IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 April 2023, accepted 9 May 2023, date of publication 19 May 2023, date of current version 30 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3277967

RESEARCH ARTICLE

A Novel Algebraic Inverse Kinematics Based Approach to Gaze Control in Humanoid Robots

AHMED NIZAR ALSHAKHS[1], MUHAMMAD FAIZAN MYSOREWALA[1,2], (Member, IEEE), ABDUL-WAHID A. SAIF[1,2], AND KHALED ALSHEHRI[1,2], (Member, IEEE)

[1]Department of Control and Instrumentation Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia
[2]Interdisciplinary Research Center for Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia

Corresponding author: Muhammad Faizan Mysorewala (mysorewala@kfupm.edu.sa)

This work was supported in part by Interdisciplinary Research Center for Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, under Grant INML2202.

ABSTRACT The vision system of humanoid robots is one of the essential components that allow them to do complex activities. This has led to numerous studies on artificial gaze stabilization, many of which are based on biological vision system reflexes. The gaze stabilization solution includes image capture and processing, inverse kinematics, and feedback control to achieve the desired gaze behavior. Generally, the multi-solution problem of finding the desired gaze direction and the inverse kinematics problem of finding the desired joints are solved by using cost functions and numerical solvers. This results in disadvantages that include long processing time, uncertain number of iterations, and risk of numerical instability. In this work, an alternative algebraic method is introduced by exploiting the cascading structure of the neck and eye. By using direct equations instead of numerical solving, the multi-solution and inverse kinematics problems are solved utilizing the geometrical structure and the proposed virtual configurations. Moreover, a sliding mode controller is designed to move the neck-eye joints to their desired positions. The developed scheme is simulated in the MATLAB/SIMULINK environment. Results for various scenarios show that the system can effectively gaze at different stationary and moving targets, showing human-like gaze behavior. In terms of timing, the desired gaze direction is quickly achieved and stabilized, whereas the neck-eye system dynamics stay longer. For one of the simulation cases presented, it took 0.3 seconds to achieve the desired gaze direction for the dynamics of more than 1 second.

INDEX TERMS Degrees of freedom (DOF), face direction, gaze control, gaze stabilization, humanoid, iCub, inverse kinematics (IK), numerical solution, vestibulo-ocular reflex (VOR).

I. INTRODUCTION

Image-dependent applications are increasing with the emergence of artificial intelligence and robots. This visual dependency is very high in humanoid robots as they are required to perform complicated tasks, mimicking normal human behavior. By Digital Image Stabilization (DIS) and Optical Image Stabilization (OIS), the visual input is stabilized to get the most information. Although DIS is responsible for processing the image, the information contained in an image is governed by the mechanism of getting it, which is the responsibility of OIS. In this work, the neck and eyes of a humanoid robot form the responsible vision system. Through coordinated motion between the neck and eyes, gaze stabilization can be achieved. However, the humanoid vision system should resemble the normal Human Vision System (HVS) to some extent.

HVS performs gaze stabilization by exploiting complicated biological techniques called reflexes. Vestibulo-Ocular Reflex (VOR) is the reflex responsible for keeping the gaze stabilized at the current gaze point in space, whereas Optokinetic Reflex (OKR) is the reflex responsible for keeping the gaze on track with the target [1]. Many researches have developed gazing techniques based on HVS to make the robot exhibit a gazing behavior similar to human gaze behavior [2]. It should be noted that gaze control, generally, includes image capturing, image processing, and system control [3]. Gaze

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Alshabi.

stabilization is done in the system control stage by considering the information of the target and elements of the vision system [4]. On the other hand, image stabilization is done in the image processing stage to reduce the blur problem [5]. An eyeball or camera serves as the vision sensor that feeds the system with visual input.

Research in the field goes back to the previous century, such as research on eye motion, gaze stabilization, and humanoid vision systems [1], [6], [7], [8], [9], [10]. One of the common problems addressed in the field is compensating for vibrations affecting the captured image. This problem was addressed in [11] by designing a compensating electromagnetic actuator, called 3 Degree of Freedom Electromagnetic Actuator for Image Stabilization (3D-EAIS), on artificial eyeballs that could isolate them from vibrations of body frame. Modelling and control of 3D-EAIS were done in [12], whereas experimenting on a fabricated model is shown in [13]. Approaching the same problem differently in [14], stabilization of bionic eyes was done mechanically to compensate for vibrations and gravity. The system considered has 3 degrees of freedom (DOF) for the neck and 2 DOF for each eye.

Researches also considered motion and rotation behaviors during performing tasks. In [15], an analysis of gaze stability was conducted on patients with vestibular loss and healthy people by observing the visual acuity during a gaze shift for standing, rotating the head, and walking cases. Furthermore, movements of human head were emulated and analyzed in [16]. Position variables were obtained by using geometric relationships instead of fixed reference methods. In addition, kinematic models for coordinated movements of eye-head were derived in [17] for a general model of two-eye humanoid head with 6 DOF. Another work, in [18], has addressed the problem of keeping a distance by gaze and motion stabilization. Model reference adaptive control was used with adaptive feedforward gains and visual information as feedback.

Several studies have been extended to research animal vision systems. Some experiments were conducted on mice to monitor their eye movements [19]. A technique for stabilizing the gaze of a robobee during a maneuver was developed and inspired by some insects' techniques [20]. However, studies on HVS have resulted in a huge impact on the improvement of the humanoid vision system by taking inspirations from stabilization techniques such as VOR and OKR. For example, VOR has been analyzed by OpenGL visualization for camera movements to evaluate image stabilization [21]. Moreover, the visual stimulus has been analyzed during a car drive to develop a model based on VOR and OKR [22]. Discussion of researches on the topic that address used models, problems and solutions, designs of robots, control methods, and system behavior can be seen in comprehensive reviews, such as [2] and [23].

Many solutions have been proposed and developed for the control problem of humanoid gaze stabilization throughout the literature. For ARMAR-III humanoid robot, a method using saliency maps was developed in [24] to have the gaze fully automated with a combination of gaze stabilization and view selection to achieve required tasks. On the same robot model, another method has been developed to be more robust by including more sources of disturbances, namely, visual, proprioceptive, vestibular, and inertial information [25]. A different approach was taken by considering the problem of tracking the fixation point of the eyes on iCub humanoid robot by developing a passive gaze stabilization with VOR and saccadic motion [26].

In the literature, deep learning methods were also used in the gazing problem to address unmodeled nonlinearities [27]. In addition, it is used to estimate the gaze, as discussed in [28], for various applications. One of the applications is gaze gesture-based control, such as controlling a wheelchair or a robotic car by reading the eye movements [29], [30]. Another application is Human-Robot Collaboration, where people and robots cooperate to accomplish specific goals [31]. Similarly in the medical field, deep learning approaches have been used to develop gaze-controlled surgical robots [32]. Additional to gaze estimation, the learning process has been used to develop a learned gaze control humanoids such as iCub [33]. Humanoid robots also use deep learning to form Human-Robot Interaction (HRI) behaviors by learning gaze expressions [34].

HRI research has also used fuzzy logic to develop humanoid robot behavior in various aspects. For instance, a fuzzy logic controller with seven criteria was developed to control the gazing behavior of a humanoid and make it more natural [35]. In addition, fuzzy logic was utilized in developing interactive emotion expressions [36]. Also, gaze-based control has been approached by fuzzy logic through reading of gaze and siliency maps [37].

Overall, the gaze system is controlled to be utilized in different areas with various approaches that must address two main problems. The first one is the redundant solution problem, which is the result of possible different actions to reach an ultimate goal. The second problem is the inverse kinematics (IK) problem, which translates desired actions into desired joint positions. These are usually solved by cost functions and numerical solvers [38]. Even though many different numerical methods are found in the literature such as Newton-Raphson and Levenberg-Marquardt methods, finding a numerical solution also includes problems such as numerical instability risks, uncertain number of iterations depending on the initial guess, and the need of performing a time-consuming process [39]. Therefore, different approaches are being developed to avoid related problems. In [40], the fruit fly optimization algorithm was used to optimize the parameters of the back propagation neural network in order to solve the inverse kinematics problem. Another study introduces an adaptive part to the particle swarm optimization to achieve more efficient and accurate solutions of IK [38]. On the other hand, in [41], it was proposed to use direct calculations rather than solvers by developing a 3D triangulation approach.

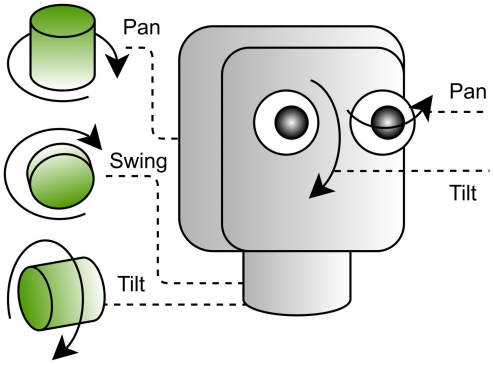


FIGURE 1. Joints' rotations of iCub humanoid robot.

