A Cartesian 6-DoF Gaze Controller for Humanoid Robots

Alessandro Roncone*†, Ugo Pattacini*, Giorgio Metta*, and Lorenzo Natale*
*iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy
†Social Robotics Lab, Computer Science Department,
Yale University, New Haven, CT, U.S.A.
Email: alessandro.roncone@yale.edu

***Abstract*—In robotic systems with moving cameras control of gaze allows for image stabilization, tracking and attention switching. Proper integration of these capabilities lets the robot exploit the kinematic redundancy of the oculomotor system to improve tracking performance and extend the field of view, while at the same time stabilize vision to reduce image blur induced by the robot's own movements. Gaze may be driven not only by vision but also by other sensors (e.g. inertial sensors or motor encoders) that carry information about the robot's own movement. Humanoid robots have sophisticated oculomotor systems, usually mounting inertial devices and are therefore an ideal platform to study this problem. We present a complete architecture for gaze control of a humanoid robot. Our system is able to control the neck and the eyes in order to track a 3D cartesian fixation point in space. The redundancy of the kinematic problem is exploited to implement additional behaviors, namely passive gaze stabilization, saccadic movements, and vestibulo-ocular reflex. We implement this framework on the iCub's head, which is equipped with a 3-DoFs neck and a 3-DoFs eyes system and includes an inertial unit that provides feedback on the acceleration and angular speed of the head. The framework presented in this work can be applied to any robot equipped with an anthropomorphic head. In addition we provide an open-source, modular implementation, which has been already ported to other robotic platforms.**

I. INTRODUCTION

Gaze control is a fundamental behavior for any robotic system with moving cameras, and it is particularly important in the field of humanoid robotics. Its goal is to allow the robot to focus its gaze on objects of interest, or just to prompt the robot curiosity towards the exploration of the surrounding environment. To this end, the coordinated motion of the eyes and the neck system must be as fast and smooth as possible, in order to track rapidly moving targets and reduce vibrations and image blur.

This problem has been widely investigated in the literature. There is a large body of work that addresses the problem of controlling anthropomorphic heads. The majority of this work draws inspiration from human studies and models the coordinated movements of the human eye-head system (cf. e.g. Guitton and Volle [10]). Maini et al. [14] designed a controller on the basis of the independent gaze control model introduced by Goossens and Van Opstal [8] as the outcome of experimental data gathered on humans. By contrast, Lopes et al. [13] applied well-established control theories – such as optimal control and feedback controller design – to realize a system exhibiting the peculiar features of the human gaze. Duran et al. [6] follow an original approach that makes use of nonlinear dynamics and chaos theory with a formalism introduced in Kaneko and Tsuda [11]. It employs Coupled Map Lattices (CML), and demonstrates the emergence of a self-organized capability of the system in controlling the axes of a simulated eye to perform smooth pursuit. It is worth noting, though, that most of this work has been carried out in simplified environments, i.e. either by using anthropomorphic heads with a relatively low number of degrees of freedom (e.g. Agravante et al. [2] used a 2-DoF head, Shaw et al. [22], Bernardino and Santos-Victor [3] worked with a 4-DoF system), or by implementing a limited set of features: indeed the majority of these works focus on tracking tasks in the image plane (e.g. [12, 17]), or design a constrained framework that does not take explicitly into account the control of the fixation point in 3D space (e.g. [3, 12, 17, 9]).

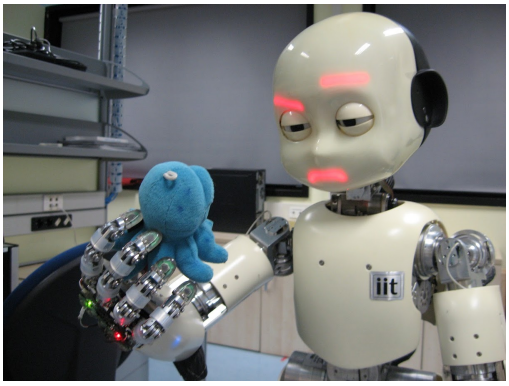

Fig. 1: The iCub humanoid robot [16] inquisitively gazing at a toy octopus.

Regrettably, in the literature related to gaze control little attention has been given to one of the main components of the human oculomotor control, i.e. gaze stabilization. It is a passive mechanism that counteracts for both self-generated motions and external disturbances, with the goal of reducing image blur and generally aiding vision-based tasks. Gaze stabilization in humans is a relatively well-understood mechanism

[5], and it has been demonstrated how its role is crucial during active gaze control tasks [15]. However, to date gaze controllers largely concentrate on the active component of the control loop, and no integration between these two systems has been proposed. In a previous work, a gaze stabilization framework has been implemented on the iCub humanoid robot [20], and we are indeed in the position of integrating active gaze control with passive gaze stabilization.

The iCub head [16] is equipped with 6-DoFs whose kinematics approaches the complexity of the human oculomotor system. For this reason it has been chosen as the testbed where to implement and validate our architecture. We focus our attention in the explicit control of the fixation point, determined as the 3D point that is fixated by both eyes of a generic binocular system. Without loss of generality, it is possible to define the fixation point as the virtual end-effector of the kinematic chain composed by the neck and the eyes. Consequently, control of the fixation point can be achieved using techniques for inverse kinematics and trajectory generation normally adopted for controlling robotic arms, e.g. [19]. We exploit the redundancy of the task under consideration by implementing additional capabilities to our system. To this end, we draw inspiration from the human gaze, and we enrich the system with three bio-inspired behaviors: vestibulo-ocular reflex (VOR), saccadic behavior, and gaze stabilization.

The contribution of this paper is a comprehensive architecture for controlling the fixation point of a binocular system. This architecture includes a rich set of behaviors for smooth pursuit and attention switching using open loop saccades. The system coordinates eyes and head DoFs to achieve fixation with large speed and it includes strategies for stabilization that use inertial feedback as well as feedback from joint encoders and feedforward commands. In addition, the architecture defines a generic software interface that can be implemented on different robots. The control framework is implemented and validated experimentally on the iCub platform; the software is released with an open source license for the benefit of the whole community.

