IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 July 2022, accepted 23 August 2022, date of publication 1 September 2022, date of current version 5 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203734

RESEARCH ARTICLE

Visual Error Constraint Free Visual Servoing Using Novel Switched Part Jacobian Control

ZUBAIR ARIF[1], YILI FU[1], (Member, IEEE), MUHAMMAD KASHIF SIDDIQUI[2], AND FUHAI ZHANG[1]

[1]State Key Laboratory of Robotics and Systems, School of Mechatronics, Harbin Institute of Technology, Harbin 150001, China
[2]Department of Fluid Flow Control and Automation, School of Mechatronics, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Yili Fu (meylfu@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62073097.

ABSTRACT This research advances the state of the image-based visual servoing (IBVS) of robotic arms to handle very large visual errors without the camera advance/retreat problem. Conventional visual servoing schemes either consist of a partitioned or a switched system that relies on the feature Jacobian to find a unique feature for partitioning control along the specific DoF. We suggest a new IBVS scheme based on part-manipulator Jacobian approach for building a hybrid switched-partitioned task jacobian without the need to define new features. Utilizing this computationally efficient, directly defined Part-manipulator Jacobian an efficient second order minimization(ESM) based adaptive switching controller was constructed. The proposed scheme was tested in the eye-in-hand configuration on a 6-DoF simulated robotic arm and a 7-DoF real robotic arm for a set of large visual errors between the initial and the desired frames, including a rotational error of 180° around the camera optical axis. Compared to other IBVS schemes under various simulation conditions, the performance of the proposed scheme remained superior to that of the Jacobian-pseudo-inverse and other ESM-based IBVS schemes. The experimental results showed a notable expansion of the convergence zone up to 180° rotational errors with a 40% improvement in the convergence rates with a significant 90% reduction in the joint velocities and joint energies required to complete the task. The proposed controller possesses no camera advance/retreat motion, has a task Jacobian matrix that is well-conditioned, and it is computationally efficient. Moreover, the method is independent of the robot's DoF and is extendable to other visual servoing schemes.

INDEX TERMS Camera advance/retreat problem, image-based-visual servoing (IBVS), large orientation error, partitioned Jacobian, switching control, vision-based control.

I. INTRODUCTION

Robotics engineering is growing at a rapid pace, everyday new and better systems are replacing the older ones. With the advancement in high-speed computing and image processing techniques, it is no more a miracle for a robot to see. Recently robots have left the industrial setups and have entered into the dynamic, unstructured environment of our homes and hospitals [1], [2] which were once, only considered fit for humans. However, this evolution has created new challenges for the research community, because robots were made to operate in a structured environment. In the dynamic environment of a home or hospital, one cannot guarantee the robot to follow straight paths and find objects always placed in a certain place or pose.

Therefore, robots need to develop a vision to perceive and interact with their environment. Visual servoing is a branch of engineering encompassing robotics, vision, and control engineering. It enables the robots to see, perceive, and interact with their environment by moving the end effector using visual feedback while extracting feature information using the camera [3]. The recent progress of robots moving in dynamic environments has created new challenges in the field of visual servoing.

Traditionally visual servo control schemes are primarily divided into two different approaches, one that realizes visual

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

control using pose information in the 3D operational space, called Pose-Based Visual Servo (PBVS) [4], and another that realizes the visual servo control using pure image features in the 2D image space, referred to as Image-Based Visual Servoing (IBVS). IBVS utilizes the 2D image features directly to derive the feature error zero in the image plane but provides no control over the 3D trajectory of the robotic arm [3]. Conversely, PBVS uses 3D information to compute the pose of the end effector and then moves towards the target goal pose. PBVS seems a more obvious choice for robotic end-effector control.

Nevertheless, PBVS schemes have no control over the image trajectory of features and the convergence depends directly on the pose estimation accuracy and camera calibration parameters, also target may leave the camera's field of vision, and pose estimation efficiency needs to be high which is difficult to ensure in unstructured environments. Hence it is not optimal to use PBVS for dynamic unstructured environments.

On the other hand, IBVS is closer to the human sense of vision-based motion, as it drives the visual error to zero, just by seeing the object image. However, this technique also has the drawback that it cannot handle substantial initial feature errors, particularly large orientation errors around the optical axis that result in abnormal camera advance or retreat motion [5]. This advance or retreat motion is also unacceptable in robotics for instance assistive and collaborative robotics [6]. A comparison of visual servoing schemes and their shortcomings is discussed in [5]. The current state of vision-based control can be seen in [7].

In general, visual servo control methods are more computationally intensive than open-loop 'look and move' methods [8], [9]. Look and move systems uses pose estimation techniques to acquire the desired pose that is achieved utilizing joint feedback from the manipulator [10]. Many existing applications are designed using open-loop look and move methods [11], [12]. However, using an open-loop look and move method to position a robotic arm for probabilistic grasping carries its shortcomings, for instance, measurements are made in an open-loop manner, hence the system becomes sensitive to uncertainties, such as the lack of positional accuracy of the robotic manipulator due to errors in the kinematic model of the robotic arm, internal errors like wear, backlash and, other external reason may imply such as weak camera calibration. Consequently, the reliability and precision of open-loop look-and-move systems remain lesser than visual servo control systems [9]. Still, visual servo control has not found much acceptance in real-world applications due to its computationally intensive nature and other associated problems [13]. Therefore, any solution to the existing problems in IBVS schemes should not add unnecessary complexity and computational load on top of the visual servo control algorithms and the desired solution should be simple yet, efficient.

In this paper we propose a novel hybrid-switched–partitioned manipulator Jacobian-based visual servoing scheme that is free from camera advance/ retreat problems and offers a wider convergence range with an improved quadratic convergence rate based on an efficient second-order minimization method. The proposed method does not need to define new features in the image plane and also the computational cost of the loop does not increase. The proposed solution is effective, efficient, and invariant to large initial target pose errors, especially around the camera optical axis, which can create camera advance/retreat motion during the conventional IBVS control.

A. IBVS SCHEMES DEALING WITH LARGE ORIENTATION ERROR CONSTRAINT

The difficulty encountered by IBVS schemes in the presence of significant orientation error, particularly around the camera optical axis, was identified earlier in [14]. The camera advance/retreat problem in image-based visual servoing appears when Cartesian coordinates are used to form the image interaction matrix in presence of a large initial orientation error, especially in the pure rotation cases around the camera optical axis.

Standard control laws tend to produce a straight-line feature trajectory in the image plane. Trying to reduce the rotational effect by increasing the translational motion and straightening the image feature trajectory, it simultaneously rotates and advances or retreats along the optical axis. This advance or retreat motion of the camera depends upon the choice of interaction matrix i.e. camera will advance if the feature jacobian is formed using current image feature values (varying jacobian method) or it will retreat if the desired features (constant jacobian) were used in the formation of image jacobian. In fact, both the current and desired feature jacobian methods do not correctly approximate the Hessian of image jacobian which modifies the direction of the steepest descent resulting in a misleading direction of the displacement of the camera, which induces a backward motion (the retreat problem) or a forward motion (the advance problem) of the camera. Mathematically this happens due to the coupling between the third and the sixth column of the feature jacobian matrix in the pure rotation cases around the optical axis. The feature jacobian becomes singular while losing its degree of freedom. Detailed mathematical explanations of this problem can be found in [15]. A severe case known as the Chaumette Conundrum [14] appears when the desired camera pose requires a pi rad rotation around the camera's optical axis and the camera advances or retreats infinitely as a curved motion can only appear as a straight line when viewed from the infinity.

The effects of camera advance/retreat become noticeable around 45-degree rotation and get worsen in the visual task requiring 90 degrees or more rotation around the optical axis. It is a major problem in deploying IBVS schemes in an unstructured environment where the orientation of the object cannot be ensured within a specified range. For instance, for use with autonomous assistive robotic arms [16]

providing care to patients or people with disabilities sitting on a wheelchair [17].

Soon after the establishment of this problem, researchers started proposing solutions to handle this problem. Schemes that are free from camera advance /retreat motion and can handle large rotation errors. One of the first approaches called 2-1/2 D [18] used hybrid features by combining the point image feature and the logarithm of its depth, as a derived feature. Another partitioning scheme was developed in which image feature Jacobian was partitioned along the specific DOF for treating rotational and translational errors separately while using a derived feature of an area of a polygon, defined as a feature in the image plane [19]. Thereafter a more stable switched hybrid visual servoing control scheme was presented in which IBVS and PBVS schemes were used side by side, using a switching controller while monitoring the system stability through the Lyapunov function and continuously switching from one scheme to another according to the system stability criteria [20], [21].

Considering that the camera advance retreat problem occurs in IBVS due to the use of Cartesian coordinates in the interaction matrix, then the natural choice is to build a visual servoing scheme using the cylindrical coordinates. IBVS schemes based upon the cylindrical coordinate system are free from camera advance retreat problems [22]. However, its curvilinear image trajectory for pure translation cases and its decoupling behavior in complex roto-translational motion was unsatisfactory, which is desired in many cases. Advanced polar IBVS schemes use shift-able origins [23] which provides a solution to such inherent problems with cylindrical IBVS schemes but at a higher computational cost and requires the computation of weights at each step.

For avoiding the inherent problem of IBVS schemes, the benefits of combining polar and Cartesian coordinates in the formation of feature jacobian were reported in [24]. However, it increases the size of the interaction matrix and the desired decoupling behavior was also not remarkable for the translational and rotational motion. Therefore, it was also not a feasible solution.

Later [25] designed an IBVS switching scheme between the cylindrical and Cartesian coordinates to form an interaction matrix using the homography of the current and the reference image, where the switching criteria was the rotational angle between the current and desired frame. The scheme combines the advantages and shortcomings of the partitioned and the polar IBVS systems. The author showed simulation results that performed well for large orientation errors, but the problem of ill-condition image jacobian matrix persists and no improvements was reported in the convergence rate.

Recently other methods for addressing the camera advance retreat problem were proposed in [26] and [27]. In these schemes, a three-stage, adaptive switching controller was proposed which breaks down the visual control into three stages i.e. rotational, translational, and fine motion by locking the rotational and translational joints during the servo control. It is a sub-optimal technique that requires too much switching, also it uses a derived feature of an angle defined in the image plane, and hence suffers from the same drawbacks of the conventional partitioned controllers.

Another recent work proposed an enhanced IBVS controller with an adaptive weighted mean of the Jacobian (AWMJ) [28]. It uses a weighted formation of the feature interaction matrix between the current and the desired image, switching depending upon the sigmoid trend of the image error norm. In this control scheme, three new parameters were calculated and multiplied in the control law, these weights depend upon the sigmoid trend of the global visual feature error. Despite adding 3 new parameters which need to be calculated and tuned beforehand, the controller was unable to handle orientation error beyond 165 degrees without noticeable camera advance retreat motion. Cases with pure rotation of pi rad around the optical axis were not discussed.

As the camera advance or retreats depends upon the choice of forming an interaction matrix from the current or the desired image features. Hence, the efficient second-order minimization (ESM) approaches came into play which utilizes the mean of the current and the desired features for forming an image interaction matrix using the pseudo-inverse of the mean of jacobians (PMJ) or the mean of the pseudo inverse of the feature jacobian (MJP) [15]. This scheme remained quite successful in avoiding the camera advance/retreat problem. Moreover, it provides quadratic convergence rates for the visual servoing task. Our proposed scheme is also based on second-order minimization. Therefore, we shall discuss the formation of the ESM-based visual servoing scheme in Section II.

Although, the schemes discussed in this section are well established and have proven benefits in certain environments. However, we still believe that the problem is not solved optimally for utilizing in a real-world scenario for robots moving in a dynamic, unstructured environment as the previously proposed solutions were not optimal as it may require the use of additional features and its convergence depends on the pose estimation while adding computational load by stacking up the polar and Cartesian image jacobians.

An optimum solution for the problem does not increase the computational load by adding features or stacking up to feature jacobian matrix by combining polar and Cartesian features and that does not require a 3D model of an object for making feature jacobian. An optimal solution cannot switch to PBVS as again it uses pose information of the object and the convergence also directly depends on the pose estimation accuracy and the camera calibration. Partitioned techniques utilize an additional derived feature of the area of the polygon defined by the points and an angle of the line between them in the image plane [29], but it has a known shortcoming that if the target is initially skewed at an angle, the features cannot be extracted correctly such as the area of a polygon.

All of these conditions cannot be assured in an unstructured environment where the robotic arms have to move, for instance, an assistive robotic mounted on a wheelchair assists a person with a disability in the home environment [17], [30].

In quest of a solution, we consulted human behavior when performing a visual task with substantial orientation errors. When humans undertake tasks that entail a large initial pose error, especially in the rotational axis, we observed that most persons first neutralize the error in the rotational axis before fixing the translational error. Humans attempt to rotate the image points to generate a close sync between corresponding points; once they develop the correspondence in the close neighborhood of the desired features, they begin to minimize the translational and rotational error simultaneously, resulting in zero error between the initial and desired frames. Human activity may be divided into two basic phases: a rotary phase first, followed by a mixed translation-cum-rotational motion.

Therefore, we proposed our rotation first strategy based on second-order minimization of the switched-partitioned manipulator Jacobian approach by using the rotation first strategy inspired by humans for solving visual tasks involving large initial orientation errors. The effectiveness of the proposed scheme will be demonstrated in the results section IV that does not suffer from camera Advance/retreat problems or singularities even for errors of 90 and 180 degrees about the optical axis. We shall achieve this goal without adding a new feature, rather we propose a simple yet effective solution that is computationally inexpensive to perform the visual task.

The rest of this paper is organized as follows, Section II revisits the design of an efficient second-order image-based visual servo control scheme and Section III describes the design of the proposed adaptive switched controller that utilizes our novel direct partitioned manipulator Jacobian. Experimental results are presented in Section IV on 6 DOF and 7 DOF robotic arms for very large initial rotational errors involving 90° and 180° rotations about the camera optical axis. In Section V, a valuable discussion will be made on the obtained results and the computational cost analysis will be made. Section VI concludes the paper by providing major highlights of the proposed scheme while providing future work direction.