In this work, the gaze control problem of a humanoid robot is addressed by developing an algebraic method to avoid the problems related to the numerical solution. It is based on the system cascading structure, that can be used to find the joints' configuration of the desired state. In addition, the multi-gaze solution problem is tackled by introducing virtual components considering the geometrical structure. The developed method is simulated in MATLAB/SIMULINK environment using a developed sliding mode controller for smooth response. A comparison of the different features of existing and proposed approaches is presented in Table 1. The most suitable research papers were considered as only a little research has a comprehensive presentation discussing the system, inverse kinematics, controller, and dynamics.

II. METHODS

The system considered in this article is the neck and eyes of a humanoid robot, as it represents the vision system. The neck-eye system is modeled based on iCub model which has its information available in multiple sources such as [42], [43], and [44]. It consists of 3 joints for the neck and 2 joints for the eye, with 5 DOF in total. Only one eye was considered as the work is extendable to the second eye.

A. THE MODEL OF THE SYSTEM

A robotic model is dependent on the joints' location, connection, and order. The neck and eye joints of iCub model results in tilt, swing and pan rotations as shown in Fig. 1. The order of neck rotations is $R_y R_x R_z$ which are rotations around the y-axis, x-axis, and z-axis in order. Similarly, the eye joints have a rotation order of $R_y R_z$. The joints are restricted within a range of allowed rotation for each one, as detailed in Table 2.

The gaze of the vision system can be defined by a 4×4 homogeneous matrix representing the eyeball (camera) position and its gaze direction. This is done by utilizing kinematic equations using transformation matrices throughout the joints. So, the gaze can be written as

$$ {}^{B}T_{E} = {}^{B}T_{N_{J_1}}{}^{N_{J_1}}T_{N_{J_2}}{}^{N_{J_2}}T_{N_{J_3}}{}^{N_{J_3}}T_{N}{}^{N}T_{E_{J_1}}{}^{E_{J_1}}T_{E_{J_2}}{}^{E_{J_2}}T_{E} \quad (1) $$

where ${}^{X}T_{Y}$ is the transformation from X to Y, and B, N, E, and J_n are the base of the system, the neck, the eye, and the joint number in the order of rotation sequence, respectively. A transformation matrix is a homogeneous matrix that represents the change in position and orientation from one part of the system to another. These transformation matrices are found by using the Denavit-Hartenberg (DH) representation. Each transformation in Eq. (1) can be represented by the following function of DH parameters:

$$ T_{J_i}(\theta_i, \alpha_i, a_i, d_i) = \begin{bmatrix} C(\theta_i) & -S(\theta_i)C(\alpha_i) & S(\theta_i)S(\alpha_i) & a_i C(\theta_i) \\ S(\theta_i) & C(\theta_i)C(\alpha_i) & -C(\theta_i)S(\alpha_i) & a_i S(\theta_i) \\ 0 & S(\alpha_i) & C(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

where θ_i, α_i, a_i, and d_i are the DH parameters for the i^{th} joint J_i, whereas S and C are *sin* and *cos* functions. It is important to note that the only variable of the DH parameters is the joint rotation (θ) because the system has no prismatic joints. Therefore, knowing the joints configuration is enough for knowing the system state at any part. So, a general equation for obtaining the homogeneous matrix of a part of the system can be written as

$$ H_n = {}^{o}H_k{}^{k}H_n = {}^{o}H_k \prod_{i=k+1}^{i=n} T_{J_i}(\theta_i, \alpha_{oi}, a_{oi}, d_{oi}) \quad (3) $$

where T_{J_i} is the transformation matrix in Eq. (2), and ${}^{o}H_k$ is the homogeneous matrix of the reference that can be the base frame H_o or the frame of the k^{th} body part. The transformation is done from the k^{th} joint to the n^{th} joint.

The kinematic model found by Eqs. (1, 2, 3) is only concerned with position and direction, whereas the rate of change of the system state is the concern of the dynamic model, which can be represented at the joint level by the following differential equation [45]:

$$ \tau = M(J)\ddot{J} + C(J, \dot{J})\dot{J} + G(J) \quad (4) $$

where τ is the input torque, J is the system joints, $M(J)$ is the matrix of inertia, $C(J, \dot{J})$ is the Coriolis effect, and $G(J)$ is the gravity. The differential equation can be reorganized to have an expression of $\ddot{J}$ as shown below:

$$ \ddot{J} = W(J)\tau - W(J)C(J, \dot{J})\dot{J} - W(J)G(J) \quad (5) $$

where $W(J)$ is the inverse of $M(J)$.

Kinematic and dynamic equations form the system model. It is implemented as a rigid-body-tree model in MATLAB/SIMULINK environment using Robotics System Toolbox.

B. INVERSE KINEMATICS METHOD

The neck-eye system discussed in Subsection II-A is representing the vision system of a humanoid robot. Developing an IK for the whole system will only ensure the ultimate gaze direction. However, for a humanoid robot, the gazing behavior is also important to have a good HRI. Hence, in order to have more control over the gazing process, IK is divided into two IKs: Eye_IK for the gaze of the eye, and Neck_IK for the

TABLE 1. Comparison of existing and proposed schemes.

Reference	System	DOF	Numerical Solution Utilized	Feedback to the IK	Controller Implemented	Coordination between neck and eyes	VOR behavior	Tracking moving target	Disturbance
[16]	Neck System	3	Yes	2 out of 3 states	Controller dynamics not discussed	Not Applicable	Not Applicable	Not Discussed	Not Considered
[24]	Neck-Eye System	7	Yes	All states	Controller dynamics not discussed	No	No	No	Considered
[25]	Neck-Eye System	7	No	All states	Implemented, no specified name	No	Yes	Yes	Considered
[26]	Neck-Eye System	6	Yes	All states	PID	Yes	Yes	Yes	Considered
[41]	Neck-Eye System	7	No	All states	P Controller	No	Not Discussed	Not Discussed	Considered
Proposed	Neck-Eye System	5	No	3 out of 5 states (Only Neck Joints)	Sliding Mode Control	Yes	Yes	Yes	Not Considered

TABLE 2. Allowed range of joints rotation.

Joints	Rotation Type	Lower Limit	Upper Limit
Neck Joint 1	Tilt	-50^o	40^o
Neck Joint 2	Swing	-40^o	40^o
Neck Joint 3	Pan	-55^o	55^o
Eye Joint 1	Tilt	-40^o	40^o
Eye Joint 2	Pan	-50^o	50^o

face direction determined by the neck. An overall view of the controlled system is presented in Fig. 2.

1) EYE_IK

The main goal of IK is to have a stable gaze at the target. It begins by finding the desired position and orientation of the eye, then provides the corresponding desired joints to the controller. However, Eye_IK only considers the eye submodel consisting of 2 joints. The two rotational joints only affect the gaze direction, whereas the eye position is predefined by the neck, considering it as a pre-reference frame. An illustrative sketch of the model is introduced in Fig. 3 to be used for illustrating examples throughout the method explanation of IK.

Because Eye_IK considers the position of the eye constant, the angle between vectors method can be used to find the needed rotation between the actual and desired gaze direction. In 2D space, the first vector is the actual gaze, whereas the second vector v is found by the following equation:

$$v = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} T_x - E_x \\ T_y - E_y \end{bmatrix} \tag{6}$$

where (Tx, Ty) are the coordinates of the target, (Ex, Ey) are the coordinates of the Eye, and x and y are components of the vector v (z component is also included in 3D space). Then, it can be represented by length and direction using the polar coordinates as shown below:

$$P(r, \theta) = r\left[\cos(\theta)\ \sin(\theta)\right] \tag{7}$$

where r is

$$r(x, y) = \sqrt{x^2 + y^2} \tag{8}$$

and θ is defined by

$$\theta(x, y) = \tan^{-1}\left(\frac{y}{x}\right). \tag{9}$$

So, the desired gaze can be expressed in terms of the current gaze rotated by an angle as shown in the following:

$$\theta_{DG} = \theta_{CG} + \theta_R \tag{10}$$

where θ_{DG} is the desired gaze, θ_{CG} is the current gaze, and θ_R is the desired rotation found by

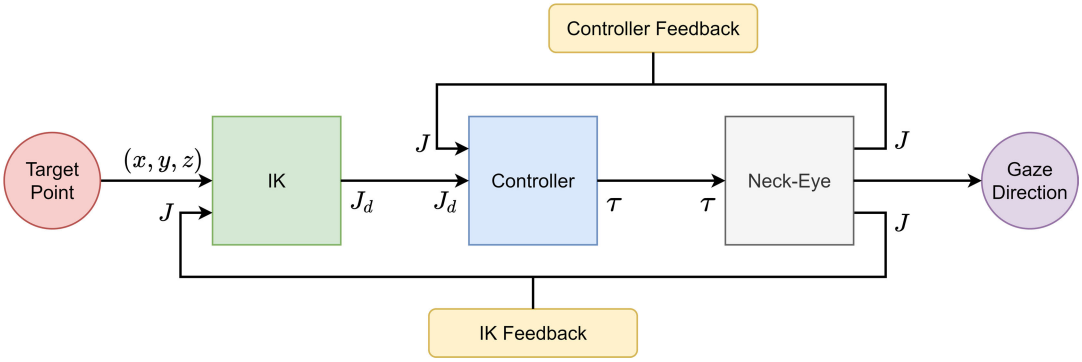
$$\theta_R = \theta(x_2, y_2) - \theta(x_1, y_1). \tag{11}$$

An illustrative example in 2D space is shown in Fig. 4, where θ_R is found using Eqs. (9) and (11).

