

## Shared Control of a Robot Arm Using BCI and Computer Vision

Mohammed Qasim\*, Omar Y. Ismael

Department of Systems and Control Engineering, Ninevah University, Mosul 41001, Nineveh, Iraq

Corresponding Author Email: [mohammed.qasim@uoninevah.edu.iq](mailto:mohammed.qasim@uoninevah.edu.iq)



<https://doi.org/10.18280/jesa.550115>

### ABSTRACT

**Received:** 4 December 2021

**Accepted:** 13 January 2022

#### Keywords:

*shared control, vision-based control, image-based visual servoing, BCI, brain-computer interface, CoppeliaSim, computer vision, robot arm*

Brain-Computer Interface (BCI) is a device that can transform human thoughts into control commands. However, BCI aggravates the common problems of robot teleoperation due to its low-dimensional and noisy control commands, particularly when utilized to control high-DOF robots. Thus, a shared control strategy can enhance the BCI performance and reduce the workload for humans. This paper presents a shared control scheme that assists disabled people to control a robotic arm through a non-invasive Brain-Computer Interface (BCI) for reach and grasp activities. A novel algorithm is presented which generates a trajectory (position and orientation) for the end-effector to reach and grasp an object based on a specially designed color-coded tag placed on the object. A single camera is used for tag detection. The simulation is performed using the CoppeliaSim robot simulator in conjunction with MATLAB to implement the tag detection algorithm and Python script to receive the commands from the BCI. The human-in-the-loop simulation results prove the effectiveness of the proposed algorithm to reach and grasp objects.

## 1. INTRODUCTION

Recent developments in assistive robotic make it possible for individuals with a disabled upper limb to pick and place objects. However, controlling robot arms remains challenging for reach and grasp activities [1]. Robot arms are typically controlled by either traditional manual control or by automatic control. It is preferred for disabled people to automatically control the robot arm with a method that is based on the intention and can be supervised by them to avoid a risk of human injury and environmental damage.

The BCI is used in assistive robot control strategies to simplify the communication between the disabled person and the robot arm [2-6]. The non-invasive BCI is preferred to use as it is safer and easier to record the electroencephalography (EEG) over the scalp than the invasive BCI that has technical difficulties and clinical risks. However, it offers a lower signal to noise ratio [7-10]. It is still challenging to utilize the BCI in controlling a multiple degrees-of-freedom (DOFs) robot arm for reach and grasp activities in a three-dimensional space (3D) [11].

The shared control strategy is used to reduce the user's mental effort and can allow the subject to reach, grasp and move objects without moment-by-moment supervision. An effective strategy of shared control is to utilize the BCI and computer vision. The value of computer vision in BCI-controlled robotic arms was demonstrated in many works in the literature.

Chen et al. [12] examined the usage of a robotic arm controlled by a high-frequency steady-state visual evoked potential (SSVEP)-based BCI and computer vision to accomplish a pick and place task without the user's continuous supervision

Xu et al. [13] used shared control to achieve dexterous robotic arm control using a motor imagery-based (MI-based)

BCI with computer vision guiding. In Xu et al. [13], the robotic arm's end-effector grips the object independently once it enters the pre-defined vision-guided area.

To tackle the major difficulties of target item selection, robot intelligent planning, and shared control by both human intention and the robot, Zhang et al. [14] presented a pipeline of hybrid EEG-based BCI for robot grasping via shared control and computer vision.

For the brain-actuated manipulator, Tang and Zhou [15] devised a shared-control method in which the computer vision module was created to identify and locate targets, allowing humans to focus on deciding which target is selected. Four participants in Tang and Zhou [15] were given the task of controlling the robotic arm so that it could grip the target bottle and deliver it to the subject's lips.

As in the above works, the effectiveness of utilizing the computer vision in controlling the BCI-robot arm can be emphasized in making the BCI system simpler and more straightforward to use, the ability to accomplish a 3D movement with fewer mental commands, reducing the amount of time required for user training which will aid in the use of BCIs in daily life, and recognizing a huge number of objects simultaneously.

However, the advantages of BCI-based computer vision in the above works, there is still an issue. Many researchers have claimed that in unstructured environments, no computer vision technique appears to be satisfactory for object manipulation such as reach and grasp activities due to the lack of communication between the object and the robot arm [16].

It is required to develop a shared control strategy in which objects can provide robot arms with knowledge on how to reach and grasp them when using both BCI and computer vision. The development of the proposed algorithm is motivated by the above.

This paper presents a shared control scheme with a novel

algorithm to automatically control the movement of the robot end-effector to assist disabled individuals in controlling a robot arm through non-invasive BCI for reach and grasp activities. The proposed algorithm generates a suitable trajectory (position and orientation) for the end-effector to reach and grasp the object based on a uniquely designed color-coded tag placed on the object.

The main contributions of this paper include i) designing a unique color-coded tag that assists in determining a suitable end-effector trajectory to reach and grasp objects, ii) developing an image-based controller, with the help of the tag, that does not need to calculate the inverse of the interaction matrix and is simple to implement, iii) controlling each velocity component of end-effector independently; thus, it can be ensured that the tag does not leave the camera field of view by moving the tag (moving the end-effector in the  $x$  and  $y$  direction of its reference frame) to the center of the image first before rotating the end-effector and iv) proving local asymptotic stability of the controller.

The rest of the paper is organized as follows: The color-coded tag design and placement are presented in Section 2. The tag detection algorithm is shown in Section 3. The vision-based control method is presented in Section 4. Section 5 presents the arbitrator. The simulation's findings are demonstrated and discussed in depth in Section 6. Section 7 summarizes the paper's findings and suggests areas for future research.

