generates \( \mu \) and \( \tau \) consequently. The NARX model simulate behaviour of an eye system given the input \( \tau \) and generates its corresponding states, finally we compute the costs according to the states \( \mathbf{x} \) to evaluate the input saccade. We run this loop for number of iterations, or stop by meeting termination condition.

![Fig. 7. Applying sampling with step size 3ms on a motor command signals for 3000ms, its’ length is reduced to 1000ms without change in shape of the signals.](image)

We applied MATLAB function ’fmincon’ with ’sqp’ solver to optimize our constrained nonlinear problems. One of the challenges in this optimization is to define cost weights \( \lambda_s \). To do so, at first, we applied some manual setting to find out the range of candidate values, later we applied a meta-heuristic approach to search for optimum \( \lambda \) values. The detail of the proposed optimization is shown in Alg. 1, where the inputs are initial and final orientation of the eye \( (\mathbf{x}_0, \mathbf{x}_T) \). We also have parameters that should be set at the beginning of the algorithm as well, i.e. \( d \) is the saccades division number, \( ts = T/d \) is the length of the saccades for each portion of the division, \( maxiter \) points to the maximum number of iteration in the solver; \( c \) is the coefficient to define the number of basis in the GMM, and \( \lambda_s \) are the costs coefficient, each shows importance of the corresponding cost term. As you can see in Eq. 25, the first cost \( J_{D} \) indicates the length of the saccade which means the duration takes the eye to move from initial to the final orientation. To compute this cost we divide the entire saccade length into \( d \) portions, and run the optimizer for different length \( l \) in range \( l = \{ts, 2ts, 3ts, \ldots, T \} \) for a single saccade. In another word, we have \( d \) runs for every saccade. Above explanation can be found in Alg. 1 lines 2 to 6, where we define number of basis for GMM according to the length of the saccade in the current repeat \( l \), later \( \varphi \) and \( \mu_0 \) are defined given the basis number. The function which will be optimized is defined in line 6, and finally it is called using the Solver in line 6. After completing the loop the minimum cost will be our desired result which will be used to compute the optimum trajectory \( \tau \) in lines 7-8. In Alg. 2 the function Policy_cost is used in our optimizer with \( \mu \) and \( \varphi \) as its inputs and total cost \( J \) as the output. After simulating the eye behavior using the motor commands in NARX model, we are able to compute the costs in lines 3. The total cost is computed as a weighted sum of the costs and their coefficients (line 4).

**VIII. Simulation Results**

To evaluate our proposal, we will analyse several parameters of the system in its kinematic (Listing’s plane) and dynamic (velocity profile) behavior. After setting our simulation setup, we will discuss the performance of our model given the performance measurements we defined.
Fig. 9. A saccade sets (24 goal orientations) in the range of 5 to 30 degree. Cost function of a zero initial horizontal saccade (22°) in the trajectory optimization. The point indicates the minimum cost of the trajectory at 130ms. And it’s corresponding motor angles of the optimum trajectory.

Algorithm 1: Trajectory_Optimization()

Input: $x_0$, $x_T$ // Initial and final orientation
1 param ← $d$, $ts$, Maxiter Set initial values in param.

for $i < d$ do
2 $\hat{t}$ ← $i \cdot ts$ length of saccade.
3 Basis_Num ← $c \cdot l$ Number of Basis for the GMM.
4 $\varphi$ ← GMM(Basis_Num) Create $\varphi$ using the GMM.
5 $\mu_0$ ← CreateMu(Basis_Num) Initial $\mu$.
6 cost$_i$ ← Solver(policy_cost($\mu$, $\varphi$, $\mu_0$, param)) calling optimizer with policy cost function.
7 $\mu_{\text{opt}}$ ← $\min$(cost) find the optimum $\mu$ given the costs.
8 $\tau_{\text{opt}}$ ← $\mu_{\text{opt}} \cdot \varphi$ optimum trajectory.