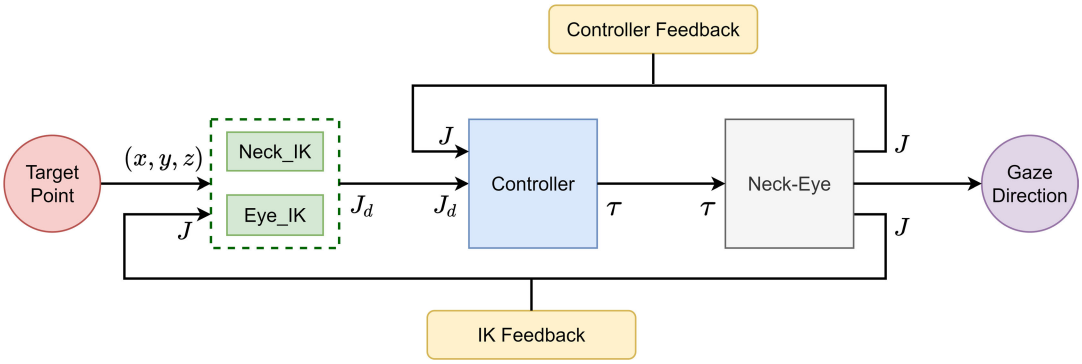
By knowing that the main goal is to find the desired gaze, the current gaze can be substituted by introducing a virtual gaze, which is an assumed gaze based on assumed joints configuration. So Eq. (10) can be modified to be

$$\theta_{DG} = \theta_o + \hat{\theta}_R \tag{12}$$

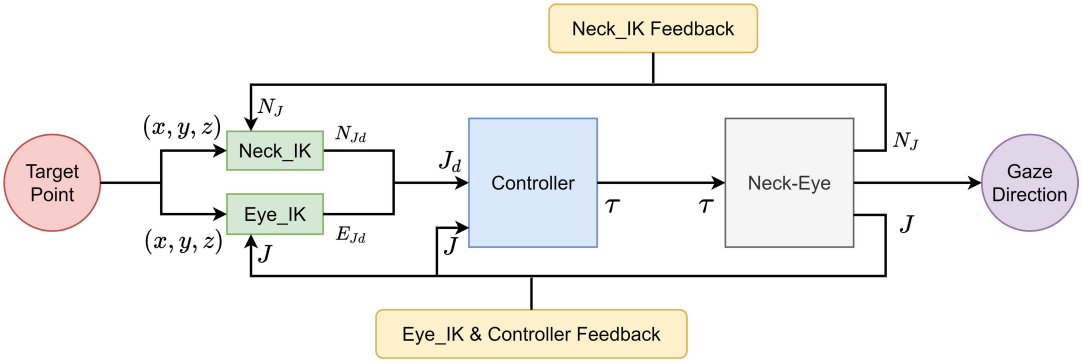
where θ_o represents the virtual gaze and $\hat{\theta}_R$ represents the rotation angle. Although the rotation from the virtual gaze will be different, the desired gaze will be the same. This means that there will be no change in the controller behavior as it only takes the joints to their desired configuration found by IK. Benefiting from the cascading structure of the system by introducing a virtual gaze makes the Eye_IK independent of the eye motion but still dependent on the reference frame,



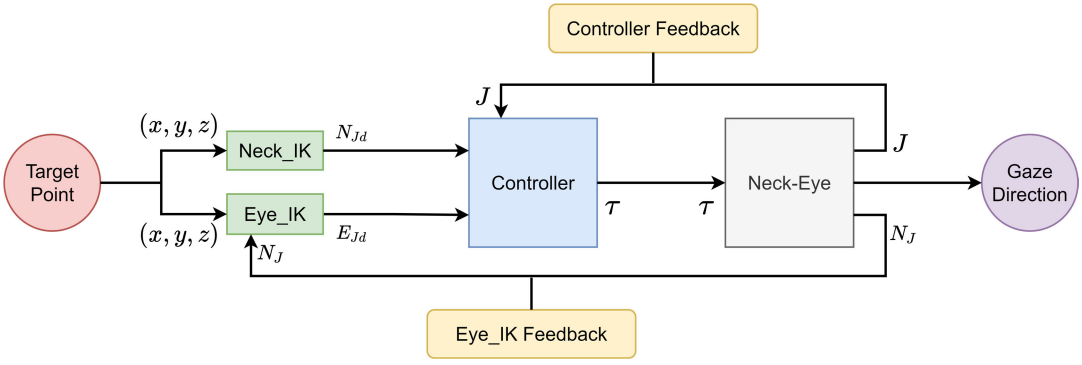
(a) A diagram of the controlled gaze system



(b) Dividing IK into two parts



(c) Introducing the new goal of face direction to Neck_IK



(d) Introducing the virtual configuration method

FIGURE 2. **Original System in (a) is modified by the division of IK in (b) then introduced face direction goal in (c) and lastly introduced the developed method in (d).**

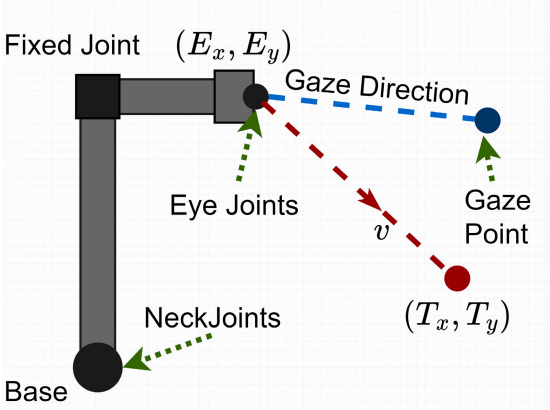


FIGURE 3. Neck-Eye model diagram.

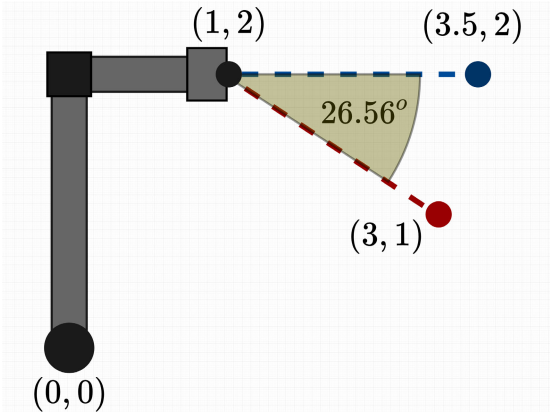


FIGURE 4. Angle between vectors.

which is only the neck in our model. Even though the procedure discussed seems basic, it is used for tackling the problem in 3D space.

In 3D space, the two angles of the spherical coordinates, θ and ϕ, can be used to represent the two rotations of the two joints of the eye submodel. The first angle of rotation is found by following the same procedure above, especially Eqs. (9) and (11). By knowing that this rotation is around the z-axis, the first rotation matrix is found as

$$R_z(\theta_R) = \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) & 0 & 0 \\ \sin(\theta_R) & \cos(\theta_R) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where θ_R is the angle of rotation. The second rotation angle for a spherical coordinates is the elevation towards the z-axis. It can be found by using r from Eq. (8) to find $\theta(r, z)$ using Eq. (9). Then, the angle of rotation is found by using Eq. (11). Because this rotation is around the relative y-axis, the second rotation matrix is found to be

$$R_y(\theta_R) = \begin{bmatrix} \cos(\theta_R) & 0 & \sin(\theta_R) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_R) & 0 & \cos(\theta_R) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

So, by knowing the defined virtual gaze, the two rotation matrices, and the sequence of rotation, the desired gaze direction is expressed as

$$R_d = R_o R_z R_y \tag{15}$$

where R_o is the virtual orientation of the eye that is defined as

$$R_o = R_N R_{VE} \tag{16}$$

where R_N is the orientation of the reference frame affected by the neck rotations, and R_{VE} is the defined virtual gaze. In the case of eye position, it is independent of eye motion. So, it can be directly found by knowing the reference frame and the relative eye location as in the following equation:

$$T_d = R_N T_E \tag{17}$$

where T_d represents the eye position and the homogeneous transformation matrix T_E represents the relative eye position. The desired homogeneous matrix of eye position and gaze direction can be written as

$$H_d = T_d R_d = \begin{bmatrix} \hat{R}_d & \hat{T}_d \\ 0 & 1 \end{bmatrix}. \tag{18}$$

By finding the desired homogeneous matrix, the desired joints can be found by using a numeric solver. However, it is important to note that an IK solver can be bypassed through knowing the needed rotations for each joint. In the above method, two rotation matrices were obtained R_z and R_y with a rotation sequence of $R_z R_y$. Although the eye has only two joints, their sequence of rotation is different, which is $R_y R_z$. Consequently, R_z and R_y found by the above procedure can not be directly used for getting the joints rotations of the eye because the rotation sequence is not commutative.