## 2. COLOR-CODED TAG DESIGN

Recent studies in BCI-based computer vision have utilized object features such as shape, color, and deep information to identify the objects in the scene [12, 14, 15, 17-19]. These studies may have limitations because of the chance of color's interference, limited number of objects and lack of information when dealing with objects (i.e., torque/force and amount of end-effector opening required for object pickup). This paper tries to solve the above limitations by designing a special tag that can be placed on objects.

To pick up an object intuitively and conveniently, e.g., cuboid shown in Figure 1(a), it is required that: i) the center of the end-effector reference frame be aligned with the center of the object (or the center of the tag as shown in Figure 1(b)) and the vision sensor as shown in Figure 1(c); ii) the plane containing the  $x - y$  axes of the end-effector be parallel to object plane 1 (see Figure 1(a)), or parallel to the plane containing the tag as shown Figure 1(b); iii) the plane containing the  $y - z$  axes of the end-effector reference frame be parallel to object plane 2 (see Figure 1(a)), or parallel to the plane perpendicular to the tag plane and passing through the centers of segments 1, 2 and 3 of the tag (see Figure 2); and iv) the end-effector should be at an appropriate distance from the object.

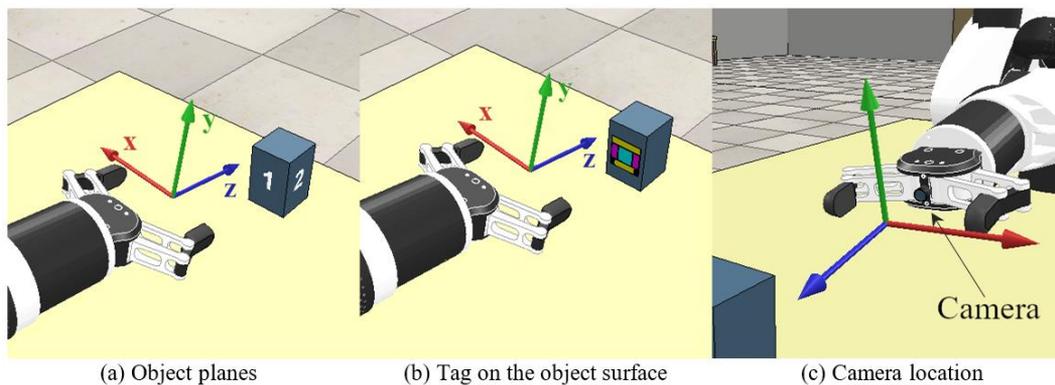


Figure 1. End-effector with its reference frame while approaching the object (tag)

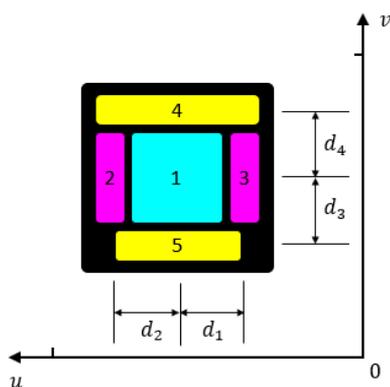


Figure 2. Color-coded tag in the image plane

The goal is to design a tag, that assists to achieve the abovementioned four requirements, to be placed on the object to be picked up. The designed tag consists of five segments colored with three distinct colors (cyan, magenta and yellow). These segments are laid on a black background, colored and arranged in a way to assist in detecting the tag and

distinguishing it from the surrounding objects even if they have the same colors as the tag. When the first three requirements are met, the following is true by the design of the tag, see Figure 2 which is the picture of the tag in the image plane: i) the center of the middle segment of the tag is at the origin of the image plane, i.e.,  $u_1 = v_1 = 0$  ii) the distances  $d_1$  and  $d_2$  are equal, and the distances  $d_3$  and  $d_4$  are equal; and iii) the  $v$ -coordinates of segments 2 and 3 of the tag are equal, i.e.,  $v_2 = v_3 = 0$ . Note that  $d_2$  and  $d_1$  are the horizontal distances between segments 2, 3 and 1 respectively and  $d_4$  and  $d_3$  are the vertical distance between segments 4, 5 and 1 respectively.

The coordinates of the five segments of the tag with respect to the image plane are known by the Tag Detection Algorithm (TDA). When the center of the middle segment is not aligned with the center of the end-effector reference frame (requirement (i) is not met), the end-effector is moved along the  $x$  and  $y$  axes of its reference frame until the alignment is done. When  $d_1 \neq d_2$  and/or  $d_3 \neq d_4$  (requirement (ii) is not met), the end-effector is rotated around  $y$  and  $x$  axes of its reference frame until  $d_1 = d_2$  and  $d_3 = d_4$  respectively. Note that if angled pickup is required, the end-effector is rotated

around  $y$  and  $x$  axes of its frame until  $d_1 = d_2$  and  $d_3 - d_4 = d_\theta$  respectively. Where  $d_\theta$  denotes the vertical distance that gives the desired pickup angle. When the  $v$ -coordinates of segments 2 and 3 are not equal (requirement (iii) is not met), the end-effector is rotated around  $z$ -axis of its reference frame until  $v_2 = v_3$ . When the vertical distance between the segments 4 and 5 are not equal to the desired one (requirement (iv) is not met), the end-effector is moved along the  $z$ -axis of its reference frame until  $v_4 - v_5 = d_d$ . Where  $d_d$  is the desired vertical distance between segments 4 and 5 at the pickup operation. The designed tag is simple to be made and passive, i.e., it does not require a power source to operate.

## 2.1 Tag placement

Since the end-effector adjusts its pose to center the tag in the middle of the image, i.e., the end-effector center is aligned with the tag center, it is reasonable to place the tag on the object at the point where it is the best for it to be picked up from. The reason behind this tag placement is that the robot end-effector can adjust its. The object may contain multiple tags of the same type. The multiple tags are placed at multiple locations on the object so that the object can be seen and picked up if the end-effector reaches it from any direction. For instance, the cuboid objects may have four tags of the same type, one on each side face of them. Each object has its unique tag. If two or more tags of the same type are detected, the largest tag (the one with more pixels in all its segments) is considered. If different tags are detected at the same time, the closer (larger in terms of pixels) tag is considered. Since each tag is unique, useful information can be stored in the tag such as the amount of end-effector opening required for the object and the end-effector torque/force required to pick the object.