II. IMAGE-BASED CONTROL USING EFFICIENT SECOND-ORDER MINIMIZATION TECHNIQUE

Consider an 'n' DoF robotic arm with a camera mounted on its end effector in an eye-in-hand configuration as observing a motionless target as shown in Fig. 2. The camera frame has a velocity $\mathbf{V_c}(v_c, \omega_c)$ where v_c is the translational velocity and ω_c is the angular velocity of the camera frame. Generally, the IBVS schemes utilize '$k > 3$' number of 2D features such as point, line [31], [32], etc. in the image plane and derive the visual feature error to zero in the image plane. The visual feature error '$\mathbf{e}$' is the difference between the current feature values '$\mathbf{s}$(t)' and the desired feature values '$\mathbf{s}^*$' in the image plane, given by

$$\mathbf{e}\,(\mathrm{t}) = \mathbf{s} - \mathbf{s}^* \tag{1}$$

For a stationary target, the relationship between, the time variation of the visual feature $\dot{\mathbf{s}}$, and the camera velocity screw $\mathbf{V_c}$ is given by

$$\dot{\mathbf{s}} = \mathbf{L_s}.\mathbf{V_c} \tag{2}$$

where $\mathbf{L_s}$ is the image interaction matrix often also known as feature jacobian, $\mathbf{L_s} = k \times 6$, '$k$' is the number of features, n is the number of robot joints.

The analytical form of this feature jacobian for instance for a point feature can be given by [3]

$$\mathbf{L_s} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{3}$$

As $\mathbf{L_s}$ comprises of the 2D coordinate of the object (x, y) in the image plane (which depends on the camera intrinsic parameters) and depth Z of the feature points that need to be either estimated, or approximated during the visual servo loop. There are several methods available in the literature for estimating the current features and depth of feature points such as [33], [34].

The task of an IBVS system is to derive the current visual feature to the desired ones by an exponential decoupled decrease of the visual error to zero. To achieve this task a conventional proportional IBVS controller [3] can be derived using a positive decay factor 'λ'

$$\mathbf{V_c} = -\lambda \widehat{\mathbf{L}_s}^{+} \boldsymbol{e} \tag{4}$$

where $\widehat{\mathbf{L}_s^{+}} \in \mathrm{R}^{(6\times k)}$ is chosen as the left Moore–Penrose pseudo-inverse of the approximation of $\mathbf{L_s}$ such that $\widehat{\mathbf{L}_s^{+}} = \left(\mathbf{L_s^T L_s}\right)^{-1} \mathbf{L_s^T}$, when $\mathbf{L_s}$ is of full rank.

In (4) $\widehat{\mathbf{L}_s^{+}}$ is the approximation of $\mathbf{L_s}$. Three popular choices are available in the literature [3], [33] for the approximation or estimation of the image interaction matrix which includes forming the feature jacobian matrix using the current image features values such that $\widehat{\mathbf{L_s}} = \mathbf{L_s}\,(f_c, Z_{est})$ i.e. by using the estimated depth and the current values of the visual features in the image plane.

Secondly by using the constant desired features for approximating the feature jacobian such that $\widehat{\mathbf{L}_s} = \mathbf{L}_{s^*}(f_d, Z_d)$ while utilizing the fixed desired feature depth and constant desired feature values of the reference frame. Thirdly, a mean of the desired and the current feature jacobian as $\widehat{\mathbf{L_s}} = (\mathbf{L_s} + \mathbf{L}_{s*})/2$, that is utilized in the efficient second-order minimization-based visual servoing techniques.

The control in (4) is called Jacobian Pseudo Inverse (JPC) control. JPC method ensures that $\mathbf{L_s}$ is invertible and calculates minimum norm camera velocity as input to the visual servo controller. JPC method has its shortcomings, it has a low convergence rate, particularly for large inter-frame movements. Moreover, it has a narrow convergence zone and cannot handle large orientation errors specifically around the camera optical axis. When encountering large initial orientation errors JPC IBVS controller produces undesirable camera advance or retreat motions as discussed in the Introduction section.

A. IBVS-BASED ON EFFICIENT SECOND-ORDER MINIMIZATION (ESM)

To address this problem of narrow convergence zone, camera advance/ retreat motion and to increase the convergence rates with improved stability, [15] proposed utilization of efficient second-order (ESM) minimization techniques for IBVS, where second-order Taylor series expansion of the visual features was utilized for estimating the camera displacement, while only taking the first derivative.

ESM IBVS technique utilizes $\widehat{\mathbf{L}_\mathbf{s}} = (\mathbf{L_s} + \mathbf{L_{s*}})/2$, the mean of the current and the desired image jacobian for approximating the feature jacobian which provides quadratic convergence and wider convergence zones without experiencing the camera advance/retreat problem, showed that this ESM technique is superior to other conventional control methods. Two methods were developed under the ESM scheme namely the Pseudo-Inverse of the mean of the feature jacobian (PMJ) and the mean of the pseudo inverse of the feature Jacobian (MJP), detailed mathematical derivation of the ESM-based IBVS can be found in [15] and [35]. We recall from ESM-based IBVS formulation [35] (18, 19) that defines

$$\dot{\boldsymbol{q}} = -\frac{1}{2}\lambda(\mathbf{L}_\mathbf{s}^{+} + \mathbf{L}_{\mathbf{s}^*}^{+})(\boldsymbol{s} - \mathbf{s}^*) \tag{5}$$

$$\dot{\boldsymbol{q}} = -2\lambda\,(\mathbf{L_s} + \mathbf{L_{s^*}})^{+}\,(\boldsymbol{s} - \boldsymbol{s^*}) \tag{6}$$

It was proved that the PMJ method is superior to JPC and even MJP ESM methods [15]. Therefore, we shall use the PMJ method to build our proposed scheme. However, both ESM techniques developed were based on an assumption that the velocity required to move from the current frame 'c' to the desired 'd' image frame are equal but in the opposite directions. For the MJP method, the assumption was taken as $v_{cd}.dt \approx -v_{dc}.dt$, similarly, for the PMJ method the error between the current and desired image as $e_{cd} \approx -e_{dc}$ was considered.

This assumption was not true globally and it holds only in a narrow region i.e. it holds correctly only for pure translational motion but when complex roto-translational coupled motion is encountered, this assumption fails especially in the presence of a large orientation error around the camera optical axis and a singularity arise at a rotation of 90 Degrees and problems of local minima around 120 Degrees rotation were reported and a modified ESM technique [35] was proposed by modifying (5), (6).

$$\dot{\boldsymbol{q}} = -\frac{1}{2}\lambda\left(\mathbf{L}_\mathbf{s}^{+} + \mathbf{T}\mathbf{L}^{+}{}_{\mathbf{s}^*}\right)(\boldsymbol{e}) \tag{7}$$

$$\dot{q} = -2\lambda\left(\mathbf{L_s} + \mathbf{L_{s^*}}\mathbf{T}^{-1}\right)^{+}(\boldsymbol{e}) \tag{8}$$

In (7), (8) the modified ESM method requires multiplying a transformation matrix $\mathbf{T}$ to transform motion from the desired frame to the current frame. Where $\mathbf{T}$ is a block diagonal matrix given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_d^c & [t_d^c] \times \mathbf{R}_d^c \\ 0_{3\times3} & \mathbf{R}_d^c \end{bmatrix}_{6\times6} \tag{9}$$

$\mathbf{R}_d^c$ is a 3×3 rotational matrix and $[t_d^c]$ is the skew-symmetric matrix of the translation, from the desired to the current frame. This $\mathbf{T}$ can be computed from the homogeneous transformation matrix that can be estimated using different methods available in the literature [34]. However, multiplying a (6×6) transformation matrix with the $(k\times6)$ desired feature jacobian at each iteration is a computationally intensive task that adds unnecessary complexity to the controller. Where we shall see in the experiments that the MESM method will not always necessarily performs better than the ESM IBVS controller.

B. JOINT SPACE ESM-BASED IBVS CONTROLLER

As the camera is mounted on the robotic arm's end effector that is manipulated in the joint space through a vector of joint velocities. Therefore, the features velocity is also connected with joint velocity, (2) can be rearranged in the joint space configuration given by

$$\dot{\mathbf{s}} = \mathbf{J_s}.\dot{\boldsymbol{q}} \tag{10}$$

where $\mathbf{J_s}$ is the task image Jacobian given by

$$\mathbf{J_s} = \mathbf{L_s}.\mathbf{V_e^c}.\mathbf{J_e^e} \tag{11}$$

Task image Jacobian $\mathbf{J_s}$ is a combination of feature Jacobian or interaction matrix $\mathbf{L_s}$, the end effector manipulator Jacobian $\mathbf{J_e^e}$in the end effector frame and the spatial motion transformation matrix $\mathbf{V_e^c}$ which transforms velocity from the end effector to the camera frame. Indeed the term $\mathbf{V_e^c}.\mathbf{J_e^e}$ refers to the manipulator inverse kinematics expressed in the camera frame [35].

Now, we can devise a standard proportional joint controller for this IBVS system to derive the feature error to zero

$$\dot{\boldsymbol{q}} = -\lambda\mathbf{J}_\mathbf{s}^{+}\boldsymbol{e} \tag{12}$$

where $\dot{\boldsymbol{q}}$ is the robotic arm joint velocity vector and λ is a positive gain scaler in the control law for convergence of error vector $\boldsymbol{e}$ to zero.

As discussed for (4), (12) is a similar JPC controller in the joint space that has the same associated limitations of slower convergence rates and narrower convergence zones with known camera advance/retreat problems. Therefore, the joint IBVS controller of (12) can be improved using an efficient second-order PMJ IBVS controller by using the pseudo-inverse of the mean of the current and the desired image jacobian by combining (6) with (12) we get,

$$\dot{\boldsymbol{q}} = -2\lambda(\mathbf{J_s} + \mathbf{J_{s^*}})^{+}.\boldsymbol{e} \tag{13}$$

where $\mathbf{J_s} = \mathbf{L_s}.\mathbf{V_e^c}.\mathbf{J_e^e}$ and $\mathbf{J_{s^*}} = \mathbf{L_{s^*}}.\mathbf{V_e^c}.\mathbf{J_e^e}$, which implies

$$\dot{\boldsymbol{q}} = -2\lambda[\mathbf{L_s}.\mathbf{V_e^c}.\mathbf{J_e^e} + \mathbf{L_{s^*}}.\mathbf{V_e^c}.\mathbf{J_e^e}]^{+}.\boldsymbol{e} \tag{14}$$

$$\dot{\boldsymbol{q}} = -2\lambda\left[(\mathbf{L_s} + \mathbf{L_{s^*}})\,\mathbf{V_e^c}.\mathbf{J_e^e}\right]^{+}.\boldsymbol{e} \tag{15}$$

As (15) is an ESM-based IBVS PMJ joint controller for an eye-in-hand system. This controller is better than (12) which is a JPC controller. However, (15) still has shortcomings that

it cannot handle large initial orientation error around the camera's optical axis, as indicated by [35] who proposed to use a modified form of this controller to get rid of that undesired behavior of local minima and instability at 90-degree optical axis rotation case. The proposed solution by the MESM method in our case can be given by

$$\dot{\boldsymbol{q}} = -2\lambda \left[(\mathbf{L_s} + \mathbf{L_{s^*}}\mathbf{T^{-1}})\mathbf{V_e^c}.\mathbf{J_e^e} \right]^{+} \boldsymbol{e} \tag{16}$$

In Section III, we shall present a different solution than (16) while making use of our proposed switched-part Jacobian (SPJ) scheme based on the efficient second-order minimization (ESM) using the Pseudo-Inverse of the mean of the image Jacobian (PMJ) method. The proposed scheme will remain effective throughout the visual servo control loop, unlike the ESM and modified MESM methods which are not effective over pi rad rotation errors around the camera optical axis. The proposed method will avoid singularity arising at pi rad and pi/2 rad, and can also avoid the local minima problem of 120 degrees as was the case in the JPC controller. The proposed scheme does not require multiplying a 6×6 transformation matrix $\mathbf{T}$ at each iteration in the control law to avoid the camera advance/retreat problem as proposed in the MESM technique.

Our proposed SPJESM scheme will use a novel part manipulator Jacobian and a switching controller to switch between two control schemes providing an increased convergence zone and an improved convergence rate, highlights of our proposed schemes are;

1) It will avoid camera advance/retreat problems at very large orientation errors even for a classical case of pi rads rotation around the optical axis.
2) It will avoid singularity appearing in ESM methods about pi/2 rad due to the singular configurations of the image jacobian.
3) As compared to conventional IBVS schemes, it will increase the convergence rates by at least quadratic for the visual servoing task.
4) It can effectively handle cases including pure translational, pure rotational, and complex Roto-translational motion even for objects placed on a tilted plane.
5) It is computationally efficient by using a directly defined part manipulator Jacobian in the control law.
6) It is energy efficient as it will consume lower Joint energy to perform the task.
7) The task jacobian will remain well-conditioned during the task.
8) The control law will be able to cater to the angular velocity limits of the robot's joints by directly controlling the magnitude of the joint velocities.

III. DESIGN OF AN IBVS SCHEME USING ADAPTIVE SWITCHED-PART MANIPULATOR JACOBIAN

A. PART MANIPULATOR JACOBIAN

At the core of our switching scheme resides a novel concept of using a part end effector Jacobian, directly defined once by the user at the design stage of the IBVS system.

Starting with an ESM-based IBVS PMJ joint controller as developed in the previous section given by

$$\dot{\boldsymbol{q}} = -\lambda \mathbf{J_{sm}^{+}} \boldsymbol{e} \tag{17}$$

where $\mathbf{J_{sm}}$ is given by

$$\mathbf{J_{sm}} = (\mathbf{L_s} + \mathbf{L_{s^*}}).\mathbf{V_e^c}.\mathbf{J_e^e} \tag{18}$$

To design a part jacobian controller, we need to modify the image Jacobian $\mathbf{J_{sm}}$. The Image Jacobian matrix is composed of the current and the desired feature jacobian or image interaction matrix $\mathbf{L_s} + \mathbf{L_{s^*}}$, manipulator Jacobian $\mathbf{J_e^e}$ expressed in the end effector frame and the velocity transformation matrix $\mathbf{V_e^c}$ between the camera and the end-effector frame. Conventionally, partitioning schemes were focused on partitioning the feature Jacobian or interaction matrix by partitioning the feature jacobian $\mathbf{L_s}$ in the rotational and translational DoF elements by choosing specific features that deal with a single degree of freedom.