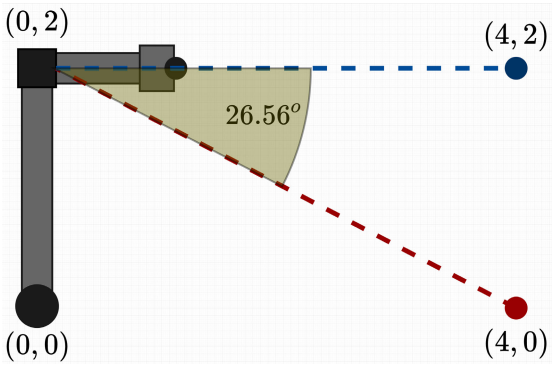
By doing the same procedure but with some modifications, the IK solver can be bypassed. The two angles of the spherical coordinates are modified to be of sequence $R_y R_z$ instead of $R_z R_y$. With this new sequence, the first angle is calculated by Eq. (9) but for $\theta(z, x)$ instead. Similarly, Eq. (9) is used to find the second angle by calculating it for $\theta(r, y)$ where $r(z, x) = \sqrt{x^2 + z^2}$. Each rotation angle can be represented by its corresponding rotation matrix to find the desired rotation matrix using the new rotation sequence as shown in the below equation:

$$R_d = R_o R_y R_z. \tag{19}$$

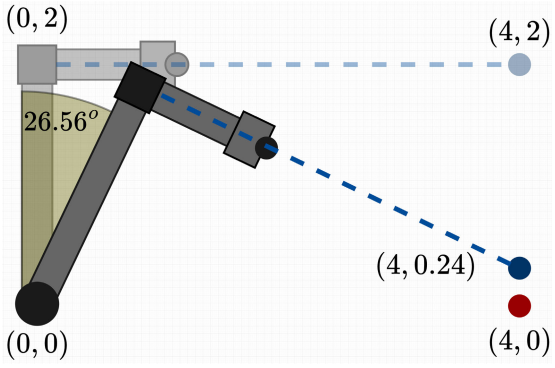
Then, it is combined with the position found by Eq. (17) to form the desired homogeneous matrix presented in Eq. (18). However, the two rotation angles can be directly applied to the two joints as they have the same sequence of rotation. With this, the numeric solver is bypassed.

2) NECK_IK

The second part of IK is Neck_IK, which considers the 3 joints of the neck. As discussed at the beginning of the section, Neck_IK is concerned about the face direction. In contrast to the Eye_IK problem, the Neck_IK problem



(a) Determining the angle of rotation



(b) Applying the rotation angle

FIGURE 5. Demonstrating the indirect relationship between the face direction and neck joints.

has two main differences. The first difference is the different number of DOF. First, Neck_IK considers 3 DOF instead of 2. Second, the configuration of neck joints affects the face direction and head position. Consequently, Neck_IK has a redundant solution problem due to having 3 DOF. Also, there is an indirect relationship between the neck joints and the face direction.

The angle between two vectors method can not be used directly in the case of Neck_IK due to the indirect relationship problem. It will result in an inaccurate solution, such as that shown in Fig. 5, because the reference of rotation angle is dependent on the configuration of the neck joints. In order to have a correct rotation, another method should be used. One way to tackle the problem is by using the geometrical structure of the system. The link at the fixed joint connecting the base with the face is always making a 90 degree angle. In addition, the length from the base to the fixed joint is known from the system structure. Also, the distance between the base and the target is always known as both of them are known. Therefore, the desired distance from the fixed joint to the target can be found by using trigonometry as in the following equation:

$$r_d = \sqrt{r^2 - r_f^2} \tag{20}$$

where r is the distance from the base to the target, and r_f is the length of the link. So, when the vision system faces the target point, the distance between the fixed joint and the target is r_d. Therefore, to find the desired rotation angle, a virtual point is introduced with the same distance of r_d along the current face direction as follows:

$$V_p = R_{vf} \begin{bmatrix} r_d \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

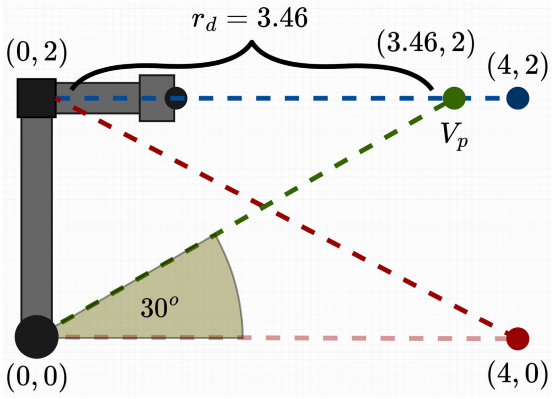
where R_{vf} represents the face direction, and V_p represents the virtual point. Then, two vectors are formulated as one from the base to the virtual point and another one from the base to the target point. Thus, the angle between two vectors can be used to find the desired rotation angle.

Considering the same illustrative example shown in Fig. 5, a correct rotation angle is obtained by the virtual point method in Fig. 6. However, in 3D space, there are two angles of rotation. In addition, there will be a multi-solution problem as the same r_d exists for more than one configuration of neck joints, resembling a circular solution set. It is important to note that for every unique target point, the circular solution set forms a unique cone-shaped directions towards that target, representing the possible solutions of the desired face direction. In addition, rotating the face direction using the vectors of old and new target points results in a new face direction but the same part of the solution set, meaning the system state will be the same. So, for a face direction with no side tilts, the solution will always be the part of the cone that is directed to the target point and has no side tilts. Taking advantage of that, a virtual face direction is introduced so that it avoids the redundant solutions problem. Also, it will make the Neck_IK independent of the neck motion. Moreover, using a virtual face direction with no side tilts gives better HRI as it has more resemblance to human behavior. The virtual direction is implemented by using a virtual R_{vf} in Eq. (21) to find the virtual point V_p with respect to the virtual face. Consequently, the angle between the virtual point and the target can be found, representing the desired rotation. Similar to Eye_IK, spherical coordinate angles are used to represent the rotation angles. Each point of virtual and target points is projected into two virtual points as shown below:

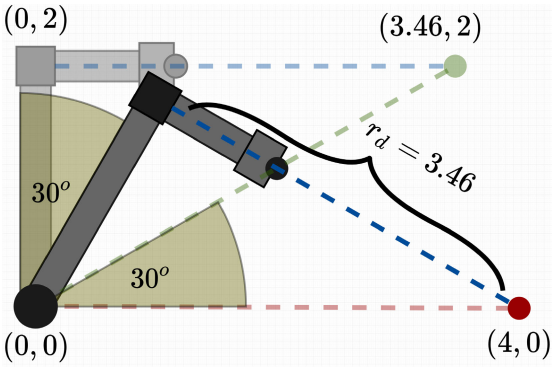
$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \implies P_{xy} = \begin{bmatrix} x \\ y \end{bmatrix}; \quad P_{rz} = \begin{bmatrix} r \\ z \end{bmatrix}, \quad r = \sqrt{x^2 + y^2} \tag{22}$$

where P is a point in the 3D space.

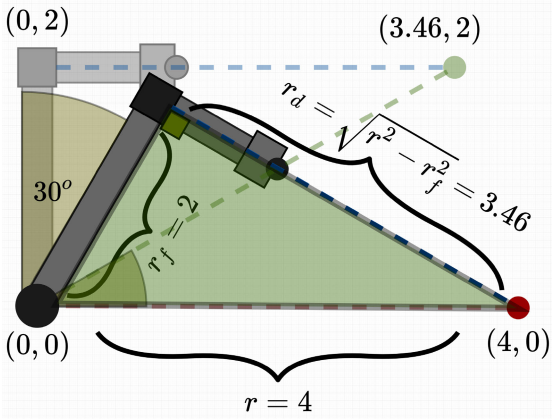
The first angle of rotation is obtained through using projected target point T_{xy} and virtual V_{xy} in Eqs. (9) and (11). Then, Eq. (13) is used to find the rotation matrix R_z. Likewise, the other angle is found by using T_{rz} and V_{rz} in in Eqs. (9) and (11). After that, the rotation matrix R_y is calculated by Eq. (14). Therefore, the desired homogeneous matrix can be found by applying the two rotation matrices to the virtual neck configuration that corresponds to the virtual face direction. First, the homogeneous matrix of the virtual face is written



(a) Defining the virtual point by r_d



(b) Applying the rotation angle



(c) Illustrating r_d in the right triangle

FIGURE 6. **Example showing the virtual point method.**

as:

$$H_{vf} = H_o H_f \tag{23}$$

where H_o represents the reference frame defined by the virtual configuration of the neck, and H_f represents the transformation from the neck to the face. By applying the obtained rotation of R_z and R_y to Eq. (23) in order, the homogeneous matrix of desired face direction and position is

$$H_{df} = H_o R_z R_y H_f . \tag{24}$$

All of this is summarized in Algorithm 1 below.