## 3. TAG DETECTION ALGORITHM

A robust and reliable computer vision algorithm decreases the chance of false positives by restricting the selected objects in the workspace to only those that should or could be acted upon [17]. This section concentrates on the detection of the artificial color-coded tag designed in section 2 which is based on a mixture of five segments with three distinct colors allocated near one another as depicted in Figure 2. These colors are Cyan, Magenta and Yellow (CMY).

Each video frame is captured by the RGB camera that is mounted in the end effector of the robot arm in the CoppeliaSim environment. Then, it is sent to MATLAB which is used to identify tags. The suggested framework of tags detection is separated into three stages.

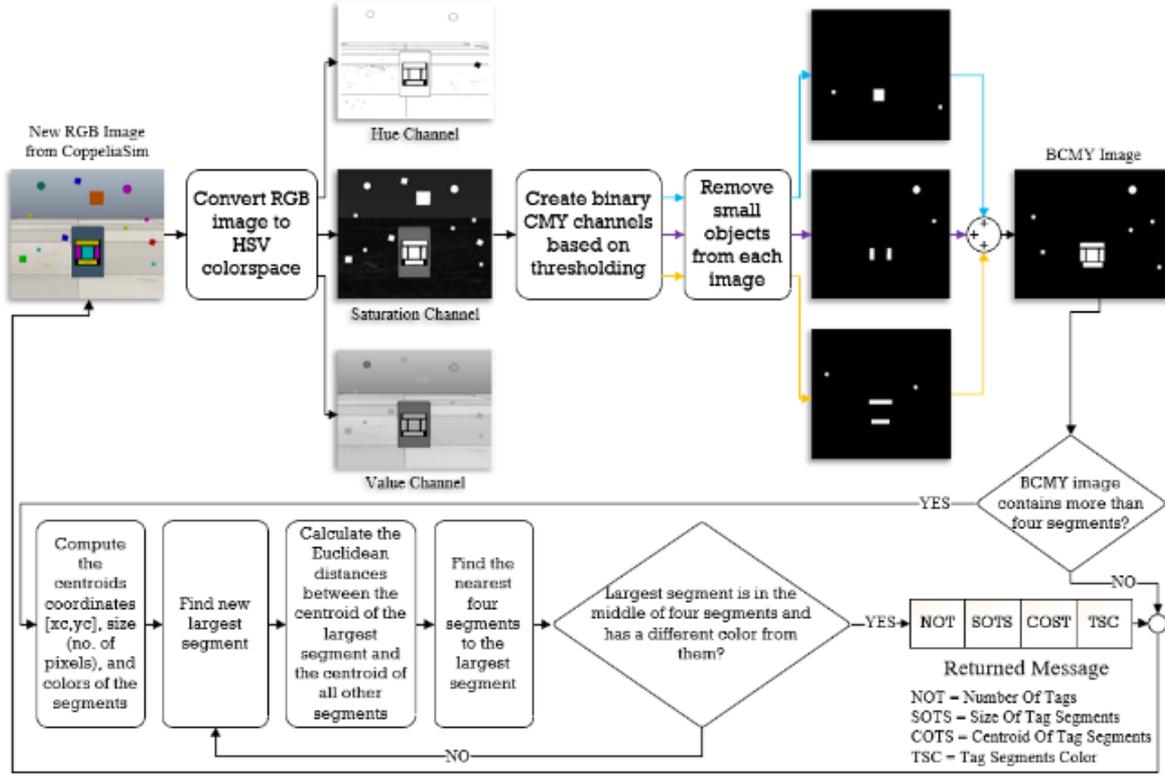
In the first stage, MATLAB takes the RGB image and converts it to the Hue, Saturation, and Value (HSV) color model and returns it as  $M \times N \times 3$  numeric array. HSV color model is utilized in this work since its performance is superior to the RGB color model used in Noaman et al. [20] in a variety of lighting conditions and the way it deals with noise. HSV color model allows filtering by the color intensity and brightness in addition to the pixel color. RGB color space is transformed into HSV color space using:  $H = \cos^{-1} \frac{\frac{1}{2}(2R-G-B)}{\sqrt{(R-G)^2 - (R-B)(G-B)}}$ ,  $S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)}$ , and  $V = \max(R, G, B)$ . where Hue ( $H$ ) refers to a number from 0 to 360 degrees and represents the color's location on a color wheel,

Saturation ( $S$ ) is the amount of color or deviation from neutral which also describes the amount of grey in a particular color, from 0 to 1, and Value ( $V$ ) refers to a color's maximum value among its red, green, and blue components which also describes the brightness or intensity of the color in the range of  $[0, 1]$ .

HSV image is then separated into three HSV channels, called H-channel, S-channel, and V-channel. Based on the thresholding of the HSV channels, the CMY channels are created. The Hue threshold is selected as  $170 > H \geq 190$  for Cyan,  $270 > H \geq 330$  for Magenta and  $50 > H \geq 90$  for Yellow. The saturation threshold is chosen as  $0.3 < S \leq 1$ . The value threshold is selected as  $0.2 < V \leq 1$ . The binary equivalent image of each of the CMY channels is then generated according to the attendance of each of the CMY colors in the scene. The C, M, and Y components of these binary channels are referred to as BC, BM, and BY, respectively.