II. METHOD

We define the gaze control problem as the task of steering the head motors of a generic robot in order to move its fixation point toward a desired 3D position $\hat{\boldsymbol{x}}_{FP}$. In the context of this work, we employ the iCub humanoid robot [16], which is equipped with a six-degrees of freedom (DoFs) head, as depicted in Figure 2. As a consequence of this, the gaze task is inherently redundant. The goal of the controller presented in this work is to leverage on this redundancy in order to accomplish a number of corollary behaviors, which are aimed at improving the performance of the system under different scenarios. To this end, a number of additional modules have been introduced, namely a vestibulo-ocular reflex (VOR), a gaze stabilizer, and a saccadic system.

The basic skeleton of the proposed architecture is depicted in Figure 3. Given the 3D cartesian position of the target $\hat{\boldsymbol{x}}_{FP}$, the gazing problem is decoupled into two separate tasks,

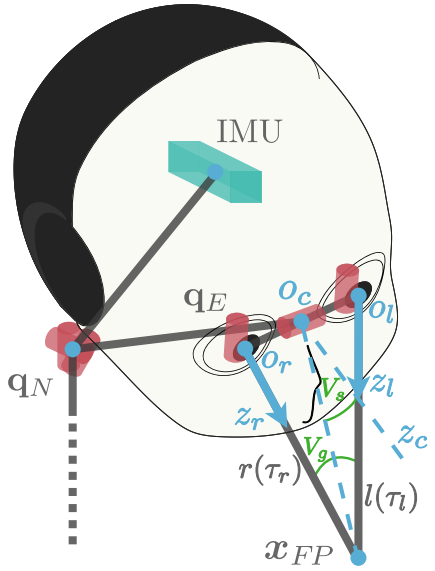

Fig. 2: Kinematics of the head system for the iCub humanoid robot. It is composed of a 3-DoFs neck and a 3-DoFs binocular system, for a total of 6-DoFs (depicted in red). Each of these joints is responsible for the motion of the fixation point. The Inertial Measurement Unit (IMU) is the green rectangle, whose motion is not affected by the eyes' movements.

which account for the neck and the eyes respectively. This is beneficial for the performance of the system, since the neck and the eyes exhibit different dynamics – the latter being faster than the former. The independent control of the two subsystems speeds up the convergence of the gaze toward the desired target. Further, each of these two subtasks is divided into two different stages. The first stage deals with the inverse kinematic task, i.e. finding a set of suitable joint configurations $\boldsymbol{q}^*_{N,E}$ that achieve the desired fixation point. The second module consists of a biologically inspired kinematic controller that computes the joint velocities $\dot{\boldsymbol{q}}^*_{N,E}(t)$ needed to generate a minimum-jerk, quasi-straight trajectory of the fixation point. The three secondary tasks – VOR, saccades, stabilization – are aimed at exploiting the redundancy of the problem, and complement the block diagram of Figure 3 at various stages. For simplicity, they will be detailed in their respective Sections.

This Section will be divided as follows: Sections II-A and II-B detail the forward and inverse kinematic problem of computing the fixation point given the neck and eyes configuration of the robot (and vice-versa). Section II-C describes the vestibulo-ocular reflex, and finally Section II-D illustrates the control problem, that is the computation of the head velocities needed to move the fixation point to the desired target.

A. Gaze Forward Kinematics

The iCub head system is depicted in Figure 2. It is provided with a three degrees of freedom (DoFs) neck, a 3-DoFs eyes system, and two cameras, allowing for tracking and

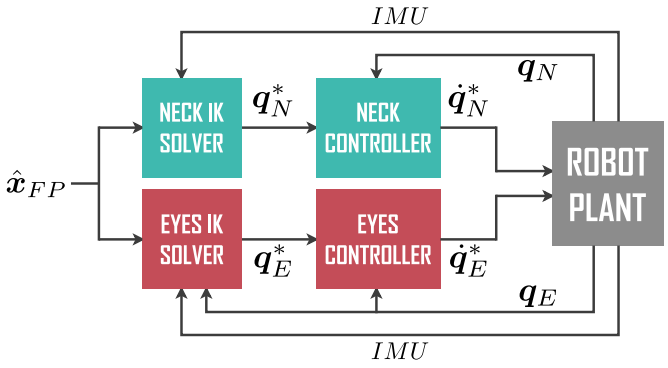

Fig. 3: Block diagram of the presented framework. The control task is decoupled into two independent subsystems, which account for the neck and the eyes respectively. For each of them, a two-stage design (consisting of an inverse kinematic solver and a velocity controller) computes the set of joint velocities needed for moving the fixation point from its current position to the desired target.

vergence behaviors. The movement of the eyes is coupled, following an anthropomimetic arrangement. With appropriate calibration, depth information can be extracted from binocular disparity. With the exception of the eyes, it is possible to express the kinematics of the head system according to the standard Denavit-Hartenberg (DH) representation. Indeed, for what concerns the eyes, three DoFs (the tilt t_c, the version v_s and the vergence v_g) are responsible of four coupled joints (namely the tilt and pan for the two cameras $[\mathtt{t}_l,\ \mathtt{p}_l]^T$ and $[\mathtt{t}_r,\ \mathtt{p}_r]^T$). In particular, the eyes configuration $\mathbf{q}_E$ is given by:

$$\mathbf{q}_E = \begin{bmatrix} t_c \\ v_s \\ v_g \end{bmatrix} = \begin{bmatrix} \mathtt{t}_l = \mathtt{t}_r \\ \dfrac{\mathtt{p}_l + \mathtt{p}_r}{2} \\ \mathtt{p}_l - \mathtt{p}_r \end{bmatrix}, \tag{1}$$

which results in the inverse relations: $\mathtt{t}_l = \mathtt{t}_r = t_c$, $\mathtt{p}_l = v_s + v_g/2$, and $\mathtt{p}_r = v_s - v_g/2$.

We define the fixation point $\boldsymbol{x}_{FP}$ as the intersection of the lines $l(\tau_l)$ and $r(\tau_r)$ that originate from the left and right eyes, as described in Figure 2. The forward and differential kinematic problems for a binocular system have been addressed in related work [20]. Consequently, we are already provided with a suitable DH representation for the neck and eyes system, along with higher order Jacobian and Hessian matrices. Collectively, they have been implemented into an open source kinematics library called *iKin*[1].