Algorithm 1 Virtual Point Method for Desired Face Direction

Require: T_p ▷ Target point
$H_{vf} = H_o H_f$ ▷ Virtual face direction
$V_p = H_{vf} r_d$ ▷ Virtual target point of Eq. (21)
$[V_{xy}\ V_{rz}] = P_2(V_p)$ ▷ Two projection points
$[T_{xy}\ T_{rz}] = P_2(T_p)$
$\theta_z = \alpha(T_{xy}) - \alpha(V_{xy})$ ▷ Rotation Angle
$\theta_y = \alpha(T_{rz}) - \alpha(V_{rz})$
$H_z = R(\theta_z, z)$ ▷ Rotation Matrix
$H_y = R(\theta_y, y)$
$H_{df} = H_o H_z H_y H_f$ ▷ Desired
function $[u_{xy}\ u_{rz}] = P_2(u)$
$u_{xy} = \begin{bmatrix} u(x) \\ u(y) \end{bmatrix}$
$r = \sqrt{u(x)^2 + u(y)^2}$
$u_{rz} = \begin{bmatrix} u(r) \\ u(z) \end{bmatrix}$
end function
function $\theta = \alpha(u)$
$\theta = \tan^{-1}\left(\frac{u(y)}{u(x)}\right)$
end function
function $H = R(\theta, c)$
$H = R_c(\theta)$ ▷ Eqs. (13) and (14)
end function

The obtained H_{df} from Algorithm 1 and Eq. (24) can be used by a numeric solver to find the desired neck joints. However, to bypass the numeric solver, certain things must be noted. The desired homogeneous matrix H_{df} has been obtained by finding two rotations, whereas the neck is capable of three rotations. Consequently, the two obtained rotations can not be applied to the neck unless they have the same rotation sequence with fixing the third joint. This is also not applicable as the rotation sequence of the neck is $R_y R_x R_z$, whereas the sequence of the rotations found is $R_z R_y$.

Even so, the Neck_IK numeric solver can be bypassed by following the same steps of Algorithm 1 but with some additional ones. The algorithm ends with obtaining the desired face direction and position represented by the homogeneous matrix H_{df}. This can be decomposed into two subproblems. One is finding the joints configuration that corresponds to the desired face position. Another is finding the joints configuration that corresponds to the desired face direction. Because of the structure of the system, only the first two neck joints affect the position of the head. Therefore, only two rotations are needed for positioning the head. So, by using the desired position from H_{df} and the virtual position from the virtual neck configuration, the two rotations can be found using the same Eye_IK solver bypassing method for the rotation sequence of $R_y R_x$. These two rotations are unique.

As for the third rotation, it is a rotation around the relative z-axis of the plane reached by the first two rotations. By knowing that the third joint does not affect the head position but rotates the face direction, there will be one solution

that faces the target, as it has already been found in H_{df}. To find the rotation angle, a vector from the desired head position (P_{head}) towards the target is defined as shown below:

$$v_o = T_p - P_{head}. \tag{25}$$

After that, the direction vector should be projected onto the relative reference plane of rotation as follows:

$$v = R_x(\theta(r, y))^T R_y(\theta(z, x))^T v_o. \tag{26}$$

Then, by knowing that the rotation is around the z-axis, the rotation angle is directly calculated by the following equation:

$$\theta(v_x, v_y) = \tan^{-1}\left(\frac{v_y}{v_x}\right). \tag{27}$$

Consequently, the three rotations found can be directly applied to the joints as they are of the same rotation sequence $R_y R_x R_z$.

A summary of the whole developed scheme is provided in Fig. 7, where a diagram is presented with the eye part on the left and the neck part on the right. The flow from top to bottom shows the IK solution for a desired gaze that includes the generation of virtual points, needed projections, and rotations. As a result, five joint angles are calculated for the neck-eye system, which are provided to the controller. The calculation process for this IK solution has a bounded computational complexity depending on the number of algorithm steps and the processing time of calculation, where the maximum algorithm steps are 25 (Eqs. ((9), (25), (26), (27)) and Algorithm 1).

C. CONTROLLER

The purpose of the controller is to take the system joints to their desired positions that are decided by IK. The controller should control the system dynamics represented by a differential equation shown in Eq. (5). A Sliding Mode Control (SMC) approach is utilized to build the system controller [46]. First, a sliding surface is defined by

$$S = \dot{\tilde{J}} + \lambda \tilde{J} \tag{28}$$

where λ is a chosen parameter and $\tilde{J} = J_d - J$. Then, the derivative of the sliding surface is found and equated to zero as shown in the following equation:

$$\begin{aligned} \dot{S} &= \ddot{\tilde{J}} + \lambda \dot{\tilde{J}} = 0 \\ \rightarrow \ddot{\tilde{J}} &= -\lambda \dot{\tilde{J}}. \end{aligned} \tag{29}$$

Therefore, along the surface, the joints go to their desired values. To converge and stay on the sliding surface, a Lyapunov equation is defined by

$$V = \frac{1}{2} S^2. \tag{30}$$

Then, the stability condition is found by restricting the Lyapunov derivative to be less than zero as written below:

$$\begin{aligned} \dot{V} &= S\dot{S} \leq 0 \\ &= S(\dddot{\tilde{J}} + \lambda \ddot{\tilde{J}}) \leq 0 \\ &= S(\lambda \ddot{\tilde{J}} - \dddot{J}) \leq 0. \end{aligned} \tag{31}$$

So, the input τ that meets the stability condition is found by substituting Eq. (5) in Eq. (31) to get

$$\tau = M(J)(\lambda \dot{\tilde{J}} + kSign(S)) + C(J, \dot{J})\dot{J} + G(J) \tag{32}$$

where the term $kSign(S)$ represents the reaching condition using a signum function. The signum function outputs 1 or -1 based on the sign of the sliding surface. The only case where it outputs zero is at the sliding surface. This causes an oscillation around the surface. To avoid this problem, the signum function is modified to be

$$Sign(S) = \begin{cases} +1; & S > \epsilon, \\ S; & \epsilon \geq S \geq -\epsilon, \\ -1; & S < \epsilon \end{cases} \tag{33}$$

where ϵ is a chosen parameter that decides the gray area between the positive and negative outputs of a value of 1 in the signum function. Consequently, the stability condition is met by substituting Eq. (5) in Eq. (31) with the τ found in Eq. (32). The resulting equation will be as following:

$$\begin{aligned} \dot{V} &= S(-kSign(S)) \leq 0 \\ &= -k|S| \leq 0. \end{aligned} \tag{34}$$

Finally, it is desired to have the eye controller approximately twice as fast as the neck controller, imitating the biological model. This can be determined by mainly tuning λ and k. A higher λ gives a higher convergence rate along the sliding surface, and a higher k provides more aggressive control taking the states towards the sliding surface. As a result, in general, the eye controller should have higher parameter values.

III. RESULTS

Before discussing the simulation results, some important notes should be considered. It is assumed that the camera focus is of the same distance from the camera to the target. In addition, the zero configuration of gaze and face directions has a direction towards the negative y-axis. Furthermore, millimeters and radians are used for distance and angle, respectively. Moreover, the joints are restricted by the data of Table 2. Lastly, Multiple tests have been performed with different values of lambda and k in order to get the desired response. It also includes running the eye controller twice as fast as the neck controller, which is achieved by doubling the k and lambda values for the eye controller. The parameters of each controller are tuned to be $\lambda = 3.5$ and $k = 15$ for the neck controller, and $\lambda = 7$ and $k = 30$ for the eye controller.

So, considering all of the above, the first test case is shown in Fig. 8, which has a target of [1200 -100 1700] for the first