Taking a fresh approach, we propose to partition the control using the end effector Jacobian for which we propose a new directly defined part end effector Jacobian. The idea derives from the fact that the camera advance/retreat problem is associated with the rotation about the optical axis or roll angle only, therefore we only need the z-axis rotation component of the Jacobian to operate the roll axis of the robotic arm around the camera optical axis.

As the camera advance/retreat motion is undesirable, therefore, the translational velocity components associated with the x, y, and z-axis are not needed until the controller first neutralizes the orientation error about the z-axis. For achieving this, roll velocity in the camera-mounted joint is needed only. Once the orientation error about the camera optical axis gets neutralized, which was actually causing the camera advance/retreat motion. The controller switches to the second stage, where translational and orientation errors in all axis are simultaneously minimized while utilizing all the joints of the robotic arm.

Actually, while using a partitioned roll first control for controlling the camera joint motion, we do not need to compute the full manipulator Jacobian in the end effector frame for the robotic arm. For controlling the orientation of the end effector, we only need to control the roll and pitch axis which can effectively minimize the orientation error as only two-axis (roll and pitch) joints are enough to minimize the 3D orientation error. Therefore, for neutralizing the orientation error only, instead of computing the full $6 \times n$ jacobian matrix, we only need the last elements of the manipulator Jacobian matrix.

Consider first only the roll joint part jacobian case for partitioning the control along the roll axis and other axes of the robotic arm. We propose a new directly defined part manipulator Jacobian for the robotic arm in the end-effector

frame given by

$$\mathbf{J_p^e} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha \end{bmatrix}_{6\times n} \tag{19}$$

where $\alpha \in \mathcal{R}$ and 'n' is the number of robot joints.

This directly defined part end effector Jacobian $\mathbf{J_p^e}$, by virtue of its last element α, which is a non-zero real number, will allow the robotic arm to minimize the orientation error between the current and desired image frames. In our case, α will move the last joint of the end effector carrying the camera and, resultantly move the camera mounted on the end effector to roll about the optical axis.

The value and position of α in the manipulator jacobian matrix are flexible and can be directly inducted by the user according to the kinematic structure of the robot and according to the joint carrying the camera in the eye-in-hand configuration. A general form of this directly defined part manipulator Jacobian can be given as

$$\mathbf{J_p^e} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha \end{bmatrix}_{6\times n} \tag{20}$$

where for a 6-DoF robotic arm we define manipulator Jacobian sensitivity parameters like α and other in (20), these parameters relate the feature error to the rotational velocity of the corresponding joint of the manipulator

$$\begin{aligned} a &= joint\ 1, \quad \gamma = joint\ 4,\ (yaw) \\ b &= joint\ 2, \quad \beta = joint\ 5,\ (pitch) \\ c &= joint\ 3, \quad \alpha = joint\ 6\,(roll) \end{aligned}$$

This directly defined end effector part-Jacobian $\mathbf{J_p^e}$ can be used in the visual servo control law developed in (18) for joint space configuration for defining the feature Jacobian for calculating visual servoing joint velocity using PMJ control. We can define a new part image jacobian $\mathbf{J_{sp}}$ using the part manipulator jacobian $\mathbf{J_p^e}$ given as

$$\mathbf{J_{sp}} = (\mathbf{L_s} + \mathbf{L_{s^*}}).\mathbf{V_e^c}.\mathbf{J_p^e} \tag{21}$$

This will ensure in the control loop that as long as (21) is used, only roll velocity will be calculated for the robotic arm corresponding to the current feature error. Please note, partitioning the manipulator Jacobian is different from partitioning the feature Jacobian $\mathbf{L_s}$, partitioning the manipulator Jacobian while keeping the image Jacobian intact will allow considering the whole feature error.

The direct induction of a partitioned manipulator Jacobian approach does not require assigning specific new image features related to the partitioned degree of freedom as required in [26] and [29]. It is computationally inexpensive to define a direct partitioned Jacobian matrix instead of using the conventional approach and defining full end effector manipulator Jacobian given by

$$\mathbf{J_e^e} = \begin{bmatrix} {}^{e}_{f}\mathrm{R} & 0_{3\times3} \\ 0_{3\times3} & {}^{e}_{f}\mathrm{R} \end{bmatrix}_{\mathbf{6\times6}} \cdot \left[\mathbf{J_e^f}\right]_{\mathbf{6\times n}} \tag{22}$$

While $\mathbf{J_e^f}(q)$ is a manipulator jacobian in the base frame and a function of forward-kinematics of the robotic arm $f(q)$. $\mathbf{J_e^f}(q)$ that increases with the number of joints of the robot which adds computational complexity and load on the task proportionally with an increase in DOF. A comparison of the computational load of various IBVS schemes will be presented in Table 10.

This proposed new Jacobian also has some additional desirable properties in the control loop, such as joint velocities can be accommodated to joint velocities limits by controlling the value of the joint sensitivity values i.e. α, β etc. values in (20) for instance (1 normal, 0.5 fast, 1.5 slow). This will be further discussed in the experiment Section IV.

A better joint configuration of the manipulator can also be achieved in the null space of the visual servo task during the proposed control, as the first stage uses only one or two wrist joints for the visual servo control. Meanwhile, other joints can be used in the null space motion to improve the posture of the robotic arm to achieve the target pose avoiding joint limits, singularities, and self-collision, using developments made in [36] and [37].

B. ADAPTIVE GAIN CONTROL

The basic IBVS proportional JPC controller (4) suffers from the problem of residual steady-state error and occasionally attracts towards local minima where joint velocities become zero while the feature error is not zero. This behavior has been improved by adding an SPJ-ESM-based PMJ controller which increases its convergence rate to at least quadratic and it also expands the convergence zone to include large orientation errors without camera advance/ retreat motion. However, the problem of the slow response of the controller near the convergence zone and attraction towards the local minima still needs to be addressed.

These problems can be improved using an integral part of the control law. However, adding an integral component in the control law will make the controller sluggish [38] to handle as the control response changes near the convergence zone where velocity components change their signs rapidly. Therefore, we shall add an adaptive gain for tackling the slow response problem in the convergence zone, making the controller adaptive so it can converge faster while remaining responsive in the convergence zone.

For designing an adaptive proportional gain controller, we will utilize the inherent infinity norm of the visual task. We shall choose two gain values tuned for the two peaks of the visual task. First, one for the case when feature error is very large or near infinity when camera velocities are higher. Another case is near the convergence zone where the camera

velocity is very small. The gain values changes according to the slope of the line defined at an angle to the zero planes. The values of gain will change adaptively to the changes in the feature error at each iteration of the control law well adjusting to the need of the controller to produce viable smooth velocity changes throughout the visual servoing task, unlike the constant gain cases where the velocities are abrupt and high in the beginning and decay rapidly as the controller approaches the convergence zones.

For ensuring an adaptive gain response in the controller, we can replace the constant gain term in (17) with the adaptive gain based on the infinity norm of the feature error [39]. The proportional controller in (17) can be updated as an adaptive gain SPJ PMJ Controller given by

$$\dot{\boldsymbol{q}} = -2\lambda_{adp}\mathbf{J}_{\mathbf{sp}}^{+}\boldsymbol{e} \tag{23}$$

where λ_{adp} is the adaptive proportional gain is given by,

$$\lambda_{adp}(x) = (\lambda_0 - \lambda_\infty)\, e^{-\frac{\lambda_0'}{\lambda_0 - \lambda_\infty}x} + \lambda_\infty \tag{24}$$

where, $x = \|e\|_\infty$ is the infinity norm of the task, $\lambda_0 = \lambda(0)$ is the gain in 0 for very small values of $\|e\|$, λ_0' is the slope of λ at $\|e\| = 0$, $\lambda_\infty = \lambda_{\|e\| \to \infty}$, $\lambda(\|e\|)$, is the gain to infinity, that is for very large values of $\|e\|$.

C. SWITCHED PART JACOBIAN ADAPTIVE GAIN IBVS SCHEME

A novel hybrid switched-partitioned system approach is presented in this section. The controller performs visual servo control while partitioning along specific dimensions of the state space using a hybrid switched partitioned manipulator Jacobian approach. A hybrid switched control law system as proposed in this paper can be expressed using a differential equation as

$$\dot{\mathbf{q}} = f_{\sigma(t)}\left(q, R_c^d, e(t)\right) : \sigma \in \{1 \ldots .n\} \tag{25}$$

where f_σ is a collection of distinct functions, $q(t)$ is the robotic arm joint angles, $e(t)$ is the visual feature state error between the desired and the current image features and R_c^d is the rotation matrix comprising the difference of orientation error between the current and the desired image frame. We considered σ to be a discrete signal switching state having a discrete values in $\{1 \ldots . \text{n}\}$. The value of σ at a given time t decides which function of $f_{\sigma(t)}\left(q, R_c^d, t\right)$ will control the system. The switch signal σ is a state-dependent variable that can cause switching by monitoring the state of the system by comparing it to a user-defined value ε. $\sigma\ (t)$ is the state of the controller according to the predefined threshold ε which is (in our case) the difference in the rotational angle around the optical axis between the current and the desired frame.

The switching controller can be designed starting from (17) with switching between ESM and SPJASM controller given as

$$\dot{\boldsymbol{q}}(t) = \begin{cases} -2\lambda_{adp}\left[\mathbf{J_{sm}}\right]^{+} e, & \sigma = 1 \; if \; R_d^c\,(roll < \varepsilon) \\ -2\lambda_{adp}\left[\mathbf{J_{sp}}\right]^{+} e, & \sigma = 2 \; otherwise \end{cases} \tag{26}$$

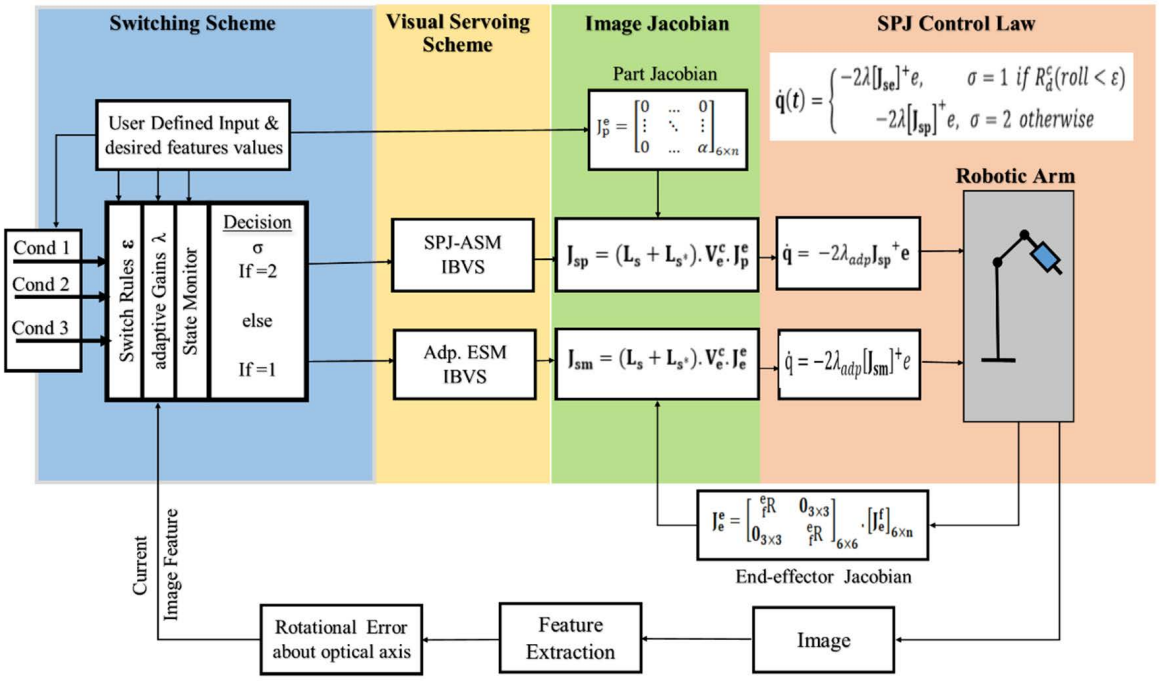

FIGURE 1. Flow diagram of the proposed switched part Jacobian control scheme.

where, $\mathbf{J_{sp}}$ is the new image jacobian formed using the part-manipulator Jacobian as developed in (21) and $\mathbf{J_{sm}}$ is given by (18)

$$\mathbf{J_{sm}} = (\mathrm{L_s} + \mathrm{L_{s^*}}).\mathbf{V_e^c}.\mathbf{J_e^e}$$
$$\mathbf{J_{sp}} = (\mathrm{L_s} + \mathrm{L_{s^*}}).\mathbf{V_e^c}.\mathbf{J_p^e}$$

And, $\mathbf{J_p^e}$ is the directly defined part end effector Jacobian using α from (22). The value of the switching threshold ε depends upon the user-defined roll angle threshold set for rotation first mode and the value of α *is* the sensitivity parameter that relates the feature error to the rotational velocity of the corresponding joint of the robotic arm. The value of switching state σ decides which controller is to be used while monitoring the difference of rotation angle between the current and the desired frame and comparing it to predefined roll first only threshold value ε. For the values of the orientation error which are greater than ε about the optical z-axis between the current and the desired frame, the SPJASM controller will be utilized as shown in Fig. 1.

The roll first mode (RFM) uses a SPJASM part-Jacobian IBVS controller to compute $\dot{\boldsymbol{q}}\,(t)$ and, for values below the threshold ε, an adaptive gain second order minimization (ASM) PMJ based IBVS joint controller will be used. To avoid the camera advance/retreat problem and handle large orientation errors with the proposed scheme, the value of the roll threshold ε is chosen such that it covers the maximum rotational error in the first stage of the control scheme. Once the orientation error is settled to an extent where it would not cause any camera advance retreat motion, the controller switches to the second stage where the ASMPMJ controller takes charge of a combined translational and rotational motion to minimize all the feature errors to zero.

Just like previous solutions dealing with significant rotational errors such as MESM [35] and partitioned methods like 3-stage adaptive switching [26]. Our proposed schemes also need knowledge of the object's pose with respect to (w.r.t.) the camera to solve this problem. However, our method only needs the difference in orientation between the object's initial (current) and desired pose relative to the camera frame. Particularly to evaluate the roll angle difference between the current and desired frame and compare it to a predefined roll angle threshold to switch the controller from the first stage (SPJASM) to the second stage (ASM). The controller evaluates the camera's pose using Partial pose estimation techniques, such as those used in simulation, which can be used to determine the object's pose w.r.t the camera by using a set of non-coplanar 2D coordinates [10] or by using fiducial markers like the April tag [40]. Other methods for determining an object's pose include [34], [41] which can be used to create advanced markerless model-based or model-free pose estimation.