Algorithm 2: Policy_cost($\mu$, $\varphi$)

Input: $\mu$, $\varphi$ // Inputs are $\mu$ and $\varphi$
Output: $J$ // total cost

1 $\tau$ ← $\varphi \cdot \mu$ Create trajectory.
2 $x$ ← NARX($\tau$) Simulate the NARX model.
3 $J$ ← compute_cost($x$)
4 $J$ ← $\lambda_a J_a$

A. Simulation Setup

To have a fair condition for the proposed methodologies (linear and nonlinear control), we created a saccade set (goal orientations) containing 24 different eye movements in horizontal, vertical and oblique directions with amplitudes from 5 to 33 degrees (see Fig. 9 a)). It should be mentioned that we consider two types of motion on the saccade set: zero-initial saccades, where every saccade starts from the origin ([0, 0, 0]), and continuous one, where the next saccade starts from the final orientation of previous saccade and so on. All of our simulation are done in MATLAB in a laptop with Windows 10, 16GB Ram and CPU core i7.

The cost functions $J$ in our optimization equations (Eq. 25 and Eq. 24) in both linear and nonlinear controls are defined in a similar way, comprising Accuracy, Duration and Energy costs, thus:

$$J = \lambda_A J_A + \lambda_D J_D + \lambda_E J_E$$

However, their multipliers $\lambda$ are different due to different time sampling, forward model and optimization algorithm. These multipliers for each of the costs were manually calibrated in an attempt to have human-like dynamic characteristics emerge from the system. This calibration was one of the most challenging parts in our proposal, i.e by increasing the weight for the accuracy cost for a saccade, more energy and duration might be needed. The value of the multipliers are shown in Table II. An example cost function is shown in Fig. 9 b), where the optimum point is detected on the convex curve indicating the total weighted cost. It can be seen that we run the optimizer for 10 different duration in the range 30 to 210ms, and the run with minimum total cost (6th run with duration 130ms in this example) is selected as the optimum result. In Fig. 9 c), motor angles for the saccade corresponding to the optimum point is shown. It represents a purely horizontal saccade, reason why the vertical cables (m1 and m3 pair) do not move.

**Table II**

| Parameter | $[\lambda_A, \lambda_D, \lambda_E]$ |
|-----------|-----------------------------------|
| linear    | $[0.35, 1, 0.06]$                |
| nonlinear | $[1, 0.04, 0.002]$               |

The architecture of the NARX network in our experiment has been defined with 1 hidden layer including 55 nodes in this layer, with Levenberg-Marquardt backpropagation technique as training function. According to Alg. 1 the value of $d = 10$, $ts = 100$, maxiter = 15 and $c = \frac{1}{\Delta}$, in our experiments. As we mentioned, $T$ represents maximum saccade duration, given the studies in real eye systems [18] the maximum duration of a saccade is around 250 – 300ms. However for the sake of simplification described in the next paragraph, after dividing to step size 3, we set $T = 100ms$. Regarding the simulation results, in order to have a better understanding, however, all the figures are shown in actual length (out of 250 – 300ms). By fitting the dataset (explained
in Sec. VII-B) to the NARX model, the best result is achieved after 96 epoch with MSE 0.0018. In Fig. 10, the result of testing the trained NARX network given a random set of saccades with length 1000ms is shown, which verifies how well the network learned the forward dynamic of our system.

Regarding the linear model, the most important feature to setup is the pre-tension. This set of values had to be iterated and constrained to make sure the eye is able to reach any orientation in the oculomotor range, while not drifting due to an excess of force produced by the cables. In order to obtain said behavior, we optimize the initial motor angles in the following way.

$$\min_\theta F(\theta) = \|\theta\|^2$$
subject to
$$f(x, \theta) > 0$$
$$\tau_{total}(x, \theta) = 0$$

(25)

As mentioned above, with the aim of not having slack by keeping the cables taut, we make sure that the angles provided by the solver create a force bigger than zero and that said force generates an overall null torque. This way it is made sure that our eye model is kept in equilibrium while static.