B. Gaze Inverse Kinematics

By resorting to the formulation provided in Pattacini et al. [19], a generic inverse kinematic problem can be defined as the computation of a suitable set of joint angles $\boldsymbol{q}^* \in \mathbb{R}^n$ such that the end effector reaches a desired position $\hat{\boldsymbol{x}} \in \mathbb{R}^m$. Further, $\boldsymbol{q}^*$ has to usually satisfy a number of constraints (e.g. joint limits), expressed as a set of inequalities. In its simplest formulation, the problem can be stated as follows:

$$\boldsymbol{q}^* = \arg\min_{\boldsymbol{q}\in\mathbb{R}^n} \left\| \hat{\boldsymbol{x}} - K_{FP}(\boldsymbol{q}) \right\|^2 \quad \text{s.t.} \left\{ \boldsymbol{q}_l < \boldsymbol{q} < \boldsymbol{q}_u \right., \tag{2}$$

where $K_{FP}(\boldsymbol{q})$ is the forward kinematic function, and $\boldsymbol{q}_l$ and $\boldsymbol{q}_u$ are the joints' lower and upper limits. For the gaze problem, the input space is $n = 6$ (equal to the number of DoFs at the neck + eyes plant), whereas the output space is $m = 3$ (i.e. the desired fixation point is simply defined as a 3D position). Hence, the optimization problem is intrinsically redundant. As detailed in Section II, Eq. 2 has been divided in two independent subproblems that solve the task for the head and the eyes subsystems, as described in the following Sections.

1) Neck Solver: The neck solver is tasked with redirecting the robot's forehead toward the target fixation point. To this end, it is convenient to consider the line $z_c(\boldsymbol{q}_N)$ that originates at the virtual point O_c placed at the midpoint of the baseline that connects the two eyes, and orthogonal to it, as depicted in Fig. 2. The point O_c is given by:

$$O_c = \frac{O_l + O_r}{2}. \tag{3}$$

Under this condition, the inverse kinematic problem can be defined as finding the neck configuration $\boldsymbol{q}_N^*$ that minimizes the following cost function:

$$\begin{aligned} \boldsymbol{q}_N^* &= \arg\min_{\boldsymbol{q}_N\in\mathbb{R}^3} \{cos(\theta)\} = \\ &= \arg\min_{\boldsymbol{q}_N\in\mathbb{R}^3} \left\{ \frac{z_c(\boldsymbol{q}_N)\cdot\left(O_c - \hat{\boldsymbol{x}}_{FP}\right)}{\left\|O_c - \hat{\boldsymbol{x}}_{FP}\right\|} \right\} \\ &\quad \text{s.t.} \left\{ \boldsymbol{q}_l < \boldsymbol{q}_N < \boldsymbol{q}_u \right. \end{aligned}, \tag{4}$$

where θ is the angle between the unitary axis $z_c(\boldsymbol{q}_N)$ and the vector $O_c - \hat{\boldsymbol{x}}_{FP}$ that goes from the desired fixation point to the forehead. By minimizing its cosine, the resulting configuration at the neck joints will keep these two vectors anti-parallel, thus achieving the desired goal. Eq. 4 is characterized by an input space dimensionality of 3 (equal to the DoFs at the neck), and an output space dimensionality of 2, since the optimization problem concerns the goal of aligning two vectors into the 3D space. The task is therefore redundant. In order for the inverse kinematic solver to exploit the available redundancy, Eq. 4 has been transformed into the following, without loss of generality:

$$\boldsymbol{q}_N^* = \arg\min_{\boldsymbol{q}_N\in\mathbb{R}^n} \left\| \hat{\boldsymbol{q}_N} - \boldsymbol{q}_N \right\|^2 \quad \text{s.t.} \begin{cases} \cos(\theta) < -1 + \epsilon \\ \boldsymbol{q}_l < \boldsymbol{q}_N < \boldsymbol{q}_u \end{cases}, \tag{5}$$

[1] The *iKin* library, originally developed in [18], is available in the iCub software repository at `http://wiki.icub.org/iCub/main/dox/html/group__iKin.html`

with ϵ an arbitrary small number (we chose it to be in the range $\left[10^{-5}, 10^{-4}\right]$). Equation 5 wraps the minimization criterion defined by Eq. 4 as a nonlinear constraint and introduces a new minimization task that accounts for the distance of the solved configuration from a set of desired neck joints values $\hat{\boldsymbol{q}_N}$. A constraint has higher priority than the objective function and thus plays the role of a primary task: this entails that the objective function is minimized only after all the given constraints are satisfied, if possible. We choose the resting configuration $\hat{\boldsymbol{q}_N}$ in order to maintain the head posture vertical with respect to the gravity as much as possible, as in Milighetti et al. [17]. To this end, $\hat{\boldsymbol{q}_N}$ is determined at run-time according to the current output of the accelerometer sensor built into the inertial measurement unit (cf. Figure 2). Through the inertial sensor readouts and the forward kinematics library (*iKin*, cf. Section II-A)), the gravity vector can be expressed in the head reference frame, and the desired resting configuration can be adequately modified to compensate movement of the other joints of the body (i.e. the torso or in general the whole-body). This is beneficial in many practical situations when the robot executes tasks that require visual input and we want the head to remain stable with respect to the direction of gravity (i.e. during locomotion or whole-body movements).