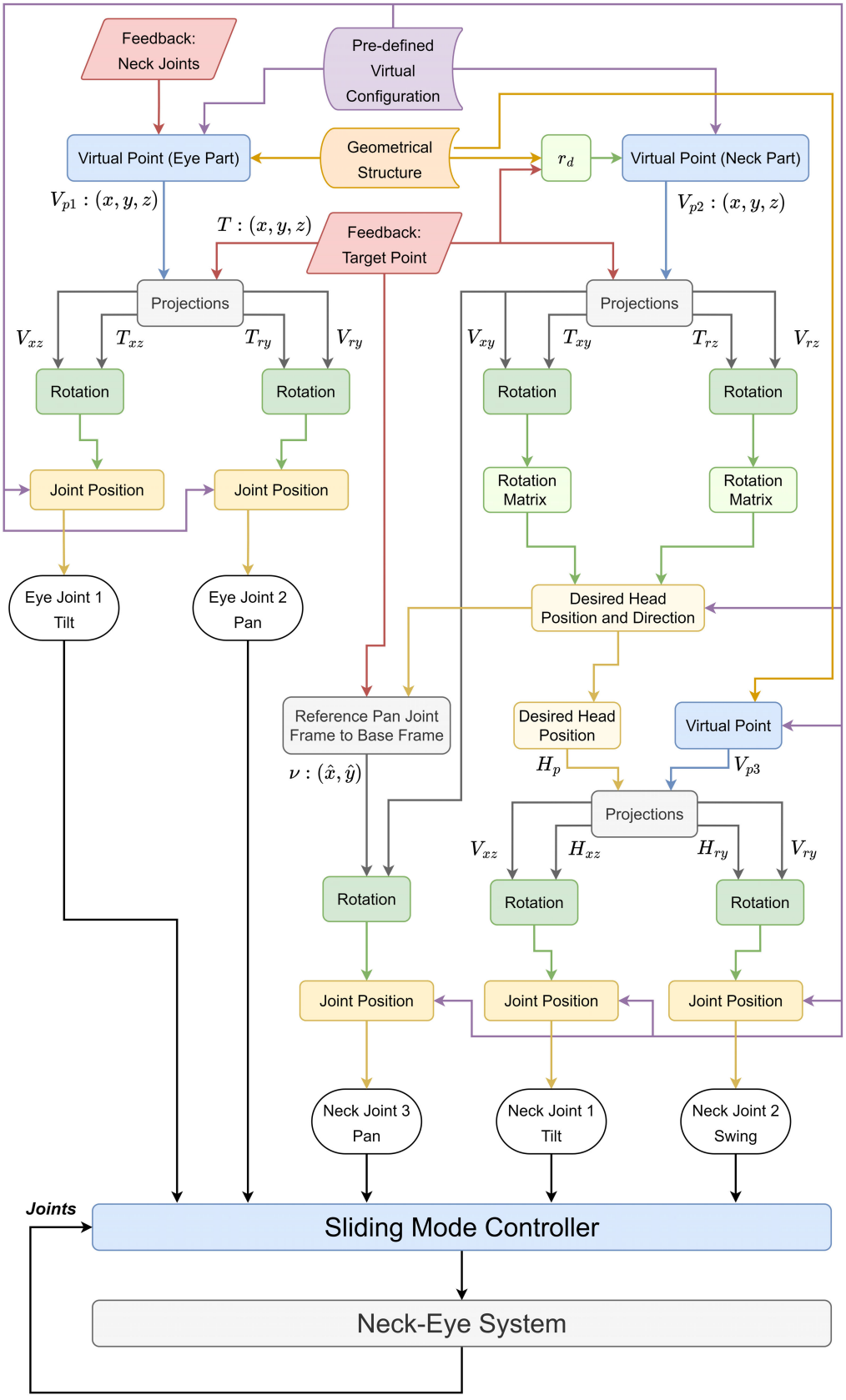


FIGURE 7. Comprehensive diagram of the method.

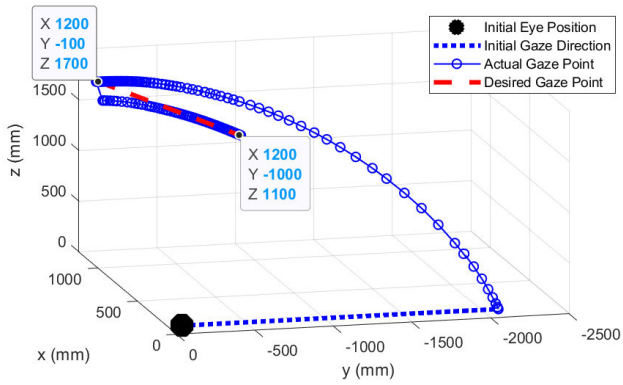


FIGURE 8. 3D View of the system response to gaze change sequence.

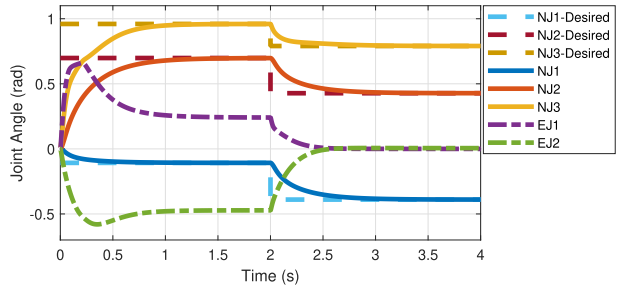


FIGURE 9. The system joints response to gaze change sequence.

2 seconds and then [1200 -1000 1100]. The gaze direction can be followed by observing the gaze point change to the first and second targets, which does not show any abnormality. This can be further discussed by looking at the joint response shown in Fig. 9. The first thing to notice is the fast reaction of eye joints EJ1 and EJ2 compared to the neck joints NJ1, NJ2, and NJ3, which is because of the faster controller that controls the eye joints. In addition, eye joints angles begin by increasing, then quickly decreasing to be steady at some point. The increasing part represents the quick eye motion that tries to reach the desired gaze direction as quickly as possible. However, at approximately 0.3 seconds, the desired direction is reached. So, after that point, the eye joints angles begin to decrease to compensate for the neck joints motion to stabilize the gaze at the target point. The eye joints reach a steady state when the neck joints reach their steady state, meaning the face is directed towards the target in normal cases. Therefore, the eye joints angles should normally reach zero at steady state but, in the shown case, eye joints angles reach a non-zero steady state for the first desired target. This is because the neck joints, specifically NJ2 and NJ3, have reached their limit angles resulting in the inability to direct the face towards the target. So, the non-zero steady state of eye joints represents continuous compensation for that inability. On the other hand, the second target case shows a zero steady state of eye joints at 2.5 seconds, meaning that the face direction has been reached. This shows that the reaction of the eye joint to neck motion resembles the VOR reflex of HVS.

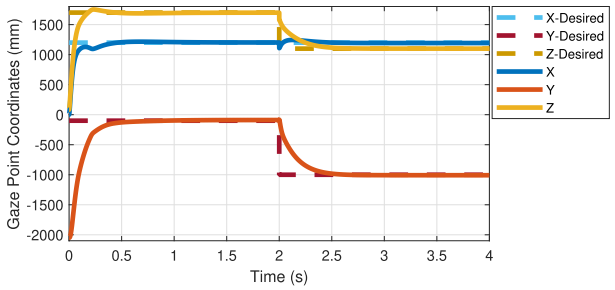


FIGURE 10. Cartesian coordinates of target and gaze point.

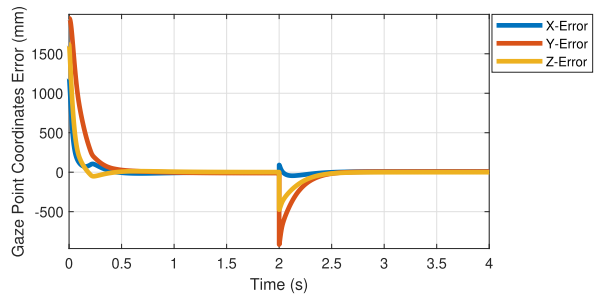


FIGURE 11. Error between target and gaze point.

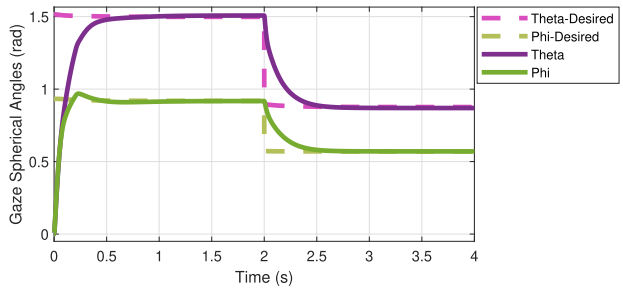


FIGURE 12. Spherical angles of target and gaze direction.

Another view of the same problem can be seen by considering the gaze point coordinates. In Fig. 10, it can be observed that the gaze point smoothly reaches the target point at around 0.3 seconds. This shows that the eye joints have succeeded in quickly reaching the target point and stabilizing it during the neck motion. The quick eye motion, resembling saccadic motion, shifts the eye towards the target point, while the eye joints compensation, resembling VOR, keeps the gaze stabilized at it. The error presented in Fig. 11 confirms that the desired gaze point has been reached as it is of zero error.

Similarly, the gaze direction can be viewed by using the spherical coordinates as shown in Fig. 12. Both angles reach the desired, meaning the direction of the gaze is also reached. There is no observation of an error in the steady state as presented in Fig. 13.

Other cases should be tested to ensure the validity of the work. One case is object tracking as shown in Fig. 14. The target moves with constant motion of around [300 −100 100] mm/s during the first interval of 2 seconds. In the

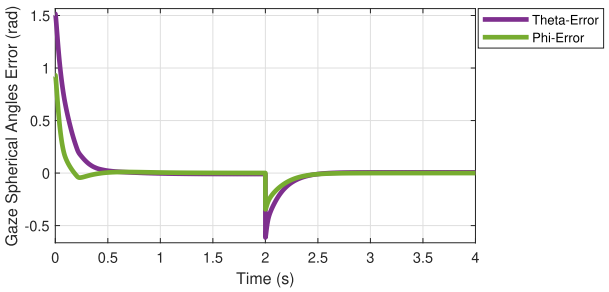


FIGURE 13. Error between target and gaze direction.