For this approach we also consider a timestep of 1ms in order to minimize discretization error and also to always permit the system to reach a convex solution in our cost minimization.

### B. Kinematic behavior Analysis

In order to study kinematic behavior of the saccades in our controllers, we check trajectory deviation from the Listing’s plane. Such that, zero-initial and continuous movements for both controllers are shown in XY planes in Fig. 11. According to eye movement in XY, it can be shown that the saccades follow Listing’s law (blue vertical line) in general. However, in zero initial setup, trajectories for the longer saccades in the nonlinear control has less deviation than the linear one. The same way, continuous saccades has more variation than zero initial ones in both controllers. For a better understanding, we show a precise numerical comparison with a division on the saccade set amplitude in three categories: short saccades with amplitude less than 10 degree, medium saccades in range 10 to 20 degrees, and large saccades with range 20 to 50 degrees. In Table III, the average standard deviation of the eye movement around the Listing’s plane is shown. Although the scale and difference of the errors is low, we have smaller error for the continuous saccades in nonlinear control in comparison to the linear one.

In Fig. 11, the trajectories of the eye movement in YZ planes are visualized. While in the linear control trajectories are more straight, in the nonlinear one they are with a curvature.

The accuracy of our models, for the two scenarios, are shown in Fig. 12, where represents the distance of the point reached by our models to the desire goal points from the saccade set. In the plots, the error range is between 0 to 6 degrees, and as expected longer saccade has bigger error. In general given the trajectories in YZ plane (Fig. 11) and the accuracy vs amplitude plots (Fig. 12) linear control is a little bit more accurate movement than the nonlinear one.

### C. Dynamic behavior Analysis

As it was mentioned before about the main sequence properties, skewness is one of the metrics used to evaluate proper functionality of an eye system. Velocities in x, y and z direction for the zero initial in linear and nonlinear controls are shown in Fig. 13. However, the skewness of the velocity profiles is not visible, given that we have symmetric velocity profiles. It is thought that the skewness witnessed in humans is due to the noise in the control signal sent by the brain, and since in our models there is no random noise, this skewness does not emerge from the system. In humans, peak velocity saturates with increasing saccade amplitude but saccade durations increase almost linearly with amplitude. Furthermore,
the acceleration duration is similar for all amplitudes, which means that the difference in saccade duration is due to their deceleration phase. This is called velocity profile skewness. Summarily, higher amplitude saccades have the same peak velocity, but take longer to reach their goal we can either just use this sentence and remove the long description or remove this. The controlled simulator respects part of this behavior, as seen in Fig. 14. It is clear that duration increases linearly with saccade amplitude and that peak velocity tends to saturate after a given amplitude.

**D. Discussion**

Given the experiments we have done, we realized that most of the properties depend on saccade amplitude, in general, thus the models have more difficulty to reach to the larger saccade with a longer distance to the goals. We also observed that the linear model has a more accurate movement in compression to the non-linear, although it has more deviation on longer saccades, while the non-linear moves with less deviation from the Listing’s plan. In both models we have drifting in the velocity profile at least in one of the components, which causes a movement close to zero velocity while the signal is relaxing. One important property in our models is pretension which is directly related to initial motor angles, such that depend on the initial pretension the output signal may change. So, finding optimal initial motor commands is another challenging in our proposals.

**IX. Conclusion**

The 6-independent muscles solution is closer to the biological counterpart and potentially lead to highly realistic gaze behaviours. However, it is significantly more challenging that the coupled agonist-antagonist approach of [16]. Not only we have to solve for the redundancy of the added degrees of freedom, that must tightly coordinate to implement an implicit agonist-antagonist strategy, but also take into account muscle pretension to prevent slack in the cables during the whole saccadic movement. In this paper we developed a full 3D non-linear dynamical model of the system of our cable-driven biomimetic robot and developed both linear and nonlinear approximations for the open-loop optimal control when generating human-like saccadic eye movements in all directions. For the continuation we would implementation the optimal controls which we proposed in this paper on a real eye model. Moreover, extend the controllers to include neck as well.