2) The Eyes Solver: It is possible to specialize Eq. 2 for the eyes plant as follows:

$$\boldsymbol{q}_E^* = \arg \min_{\boldsymbol{q}_E \in \mathbb{R}^n} \left\| \hat{\boldsymbol{x}}_{FP} - K_{FP}\big(\boldsymbol{q}_E\big) \right\|^2 \quad \text{s.t.} \left\{ \boldsymbol{q}_l < \boldsymbol{q}_E < \boldsymbol{q}_u \right. , \tag{6}$$

where K_{FP} computes here the 3D position of the fixation point considering only $\boldsymbol{q}_E$ as independent variables. This task does not allow any redundancy and for this reason it is sufficient to solve it resorting to the inverse Jacobian algorithm, i.e.:

$$\dot{\boldsymbol{q}_E} = G \cdot J^{-1} \cdot \big[\hat{\boldsymbol{x}}_{FP} - K_{FP}\big(\boldsymbol{q}_E\big)\big], \tag{7}$$

where J^{-1} is the inverse of the Jacobian of the gaze forward function (cf. Section II-A), and G is a suitable gain. For geometrical reasons, the columns of the $[3 \times 3]$ matrix J are always linearly independent due to the kinematics of a binocular system, hence the Jacobian is always full rank and thus invertible. Equation 7 is solved via iterative time-discrete methods, whose convergence rate is quick thanks to the low dimensionality of the search space. It is therefore possible to obtain the reference trajectory as it evolves over time to be tracked by the second-stage eyes' controller depicted in Fig. 3.

C. The Vestibulo-Ocular Reflex (VOR)

Eq. 7 can be adapted to incorporate a term $\dot{\boldsymbol{q}}_c$ that accounts for the counter-rotation required to compensate for head movements. The complete formulation is thus the following:

$$\dot{\boldsymbol{q}_E} = G \cdot J^{-1} \cdot \big[\hat{\boldsymbol{x}}_{FP} - K_{FP}\big(\boldsymbol{q}_E\big)\big] - \dot{\boldsymbol{q}}_c. \tag{8}$$

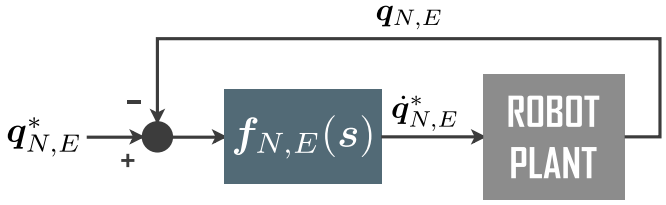

Fig. 4: Two identical (albeit independent) controllers have been developed for the neck and eyes systems. They take into account the desired joint configuration for either the neck or the eyes subsystems, and thanks to a minimum-jerk Laplace transfer function (Eq. 9), they output a smooth, minimum-jerk velocity trajectory in the joint space, that can be directly tracked by the low-level motor controllers of the robot.

The component $\dot{\boldsymbol{q}}_c$ is obtained from the velocity of the fixation point $\dot{\boldsymbol{x}}_{FP}$. To this end, we are currently employing two techniques for the computation of $\dot{\boldsymbol{x}}_{FP}$:

1) a cartesian implementation of the vestibulo-ocular reflex (VOR), which is a crucial element in the human eye-hand coordination mechanisms (see e.g. Rosander and von Hofsten [21]). It can be implemented by means of the gyroscope output $\boldsymbol{\omega}_{IMU} = [\omega_x, \omega_y, \omega_z]^T$ provided by the inertial measurement unit (IMU), which moves jointly with the head;
2) alternatively, the velocity of the fixation point can be computed thanks to the kinematic feed-forward function of the torso and head system.

The counter-rotation vector $\dot{\boldsymbol{q}}_c$ is then computed from $\dot{\boldsymbol{x}}_{FP}$ by means of the standard differential kinematics methods available in the *iKin* library (cf. Section II-A).

D. Control Problem

1) Neck and Eyes Controller: With reference to Figure 3, after the inverse kinematic step (described in Section II-B), the desired neck and eyes configurations are then passed to the second-stage controllers. These controllers are in charge of computing velocity profiles to command the motors, given the desired attractor point in the neck configuration space (Section II-B1), and the reference trajectory in the eyes space (Section II-B2). To realize the controllers we reuse the design proposed in Pattacini et al. [19], where a linear time-invariant (LTI) system is employed to best approximate the time-varying (LTV) feedback implementation of the minimum-jerk trajectory generator. The underlying rationale is that numerous neurophysiological findings (e.g. Abend et al. [1], Flash and Hogan [7]) indicate that human movements can be essentially described by means of the so-called minimum-jerk model, which captures the property of the human limbs of moving with symmetric bell-shaped velocity profiles, characterized by a smooth onset as well as a gentle ending part of the trajectory while approaching the set-point. Thereby, the controllers' Laplace transfer function, which applies to all the six head

joints independently, is the following:

$$\frac{\dot{q}_{N,E}}{q^*_{N,E} - q_{N,E}} = \frac{a/T_{N,E}}{s^2 + \left(c/T^3_{N,E}\right) \cdot s + b/T^2_{N,E}}, \quad (9)$$

where the coefficients are chosen as in [19] to be $a \approx 151, b \approx 85, c \approx 16$, and T_N and T_E specify the point-to-point execution time for the neck and the eyes joints, respectively. In general, T_E is lower than T_N; we use as default values $T_N = 0.75\,\text{s}$ and $T_E = 0.25\,\text{s}$, although they can be conveniently tuned to meet particular requirements. As sketched in Figure 4, Equation 9 provides the desired bell-shaped velocity profiles (three velocities $\dot{q}_N$ for the neck and three velocities $\dot{q}_E$ for the eyes) to command the joints, given the error in the configuration space between the reference $q^*_{N,E}$ and the current joint position $q_{N,E}$. While the attractor points $\boldsymbol{q}^*_N$ are directly computed by the neck's solver, we have to integrate the reference velocity resulting from Equation 8 to obtain the equivalent set-points $\boldsymbol{q}^*_E$ for the eyes joints. Remarkably, the reference velocities generated via Equation 8 have exponential profiles, characterized by highly asymmetric bell shapes with snap onset and slower decay toward the target. Therefore, they could be fed straight to the motors for direct control purposes, hence bypassing the integration step so as the LTI block described above, but would produce movements significantly different from those observed in humans, which are instead minimum-jerk. Further, this difference appears more prominent when slower movements of the eyes are requested (i.e. higher T_E). We refer the reader to the experimental evaluation detailed in Pattacini et al. [19] for deeper insights on the comparison between exponential and minimum-jerk trajectories.