The second stage is the image segmentation, in which for each binary image, a collection of pixel areas is created, allowing just the most essential portions of the image to be processed rather than the full image. The region-growing approach is used in the segmentation process, which produces clean edges and effective segmentation results. The nearby pixels of the initial seed points are assessed according to an 8-connected neighborhood in this segmentation technique and repeatedly computed if the pixel's neighbors need to be added to the region. The segment is a collection of binary data found in each region. Depending on what the robot camera sees in the surroundings, segments with various numbers of pixels are stored in the BC, BM, and BY channels. These channels are combined to generate a binary image called BCMY image. Furthermore, small segments with less than  $P$  pixels are eliminated from the BCMY image.

In the third stage, the tag is recognized. Firstly, a conditional operation is performed on the BCMY image to check if the camera sees more than four segments with CMY colors. Then, the detection of the tags is based on size, distance, color and location of the segments in the BCMY image. Consequently, two matrices called  $C_t$  and  $D_e$  is generated with this information. Matrix  $C_t$  includes the centroids coordinates, sizes and colors of the segments. The tag design in section 2 assumes that the middle segment has the largest size and a different color compared with the other segments of the tag. Matrix  $D_e$  contains the Euclidean distances between the centroid of the largest segment and the centroid of all other segments in  $C_t$ . After that,  $D_e$  is arranged descendingly to find the nearest four segments to the largest segment. Next, a conditional procedure is executed to check if the largest segment has a different color from these four segments and is in the middle of them. This is done by comparing the distance between the centroid of the largest segment and the average of the centroids of the nearest four segments to it and this distance should not exceed the minimum of the distances between the largest segment and every one of the nearest four segments to it. If the previous test is passed, a tag is detected and the information of these tag segments are omitted from  $C_t$ . If the test is not passed, the information of the largest segment is omitted only from  $C_t$ . The steps of the third stage may be repeated many times until finding all the tags in the image. When there are less than five segments in  $C_t$ , the counts of the tags and the size of each segment inside the tag with their centroids and colors are returned to CoppeliaSim. The steps of the proposed method of tags detection are shown in Figure 3.



**Figure 3.** The steps of the tag detection algorithm

After detecting the tag, it is necessary to identify each one of the five segments of the tag since it is used in designing the image-based controller. This is done based on the size and position of the segment as follows: segments 1, 4 and 5 have the largest, the second largest and the third largest sizes among the other segments respectively; segments 2 and 3 can be easily identified based on their positions and the positions of segments 4 and 5 in the image plane.

## 4. VISION-BASED CONTROL

### 4.1 Trajectory generating controller

In this section, the image-based control law is designed in which the image data from a single camera is used to control the manipulator motion directly. It is assumed that the manipulator is equipped with a tracking controller, e.g., similar to the ones in the references [21-24], to track the control commands (end-effector trajectory) generated by the image-based controller.

Let  $\xi_c^c(t) = [\mathbf{v}_c(t), \mathbf{w}_c(t)]^T$  be the camera velocity which consists of linear and angular velocities of the camera reference frame where  $\mathbf{v}_c(t) = [v_{cx}(t), v_{cy}(t), v_{cz}(t)]$  and  $\mathbf{w}_c(t) = [w_{cx}(t), w_{cy}(t), w_{cz}(t)]$  and  $[u_{1-5}(t), v_{1-5}(t)]$  be the coordinates of the centroids of the tag segments (see Figure 2) in the image plane. These centroids serve as image feature points and are utilized to design the control laws that compute the camera velocity.

To align (requirement (i) of section 2) the end-effector reference frame with the center of segment 1 of the tag and since the vision sensor is already aligned with the end-effector frame by the design, the tag must be in the center of the image. The proportional control laws that center segment 1 of the tag in the image plane are formed as follows:

$$v_{cx}(t) = K_{pvx}e_1(t) \quad (1)$$

$$v_{cy}(t) = K_{pvy}e_2(t) \quad (2)$$

where,  $e_1(t) = u_1 - 0$ ,  $e_2(t) = v_1 - 0$ ,  $K_{pvx} > 0$  and  $K_{pvy} > 0$ .

Next, it is necessary to move the end-effector towards the object and stop it before the object before the pickup operation. The distance from the end-effector to the tag (object) can be approximated based on the vertical distance  $d_d$  between segments 4 and 5 of the tag. To calculate  $d_d$ , the end-effector is moved manually to a suitable location before the object to be picked (the four requirements of section 1 must be satisfied) and then the  $d_d$  is calculated and recorded to be used in the control law. The proportional control law that moves the end-effector towards the object can be formed as

$$v_{cz}(t) = K_{pvz}e_3(t) \quad (3)$$

where,  $e_3(t) = d_d - \text{sign}(v_4)(v_4 - v_5)$  and  $K_{pvz} > 0$ . The sign function is used in Eq. (3) to make the vertical distance  $(v_4 - v_5)$  always positive since it may be negative when the camera rotates around z-axis of its reference frame.

To make the  $x$ - $y$  axes plane of the end-effector parallel to tag plane (requirement (ii) of section 1), the camera is rotated around  $x$  and  $y$  axes of its reference frame based on following proportional control laws:

$$w_{cx}(t) = K_{pwx}e_4(t) \quad (4)$$

$$w_{cy}(t) = K_{pwy}e_5(t) \quad (5)$$

where,  $e_4(t) = d_4 - d_3 = -v_4 + 2v_1 - v_5$ ,  $e_5(t) = d_1 - d_2 = u_2 - 2u_1 + u_5$ ,  $K_{pwx} > 0$  and  $K_{pwy} > 0$ .