**Acknowledgment**

This research was supported by the European Union Horizon 2020 program, ERC Advanced Grant, 879 2016 (project ORIENT, nr. 693400; AJvO, AJ, AB) and the Instituto Tecnico Superior, 880 Lisbon (CA)

**References**

[1] M. Argyle, R. Ingham, F. Alkema, and M. McCallin, “The different functions of gaze,” 1973.
[2] H. Admoni and B. Scassellati, “Social eye gaze in human-robot interaction: a review,” Journal of Human-Robot Interaction, vol. 6, no. 1, pp. 25–63, 2017.
[3] K. Kompatsiari, F. Ciardo, V. Tikhonoff, G. Metta, and A. Wykowska, “It’s in the eyes: The engaging role of eye contact in hri,” International Journal of Social Robotics, vol. 13, no. 3, pp. 525–535, 2021.
[4] K. S. Lohan, E. Sheppard, G. Little, and G. Rajendran, “Toward improved child–robot interaction by understanding eye movements,” IEEE Transactions on Cognitive and Developmental Systems, vol. 10, no. 4, pp. 983–992, 2018.
Fig. 13. Velocity profile of the zero initial saccade set in x, y and z direction, created by the linear (a,b,c), and nonlinear (a,b,c) control.

Fig. 14. Ratio between amplitude vs duration and peak velocity for linear control (two most left) and nonlinear (two most right) controllers (data points are for both zero initial and continuous saccades).

[5] N. F. Duarte, M. Raković, J. Tasevski, M. I. Coco, A. Billard, and J. Santos-Victor, “Action anticipation: Reading the intentions of humans and robots,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4132–4139, 2018.

[6] C. McGinn, “Why do robots need a head? the role of social interfaces on service robots,” International Journal of Social Robotics, vol. 12, no. 1, pp. 281–295, 2020.

[7] E. De Wolf and J. Li, “Double trouble: The effect of eye gaze on the social impression of mobile robotic telepresence operators,” in International Conference on Social Robotics. Springer, 2020, pp. 357–368.

[8] L. M. Ward and Z. Kapoula, “Differential diagnosis of vergence and saccade disorders in dyslexia,” Scientific Reports, vol. 10, no. 1, p. 22116, Dec 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-79089-1

[9] M. J. Thurtell, R. L. Tomsak, and R. J. Leigh, “Disorders of saccades,” Current neurology and neuroscience reports, vol. 7, no. 5, pp. 407–416, 2007.

[10] J. Peng, A. Srikaew, M. Wilkes, K. Kawamura, and A. Peters, “An active vision system for mobile robots,” in Smc 2000 conference proceedings, 2000 ieee international conference on systems, man and cybernetics, cybernetics evolving to systems, humans, organizations, and their complex interactions /cat. no. 0, vol. 2. IEEE, 2000, pp. 1472–1477.

[11] A. Vasilyev, “Optimal control of eye movements during visual search,” IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 4, pp. 548–559, 2019.

[12] T. Kanda, H. Ishiguro, T. Ono, M. Imai, and R. Nakatsu, “Development and evaluation of an interactive humanoid robot ‘robevie’,” in Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2. IEEE, 2002, pp. 1848–1855.

[13] Y. Mohammad, S. Okada, and T. Nishida, “Autonomous development of gaze control for natural human-robot interaction,” in Proceedings of the 2010 workshop on Eye gaze in intelligent human machine interaction, 2010, pp. 63–70.

[14] E. S. Maini, L. Manfredi, C. Laschi, and P. Dario, “Bioinspired velocity control of fast gaze shifts on a robotic anthropomorphic head,” Autonomous Robots, vol. 25, no. 1, pp. 37–58, 2008.