2) Gaze Control in presence of inertial stabilization: Gaze stabilization has the goal of maintaining the vision system during locomotion or generic whole body control as steady as possible. Related work has recently presented a gaze stabilization system which takes advantage of the inertial feedback coming from the IMU sensor [20]. In this paper we describe its integration in the proposed control architecture. Figure 5 depicts the stabilization system and how it interacts with the neck subsystem (solver and controller) detailed in Figure 3. The gaze stabilizer in this case actuates only the motors of the neck, since for the eyes a vestibulo-ocular reflex has been implemented differently (cf. Section II-C).

The stabilization is achieved by means of a closed-loop velocity control stage which uses the movement of the head as it is sensed by the inertial unit. The stabilization commands are computed as the neck motor commands $\dot{\boldsymbol{q}}_N$ that compensate for this movement. The gyroscope measurements coming from the inertial sensor are first projected on the reference frame of the fixation point; this quantity ($\dot{\boldsymbol{x}}_{FP}$) is then reprojected on the neck joints using inverse kinematics to compute the stabilization command. It is important to notice that this part of the control is open-loop and it is composed only of algebraic conversions.

The design of the system detailed in Figure 5 envisages the identification of the plant (robot plus IMU) with a PID

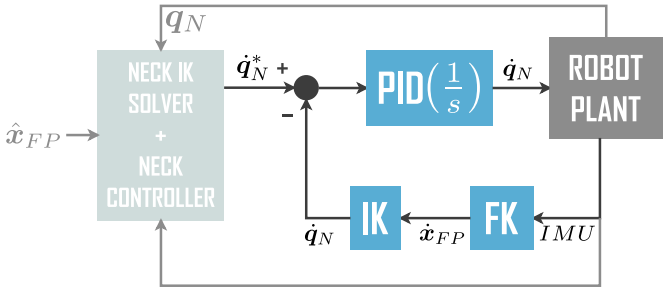

Fig. 5: Block diagram of the neck stabilization system integrated into the gaze control loop. The neck velocities $\dot{\boldsymbol{q}}^*_N$ computed by the neck controller are combined with the velocities needed by the system to keep the fixation point as stable as possible.

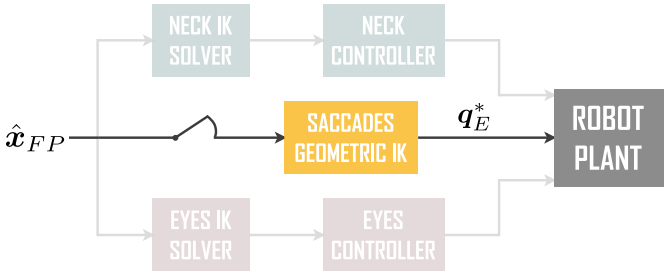

Fig. 6: Block diagram of the controller in presence of saccadic behavior. Upon user request, and according to various contextual parameters, a geometric inverse kinematic is tasked with generating feedforward eyes configurations able to reach the target fixation point as much as possible.

controller. The design of the discrete PID controller (running at a frequency of 100 Hz) has the objective to minimize the response time of the plant when it undergoes an input stepwise load disturbance. After the identification and tuning stage, a simple integral controller has been found to be sufficient.

The stabilization module is integrated into the main control architecture by means of a classic cascade control system. The outer loop (Fig. 3) is tasked with actively controlling the position of the fixation point with minimum-jerk trajectories, whereas the inner loop in Figure 5 allows the system to effectively maintain the reference velocities even when the plant is subject to external disturbance.

3) Saccadic behavior: Saccadic behavior is crucial for the gazing in humans and higher level animals. It is a quick feedforward command that exploits the fast dynamics of the eyes system in order to efficiently foveate a visual target. A saccadic movement is normally followed by the movement of the head towards the target. This movement allows centering the eyes with respect to the head while maintaining fixation on the target. To maintain fixation, the eyes are controlled by the vestibular input to counter rotate and compensate the movement induced by the head.

In this work, saccades are achieved by means of an inverse

kinematics block, as described in Figure 6. It shortcuts the gaze controllers and directly computes the eyes configuration $\boldsymbol{q}_E$ needed to fixate the desired point. Using the *iKin* kinematic library, it is possible to express the fixation point $\hat{\boldsymbol{x}}_{FP}$ into a convenient reference frame centered in the virtual point O_c mentioned in Section II-B, whose $z-$axis is normal to the robot's forehead (cf. Figure 2). The desired eyes' tilt and version angles are then computed by expressing the transformed fixation point $\hat{\boldsymbol{x}}'_{FP}$ into spherical coordinates $[\theta, \varphi]$:

$$\begin{aligned} t_c = \theta &= \arctan\left(\frac{\hat{x}'_{FP,y}}{\hat{x}'_{FP,z}}\right) \\ v_s = \varphi &= \arctan\left(\frac{\hat{x}'_{FP,x}}{\hat{x}'_{FP,z}}\right) \end{aligned} . \tag{10}$$

The vergence v_g can be estimated in open loop by means of geometrical computations. We first apply as further transformation to $\hat{\boldsymbol{x}}'_{FP}$ the rotation given by the predicted tilt t_c, resulting in $\hat{\boldsymbol{x}}''_{FP}$. In this way, we are in the condition of estimating the vergence as $\mathtt{p}_l - \mathtt{p}_r$ (see Equation 1), where the two components $\mathtt{p}_l$ and $\mathtt{p}_r$ can be still retrieved using trigonometric functions. For example, in case $\hat{x}''_{FP,x}$ is greater than $b/2$, with b representing the length of the eyes baseline (i.e. the robot is looking rightward), we have:

$$\begin{aligned} \mathtt{p}_l &= \frac{\pi}{2} - \arctan\left(\frac{\hat{x}''_{FP,z}}{\hat{x}''_{FP,x} + b/2}\right) \\ \mathtt{p}_r &= \frac{\pi}{2} - \arctan\left(\frac{\hat{x}''_{FP,z}}{\hat{x}''_{FP,x} - b/2}\right) \end{aligned} . \tag{11}$$

This approximation of the final vergence serves to provide only a initial coarse hint (feed-forward term) where to foveate. At the end of the motion, the vergence will converge onto the target anyway, thanks to the eyes' configuration provided by Equation 7 combined with the closed loop control described in Section II-D.