To achieve requirement (iii) of section 1, the camera is rotated around z-axis of its reference frame based on following PID control law

$$w_{cz}(t) = K_{pwz}e_6(t) \quad (6)$$

where,  $e_6(t) = v_2 - v_3$  and  $K_{pwz} > 0$ . The control law (6)

$$L = \begin{bmatrix} -\frac{\lambda}{z_1} & 0 & \frac{u_1}{z_1} & \frac{u_1 v_1}{\lambda} & \frac{\lambda^2 + u_1^2}{-\lambda} & v_1 \\ 0 & -\frac{\lambda}{z_1} & \frac{v_1}{z_1} & \frac{\lambda^2 + u_1^2}{\lambda} & \frac{-u_1 v_1}{\lambda} & -u_1 \\ 0 & \frac{\lambda}{z_4} - \frac{\lambda}{z_5} & \frac{v_5}{z_5} - \frac{v_4}{z_4} & \frac{-(v_4^2 - v_5^2)}{\lambda} & \frac{(u_4 v_4 - u_5 v_5)}{\lambda} & u_4 - u_5 \\ 0 & \frac{\lambda}{z_4} - \frac{2\lambda}{z_1} + \frac{\lambda}{z_5} & \frac{2v_1}{z_1} - \frac{v_4}{z_4} - \frac{v_5}{z_5} & \frac{-2v_1^2 + v_4^2 + v_5^2}{-\lambda} & \frac{u_4 v_4 - 2u_1 v_1 + u_5 v_5}{\lambda} & u_4 - 2u_1 + u_5 \\ -\frac{\lambda}{z_2} - \frac{\lambda}{z_3} & 0 & \frac{u_2}{z_2} - \frac{2u_1}{z_1} - \frac{u_3}{z_3} & \frac{u_2 v_2 - 2u_1 v_1 + u_3 v_3}{\lambda} & \frac{-2u_1^2 + u_2^2 + u_3^2}{-\lambda} & v_2 - 2v_1 + v_3 \\ 0 & \frac{\lambda}{z_3} - \frac{\lambda}{z_2} & \frac{v_2}{z_2} - \frac{v_3}{z_3} & \frac{v_2^2 - v_3^2}{\lambda} & \frac{u_2 v_2 - u_3 v_3}{-\lambda} & u_3 - u_2 \end{bmatrix}$$

To ensure that the tag remains inside the image plane when first detected, the gains  $K_{pvx}$  and  $K_{pvy}$  are tuned to be larger than the other gains to ensure that the tag moves quickly to the center of the image before performing any camera rotation or moving closer to the tag which may cause the tag to leave the image plane.

The relation between the camera velocity  $\xi_c^c$  and the end-effector velocity  $\xi_e^e$  is [25].

$$\xi_e^e = \begin{bmatrix} R_c^e & S(d_c^e)R_c^e \\ 0_{3 \times 3} & R_c^e \end{bmatrix} \xi_c^c \quad (7)$$

where,  $R_c^e$  and  $d_c^e$  denote the fixed relative orientation and position of the camera frame with respect to the end-effector frame and  $S()$  is a skew-symmetric matrix. The end-effector velocity can be expressed with respect to the robot base frame as follows: [25].

$$\xi_e^0 = \mathbf{R}_e^0 \xi_e^e = \begin{bmatrix} R_e^0 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_e^0 \end{bmatrix} \xi_e^e \quad (8)$$

The proposed controller (slow, outer loop) generates the desired end-effector velocity  $\xi_e^0$ . If the manipulator does not

rotates the camera around z-axis of its reference frame to constantly make  $v_4 > v_5$  in the image plane.

Note that the proposed controller is linear and easy to implement. Moreover, each velocity component of the camera is controlled independently. Thus, unlike traditional image-based controllers, the proposed one does not need to calculate or estimate the inverse of the interaction matrix that requires depth information of the feature points.

have a controller that can follow velocity commands directly, an end-effector pose can be generated at each time step and passed to an inverse kinematic solver to generate the desired robot joint angles ( $q_d$ ). The desired joint angles are then passed to a tracking controller (fast, inner loop) to achieve the desired joint angles, i.e., follow the desired trajectory of the end-effector, as shown in Figure 4.

#### 4.2 Stability analysis

To analyze the stability of the trajectory generating controller, the behaviors of the errors  $e_{1-5}$  need to be studied. Let  $\mathbf{e} = [e_1, e_2, e_3, e_4, e_5, e_6]^T$  be the vector of errors. Then, taking the time derivative (when  $v_4 > 0$ ) of  $\mathbf{e}$  and knowing that [25]  $\dot{u} = -\frac{\lambda}{z}v_{cx} + \frac{u}{z}v_{cz} + \frac{uv}{\lambda}w_{cx} - \frac{\lambda^2 + u^2}{\lambda}w_{cy} + vw_{cz}$  and  $\dot{v} = -\frac{\lambda}{z}v_{cy} + \frac{v}{z}v_{cz} + \frac{\lambda^2 + u^2}{\lambda}w_{cx} - \frac{uv}{\lambda}w_{cy} - uw_{cz}$  yields.

$$\dot{\mathbf{e}} = \mathbf{L}\xi_c^c \quad (9)$$

where,  $z$  is the z-coordinate of the point in the world relative to the camera reference frame,  $\lambda > 0$  is the focal length of the camera lens and  $L$  is at the top of this page.

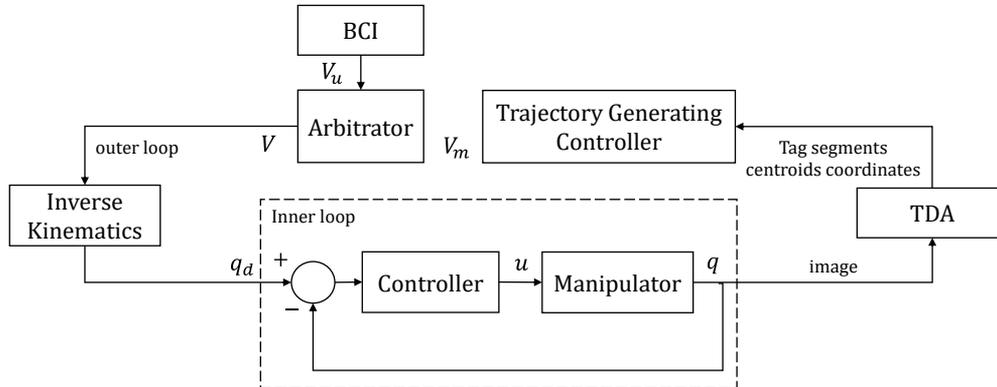


Figure 4. The proposed shared control scheme

Using Eqns. (1)-(6), it can easily be shown that  $\xi_c^e = \mathbf{K}\mathbf{e}$  where  $\mathbf{K} = \text{diag}(K_{pvx}, K_{pvy}, K_{pvz}, K_{pwx}, K_{pwy}, K_{p wz}) > 0$ . Substituting  $\xi_c^e = \mathbf{K}\mathbf{e}$  into Eq. (9) gives:

$$\dot{\mathbf{e}} = \mathbf{L}\mathbf{K}\mathbf{e} \quad (10)$$

The differential system (10) has an equilibrium point since if  $\mathbf{e} = \mathbf{0}$ , then  $\dot{\mathbf{e}} = \mathbf{L}\mathbf{K}\mathbf{e} = \mathbf{0}$ . This equilibrium point is unique because the matrix  $\mathbf{L}\mathbf{K}$  has full rank. Therefore, the error  $\mathbf{e}$  has no singularity and if it decreases, it decreases towards zero.

Tacking the first order Tylor expansion of system (10) around the equilibrium point ( $\mathbf{e}^\circ = \mathbf{0}$ , i.e., the end-effector pose right before the pickup operation) yields.

$$\dot{\mathbf{e}} \approx \mathbf{L}\mathbf{K}(\mathbf{e}^\circ)(\mathbf{e} - \mathbf{e}^\circ) \quad (11)$$

At the equilibrium point, the following are true:  $u_2 = -u_3 = u_{2e} > 0$ ,  $v_4 = -v_5 = v_{4e} > 0$ ,  $z_1 = z_2 = z_3 = z_4 = z_5 = z_e > 0$  and  $u_1 = v_1 = v_2 = v_3 = u_4 = u_5 = 0$ . Substituting these values in  $\mathbf{L}\mathbf{K}$  matrix yields  $\mathbf{L}\mathbf{K}(\mathbf{e}^\circ)$ . The linearized system (11) is locally asymptotically stable if and only if the eigenvalues of  $\mathbf{L}\mathbf{K}(\mathbf{e}^\circ)$  are negative. They are given by.

$$\begin{aligned} \lambda_1 &= -2K_{p wz}u_{2e} & \lambda_4 &= -2K_{pvz}v_{4e}/z_e \\ \lambda_2 &= -\lambda K_{pvx}/z_e & \lambda_5 &= -2K_{pwy}(-u_{2e})^2/\lambda \\ \lambda_3 &= -\lambda K_{pvy}/z_e & \lambda_6 &= -2K_{pwx}(-v_{4e})^2/\lambda \end{aligned} \quad (12)$$

Thus, the system (11) is locally asymptotically stable.

## 5. ARBITRATOR

Picking an object using BCI only is a tedious task since it is hard to control the end-effector of a six-degree of freedom robot arm. Thus, to make it effortless for the BCI user a shared control strategy is proposed. The user only controls the position of the end-effector while the proposed method performs the rest. Let  $\xi_{ue}^e(t) = [\mathbf{v}_u(t), \mathbf{0}_{1 \times 3}]^T$  where  $\mathbf{v}_u(t) = [v_{ux}(t), v_{uy}(t), v_{uz}(t)]$  be the end-effector velocity commanded by the user. The proposed strategy needs to decide when to grant control to the user or manipulator. This is done using the following:

$$V = (1 - \alpha)V_u + \alpha V_m \quad (13)$$

where,  $V$  specifies the ultimate velocity command sent to the tracking controller,  $V_u = \mathbf{R}_e^0 \xi_{ue}^e(t)$  and  $V_m$  is as in Eq. (8) and  $\alpha$  is equal to 1 when i) the end-effector is close to an object (can be estimated based on the tag size in pixels, and ii) if the user decides to activate the image-based controller by thinking of closing the end-effector and there is an object close to end-effector while  $\alpha$  is equal to 0 otherwise.

## 6. RESULTS AND DISCUSSION

The human-in-the-loop simulation is performed using CoppeliaSim robot simulation platform in conjunction with MATLAB to implement the TDA and Python script to receive the brain signals from the EMOTIV EPOC+ BCI (some of the experiments are performed using a game controller to move the manipulator end-effector due to the limited access to the

BCI). CoppeliaSim is a very powerful robot simulator due to its scalability, flexibility and versatility. It contains a large number of robot models, sensors, actuators that can be used for fast algorithms development. Additionally, it can be programmed by a variety of programming languages such as C++, Matlab, Python and Lua. CoppeliaSim provides a seamless method to communicate with MATLAB and Python and can communicate with both of them at the same time through remote APIs.

The setup for the simulation is as follows: a disabled person wearing a BCI sends commands to move the end-effector of the manipulator. These commands along with the commands from the trajectory generating controller are sent through a laptop to the manipulator.

Two scenarios are considered in the simulation. In both scenarios, it is assumed that the robot can reach and pick up the object, i.e., the inverse kinematic solver can find a feasible solution. If there are multiple objects on the table, it is assumed that there is enough free space around the object to perform the pickup operation. The simulated version of the e.do robot is utilized in the simulations. The discrete version of Eqns. (1)-(6) is implemented to generate the desired pose of the end-effector. For example, the  $x$ -axis trajectory of the end-effector is given by

$$x_e^0(n+1) = x_e^0(n) + \Delta T v_{x_e}^0 \quad (14)$$

where,  $x_e^0(1)$  is the  $x$  position of the end-effector at the time the vision-based control begins and  $\Delta T=50\text{ms}$  for the outer loop and  $5\text{ms}$  for the inner loop. The camera image resolution used in the simulations is  $512 \times 512$  pixels.