One might categorize the proposed SPJASM controller as a 2-1/2 D visual servoing technique as it requires a mix of 2D and 3D pose information for the successful completion of the task. However, our proposed SPJASM controller only uses the orientation difference information for switching purposes and not for building the feature jacobian as is the case with 2-1/2 D and PBVS schemes which use a mix of 2D and 3D features to build the feature jacobian matrix. Our proposed switching control law only needs the difference of the roll angle between the current and the desired frame about the camera optical axis (usually the Z-axis). This roll angle difference update at each iteration and compared to the pre-defined threshold value of ε that is used for switching the controller. In the proposed scheme, unlike the 2-1/2-D and PBVS schemes, neither the convergence depends upon the pose estimation nor any 3D pose feature is utilized in the formation of the feature jacobian (except the current depth and IBVS schemes are known to be robust in handling depth estimation errors).

In our scheme, the pose estimation error can only affect the image trajectory taken by the first controller, as the use of pose information was only for switching the controller and the second loop is a pure ESM-IBVS law and does not utilize any 3D feature in building the control law. Therefore, our proposed control scheme is a 2D IBVS switching scheme where the controller is pure 2D-ESM based and only the switching depends upon the roll angle difference between the current and the desired frame. On the contrary, The PBVS and MESM control methods rely heavily on the correctness of the pose estimation method due to the use of pose information directly in the control law. The MESM controller needs a 3D pose to build the control law and thus its convergence depends upon the accuracy of the pose estimation process. The MESM controller requires 3D pose information to further process it to extract the rotation matrix between the current and the desired frame using Rodrigue's formula. Using the rotation matrix, it formulates a 6×6-transformation matrix and multiplies this matrix with a 2K x6 feature jacobian matrix at each iteration. That is a quite cumbersome, and computationally expensive process.

Therefore, our proposed method is a 2D IBVS switching scheme that is simple yet efficient in handling large rotational errors about the camera's optical axis while using minimal pose information indirectly only for switching the controller. Experimental Validation of our proposed schemes is presented in section IV.

IV. EXPERIMENTS AND RESULTS

To validate our proposed scheme experiments were conducted on a simulated and a real robotic arm for visual servoing tasks comprising large initial pose errors including pure rotation, pure translation, and complex roto-translational motion. As well as for objects placed on a tilted plane. Systematic experimentation was made to evaluate all aspects of the proposed controller under various rotational and translational errors and a detailed comparison is presented in this section. The experiment was divided into the following;

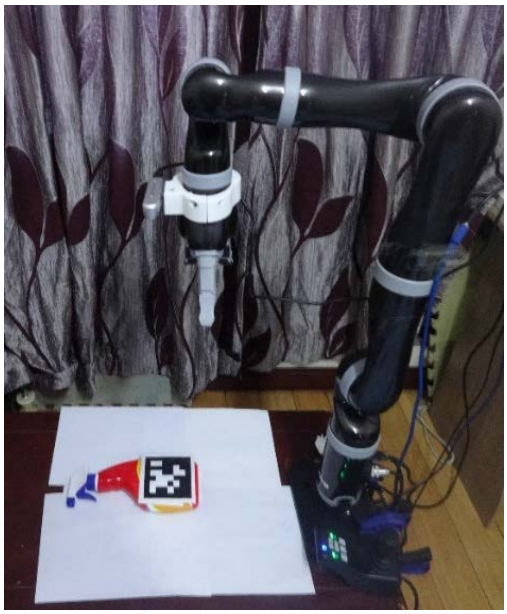

FIGURE 2. Experimental setup: Kinova robotic arm eye in hand configuration with an object carrying an April tag.

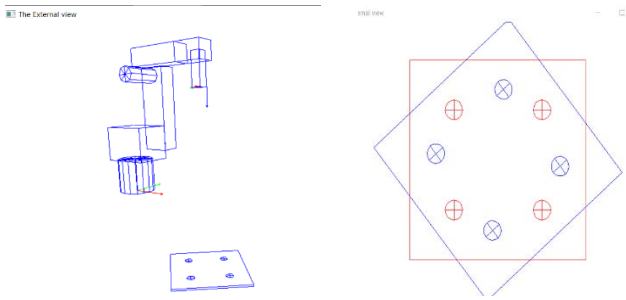

FIGURE 3. Simulation setup in VISP library: External camera view (left): a wireframe model of VIPER 850-6 DoF robotic arm and a square plate target with 4 points; Internal camera view (right): current frame (blue) and the desired frame (red).

1) Simulation of Viper850 Robotic-arm under a large rotational error of 180° degrees around the camera's optical axis.
2) Simulation of Viper850 Robotic-arm under a large rotational error of 140° degrees around the optical axis.
3) Comparison of the proposed SPJASM and the conventional controllers like ESM, MESM, and AWPJ controllers in terms of consumed joint energies, the condition number of the task Jacobian and the maximum joint velocities required for successful task completion.
4) Evaluation of the effect of the roll angle threshold ε and $\propto$ sensitivity value on the performance of the proposed SPJASM controller
5) Evaluation of the proposed controller under pure rotational, pure translational, and complex roto translational motion.
6) Evaluation of the proposed controller for objects placed on a tilted plane.
7) Evaluation of the proposed controller on a real- 7 DoF Kinova JACO-2 robotic arm.

Experiments have been carried out in simulation on a model of Viper 850 6-DOF robotic arm and a real-world 7-DoF *KINOVA JACO*-2 robotic arm equipped with an Intel RGB D415 camera mounted on the end-effector in an eye-in-hand configuration. The camera and the hand-eye calibration were performed offline. The image processing and the control law computation were executed on a PC equipped with a 1.6 GHz *Intel i5* 10^{th} generation laptop. The code was developed in C++ using the ViSP library [42].

The camera was mounted on the robotic arm's end effector in a setup as shown in Fig. 2. As in the case of several industrial and assistive robotic arms cases [43], [44], the camera roll axis around the optical center was controlled by the movement of joint-7, the pitch is controlled through joint-6, and yaw motion is controlled by joint-5 of the robotic arm A spray bottle with an April tag square of size 0.096 m, was used as a target object for detecting the current visual features. To get the desired features, virtual homography was used [45]. In the real experiments, the April-Tag pose estimation [40] was used to get the current pose and current depth of points.

Fig. 3 shows the simulation setup of a wireframe model of the Viper 850 6-DOF robotic arm with an eye-in-hand configuration available in the VISP library [42]. Four circular cross points on a square target plate were used as the visual features. The initial (blue) and the desired (red) image frame as shown in the internal camera view Fig. 3b, corresponds to a 140-degree rotational error around the camera optical axis. [See also supplementary video results for all experiments]

A. PURE ROTATIONAL ERROR CASE

A simulated model of a VIPER 850 robotic arm is shown in Fig. 3, with an eye-in-hand camera showing an external and an internal camera view of the object's current (blue) and the desired image (red), depicting 140-degree pure rotation around the camera optical axis. When the proposed SPJASM controller is deployed on this visual servoing task, the controller first evaluates the orientation difference between the current and the desired frame using the partial pose estimation technique. For the simulation case, we used a partial pose estimation [34] and for a real-world robotic arm case, we used an April tag [40] for estimating the pose of the current and the desired frame.

Once the difference between the current and the desired frame is evaluated, the program computes the difference in the roll angle between the current and the desired frame. This roll angle provides the basis for our roll first strategy and the controller begins to roll the robotic arm wrist joint to minimize the difference in the roll angle to reach a predefined roll angle threshold ε. After reaching this roll angle threshold, the controller switches to adaptive gain ESM PMJ controller to converge the remaining visual feature error to zero.

TABLE 1. Pure rotational error case; Comparison of the ESM, SPJESM and SPJASM controller under various rotational errors.

Error (Deg)	ESM METHOD (λ= 0.5)				SPJ-ESM-METHOD (λ= 0.5 α= 0.75 , ε =45°)				SPJ-ASM-METHOD α= 1, ε =45°, adp gain (3.0, 0.3, 10)			
	Max. Joint. Vel. (rad/s)	Max condition Num.	Max Energy	Iteration	Max. Joint. Vel (rad/s)	Max conditi on. Num.	Max Energy	Iteration (rfm)	Max. Joint. Vel. (rad/s)	Max conditi on. Num.	Max Energy	Iteration (rfm)
90	1.18	202527	1.415	176	1.04	5446	1.09	185 (27)	0.484	5430	0.234	102(42)
140	3.18	207308	9.272	198	1.694	4929	2.817	190 (57)	0.672	4930	0.455	119(56)
165	8.37	155997	70.86	unstable	1.584	5052	2.515	209 (61)	0.677	5052	0.501	121(65)
175	25.26	194559	644.17	unstable	1.565	5071	2.451	213 (65)	0.680	5071	0.507	130 (75)
180	3739	379286	$1.4x10^7$	unstable	1.565	5067	2.445	297 (183)	0.704	5073	0.510	178(137)

Indeed using a two-stage switching controller breaks down the complex large orientation error task into two simple tasks. First task is a pure rotation around the camera optical axis till the orientation error becomes equal to the switching threshold. Subsequently, the second task becomes a straightforward IBVS problem with a 45-degree orientation error in the roll axis which can be completed smoothly using an ESM controller.

For a large pure rotational error visual servoing task, the experiment data and the parameters were recorded in Table 1. Various initial roll angles were used ranging from 90 degrees to 180 degrees in simulation to analyze the performance of the ESM controller and a SPJ-ESM controller i.e a SPJ controller without adaptive gain was used to see the effectiveness of switched part jacobian in isolation. Finally, the proposed Adaptive gain (SPJASM) controller results are also given in Table 1. For evaluating the controller performance following parameters were recorded convergence iterations, maximum energy consumed by the controller [28]-(36), the maximum joint velocity reached by the robotic arm, and the maximum condition number of the task jacobian reached during the task. Camera /advance retreat motion was also noted if present under any condition. Image feature trajectories and camera 3D trajectories were also recorded for selected cases which were important for a comparison point of view.

Table 1 shows that for large camera orientation errors about the camera's optical axis, the ESM controller was not effective in handling large orientation errors as the controller was able to converge the task only up to 140 degrees. Moreover, the joint velocity produced is generally high and the loop was unstable due to an ill-condition task Jacobian matrix with very large condition numbers of the order of 2×10^5. The ESM controller also has a slow response in terms of convergence rates. The energy required for the task was also very high.

The second considered controller was the SPJ-ESM controller i.e switched part jacobian controller without the adaptive gain part. SPJESM controller was able to converge the visual servo task up to 180 degrees effectively and it invariably extended the convergence zone of the controller to 180 degrees. It also reduces the maximum joint velocities needed to complete the task and the maximum condition number of the loop we also reduced to unity for the first phase of control.

As compared to the ESM controller, the SPJESM controller also reduces the amount of energy consumed by the controller to achieve the tasks under all conditions. Nevertheless, the convergence rate was slow, it performed better than the ESM controller though, but still, the controller's convergence rate and energy consumption can be improved by using the proposed scheme. The purpose of sharing SPJESM controller performance was to prove that the SPJ technique is providing a solution in isolation without the help of a supplementary adaptive gain algorithm.

Using the proposed adaptive gain-switched part jacobian SPJASM scheme, the controller performance was further enhanced. The proposed SPJASM controller while utilizing switched-part jacobian combined with the adaptive gain method was able to converge visual servo task efficiently against wide ranges of rotational errors up to 180 degrees about the optical axis.

SPJASM controller improved the performance of ESM controller for rotational error cases such as the convergence zone was widened to 180 degrees, the maximum joint velocities and maximum energy consumed was further reduced for instance for 140-degree case the joint velocity reduced from 3.8 rad/s (ESM) to 0.672 rad/s (SPJASM) and maximum joint energy reduced from 9.27(ESM) to 0.455(SPJASM) which is a significant 95% decrease in joint energy consumed for the same task.

SPJASM controller also improved the convergence rate by consuming fewer iterations to converge than other (ESM, SPJESM) controllers for instance, for a 140-degree error case, as compared to the ESM controller which took 198 iterations, the SPJASM controller took only 119 iterations out of which 56 were in roll first mode (rfm) using directly induced part jacobian. Hence, increasing the convergence rate significantly by 40% and reducing the computational load of the control loop considerably.

1) PERFORMANCE OF THE ESM-IBVS CONTROLLER UNDER PURE ROTATIONAL ERROR OF 180 DEGREES

This experiment demonstrates the performance of the conventional ESM-IBVS-PMJ controller under a large orientation error of pi rad rotation around the camera optical axis. Initially, the current and the desired image frames have pi rad (0,0,0,0,0,180) degrees pure rotational error. The proposed

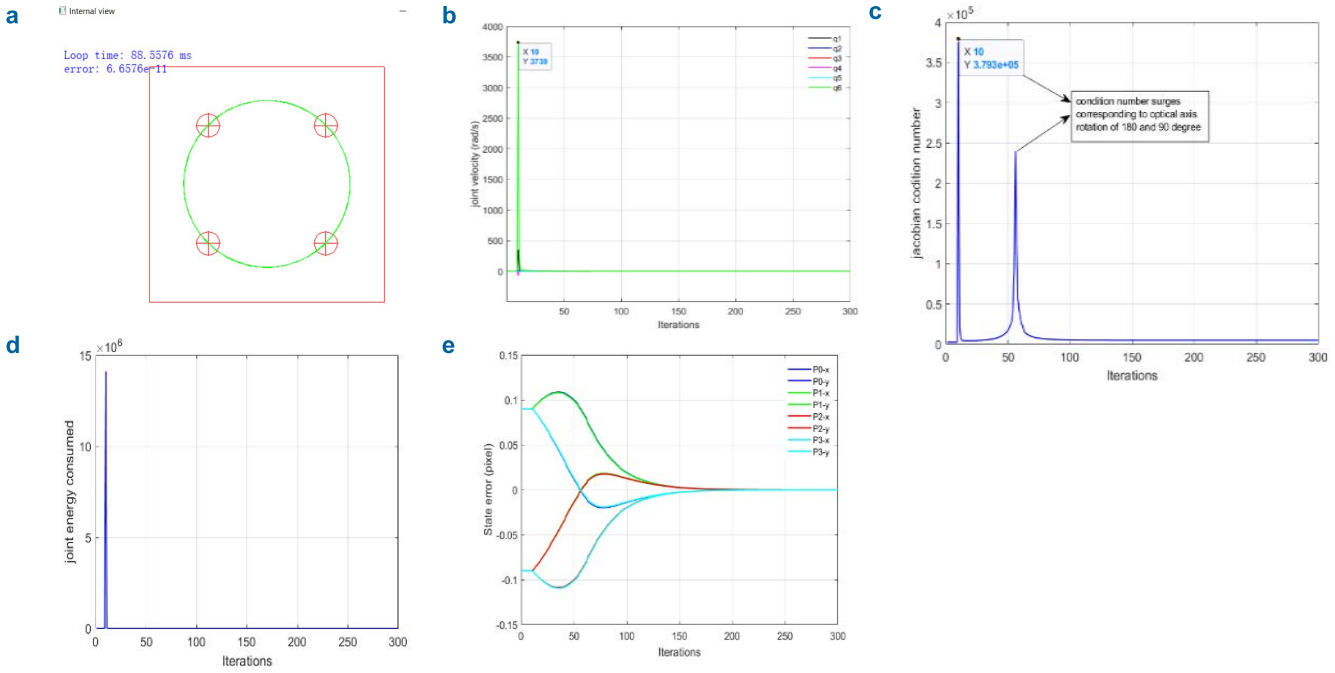

FIGURE 4. Performance of the ESM controller under the pure rotational error of 180 degrees around the camera's optical axis; a. visual feature trajectory in the image plane, b. Joint velocities, c. task Jacobian condition number, d. joint energy consumed during the task and e. visual feature error decay.

controller under simulation produces a pure circular feature trajectory on the image plane as shown by green curves in Fig. 4a.