III. EXPERIMENTAL RESULTS

The control architecture described in Section II has been implemented and released under the GPL open source license. As such, it is freely accessible on GitHub[2]. It is developed for the iCub humanoid robot, and it is readily available for any iCub robot. Nonetheless, the architecture is generically applicable to any humanoid head, and the code has been designed to be modular and easy to adapt to any other robotic platform (provided that a suitable kinematic representation is available)[3].

Particular attention has been given to the design of the software interface that is exposed to the user. For example, the desired target can be sent to the controller in different formats (e.g. either the 3D cartesian point to look at or a triangulation of the target coordinates in both the left and the right image frames). A configuration interface allows the user to tune the internal parameters of the Gaze Controller for a finer control of the software (e.g. enabling/disabling saccadic movements [Section II-D3], triggering the neck stabilization [Section II-D2], tuning the execution time for the neck and eyes controllers [Equation 9], and so on).

In the following Sections, experimental results on the iCub humanoid robot are reported. Please refer to the accompanying video for an overview of the experiments (full resolution available at `https://youtu.be/I4ZKfAvs1y0`). To validate the proposed architecture, we set up four experiments. The convergence of the system while tracking a moving 3D target is shown in Section III-A, while a comparison between the Gaze Controller with human gaze shifts is analyzed in Section III-B. Section III-C details the behavior of the Gaze Controller during saccadic behavior, as well as a comparison with human gaze shifts. Finally, Section III-D showcases the improvement of the overall performance of the system when gaze stabilization is triggered. For the purposes of this work, we have developed an independent software module (hereinafter referred as Gaze Tester, or GT), with the goal of streamlining the tests described in the following sections. It interfaces with the proposed Gaze Controller (GC) and is capable of creating repeatable position profiles in the 3D cartesian space of the robot upon user request.

A. Tracking of a moving 3D Target

The first case study accounts for of how well the Gaze Controller presented in this work performs tracking of a moving cartesian target on the real robot. To this end, we have created a number of circular trajectories into the 3D cartesian space of the robot, with varying values of speed, radius, and orientation. The 3D position profile generated by the testing module is directly fed to the Gaze Controller. By referring to the two experiments detailed Figure 7, the Gaze Controller is able to consistently track the circular trajectory with an average tracking error and delay of $[1.3\,\text{cm}, 90\,\text{ms}]$ for the first scenario, and $[1.4\,\text{cm}, 40\,\text{ms}]$ for the second circular profile.

B. Qualitative comparison with human gaze velocity profiles

The velocity profiles of the head and eyes joints of the Gaze Controller are similar to the ones observed in human subjects. Figure 8 qualitatively compares the velocity profiles of the robot and those of a typical gaze shift in healthy human subjects. In both cases, the faster dynamics of the eyes let the subject rapidly catch the target; after a while, the neck starts moving toward the fixation target, while the VOR reflex occurring at the eyes maintain the fixation point on the target by compensating the neck movements.

It is worth noting that the controller proposed in this work does not attempt to model a specific brain network, nor to replicate biological data. Rather, the design choices described in Section II (and in particular the possibility to tune most of

[2] See `https://github.com/robotology/icub-main` for the source code; a comprehensive documentation and tutorials are available at `http://wiki.icub.org/iCub/main/dox/html/icub_gaze_interface.html`

[3] In this regard, a porting onto the 6-DoF head of the Vizzy robot has been already carried out (cf. `http://mediawiki.isr.ist.utl.pt/wiki/Vizzy_Cartesian_Interface`).

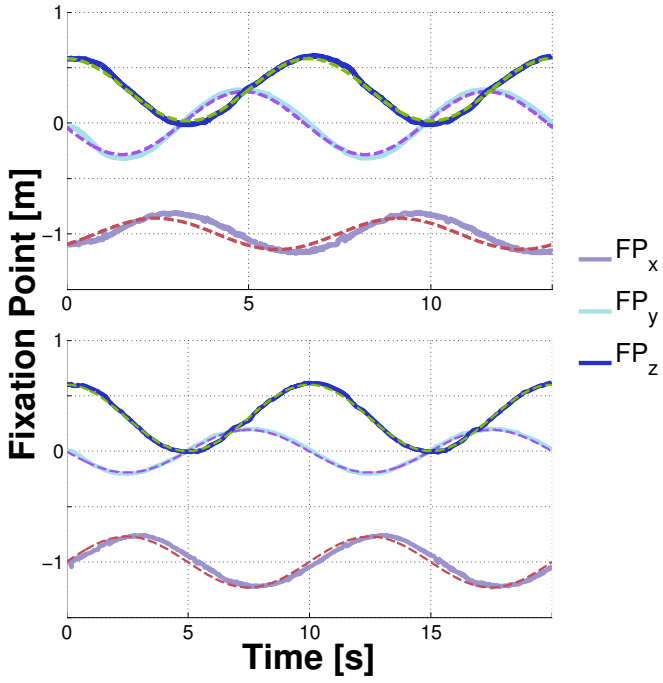

Fig. 7: Gaze Controller (with default settings for T_N and T_E) in presence of a 3D moving target. The graphs show two different circular trajectories in the cartesian space, with varying speed (15 cm/s and 10 cm/s), radius (0.3 m and 0.25 m), and orientation in spherical coordinates (equal to $[20.0°, 70.0°]$ and $[50.0°, 90.0°]$ respectively). The dashed lines are the reference trajectories requested by the GazeTester, respectively the x, y and z coordinates of the target fixation point over the time. The solid lines represent the behavior of the output trajectory computed by the Gaze Controller.