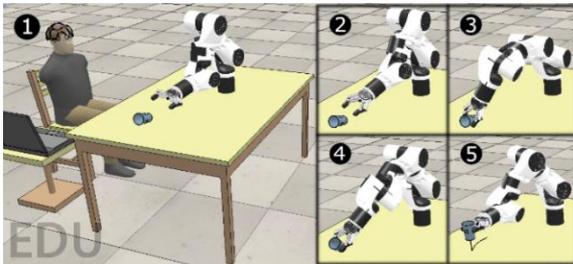
After the pickup, the  $x - z$  plane of end-effector reference frame becomes parallel to the table surface and the end-effector moves up in the direction of the  $y$ -axis of end-effector reference frame for a predefined distance and then the control returns to the human.

If the camera detects multiple tags of the same or different types, the largest (in the term in the number of pixels) tag is considered for the vision-based control. Since the camera is placed between end-effector arms, it is at full open to allow the camera to exploit the camera's full FOV when the human controls the end-effector. When the vision-based controller takes over the control of the end-effector, the gripper starts to close to a predefined distance dependent on the object size which is known from the tag placed on the object.

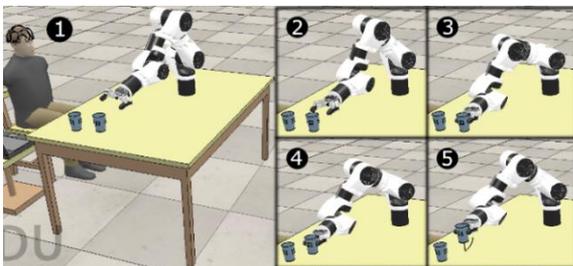
The first scenario is shown in Figure 5 where the person is required to pick up the cup that is fallen on the table surface. The cup has four tags of the same type and is placed on the circumference with equal distance between them, i.e., the robot can see a maximum of two tags at the same time. The person controls the pose of the end-effector and reaches a stage where he/she decides to grant the control to the image-based controller by thinking of closing the end-effector. Then, the process of picking up the cup is illustrated in five steps as shown in Figure 5. Step one shows the robot at the moment where the image-based controller takes control of the end-effector. The end-effector is closed to an appropriate distance while approaching the cup in step two. In step three, the four requirements in the tag design section are fulfilled and the end-effector is fully closed to grab the cup. Step four demonstrates the robot while adjusting its end-effector pose to make the  $x - z$  plane of the end-effector parallel to the table surface and elevating the end-effector to an appropriate distance from the table to reach the final end-effector pose in step five. At this

stage, the control returns to the human and now it is easy for him/her to control the end-effector from the pose in step five. The black curve in step 5 represents the path of the reference frame of the end-effector.

The second scenario is shown in Figure 6 where the person is required to pick up the cup far away from the person. The person changes the pose of the end-effector to pick up the cup and the vision sensor detects three tags: one for cup close to the person and two for the cup far away. When the end-effector reaches a predefined distance to the tag based on the size of the tag in pixels, the image-based controller is granted control and it moves the end-effector towards the cup and picks it up. The steps in the second scenario are similar to the steps in the first one.



**Figure 5.** The first scenario used in the simulation in which there is only one object in the scene



**Figure 6.** The second scenario used in the simulation in which there are two objects in the scene

## 7. CONCLUSION

In this paper, a shared control strategy was proposed to assist disabled people with the help of BCI to control a robotic arm that helps them in their daily routines. A color-coded tag is designed to help detect the object to be picked up and determine the right end-effector pose for the pickup operation. TDA was implemented inside MATLAB that helps detect the tag. The human-in-the-loop simulation was performed using the CoppeliaSim robotic simulator which is connected to both MATLAB and a Python script that receives the signals from the BCI. The stability of the proposed shared control was analyzed and proved. The results proved the efficiency of the proposed approach where the workload is greatly reduced for the disabled people making them focus on high-level tasks while the proposed algorithm manages the low-level tasks, i.e., the pickup operation. Future work will focus on implementing the proposed shared control strategy on the real e.do robot.

## REFERENCES

[1] Brose, S.W., Weber, D.J., Salatin, B.A., Grindle, G.G.,

Wang, H., Vazquez, J.J., Cooper, R.A. (2010). The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6): 509-521. <https://doi.org/10.1097/PHM.0b013e3181cf569b>

[2] Chen, X., Zhao, B., Wang, Y., Xu, S., Gao, X. (2018). Control of a 7-DOF robotic arm system with an SSVEP-based BCI. *International Journal of Neural Systems*, 28(8): 1850018. <https://doi.org/10.1142/S0129065718500181>

[3] Hortal, E., Iáñez, E., Úbeda, A., Perez-Vidal, C., Azorín, J.M. (2015). Combining a brain-machine interface and an electrooculography interface to perform pick and place tasks with a robotic arm. *Robotics and Autonomous Systems*, 72: 181-188. <https://doi.org/10.1016/j.robot.2015.05.010>

[4] Onose, G., Grozea, C., Angheliescu, A., et al. (2012). On the feasibility of using motor imagery EEG-based brain-computer interface in chronic tetraplegics for assistive robotic arm control: A clinical test and long-term post-trial follow-up. *Spinal Cord*, 50(8): 599-608. <https://doi.org/10.1038/sc.2012.29>

[5] Wodlinger, B., Downey, J.E., Tyler-Kabara, E.C., Schwartz, A.B., Boninger, M.L., Collinger, J.L. (2014). Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering*, 12(1): 016011.