The joint velocities produce surges near 180 degrees as shown in Fig. 4b where a very high joint velocity (3.730×10^3 rad/s) was produced by the controller that is unrealizable by the robotic arms under normal circumstances. This behavior is due to the highly ill-conditioned task Jacobian matrix as listed in Table 1 that is producing velocities for visual servoing.

As shown in Fig. 4c, two surges are visible for task Jacobian condition number one at 180 degrees and one near 90 degrees that are caused due to the singular arrangement of desired feature jacobian at 90 and 180-degree. This problem was also reported by [35]. The energy required for performing this movement was also very high (1.3×10^6) as shown in Fig. 4d. The required motion of the robotic arm is unrealizable in the real world but the task was completed anyhow in the simulation environment and the visual feature error converges to zero.

In real-world environments, this visual task would become unstable and cannot be completed successfully as required joint velocities are unrealizable by the robotic arm due to the joint velocities constraints in the real robotic arms.

2) PERFORMANCE OF THE SPJASM IBVS CONTROLLER UNDER 180° ROTATIONAL ERROR

The performance of the proposed SPJASM IBVS controller under a large initial orientation error is shown in Fig. 5. Initially the current and the desired image frames have pi rad (180 Degree) pure rotational error, about the optical axis. In the first phase of the control, the controller neutralizes the rotation error between the current and the desired image by following the roll-first strategy and overcoming much of the orientation error about the optical axis of the camera.

Once the orientation error is within the defined threshold range (i.e. 45 degrees for this case) the controller switched to an adaptive ESM controller and brought all the feature errors to zero, hence avoiding large surges in the joint velocity as observed using the ESM controller Fig. 1b.

The image feature trajectory can be seen in Fig. 5a, where a pure rotational path is taken by the point features for the first control phase till the rotation error threshold was achieved, followed by the curvilinear trajectory of features for the later control phase where all the joints were utilized to reduce the visual feature error to zero.

Fig. 5b shows the joint velocities produced by the controller which can be divided into three phases. The first phase is when the control output is building from zero. As for the case of pure rotation of pi rad, initially, the response of the controller is very small nearly zero, that grows gradually and after 50 iterations it becomes non-zero. In the second phase, near the 60^{th} iteration, it becomes large enough to move the robotic arm roll joint, which moves the camera in a circular motion and the orientation error started to decrease slowly and gradually.

Once the visual feature error decreases between the current and the desired frame the roll velocity decreases. In the final phase, When the visual feature error between the current and the desired frame decreases less than 45 degrees (i.e. the roll angle threshold for switching in this case), near the 137^{th} iteration the controller, switched to adaptive gain-ESM

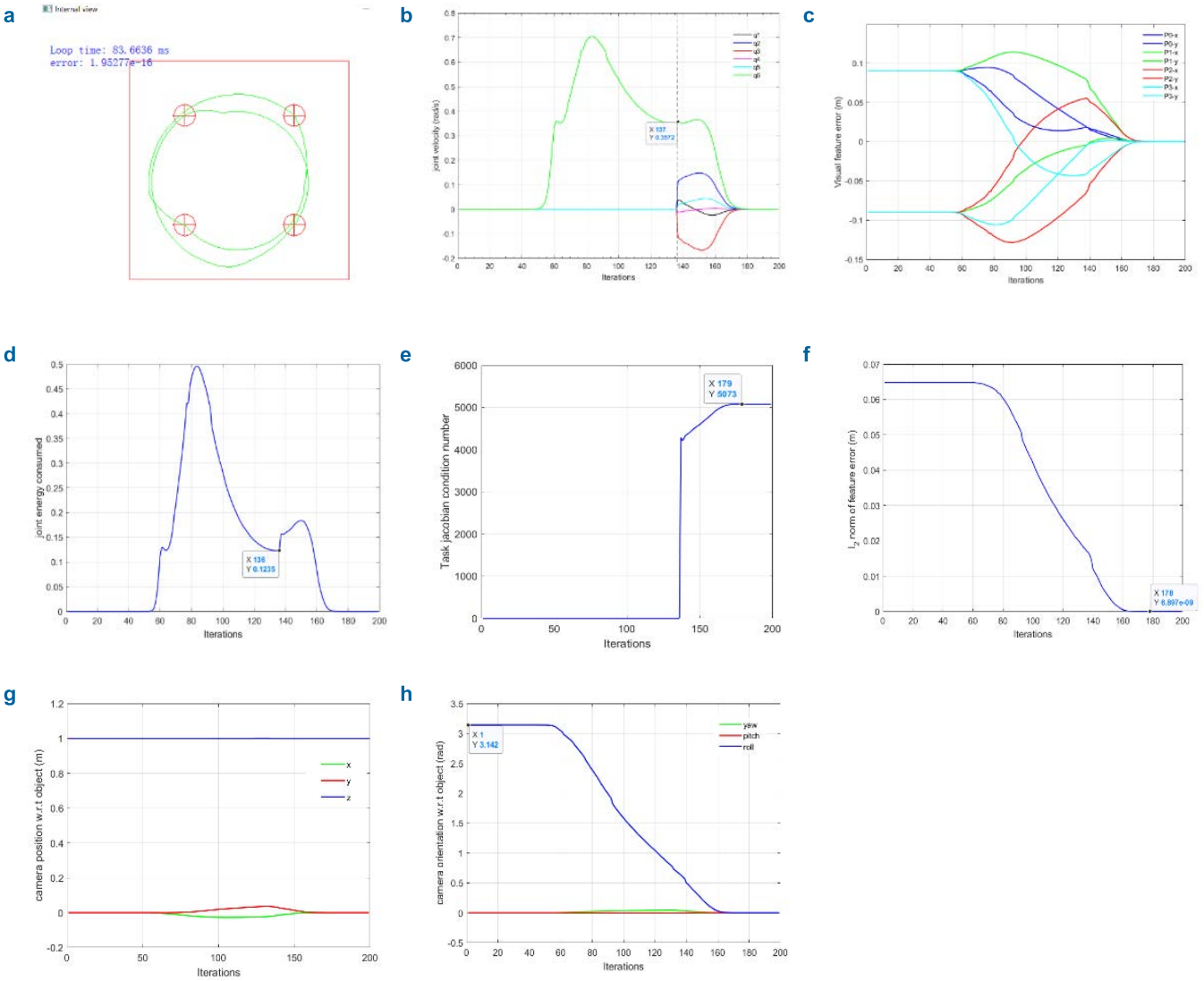

FIGURE 5. Performance of the SPJASM controller under a pure rotational error of 180 degrees around the camera's optical axis; a. visual feature trajectory in the image plane, b. Joint velocities, c. visual feature error decay, d. joint energy consumed during the task and e. task Jacobian condition number, f. l_2 norm of feature error, g. Camera position w.r.t object during the task, h. Camera orientation w.r.t object during the task.

(i.e. ASM controller) controller and all the joint came into play and their joint velocities starting contributing towards goal achievement to further reduce visual feature error to zero as shown in Fig. 5c. Both the figures showing joint velocities (Fig. 5b) and visual feature error (Fig. 5c) share common switching zones from one controller to another.

Fig. 5d shows a nice behavior of the image jacobian condition number which remains nicely equal to unity while using the proposed SPJASM controller in the first stage and in the second stage the condition number restores to the case of the ASM controller that is raised to around 5000.

The camera motion was also analyzed for the proposed scheme to see whether there is any camera advance/retreat motion during the servo control or not. The image plane feature trajectory is shown in Fig. 5a and the camera movement w.r.t the object plane is given in Fig. 5g. Fig. 5g-h shows x-y-z displacement of the camera w.r.t the object plane, no movement of the z-axis was observed which means no camera advance/retreat motion was observed. Fig. 5h shows the camera orientation part where mostly the roll joint of the camera was utilized with roll angle converging from pi radian to 0 radians smoothly. During the roll motion of the camera, other joints were not moving as per the proposed control scheme.

Consequently, there was some deviation caused in the x and y-axis in the camera plane w.r.t object pose that is covered in the second phase of control when all joints come into play to reduce feature error to zero. Finally, we achieved zero [0,0,0] orientation error and [0,0,1] x,y,z position as desired for the camera plane w.r.t object plane.

Using the proposed SPJASM control scheme, The controller took 178 iterations in total to converge, out of which 137 iterations were in roll first mode where the controller utilizes directly defined part jacobian without calculating the

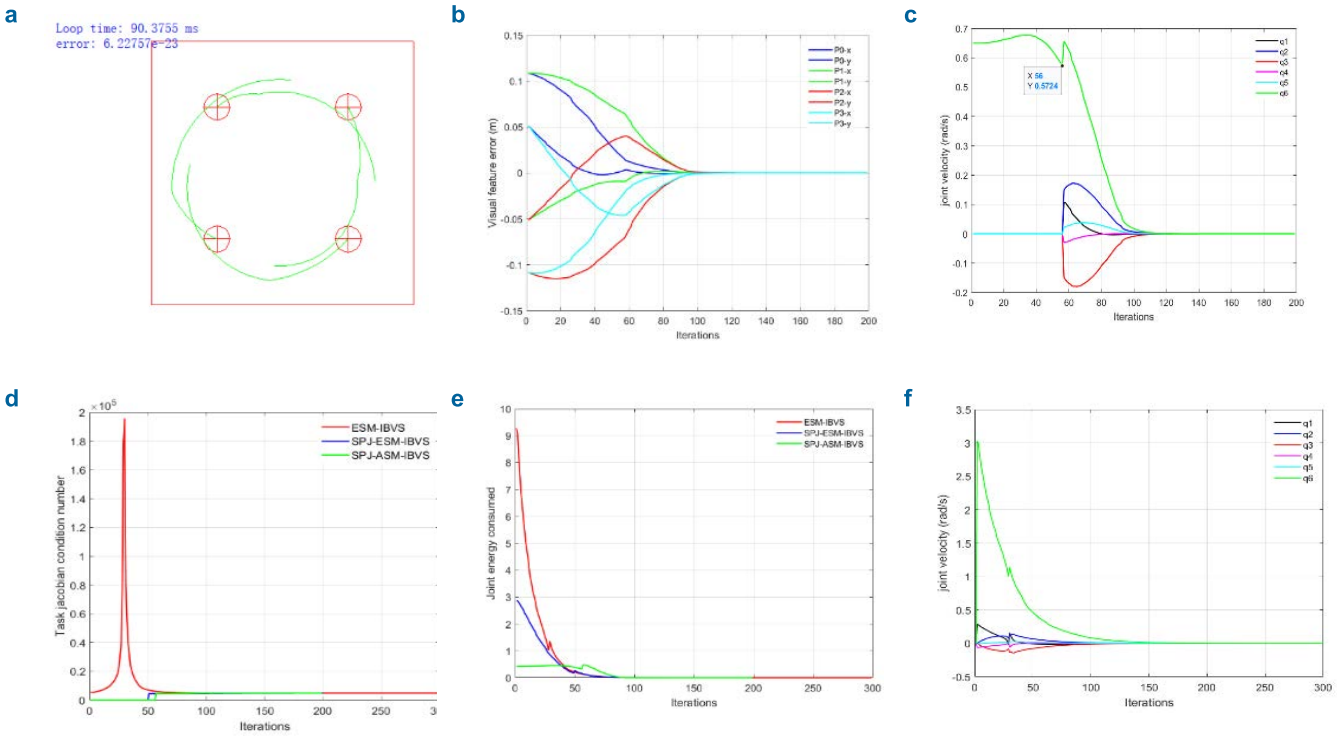

FIGURE 6. Performance of the proposed SPJASM controller under a pure rotational error of 140 degrees; a. Visual feature trajectory in the image plane, b. Visual feature error decay, c. Joint velocities, d. Task Jacobian condition number and e. Joint energy consumed during the task, f. Joint velocity profile for the ESM controller.

full jacobian and the condition number remains unity which means that the task jacobian stayed well-conditioned and the jacobian pseudo inverse is not causing any disturbance in the control scheme output. Besides getting computational ease and a well-conditioned loop, we were also able to perform the visual servo control task consuming very less energy as shown in Fig. 5d, the energy consumed in the joints is well distributed over the loop and requires very less energy as compared to the ESM method Fig. 4d.

B. COMPARISON OF THE SPJ AND THE ESM CONTROLLERS FOR A ROTATIONAL ERROR OF 140 DEGREES

In this simulation case, we compared a pure rotation case which is achievable under both the control laws i.e. SPJASM, and also for ESM, and have compared their performance in terms of image trajectories and joint velocities, Task jacobian condition numbers, and the amount of energy consumed in joints to achieve the task.