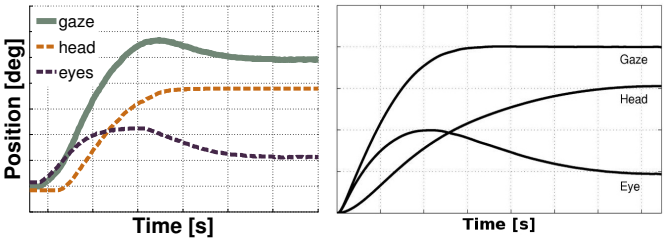

Fig. 8: Qualitative comparison between a gaze shift performed by the Gaze Controller on the iCub robot (left, $T_N = 1.0$ s, $T_E = 0.6$ s), and a typical gaze shift in healthy human subjects (right, from Sohrab et al. [23]). The faster dynamics of the eyes system lets the subject quickly shift its gazing toward a target in the image plane. In both the graphs, angular displacements of the head, the eye system, and the gaze are shown with respect to time.

the system parameters) ease off the process of adapting the GC according to the user's needs. In this context, the similarity between gaze shifts performed by the Gaze Controller and those executed by human subjects effectively helps in the context of human-robot interaction. The head and eye movements are a highly communicative part of the human body, and endowing a robotic system with similar behaviors equips the robot with higher predictability and better integration with the human collaborator. The effectiveness of the proposed control framework in signaling non-verbal cues during human-robot interaction has been analyzed in related work [4]. They demonstrated how naive human subjects are sensitive to gaze in the context of cooperative tasks, effectively improving their performance if a meaningful gazing information from the robot is provided.

C. Saccadic movements for gaze shifts in the periphery

The third experiment is an analysis of how the Gaze Controller performs if significant gaze shifts are requested. As detailed in Section II-D3, one of the main components involved in human gaze shifts is the exploitation of the faster dynamics of the eyes system with respect to the neck complex in order to achieve quick foveation on a target. In order to test this behavior, we set up an experiment similar to Section III-A, with circular trajectories sent to the Gaze Controller. At random points in time, the Gaze Tester asks for a $180°$ shift in the trajectory, effectively rupturing the smooth pursuit capabilities of the GC. Consequently, the Gaze Controller is tasked with efficiently tracking a step profile. Figure 9 shows how the solvers and the controllers successfully cope with this scenario. Even when the requested trajectory (dashed lines in Figure) is abruptly changed, the open-loop, quick response of the saccadic module lets the controller keep up with the input signal with minimum delay (30 ms) and negligible tracking error (3.3 cm on average, which reduces to 1.8 cm after the 100 ms transient of the saccadic response is finished).

D. Gaze controller in presence of gaze stabilization

In the last experiment, the stabilization capabilities of the Gaze Controller are tested. The iCub has to track a circular 3D trajectory (similarly to Experiment III-A) while the experimenter is perturbing the robot by manually moving the torso joints. As such, this experiment provides only a qualitative assessment of the contribution the gaze stabilization brings to the performance of the system. We devise three different scenario: i) a *baseline*, in which the robot is tracking the 3D trajectory without any disturbance; ii) a *no stabilization* stage, in which the user perturbs the system but no stabilization is put in place; iii) a *stabilization* scenario, where the motion inducted by the experimenter are counter-balanced by the stabilization system detailed in Section II-D2. Figure 10 compares the tracking error of the controller during the three experimental scenarios. In order to isolate the contribution of the neck stabilization from the eyes system, the tracking error has been defined as the error exhibited by the neck subsystem in solving the inverse kinematic task of Equation 5, i.e. the

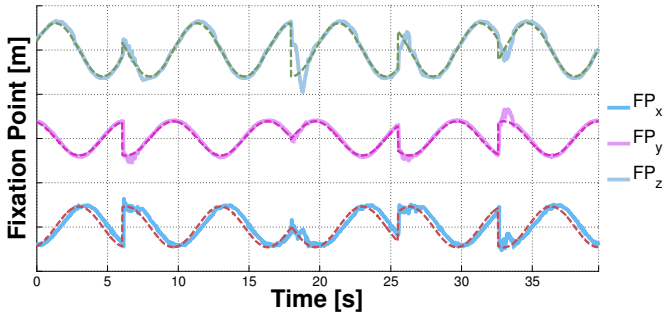

Fig. 9: Gaze Controller with saccadic behavior enabled (and default values for T_N and T_E). When abrupt changes in the desired fixation point are requested, the saccades are able to quickly converge onto the target. Please refer to Figure 7 for a description of the plot. The three coordinates have been equally spaced along the ordinate axis in order to facilitate the interpretation of the plot.

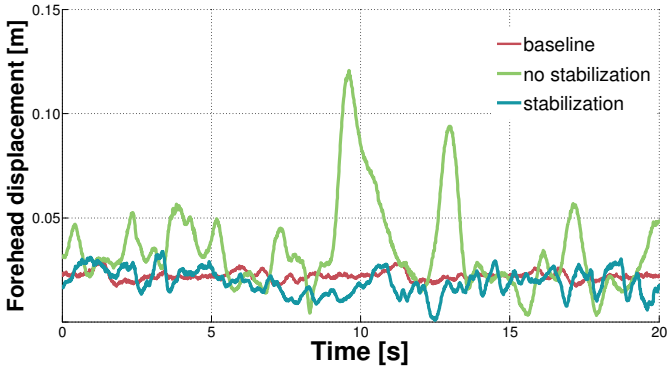

Fig. 10: Neck tracking error of the Gaze Controller with gaze stabilization enabled, expressed as displacement between the target fixation point $\hat{\boldsymbol{x}}_{FP}$ and the axis z_c centered in the midpoint O_c (cf. Figure 2). T_N and T_E are set to their default values for all the three experimental scenarios – *baseline*, *no stabilization*, *stabilization*.

euclidean distance between the target fixation point $\hat{\boldsymbol{x}}_{FP}$ and the axis z_c centered in the midpoint O_c (cf. Figure 2). As it is possible to deduce from Figure 10, there is a non-zero tracking error even during the baseline experiment – equal to 2.2 cm on average. This is either due to the kinematic unattainability of the task described by Equation 5, or to the delay exhibited by the neck in following the trajectory[4]. Most notably, when a perturbation is present without stabilization (*no stabilization* scenario), the tracking error is considerably higher (in this case, the average tracking error is equal to 3.5 cm). Further, it is evident how the stabilization module significantly improves the tracking error, with an average value of 1.9 cm – lower than the baseline. This – albeit small – improvement with respect to the baseline is due to the fact that by closing the loop with the signal provided by the IMU the controller is able to intrinsically compensate for the component of the tracking error due to the low level PIDs implemented in the robot's motor interfaces.