[15] D. Biamino, G. Cinnata, M. Maggiiali, and A. Piazza, “Mac-eye: A tendon driven fully embedded robot eye,” in 5th IEEE-RAS International Conference on Humanoid Robots, 2005. IEEE, 2005, pp. 62–67.

[16] 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. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1008975

[17] R. S. Snell and M. A. Lemp, Clinical anatomy of the eye. John Wiley & Sons, 2013.
[18] J. Van Opstal, The auditory system and human sound-localization behavior. Academic Press, 2016.

[19] F. C. Donders and W. D. Moore, On the anomalies of accommodation and refraction of the eye: With a preliminary essay on physiological dioptrics. New Sydenham Society, 1864, vol. 22.

[20] K. Hepp, “On listing’s law,” Communications in Mathematical Physics, vol. 132, no. 1, pp. 285–292, 1990.

[21] S.-N. Yang and G. W. McConkie, “Eye movements during reading: A theory of saccade initiation times,” Vision research, vol. 41, no. 25-26, pp. 3567–3585, 2001.

[22] A. Van Opstal and J. Van Gisbergen, “Skewness of saccadic velocity profiles: a unifying parameter for normal and slow saccades,” Vision research, vol. 27, no. 5, pp. 731–745, 1987.

[23] J. Van Gisbergen, A. Van Opstal, and J. Schoenmakers, “Experimental test of two models for the generation of oblique saccades,” Experimental brain research, vol. 57, no. 2, pp. 321–336, 1985.

[24] S. Saeb, C. Weber, and J. Triesch, “Learning the optimal control of coordinated eye and head movements,” PLoS computational biology, vol. 7, no. 11, p. e1002253, 2011.

[25] S. Traversaro and A. Saccon, “Multibody dynamics notation (version 2),” 2019.

[26] J. Rychlewski, “On hooke’s law,” Journal of Applied Mathematics and Mechanics, vol. 48, no. 3, pp. 303–314, 1984.

[27] G. Wu and K. Sreenath, “Variation-based linearization of nonlinear systems evolving on so(3) and s^2,” IEEE Access, vol. 3, pp. 1592–1604, 2015.

[28] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, “Lie-group methods,” Acta numerica, vol. 9, pp. 215–365, 2000.

[29] Y. Yu and X. Ding, “Trajectory linearization control on so (3) with application to aerial manipulation,” Journal of the Franklin Institute, vol. 355, no. 15, pp. 7072–7097, 2018.

[30] M. Pereira, Z. Wang, I. Exarchos, and E. A. Theodorou, “Learning deep stochastic optimal control policies using forward-backward sdes,” arXiv preprint arXiv:1902.03986, 2019.

[31] D. Clever, M. Harant, K. Mombaur, M. Naveau, O. Stasse, and D. Endres, “Cocomopl: A novel approach for humanoid walking generation combining optimal control, movement primitives and learning and its transfer to the real robot hrp-2,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 977–984, 2017.

[32] J. Mainprice, R. Hayne, and D. Berenson, “Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning,” in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 885–892.

[33] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, “Optimal and autonomous control using reinforcement learning: A survey,” IEEE transactions on neural networks and learning systems, vol. 29, no. 6, pp. 2042–2062, 2017.

[34] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, “Learning dynamic models for open loop predictive control of soft robotic manipulators,” Bioinspiration & Biomimetics, vol. 12, no. 6, p. 066003, oct 2017. [Online]. Available: https://doi.org/10.1088/1748-3190/aa839f

[35] ——, “Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators,” IEEE Transactions on Robotics, vol. 35, no. 1, pp. 124–134, 2018.

[36] R. Shadmehr, J. J. O. De Xivry, M. Xu-Wilson, and T.-Y. Shih, “Temporal discounting of reward and the cost of time in motor control,” Journal of Neuroscience, vol. 30, no. 31, pp. 10 507–10 516, 2010.