As shown in Table 1, the task was achievable by both controllers where the ESM controller took 198 iterations to complete while the proposed SPJASM controller took only 119 iterations out of which 56 were under roll first mode providing a 47% saving in the computational cost.

SPJASM controller utilized directly induced part jacobian to compute the control law and the task condition number remained unity for this phase. Fig. 6a shows the image feature trajectory taken by the SPJASM controller which is a circular motion in the first phase and the second phase is a curvilinear trajectory to achieve the task. Actually, the first part of the trajectory was produced during the roll-only motion of the camera and in the second stage, the camera moves with all other joints to produce a curvilinear path as shown in the image feature trajectory.

The visual feature error as shown in Fig. 6b also has a nice uniform decay towards zero error while using the SPJASM controller. The joint velocities uniformly decayed towards zero. Fig. 6(c, f) shows the joint velocity profiles for SPJASM and ESM controllers. The maximum joint velocity required by the SPJASM controller was 0.672 rad/s for the roll joint as compared to the 3.2 rad/s required by the ESM controller.

As both controllers were capable of achieving the desired visual task we can get an insightful comparison of the performance by comparing parameters like the maximum joint velocity required to achieve the task, maximum energy consumed by the joints and also the task Jacobian condition number during the servoing task. This will provide good insight for deciding which controller performed better for this task. Fig. 6(d-e) shows a comparison of the joint energy consumed by using an ESM controller, a SPJESM controller (which uses SPJ without adaptive gain), and finally the proposed SPJASM controller. It can be seen in Fig. 6e that the ESM controller spends too much energy to achieve the task whereas while using the SPJESM controller roll first strategy energy consumed is reduced considerably. Finally, the proposed controller minimizes the joint energy consumed

to a minimal level while using 95% less energy (Table 1) than the ESM controller. Fig. 6d shows the comparison of task Jacobian condition number. For an ESM controller, there is a surge in jacobian condition number as it passes over 90 degrees the condition number rises to 2×10^5 i.e. the controller was highly ill condition whereas the SPJESM and the SPJASM controller were nicely conditioned at unity in the start of loop till switching to the second phase, where it remained at par with ESM controller settling near 5000.

The comparison clearly showed that ESM controllers were taking very high joint velocities for the task and also they become ill-conditioned and approached a singularity of feature jacobian (at 90 degrees), producing a sudden surge in the condition numbers. On the other hand, SPJESM controllers reduced the joint energy considerably to one-third of the ESM controller. Finally, the proposed SPJASM controller improved the SPJESM controller performance by further reducing these parameters and produced the lowest joint velocities and joint energies to achieve the visual servoing task in the least iterations (119). SPJASM controller also remained well-conditioned throughout the task. In fact, during the use of the SPJ controller, the task Jacobian condition number remained at unity during the first RFM phase.

C. EFFECT OF THE ALPHA AND ROLL ANGLE THRESHOLD VALUES ON THE SPJ CONTROLLER

The effect of joint sensitivity parameter α and the roll angle threshold ε values on the performance of the SPJESM controller are given in Table 2. Indeed, the α is the component of the robotic arm manipulator jacobian matrix which defines the sensitivity of the joint velocity w.r.t to the visual feature error in the proposed visual servo control. To study the effect of these parameters only, we affixed other parameters of the experiment such as a rotational error of 165 degrees and the gain was set to constant (unity) in these simulations. A range of α values from 0.75 to 1.5 were induced in the controller to assess the effect of the α value on the performance of the controller.

In the control law (27), there is an inverse relationship of α value with the joint velocity. Therefore, increasing the value of α decreases the joint velocity and thus increases the energy consumed by the controller as also indicated in Table 2. Interestingly this is a desirable property of the proposed controller as shown in Table 2, that by changing the α value, one can choose to complete the same visual task with a range of desired parameters. Such as choosing a range of joint velocities (from 3.16 to 1.58 rad/s i.e. about a 50% decrease) and where the consumed joint energy ranges from 10 to 2.5 which is a 75% decrease, without changing the gain (constant gain = 1). Of course, this comes at the cost of increased iterations to complete the task.

The effect of rotation angle threshold ε was also studied in this simulation in Table 2. Two roll angle thresholds ε were chosen for switching SPJ control i.e. 45 degrees and 60 degrees. On one hand, setting a higher ε such as 60 degrees makes it easier and quicker for the control scheme to switch from the first phase of roll first mode to the second phase where all joints can contribute to task convergence. Thus more time is available in the second stage to converge the tasks. This is especially beneficial in cases where visual error comprises rotational and translational errors. However, choosing a wider ε produces bigger joint velocities, and lesser iterations are covered under the roll first mode which means more control iterations have to compute full end effector jacobian that increases the computational load.

On the other hand, by choosing a narrower roll angle threshold ε such as 45 degrees, the controller will remain in the first roll first mode for more time which is computationally efficient due to the use of directly induced part jacobian and the controller will be able to cover more rotational error while spending less joint velocities and consuming lesser energy. Secondly, after switching, lesser joint velocities will be required by all other joints to converge the remaining task error. By repeating the experiment numerous times and in several conditions. The insight we have developed over time is that in general motion cases, choose a wider ε threshold like 60 degrees for combined rotational + translational error. On the other hand, choose a tighter ε threshold like 45 degrees if the task involves more rotational errors and the user wants to have lesser joint velocities and to spend lesser energy. We have utilized both values of ε (45 and 60) in our simulation and experiments.

D. PERFORMANCE OF THE SPJASM CONTROLLER UNDER PURE TRANSLATIONAL ERROR

The performance of the proposed SPJASM controller under pure translational error was assessed in the simulation as shown in Fig. 7 and Table 3. The proposed controller switches from the adaptive ESM controller to the SPJASM controller in presence of rotational errors and that happens when the

TABLE 2. Effects of roll angle threshold and alpha value on the performance of the SPJESM controller.

Rotation error	SPJ Sensitivity α	Rotation threshold ε	Max. Joint vel (q6) rad/s	Task Jac. cond. Num (peak)	Energy Consumed (peak)	Convergence iterations (roll-first mode) (error norm $l_2 = 10^{-8}$)
165 Deg	0.75	45	3.16	5052	10.06	157(56)
	1	45	2.35	5052	5.56	150(58)
		60	2.37	5434	5.65	153(54)
	1.5	45	1.570	5052	2.515	159(62)
		60	1.585	5366	2.535	155(58)

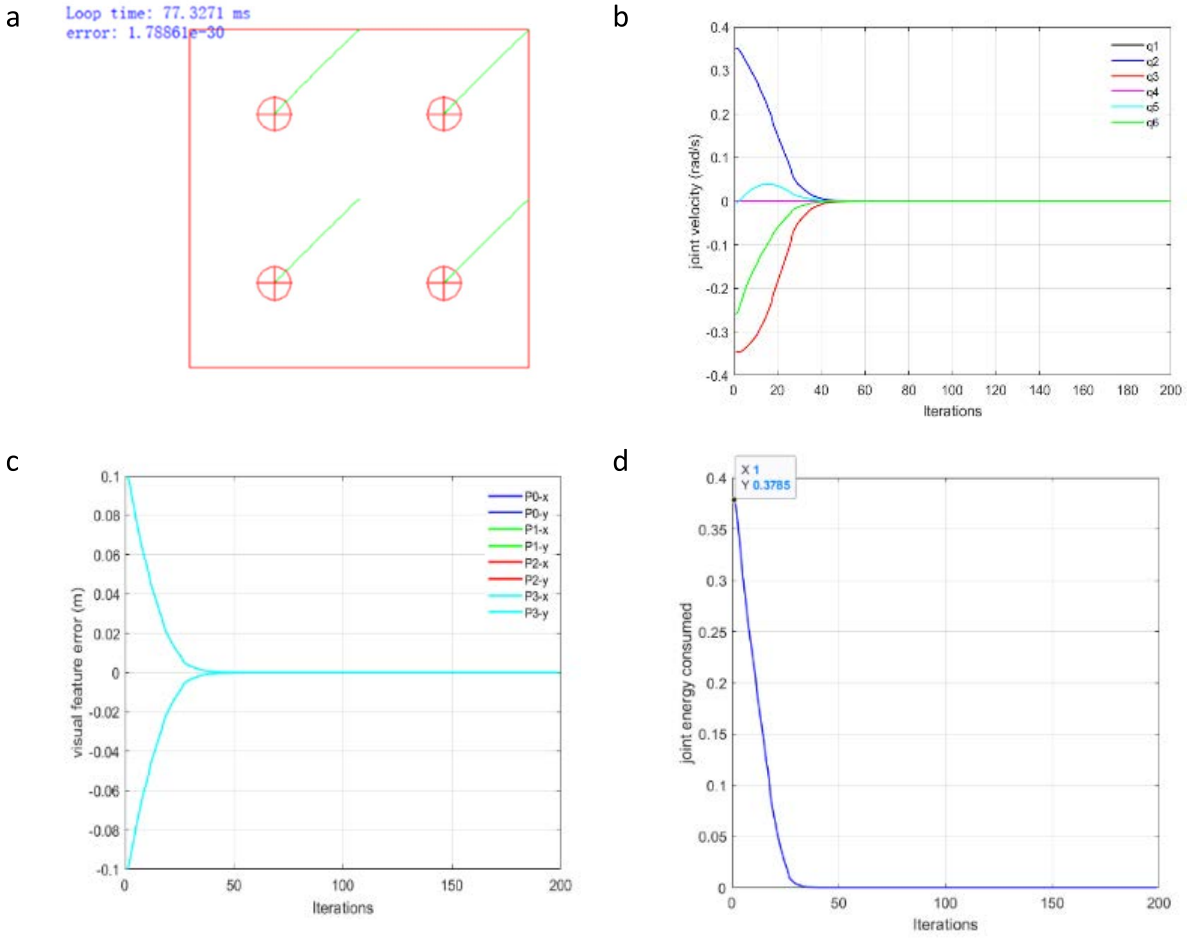

FIGURE 7. Performance of a SPJASM controller under pure translational motion; a. Visual feature error trajectory, b. Joint velocities, c. Visual feature error decay, and d. Joint energy consumed during the task. (Adaptive ESM controller has the same graph for pure translational error.)

TABLE 3. Performance of the SPJASM controller under pure translational motion.

Translation Error	Adaptive ESM Method (adapt. gain= 2.5,0.25,10)				SPJ-ASM-Method α= 1.0, ε = 45°, (adapt. Gain=2.5,0.25,10)			
	Max. Joint Vel. (rad/s)	Max condition	Max Energy	Converge Iterations	Max. Joint Vel. (rad/s)	Max condition	Max Energy	Converge Iterations
Pure Z(-0.2)	0.251	2533	0.1236	63	0.25	2533	0.1236	63
Pure X(0.15)	-0.2543	4383	0.128	59	-0.25	4383	0.128	59
X=0.1, Y = -0.1	0.354	5629	0.378	54	0.35	5629	0.378	54

rotational error exceeds the roll angle threshold ε. For the pure translational error, the two controllers act similarly. As SPJASM controller reduces to a single-stage ASM controller for this case. The simulation shows the performance of the two controllers under various pure translational errors.

Fig. 7a shows the image points feature trajectory in the image plane that is a straight line from the current to the desired image frame. This is a nice feature of this control scheme as it produces a straight line, unlike polar coordinates-based visual servoing schemes [22] which produce curved feature trajectories for the translational errors. The joint velocities for the task are nominal around 0.35 rad/s as shown in Fig. 7b also decayed smoothly. The controller only took 54 iterations to reach a convergence threshold of 0.00005 (Euclidean error norm of the feature error). Visual feature error decay was uniform as all the feature errors decayed at the same time coinciding with each other as shown in Fig. 7c for the case of pure translational with combined x and y-axis error ($x = 0.1, y = -0.1$).

Thanks to using an adaptive gain controller, the decay of feature error was smooth and fast. Finally, as shown in Fig. 7d, the controller was efficient and did not spend too

TABLE 4. Visual feature values for the object on a tilted plane.

Z FEATURE VALUE	X FEATURE VALUE	Y FEATURE VALUE
CURRENT VISUAL FEATURES		
Z= 1.01245	X= 0.04309	Y= 0.04407
Z= 1.01207	X= -0.04577	Y= 0.04142
Z= 0.9875	X= -0.04418	Y= -0.04518
Z= 0.9879	X= 0.04688	Y= -0.04244
DESIRED VISUAL FEATURES		
Z= 1.0	X= -0.045	Y= -0.045
Z= 1.0	X= 0.045	Y= -0.045
Z= 1.0	X= 0.045	Y= -0.045
Z= 1.0	X= -0.045	Y= -0.045

TABLE 5. Performance of the SPJASM controller for the tilted plane object.

SPJ-ASM-METHOD ALPHA= 1.0@60DEGREES , ADAPTIVE GAIN [3.0,0.3,10]				
DIFFERENCE OF POSE (M, DEG)	MAX. JOINT VELOCITY	MAX COND.	MAX JOINT ENERGY	CONVERGE ITERATIONS (RFM)
[0 0 0 25 0 180]	-0.685	6365	0.47	136(78)

much energy on the task, and completed the task using a minimum joint energy peak of 0.378.

E. SPJ-ASM CONTROLLER FOR OBJECTS PLACED ON A TILTED PLANE

This simulation experiment shows the effectiveness of the proposed SPJASM controller against initially tilted targets i.e. the controller is capable of handling very large rotational errors for objects that are placed on a tilted plane and the image planes are not parallel to the camera image plane. For this case, the chosen object is placed on a tilted plane (angle of −25 degrees w.r.t the camera plane y-axis), combined with a rotational error of 180 degrees about the camera optical axis (z-axis).

The initial and the desired poses of the robotic arm are shown in Fig. 9 (left). The initial (blue) and desired (red) frames with an orientation error of [0 0 0 25 0 180] (m, rad) are shown in Fig.9b (right), and the corresponding initial and the desired visual feature values are enlisted in

Table 4. Fig. 9b shows the image feature trajectory during the successful visual servoing using the SPJASM controller.

The trajectory comprises a clockwise rotation of the camera followed by a curvilinear path. The point to be noted here is that the controller is not bound to take clockwise or anticlockwise circular motion during the roll first mode, rather it depends on the nature of the error and the values of the visual features on which the controller decides whether to take anticlockwise direction (like in case of 180-degree SPJASM) or to roll in a clockwise direction (like in this case of XZ-axes orientation errors) that is a trademark feature of image-based visual servo control schemes. Feature error decay is given in Fig. 8.