[4] For both cases, in a standard scenario the eyes subsystem is tasked with overcoming the tracking error occurring at the neck (not shown in Figure 10).

IV. CONCLUSIONS AND FUTURE WORK

In this work, we presented a complete gaze control architecture for a humanoid robot. With respect to the state of the art, the proposed solution is able to explicitly control the 3D fixation point of a generic binocular head system. The proposed control framework is able to generate biologically-inspired, minimum-jerk trajectories with quasi-straight convergence on target. Additionally, we focused on the exploitation of the redundancy of the kinematic problem by adding a number of secondary tasks, i.e. gaze stabilization, vestibulo-ocular reflex, and saccadic behavior.

Our experiments demonstrate the capabilities of the system in handling different types of tasks. The Gaze Controller presented in this paper is able to successfully execute smooth pursuit behaviors and tracking of 3D points in space. Importantly, the additional capabilities added to the system proved to be advantageous under various circumstances. The saccadic behavior is able to quickly move the eyes toward the target when large gaze shifts are requested, whereas the vestibulo-ocular reflex uses the gyroscope readouts in order to compensate the neck movements, effectively maintaining the fixation point on target. Finally, the gaze stabilization system utilizes the feedback from the inertial measurement unit in order to reduce image blur both when the robot is subject to external disturbances, and during active gaze control.

V. ACKNOWLEDGMENTS

This work was supported by the European Project KoroiBot (FP7-ICT-611909).

REFERENCES

[1] W. Abend, E. Bizzi, and P. Morasso. Human arm trajectory formation. *Brain*, 105(2):331–348, 1982. doi: dx.doi.org/10.1093/brain/105.2.331.

[2] D.J. Agravante, J. Pages, and F. Chaumette. Visual servoing for the REEM humanoid robot's upper body. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5253–5258, May 2013. doi: 10.1109/ICRA.2013.6631328.

[3] A. Bernardino and J. Santos-Victor. Binocular tracking: integrating perception and control. *Robotics and Automation, IEEE Transactions on*, 15(6):1080–1094, Dec 1999. doi: 10.1109/70.817671.

[4] Jean David Boucher, Ugo Pattacini, Amelie Lelong, Gerard Bailly, Frederic Elisei, Sascha Fagel, Peter F Dominey, and Jocelyne Ventre-Dominey. I Reach Faster When I See You Look: Gaze Effects in Human-Human and Human-Robot Face-to-Face Cooperation. *Frontiers in Neurorobotics*, 6(3), 2012. doi: 10.3389/fnbot.2012.00003.

[5] Roger H.S. Carpenter. *Movements of the eyes*. Medical. Pion, London, UK, 1988.

[6] B. Duran, G. Metta, and G. Sandini. Emergence of Smooth Pursuit using Chaos. In *Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on*, pages 269–272, July 2007. doi: 10.1109/SASO.2007.23.

[7] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[8] H. H. Goossens and A. J. Van Opstal. Human eye-head coordination in two dimensions under different sensorimotor conditions. *Experimental Brain Research*, 114(3):542–560, 1997. doi: 10.1007/PL00005663.

[9] Lizhong Gu and Jianbo Su. Gaze Control on Humanoid Robot Head. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 2, pages 9144–9148, 2006. doi: 10.1109/WCICA.2006.1713769.

[10] D. Guitton and M. Volle. Gaze control in humans: eye-head coordination during orienting movements to targets within and beyond the oculomotor range. *Journal of Neurophysiology*, 58(3):427–459, 1987. ISSN 0022-3077.

[11] Kunihiko Kaneko and Ichiro Tsuda. *Complex systems : chaos and beyond*. Springer, 2000.

[12] Hyundo Kim, Boris Lau, and Jochen Triesch. Adaptive Object Tracking with an Anthropomorphic Robot Head. In *Proceedings of the 2004 International Conference on the Simulation of Adaptive Behaviors (SAB)*, 2004.

[13] Manuel Lopes, Alexandre Bernardino, Jose Santos-Victor, Kerstin Rosander, and Claes von Hofsten. Biomimetic Eye-Neck Coordination. In *Proceedings of the 2009 IEEE 8th International Conference on Development and Learning*, pages 1–8, Washington, DC, USA, 2009. IEEE Computer Society. doi: 10.1109/DEVLRN.2009.5175535.

[14] Eliseo Stefano Maini, Luigi Manfredi, Cecilia Laschi, and Paolo Dario. Bioinspired velocity control of fast gaze shifts on a robotic anthropomorphic head. *Autonomous Robots*, 25(1-2):37–58, 2008. doi: 10.1007/s10514-007-9078-z.

[15] W. P. Medendorp, J. A. M. Van Gisbergen, and C. C. A. M. Gielen. Human gaze stabilization during active head translations. *Journal of neurophysiology*, Jan 2002.

[16] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano. The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23 (8-9):1125–1134, 2010. ISSN 0893-6080.

[17] G. Milighetti, L. Vallone, and A. De Luca. Adaptive predictive gaze control of a redundant humanoid robot head. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3192–3198, Sept 2011. doi: 10.1109/IROS.2011.6094417.

[18] U. Pattacini. *Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub*. PhD thesis, Istituto Italiano di Tecnologia, Genova, Italy, 2011.

[19] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1668–1674, Oct 2010. doi: 10.1109/IROS.2010.5650851.

[20] A. Roncone, U. Pattacini, G. Metta, and L. Natale. Gaze Stabilization for Humanoid Robots: a Comprehensive Framework. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 259–264, Nov 2014. doi: 10.1109/HUMANOIDS.2014.7041369.

[21] K. Rosander and C. von Hofsten. Visual-vestibular interaction in early infancy. *Experimental Brain Research*, 133(3):321–333, Aug 2000.

[22] Patricia Shaw, James Law, and Mark Lee. A comparison of learning strategies for biologically constrained development of gaze control on an iCub robot. *Autonomous Robots*, 2013.

[23] Saeb Sohrab, Cornelius Weber, and Jochen Triesch. Learning the Optimal Control of Coordinated Eye and Head Movements. *PLoS Computational Biology*, 2011.