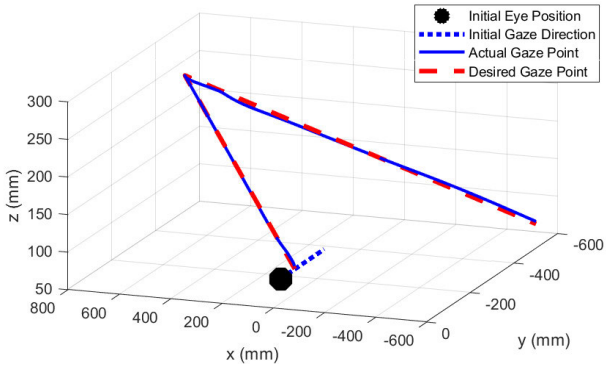


FIGURE 14. 3D view of gaze response to a moving target.

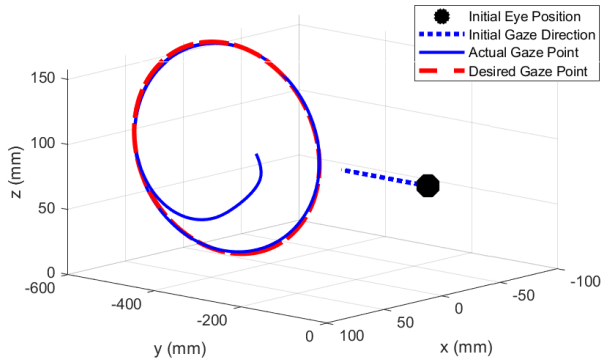


FIGURE 15. 3D view of gaze response to circular motion.

second interval, the direction and speed change to $[-600\ -100\ -100]$ mm/s. The gaze shows a smooth pursuit motion tracking the moving target, resembling the OKR reflex. In this case, if the neck joints can keep up with the movements of the target, the eye joints will not do much work. However, in the case of a faster motion, the eye will follow the target until the face direction keeps up with the target motion. Another case is tracking a circular motion as shown in Fig. 15. The target follows a circular path with an amplitude of 75 mm and a frequency of π Hz. The eye joints quickly take the gaze towards the circular path, stabilize it at the target, and compensate for neck motion as the face is also directed towards the circling target.

IV. CONCLUSION

Addressing the humanoid gaze, a humanoid neck and eye were modeled based on iCub. Additional to the gaze direction task, a face direction subgoal is introduced to have a more natural gaze. An IK was developed for the eye to achieve the desired gaze direction. Similarly, the desired face direction was achieved by developing an IK for the neck. By suggesting an algebraic approach, the developed scheme has avoided the problems related to numerical approach such as the processing time of uncertain number of iterations, and the numerical instability risks. The multiple solutions problem was adderessed by using virtual states with direct equations instead of using constraints, cost functions, or solvers. In addition, the desired joints were obtained analytically instead of using numerical solvers. Simulation results show that the gaze rejects undesirable neck motion and tracks targets, exhibiting VOR and OKR reflexes, respectively. Moreover, a gaze change from one target to another is quickly made, resembling saccadic motion. Overall, the gaze has been controlled, showing human-like gazing behavior.

For future work, recommendations are given in the following:

1) Generalization of the developed IK method.
   - The developed IK method can be generalized by considering the rotation plane of humanoid joints and their normals.
   - The number of DOF may vary between systems, but the method should be able to achieve the desired result by decomposing the problem into subproblems with certain objectives.
   - Although any subproblem should be composed based on the related system itself, position and direction are the ultimate goals for each one.
   - Therefore, by decomposing the main objective into enough subobjectives and using all the system's DOF, the method should result in a unique analytical solution.
2) Controlling the gazing behavior by introducing a gaze path objective in the controller.
   - Although neck and eye coordination affects the gazing behavior, the path taken towards the desired directions should also be considered.
   - One way is to define a maximum proportional deviation from the actual gaze with fixed or adaptive ratios for each joint so that a desired behavior will be followed.
   - Another way is to include a machine learning algorithm by making it learn certain gaze features in order to impose them on the gaze control.
   - Ultimately, by defining different gaze styles with certain features, multiple algorithms can be developed to follow a particular behavior.
3) Including an intelligent part in the controller to give more suitable actions for different cases.

- For cases like stationary target, small change in target position, huge change in target position, slow moving target, etc., theÂ fuzzification method can be utilized to provide more appropriate control actions for each defined case.
- In general, learning algorithms and artificial intelligence can also be utilized to provide high-level decisions.

REFERENCES

[1] G. Schweigart, T. Mergner, I. Evdokimidis, S. Morand, and W. Becker, "Gaze stabilization by optokinetic reflex (OKR) and vestibulo-ocular reflex (VOR) during active head rotation in man," *Vis. Res.*, vol. 37, no. 12, pp. 1643–1652, Jun. 1997.

[2] I. Farkhatdinov, H. Michalska, A. Berthoz, and V. Hayward, "Review of anthropomorphic head stabilisation and verticality estimation in robots," in *Biomechanics of Anthropomorphic Systems*. Cham, Switzerland: Springer, 2019, pp. 185–209.

[3] E. Cuevas, D. Zaldivar, and R. Rojas, "Neurofuzzy prediction for gaze control," *Can. J. Electr. Comput. Eng.*, vol. 34, nos. 1–2, pp. 15–20, 2009.

[4] A. Roncone, U. Pattacini, G. Metta, and L. Natale, "Gaze stabilization for humanoid robots: A comprehensive framework," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2014, pp. 259–264.

[5] W.-Y. Kim, H.-T. Seo, S. Kim, and K.-S. Kim, "Practical approach for controlling optical image stabilization system," *Int. J. Control, Autom. Syst.*, vol. 18, no. 4, pp. 824–833, Apr. 2020.

[6] L. R. Young and D. Sheena, "Survey of eye movement recording methods," *Behav. Res. Methods Instrum.*, vol. 7, no. 5, pp. 397–429, Sep. 1975.

[7] H. Hoshikawa, M. Takahashi, T. Fujishiro, N. Tsujita, and C. Midorikawa, "Study of gaze stabilization during stepping and running," *Nippon Jibiinkoka Gakkai Kaiho*, vol. 86, no. 8, pp. 881–885, 1983.

[8] H. Hoshikawa, "Analysis of gaze stabilization during stepping and running," *Nippon Jibiinkoka Gakkai Kaiho*, vol. 88, no. 5, pp. 678–686, 1985.

[9] S. Rougeaux and Y. Kuniyoshi, "Robust tracking by a humanoid vision system," in *Proc. IAPR 1st Int. Workshop Humanoid Friendly Robot.*, Tsukuba, Japan, 1998. [Online]. Available: https://cir.nii.ac.jp/crid/1573950399922091008

[10] Y. Kuniyoshi and S. Rougeaux, "A humanoid vision system for interactive robots," in *Proc. 1st Asian Symp. Ind. Automat. Robot.*, 1999, pp. 13–21.

[11] A. Heya, K. Hirata, N. Niguchi, T. Yoshimoto, and T. Ota, "Dynamic analysis of high-speed three-degree-of-freedom electromagnetic actuator for image stabilization," *IEEE Trans. Magn.*, vol. 53, no. 11, pp. 1–4, Nov. 2017.

[12] A. Heya, K. Hirata, and N. Niguchi, "Dynamic modeling and control of three-degree-of-freedom electromagnetic actuator for image stabilization," *IEEE Trans. Magn.*, vol. 54, no. 11, pp. 1–5, Nov. 2018.

[13] A. Heya and K. Hirata, "Experimental verification of three-degree-of-freedom electromagnetic actuator for image stabilization," *Sensors*, vol. 20, no. 9, p. 2485, Apr. 2020.

[14] W. Cheng, X. Chen, Y. Xu, and X. Liu, "Mechanical stabilization in robotic bionic eyes with gravity and disturbance compensations," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 936–941.

[15] P.-Y. Chen, Y.-C. Jheng, S.-E. Huang, L. Po-Hung Li, S.-H. Wei, M. C. Schubert, and C.-L. Kao, "Gaze shift dynamic visual acuity: A functional test of gaze stability that distinguishes unilateral vestibular hypofunction," *J. Vestibular Res.*, vol. 31, no. 1, pp. 23–32, Feb. 2021.

[16] S. Fuenzalida, K. Toapanta, J. Paillacho, and D. Paillacho, "Forward and inverse kinematics of a humanoid robot head for social human robot-interaction," in *Proc. IEEE 4th Ecuador Tech. Chapters Meeting (ETCM)*, Nov. 2019, pp. 1–4.

[17] L. Gu and J. Su, "Gaze control on humanoid robot head," in *Proc. 6th World Congr. Intell. Control Autom.*, vol. 2, Jun. 2006, pp. 9144–9148.

[18] S. Takizawa, S. Ushida, T. Okatani, and K. Deguchi, "2DoF motion stabilization of biped robot by gaze control strategy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 1102–1107.

[19] K. Voges, B. Wu, L. Post, M. Schonewille, and C. I. De Zeeuw, "Mechanisms underlying vestibulo-cerebellar motor learning in mice depend on movement direction," *J. Physiol.*, vol. 595, no. 15, pp. 5301–5326, Aug. 2017.