The joint velocity of the robotic arm is plotted in Fig. 8a where the maximum roll velocity reached (−0.685 rad/s) and it took about 136 iterations (78 in the roll first mode) to

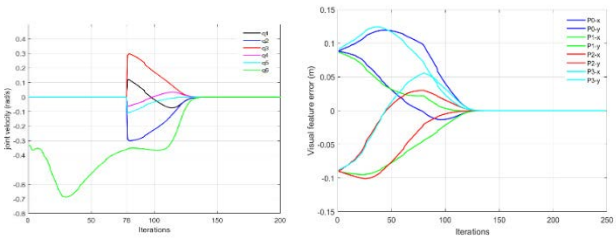

FIGURE 8. SPJASM controller performance for a tilted plane object; a. joint velocities, b. visual feature error.

converge the visual feature error to zero. The task jacobian remained well-conditioned throughout the visual servo control as the task Jacobian condition number remains unity for the roll first mode (78 iterations). While the maximum energy consumed by the joints remained quite low at 0.47 as given in Table 5.

F. COMPARISON OF THE SPJASM CONTROLLER WITH THE M-ESM AND AWPM IBVS SCHEMES

The comparison of the performance of the proposed SPJASM controller with the modified ESM (MESM) controller [35] is given in Fig. 10 and Table 6. A visual servoing task is set up in simulation with a large rotational error around the camera optical axis ranging from 60 to 180 degrees. Performance was compared in terms of maximum joint velocity, task Jacobian condition number, maximum joint energy consumed, number of iterations for convergence, and Image feature trajectories demonstrated for selected cases. For a detailed comparison, a rotational error of 165 degrees was chosen for both controllers.

Fig. 10a shows the image feature trajectory for the SPJASM controller for the rotational error of 165 degrees, which is a combination of circular and curvilinear motion. Whereas the feature trajectory for the MESM controller is a pure circle as shown in Fig. 10b. The visual feature error decay for both controllers was smooth as shown in Fig. 10(d-e).

The major difference in the controller performance was indicated by comparing the joint velocity produced by the SPJASM (0.677 rad/s) which is much lesser than the joint velocities required by the MESM controller (8.73 rad/s) for converging the task as shown in Fig. 5b. The joint energy consumed for task completion in Fig.10g where the joint energy required by the MESM controller was enormous (70.86) as compared to the joint energy (0.510) consumed by the proposed SPJASM controller with an added advantage of lower computational cost due to the use of a directly defined end effector jacobian and a lower task Jacobian number as shown in Fig. 10h.

The SPJASM controller converged faster than the MESM controller and took 121 iterations out of which 65 were in roll first mode providing computational ease in the control loop. While the MESM controller produced steep roll velocity initially and tries to converge the task hastily at the start of

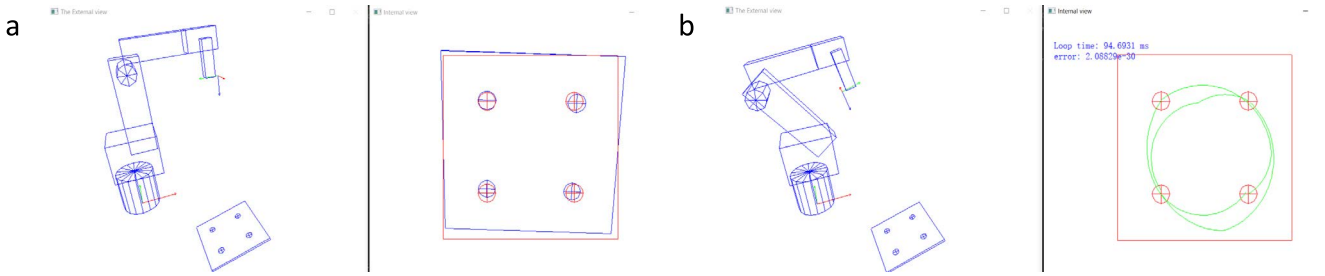

FIGURE 9. SPJASM controller case for an initially tilted object with orientation error (0, 0, 0, 0, −25, 180) (m, rad): a. Initial pose of robotic arm and initial (blue) and the desired (red) frame, b. Final pose after task and image feature trajectory.

TABLE 6. Performance comparison of the SPJASM and MESM controller.

Rotational error (Deg)	MESM (@gain 0.5)				SPJ-ASM-METHOD ALPHA= 1.0@45D @(adp gain 3.0, 0.3, 10)				
	Max. Joint Vel. Q6 (rad/s)	Max cond. num.	Max Energy	converge Iterations	Max. Joint Vel. Q6(rad/s)	Max cond. num.	Max Energy	convergence Iterations (rfm)	Camera retreat d (cm)
+60	0.636	4326	0.409	186	0.38	5580	0.132	87(20)	0
+90	1.10	4534	1.22	190	0.484	5430	0.234	102(42)	0
+140	3.03	5071	9.27	200	0.672	4930	0.455	119(56)	0
+165	8.37	5071	70.86	209	0.677	5052	0.501	121(65)	0
+175	25.26	5071	645	222	0.680	5071	0.507	130 (75)	0
+179	126.41	5071	16129	NC	0.682	5073	0.507	141(76)	0
+180	INF	INF	INF	NC	0.704	5073	0.510	178(137)	0

TABLE 7. Performance comparison of SPJASM and AWPMJ controller.

Rotational error (Deg)	AWPMJ Controller (using adp. gain , parameters from [28]) $\Lambda = 0.03$, $K_W = 0.1$, $K_C = 2.302$				SPJ-ASM-Controller using adp gain 2.5, 0.25, 10 alpha= 1.0@45Deg				
	Joint Vel. (rad/s)	Max Energy	converge Iteration	Camera Retreat (cm)	Joint Vel. (rad/s)	Max condition	Max Energy	convergence Iterations (rfm)	Camera retreat (cm)
+95	NR	0.1091	44	3.33	0.513	5356	0.252	105(45)	0
+165	NR	0.4013	44	7.19	0.585	5052	0.342	181(107)	0
+175	NR	0.5054	44	9.99	0.587	5073	0.345	191(119)	0
179	NR	0.8076	44	12.74	0.587	5073	0.355	193(135)	0
+180	Unstable	Unstable	NC	Infinite	0.587	5073	0.364	198(155)	0

the control loop but later suffered from the low residual error left in the end with very low velocities near the convergence zone, consequently MESM controller took 209 iterations to complete the task.

Supplementing the ESM controllers, The MESM controllers were found to be effective till 140 degrees in terms of producing viable joint velocities. The difference MESM creates is that it stabilizes the task Jacobian and reduces the condition number for the cases passing over 90 and 180 degrees by avoiding the singularity of the ESM controller at an additional computational cost i.e. using a multiple of motion transformation matrix (T^{-1}, a 6×6 matrix) in the control law as shown in (8).

The Task jacobian condition number comparison graph shows that the condition number for the SPJASM controller remains unity for the first phase for about half of the loop (65 out of 121 iterations) and then in the later phase it was equal to the MESM controller near 5000. The computational complexity of the control loop increases by using the MESM controller which is caused by multiplying a T^{-1} at each iteration. Whereas using the SPJASM controller one can efficiently complete the visual task.

The proposed SPJASM control scheme which can avoid camera advance retreat motion and can handle very large orientation rotational error about the camera optical axis is also compared to a recent method in [28], which uses the weighted mean of the jacobian matrix to build visual servo control scheme using the adaptive gain based on a sigmoid trend of the visual feature error as also discussed in the introduction of this paper.

The performance of an Adaptive Pseudo-inverse of the Weighted Mean of the Jacobians (AWPMJ) [28] method and the proposed SPJASM controller was compared in Table 7 under various initial camera roll errors. It can be seen in [28]-[Table 3] which shows that the AWPMJ method is effective till 179 degrees. However, the problem of camera advance retreat motion cannot be handled by the controller which is producing 12.74 cm of camera retreat motion for 179 degrees. Even for rotational angles like 95 degrees, the AWPMJ controller was not able to produce camera advance/retreat free

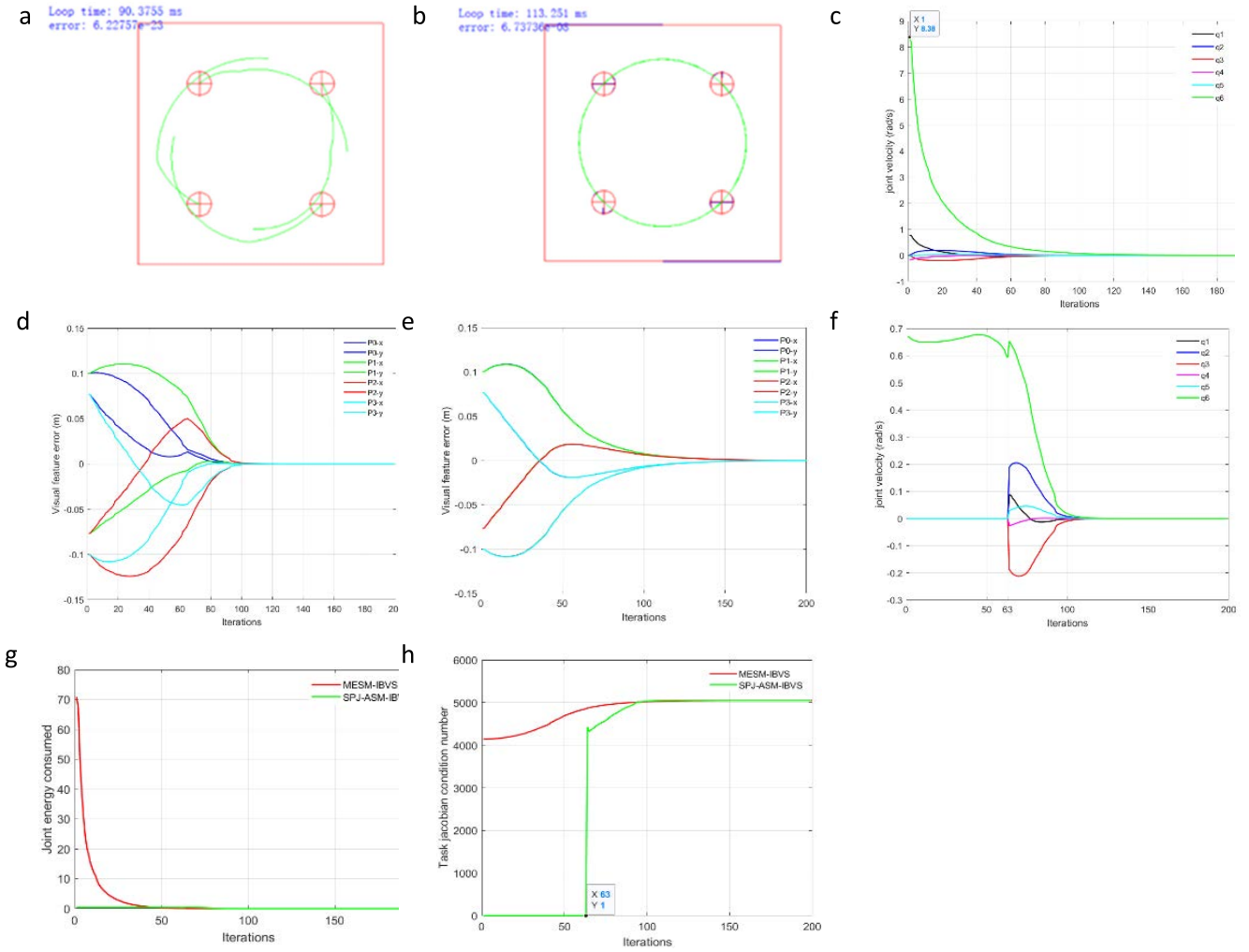

FIGURE 10. Comparison of SPJASM and MESM controller under a large rotational error of 165 degrees a. visual feature trajectory in the image plane for SPJASM control, b. feature trajectory in the image plane for MESM control c. joint velocity of the MESM controller d. feature error decay SPJASM controller, e. feature error decay MESM controller, the joint velocity of the SPJASM controller, g. comparison of the joint energy, h. Task jacobian condition number.

motion. This problem was caused due to the inadequate use of the ESM model (which requires the exact mean of the feature jacobians) rather the APWMJ model used a weighted mean its value is not half in most cases thus producing camera advance/retreat problem depending upon the weightage of the desired feature jacobian in the control law.

In addition to producing undesired camera advance retreat motion the AWPMJ controller also consumes higher joint energies to complete the visual servo task for large rotational orientation errors. Furthermore, the joint velocities were not reported in the paper, which is deemed necessary in deploying the model on real-world robotic arms to cater to the joint velocity limits. Also, no graphical plots were shared for rotational angle errors greater than 165 degrees.

On the other hand, the proposed SPJASM controller can handle large orientation rotation errors around the optical axis efficiently with low joint velocities while consuming less joint energies as shown for 165 & 180 degrees in Fig. 5 and Fig. 6. Hence, we can safely conclude that the performance

TABLE 8. SPJASM-controller performance for a real robotic arm adp. gain = [4.5, 0.3, 30] Alpha = 1.0@30 degree threshold.

Difference of Pose (m, Deg)	Max. Joint Velocity	Max Energy	Converge Iterations	Camera Advance /Retreat
[0 0 0.18 0 0 180]	0.81 (Q6)	0.66	349 (43 RFM)	0.0

of our proposed SPJASM controller remained better in all aspects than the AWPMJ controller in terms of handling camera advance retreat problem, taking lesser joint velocities, and consuming lesser energy to complete the task while covering all the rotational angles errors till 180 degrees, our proposed scheme is supported with the graphical plots in Fig. 5, 6 and data in Table-1.