[6] Zeng, H., Wang, Y., Wu, C., Song, A., Liu, J., Ji, P., Xu, B.J., Zhu, L.F., Li, H.J., Wen, C. (2017). Closed-loop hybrid gaze brain-machine interface based robotic arm control with augmented reality feedback. *Frontiers in Neurobotics*, 11: 60. <https://doi.org/10.3389/fnbot.2017.00060>

[7] Hochberg, L.R., Bacher, D., Jarosiewicz, B., Masse, N.Y., Simeral, J.D., Vogel, J., Haddadin, S., Liu, J., Cash, S.S., van der Smagt, P., Donoghue, J.P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398): 372-375. <https://doi.org/10.1038/nature11076>

[8] Collinger, J.L., Wodlinger, B., Downey, J.E., Wang, W., Tyler-Kabara, E.C., Weber, D.J., McMorland, A.J.C., Velliste, M., Boninger, M.L., Schwartz, A.B. (2013). High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866): 557-564. [https://doi.org/10.1016/S0140-6736\(12\)61816-9](https://doi.org/10.1016/S0140-6736(12)61816-9)

[9] Fifer, M.S., Acharya, S., Benz, H.L., Mollazadeh, M., Crone, N.E., Thakor, N.V. (2012). Toward electrocorticographic control of a dexterous upper limb prosthesis: Building brain-machine interfaces. *IEEE Pulse*, 3(1): 38-42. <https://doi.org/10.1109/MPUL.2011.2175636>

[10] MILLÁN, J.D.R., Ferrez, P.W., Galán, F., Lew, E., Chavarriaga, R. (2008). Non-invasive brain-machine interaction. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(5): 959-972. <https://doi.org/10.1142/S0218001408006600>

[11] Meng, J., Zhang, S., Bekyo, A., Olsoe, J., Baxter, B., He, B. (2016). Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks. *Scientific Reports*, 6(1): 1-15. <https://doi.org/10.1038/srep38565>

[12] Chen, X., Zhao, B., Wang, Y., Gao, X. (2019). Combination of high-frequency SSVEP-based BCI and

- computer vision for controlling a robotic arm. *Journal of Neural Engineering*, 16(2): 026012. <https://doi.org/10.1088/1741-2552/aaf594>
- [13] Xu, Y., Ding, C., Shu, X., Gui, K., Bezsudnova, Y., Sheng, X., Zhang, D. (2019). Shared control of a robotic arm using non-invasive brain-computer interface and computer vision guidance. *Robotics and Autonomous Systems*, 115: 121-129. <https://doi.org/10.1016/j.robot.2019.02.014>
- [14] Zhang, W., Sun, F., Liu, C., Su, W., Tan, C., Liu, S. (2017). A hybrid EEG-based BCI for robot grasp controlling. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3278-3283. <https://doi.org/10.1109/SMC.2017.8123134>
- [15] Tang, J., Zhou, Z. (2017). A shared-control based BCI system: For a robotic arm control. In 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), pp. 1-5. <https://doi.org/10.1109/EIIS.2017.8298767>
- [16] Chong, N.Y., Tanie, K. (2003). Object directive manipulation through RFID. *Proceedings of the Korean Society for Control and Robot Systems Conference*, pp. 2731-2736.
- [17] McMullen, D.P., Hotson, G., Katyal, K.D., Wester, B.A., Fifer, M.S., McGee, T.G., Harris, A., Johannes, M.S., Vogelstein, R.J., Ravitz, A.D., Anderson, W.S., Thakor, N.V., Crone, N.E. (2013). Demonstration of a semi-autonomous hybrid brain-machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4): 784-796. <https://doi.org/10.1109/TNSRE.2013.2294685>
- [18] Muelling, K., Venkatraman, A., Valois, J.S., Downey, J.E., Weiss, J., Javdani, S., Hebert, M., Schwartz, A.B., Collinger, J.L., Bagnell, J.A. (2015). Autonomy infused teleoperation with application to BCI manipulation. *arXiv preprint arXiv:1503.05451*.
- [19] Katyal, K.D., Johannes, M.S., Kellis, S., et al. (2014). A collaborative BCI approach to autonomous control of a prosthetic limb system. In 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1479-1482. <https://doi.org/10.1109/SMC.2014.6974124>
- [20] Noaman, M.N., Qasim, M., Ismael, O.Y. (2021). Landmarks exploration algorithm for mobile robot indoor localization using vision sensor. *Journal of Engineering Science and Technology*, 16(4): 3165-3184.
- [21] Alghanim, M.N., Qasim, M., Valavanis, K.P., Rutherford, M.J., Stefanovic, M. (2020). Passivity-based adaptive controller for dynamic self-leveling of a custom-built landing platform on top of a UGV. In 2020 28th Mediterranean Conference on Control and Automation (MED), pp. 458-464. <https://doi.org/10.1109/MED48518.2020.9182807>
- [22] Alghanim, M.N., Qasim, M., Valavanis, K.P., Rutherford, M.J., Stefanovic, M. (2020). Comparison of controller performance for UGV-landing platform self-leveling. In 2020 28th Mediterranean Conference on Control and Automation (MED), pp. 471-478. <https://doi.org/10.1109/MED48518.2020.9182837>
- [23] Ismael, O.Y., Qasim, M., Noaman, M.N. (2021). Equilibrium optimizer-based robust sliding mode control of magnetic levitation system. *Journal Européen des Systèmes Automatisés*, 54(1): 131-138. <https://doi.org/10.18280/jesa.540115>
- [24] Ismael, O.Y., Qasim, M., Noaman, M.N., Kurniawan, A. (2020). Salp swarm algorithm-based nonlinear robust control of magnetic levitation system using feedback linearization approach. In *Proceedings of the 3rd International Conference on Electronics, Communications and Control Engineering*, pp. 58-64. <https://doi.org/10.1145/3396730.3396734>
- [25] Spong, M.W., Hutchinson, S., Vidyasagar, M. (2020). *Robot Modeling and Control*. John Wiley & Sons.