[20] S. Mange, E. F. Helbling, N. Gravish, and R. J. Wood, "An actuated gaze stabilization platform for a flapping-wing microrobot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5409–5414.

[21] B. Adithya, B. N. P. Kumar, Y. H. Chai, and A. K. Patil, "Inspired by human eye: Vestibular ocular reflex based gimbal camera movement to minimize viewpoint changes," *Symmetry*, vol. 11, no. 1, p. 101, Jan. 2019.

[22] L. A. Son, M. Inagami, H. Hamada, T. Suzuki, and H. Aoki, "The effect of visual stimulus on voluntary eye movement based on a VOR/OKR model," *Int. J. Automot. Eng.*, vol. 8, no. 2, pp. 37–44, 2017.

[23] Z. Zhu, Q. Wang, W. Zou, and F. Zhang, "Motion control on bionic eyes: A comprehensive review," 2019, *arXiv:1901.01494*.

[24] M. Grotz, T. Habra, R. Ronsse, and T. Asfour, "Autonomous view selection and gaze stabilization for humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1427–1434.

[25] T. Habra, M. Grotz, D. Sippel, T. Asfour, and R. Ronsse, "Multimodal gaze stabilization of a humanoid robot based on reafferences," in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot. (Humanoids)*, Nov. 2017, pp. 47–54.

[26] A. Roncone, U. Pattacini, G. Metta, and L. Natale, "A Cartesian 6-DoF gaze controller for humanoid robots," in *Proc. 12th Robot., Sci. Syst.*, 2016. [Online]. Available: https://www.roboticsproceedings.org/rss12/p22.pdf

[27] T. Shibata and S. Schaal, "Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks," *Neural Netw.*, vol. 14, no. 2, pp. 201–216, Mar. 2001.

[28] M. F. Ansari, P. Kasprowski, and M. Obetkal, "Gaze tracking using an unmodified web camera and convolutional neural network," *Appl. Sci.*, vol. 11, no. 19, p. 9068, Sep. 2021.

[29] A. Rajesh and M. Mantur, "Eyeball gesture controlled automatic wheelchair using deep learning," in *Proc. IEEE Region Humanitarian Technol. Conf. (R-HTC)*, Dec. 2017, pp. 387–391.

[30] D. Saha, M. Ferdoushi, Md. T. Emrose, S. Das, S. M. M. Hasan, A. I. Khan, and C. Shahnaz, "Deep learning-based eye gaze controlled robotic car," in *Proc. IEEE Region Humanitarian Technol. Conf. (R-HTC)*, Dec. 2018, pp. 1–6.

[31] X. Zhao, Y. He, X. Chen, and Z. Liu, "Human–robot collaborative assembly based on eye-hand and a finite state machine in a virtual environment," *Appl. Sci.*, vol. 11, no. 12, p. 5754, Jun. 2021.

[32] P. Li, X. Hou, L. Wei, G. Song, and X. Duan, "Efficient and low-cost deep-learning based gaze estimator for surgical robot control," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2018, pp. 58–63.

[33] J. Law, P. Shaw, and M. Lee, "A biologically constrained architecture for developmental learning of eye–head gaze control on a humanoid robot," *Auton. Robots*, vol. 35, no. 1, pp. 77–92, Jul. 2013.

[34] U. Park, E. J. Hwang, and J. Choi, "Automatic generation of eye expressions with end-to-end learning," Univ. Sci. Technol., Korea Inst. Sci. Technol., Tech. Rep.

[35] B.-S. Yoo and J.-H. Kim, "Fuzzy integral-based gaze control of a robotic head for human robot interaction," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1769–1783, Sep. 2015.

[36] I. Lütkebohle, F. Hegel, S. Schulz, M. Hackel, B. Wrede, S. Wachsmuth, and G. Sagerer, "The Bielefeld anthropomorphic robot head 'Flobi,'" in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3384–3391.

[37] Y. Maeda and J. Niu, "Human intention estimation using fuzzy inference based on gaze tracking and saliency map," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2019, pp. 1–6.

[38] H. Deng and C. Xie, "An improved particle swarm optimization algorithm for inverse kinematics solution of multi-DoF serial robotic manipulators," *Soft Comput.*, vol. 25, no. 21, pp. 13695–13708, Nov. 2021.

[39] S. Xie, L. Sun, Z. Wang, and G. Chen, "A speedup method for solving the inverse kinematics problem of robotic manipulators," *Int. J. Adv. Robot. Syst.*, vol. 19, no. 3, pp. 1–10, 2022.

[40] Y. Bai, M. Luo, and F. Pang, "An algorithm for solving robot inverse kinematics based on FOA optimized BP neural network," *Appl. Sci.*, vol. 11, no. 15, p. 7129, Aug. 2021.

[41] D. Fan, Y. Liu, X. Chen, F. Meng, X. Liu, Z. Ullah, W. Cheng, Y. Liu, and Q. Huang, "Eye gaze based 3D triangulation for robotic bionic eyes," *Sensors*, vol. 20, no. 18, p. 5271, Sep. 2020.

[42] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta, "The design of the iCub humanoid robot," *Int. J. Humanoid Robot.*, vol. 9, no. 4, Dec. 2012, Art. no. 1250027.

[43] V. Tikhanoff, P. Fitzpatrick, F. Nori, L. Natale, G. Metta, and A. Cangelosi, "The iCub humanoid robot simulator," in *Proc. IROS Workshop Robot Simulators*, vol. 22, 2008. [Online]. Available: https://people.csail.mit.edu/paulfitz/pub/tikhanoff08icub.pdf

[44] *Home Page—iCub Tech Docs*. Accessed: Mar. 14, 2022. [Online]. Available: https://icub-tech-iit.github.io/documentation/
[45] A. Parmiggiani, "Torque control: A study on the iCub humanoid robot," Ph.D. thesis, Italian Inst. Technol., Univ. Genoa, Genoa, Italy, 2010.
[46] H. Hashimoto, K. Maruyama, and F. Harashima, "A microprocessor-based robot manipulator control with sliding mode," *IEEE Trans. Ind. Electron.*, vol. IE-34, no. 1, pp. 11–18, Feb. 1987.

AHMED NIZAR ALSHAKHS received the B.Sc. degree from the Control and Instrumentation Systems Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2018, and the M.Sc. degree from the Control and Instrumentation Engineering Department, KFUPM, in 2022. He is currently a Graduate Assistant with the Department of Control and Instrumentation Engineering, KFUPM. He taught several laboratories in automation, systems control modeling and simulation, digital systems control, and programmable logic controllers (PLC's). His research interests include humanoid robots, unmanned aerial vehicles (UAVs), interconnected systems, intelligent control, system identification, and central processing unit (CPU) design and architecture.

MUHAMMAD FAIZAN MYSOREWALA (Member, IEEE) received the B.E. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington (UTA), USA, in 2002 and 2008, respectively. From 2002 to 2008, he was a Research Assistant with the Automation and Robotics Research Institute (ARRI), UTA. He joined the Systems Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2008, as an Assistant Professor. Currently, he is an Associate Professor with the Control and Instrumentation Engineering Department, KFUPM, where he is also an affiliate Faculty Member of the Interdisciplinary Research Center of Smart Mobility and Logistics. His research interests include sampling algorithms for distributed sensors and robots, intelligent control, and emerging technologies of cyber-physical systems. He has been a principal investigator for several funded research projects, a number of these are related to sensor deployment strategies for monitoring, and leak detection in pipelines. He has published numerous journals and conference papers, delivered presentations in international conferences, and coauthored a book. He is a technical reviewer of reputable journals, such as IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and *Journal of Intelligent & Robotic Systems*.

ABDUL-WAHID A. SAIF received the B.Sc. degree from the Physics Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, the M.Sc. degree from the Control and Instrumentation Engineering Department, KFUPM, and the Ph.D. degree from the Control and Instrumentation Group, Department of Engineering, Leicester University, U.K. He is currently a Professor with the Department of Control and Instrumentation Engineering, KFUPM. He worked as a principal investigator or a co-investigator in many internal and external funded projects by KFUPM. He taught several courses in modeling and simulation, digital control, digital systems, microprocessor and micro-controllers in automation, optimization, numerical methods, PLC's, process control and control system design plus other courses in EE, physics, engineering economics, and programming. He has supervised and was a member of the thesis committees of many Ph.D. and M.Sc. students in systems engineering with the Department of Computer Engineering and Electrical Engineering Department. He has published more than 100 articles in reputable journals and conferences, patents, and technical reports. His research interests include simultaneous and strong stabilization, robust control and H_∞ optimization, wire and wireless networked control, time delay systems, and instrumentation and computer control.

KHALED ALSHEHRI (Member, IEEE) received the B.S. degree in control and instrumentation systems engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2012, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2015 and 2019, respectively. He is currently an Assistant Professor with the Control and Instrumentation Engineering Department, KFUPM. His practical experiences include working for General Electric, Saudi Aramco, ISO New England, and Commonwealth Edison (ComEd).