G. EXPERIMENTAL VALIDATION OF SPJASM CONTROLLER ON A KINOVA JACO-2 ARM

In this experiment, we have validated our proposed SPJASM control scheme on a real-world robotic arm i.e. a 7 DoF

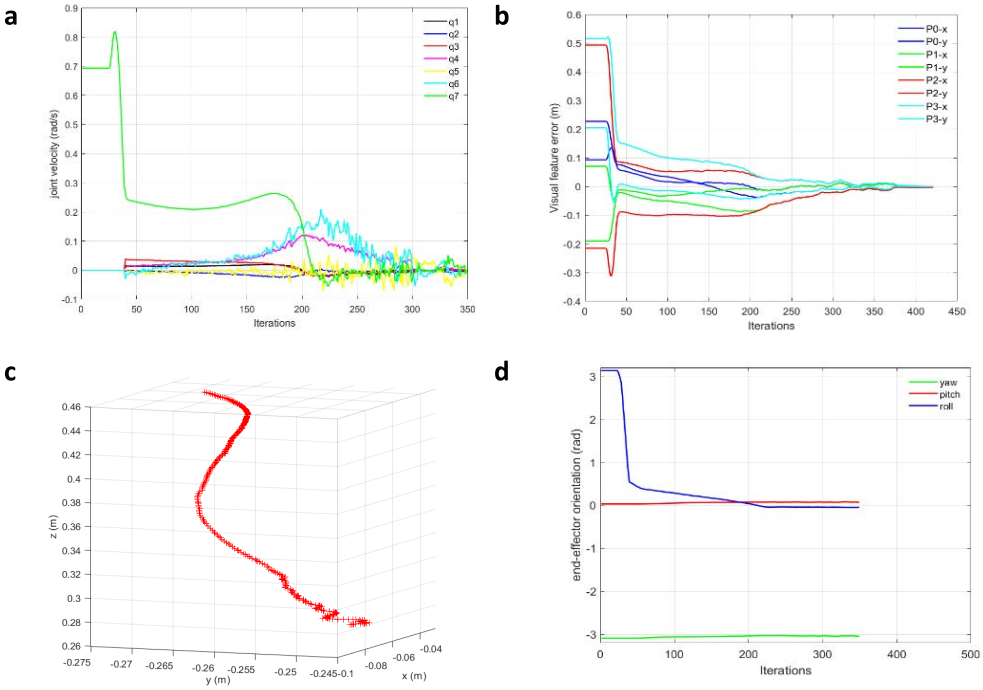

FIGURE 11. Experimental results for the SPJASM controller deployed on a real Kinova Jaco-2 robotic arm, a: joint velocities, b: visual feature error decay, c: 3D trajectory of the end effector, d: orientation of the end effector during the task.

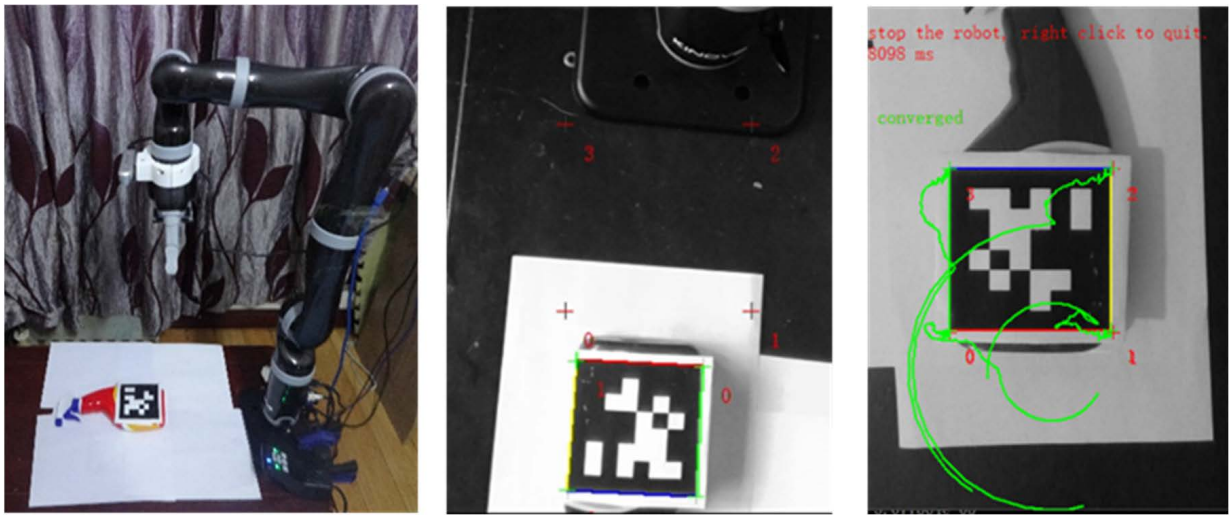

FIGURE 12. Experimental setup; Left: a KINOVA Jaco-2 robotic arm in an eye-in-hand camera showing a target object with an April tag, Center: Initial and the desired point features in the image plane, Right: image feature trajectory for the task.

KINOVA JACO-2 robotic arm with a spherical wrist, and it is mounted with a camera in the eye-in-hand configuration as shown in Fig. 12a and Table 8 shows the controller performance for a roto-translational task.

The camera intrinsic and extrinsic Camera calibration were performed offline using *VISP* [42] and external calibration results were [x = 2.6 y = 7.5 z = −11.8] mm from the end effector gripper frame. The target spray bottle was placed on a table beneath the robotic arm. The spray bottle carries an April tag for detection purposes and automatic visual feature extraction using the April tag pose estimation method [40]. Visual features were the four corner points of the detected April-tag. The initial (blue) and the desired features (red) are shown in Fig. 12b.

The corresponding values of the initial and the desired visual features are tabulated in Table 9. The orientation error comprises a rotational error of 180 degrees combined with a translational error of 180 mm in the z-axis. However, due to the positional inaccuracy of the robotic arm and unavoidable misalignment between the real surfaces involved in the

TABLE 9. Visual features values for the real robotic arm April tag target object.

Z FEATURE VALUE	X FEATURE VALUE	Y FEATURE VALUE
CURRENT VISUAL FEATURES		
Z= 0.561	X= 0.102	Y = 0.2178
Z= 0.561	X= -0.065	Y = 0.195
Z= 0.561	X= -0.09	Y = 0.368
Z= 0.561	X= 0.08	Y = 0.39
DESIRED VISUAL FEATURES		
Z= 0.384	X= -0.125	Y = 0.125
Z= 0.384	X= 0.125	Y = 0.125
Z= 0.384	X= 0.125	Y = -0.125
Z= 0.384	X= -0.125	Y = -0.125

experiment, the following pose of the camera w.r.t the object could be maintained at the start of the task cMo = [−0.003, −0.002, 0.56, 0.022, 0.027, 3.135] (mm, rad).

Nevertheless, the performance of the SPJASM controller under the pure rotational error of 180 degrees about the camera optical axis has already been analyzed under perfect conditions in Section IV (A-B). Another experimental case of the Kinova robotic arm for 100 Degree rotational error can be seen in the associated video results.

In this case, there was a 180-degree rotational error involved in the servoing task, the SPJASM controller first switches to roll-first mode, and the wrist joint rolls to neutralize the rotational error as shown in Fig. 12c by green circular lines as the feature trajectory.

Once the rotational error reached the roll angle error threshold, the controller switched to the adaptive ESM controller, and the remaining rotational and translational error in the z-direction started to decay simultaneously by using all joints, and the end effector moved in a curvilinear direction and moved downwards simultaneously using an adaptive ESM (ASM) controller and the error minimizes to zero.

The velocity profile produced by the SPJASM controller for the visual task of the Kinova robotic arm is shown in Fig. 11a. It is a trademark SPJASM switching controller velocity profile, where the curve starts with the roll velocity only and the controller switching occurs near the 43rd iteration after switching all the joint's velocities started to register in the graph. Due to the use of high gain values, the velocity profile was wavy. This problem can be avoided by delicately tuning the gains.

The visual feature error in Fig. 11b can be divided into 03 phases. Starting where the controller response was building and no movement of visual features was recorded thus getting a straight line. The second phase was the roll first mode which drastically reduces the rotational error through the movement of the roll joint of the wrist and the visual feature took a circular trajectory to minimize the rotational error. Finally, the last phase comprises the adaptive ESM (ASM) control which converges the error slowly but smoothly.

The end effector movement was also traced out in 3D space for the visual servoing task. Fig. 11c shows the 3D trajectory starting with a circular movement of the end effector at the beginning of the servoing task till the roll angle threshold is achieved, followed by a downward motion to cover the z-axis error and converge the camera on the visual feature points using the visual feature error.

The end effector pose graph in Fig. 11d showed that the end effector roll axis was used extensively in the first phase and also used slightly in the second phase along with other joints of robotic arms to reduce roll axis error from 3.13 rads to 0 rad for the roll axis.

The SPJASM controller took 349 iterations to converge this complex task with large rotational and translational errors as shown in Table 8. The convergence rate varies for this experiment as only 43 iterations were taken to neutralize large rotational error from 180 degrees to 30 degrees in the roll first mode using the direct part jacobian while the remaining error was covered under the ASM controller. As most of the iterations (about 306) were taken in the 2nd phase by the ASM controller, it can be improved by tuning well the adaptive gains for the controller to be used with a real robotic arm or by using separate gains for the two stages i.e. SPJASM and the ASM controllers as using the same gains for both controllers could produce a shaky velocity profile for the ASM controller.

Nonetheless, the task was converged successfully and most importantly there was no undesirable camera advance or retreat motion produced by the controller. Furthermore, there were no sudden surges in the velocity like the ones we have witnessed in ESM controllers while handling large orientation error tasks.

This validates that the proposed controller can be used with real robotic arms for IBVS under large rotational errors for a visual error constraint-free visual servoing experience.

V. DISCUSSION

From the simulations and the experimental results, it is evident that the proposed switched-part Jacobian IBVS scheme is effective, efficient, fast, and reliable under large orientation errors as can be seen in the associated video result [supplementary material]. Starting from the simulation results, the visual servo task with an initial camera rotation of 180° was not achievable for a standard IBVS during which the robotic arms went through chattering, and the camera retreated and finally did not converge. The same task was not achievable for ESM and a MESM controller. Whereas the proposed SPJASM controller converged without camera advance/retreat problem as shown in the associated video [supplementary material].

SPJ controller starts in the roll first mode and converged the feature error sharply for the rotational part and then in the second stage all image features decay simultaneously. This is a praiseworthy feature of our controller while in roll first mode it also considers the whole error vector and does not let other feature errors go astray. It rather keeps an eye on all

TABLE 10. Comparison of complexities of the visual servoing schemes.

Visual servoing Scheme	Limitation	Camera Advance/ Retreat motion	Can Handle 180° Error	Computational Complexity
IBVS (Desired Features) [3]	Camera Retreat	Yes Camera retreats	No	1d+1fj= 2
IBVS (Current Features)[3]	Camera Advance	Yes Camera advance	No	2c+1fj= 3
IBVS (Cylindrical)[22]	Curved trajectory for Translational motion	No Advance / retreat motion	Yes	2c+1fj= 3
PBVS [3]	Convergence depends upon pose estimation and camera parameters	No Advance / retreat motion	Yes	2pb+1fj+1ps=4
Switched Part Jacobian (Proposed)	Needs rotational error between the current and the desired frame	No Advance / retreat motion,	Yes (upto 180°) Quadratic convergence	2m+1sw+1ps-1pj =+3
Partitioned Control [29]	Uses extra features in the image plane cannot handle objects on tilted plane	No Advance / retreat motion	Yes	1d+1fj+1p+1ef=4
2-1/2- D visual servoing[18]	Use a mix of 2D and 3D features in the formation of feature jacobian	No Advance / retreat motion	Yes (upto 180°)	2c+1fj+1hf= 4
ESM[15]	Cannot handle very large rotational error about the optical axis Singularities at 90°and 180°	No Advance / retreat motion, Quadratic convergence	No	2m+1fj=3
MESM[35]	Cannot handle very large rotational error about the optical axis Have to multiply an inverse of a T6x6 transformation matrix in the control law	No Advance / retreat motion, Quadratic convergence	No	2m+1fj+1ps+1t=5
Switch Control (IBVS +PBVS)([20]	Convergence depends upon pose estimation and camera parameters	No Advance / retreat motion	Yes (upto 180°)	1c+1fj+1sw+2pb =5
Adaptive Switch partitioned control [26]	3-stage switching control, uses extra feature	No Advance / retreat motion	Yes (upto 179°)	2m+1fj+2sw+1ef=+6
AWMJ[28]	Use of 3 extra parameters to be tuned at prior. Camera retreats for large orientation errors	Low velocities, low energy consumption	No (upto 165°)	2m+1fj+3ef=+6

Complexity Measure: IBVS desired feature=1d, IBVS current feature=2c, IBVS mean of the pseudo inverse of the feature Jacobian =2m, switching control =1s, Partition control=+1p, part Jacobian = -1pj, full jacobian = 1fj, Extra Derived features= +1ef, Hybrid features 2D + 3D=+1hf, PBVS =2pb, use of 3D Pose = +1ps, use of transform matrix T = 1t.

the errors of the feature while in RFM and once getting to the defined switching threshold, it switches to the ASM IBVS controller that converges simultaneously all the remaining features error smoothly.

The proposed SPJ controller converges the visual task comprising large orientation errors by splitting the task into two stages. The first stage uses an adaptive gain SPJASM controller which neutralizes the large rotational error in the roll first mode to a roll error angle threshold. While the second stage uses an ASM controller to converge the remaining visual feature error to zero. The SPJASM controller and directly defined manipulator Jacobian have additional desirable properties like producing small joint velocities and consuming lesser joint energies for completing the tasks. Moreover, the task Jacobian condition number remains unity for the first stage during the RFM part of the control while allowing control over the joint velocities, joint directions, and manipulator motion in the 3D space. The SPJ controller makes the robotic arms follow a predictable and minimalistic trajectory in the 3D space.

The time taken to complete the task is also very short as one iteration takes approximately 90 ms in our programming (including detection, feature extraction, and program execution, which may differ from others and depends upon the computer hardware, robotic arm, programming libraries and functions called during program execution). As a measure of time, we noted that in a real robotic arm experiment, a task involving 180 Degrees of rotational error took approximately 40 sec to complete. Therefore, we have plotted our graphical results against the iterations scale to better understand how many program cycles are required to converge the task. Generally, the proposed method is faster than the ESM and MESM-based methods due to the use of computationally efficient part manipulator Jacobian as shown in Table 10.

MESM method computes a full end effector manipulator Jacobian at each iteration of the loop. This end-effector jacobian takes a lot of operations for calculating in the base frame. Then this base frame jacobian is to be transformed to end eff jacobian by multiplying to a 6×6 transformation matrix containing a diagonal rotation matrix from the base

to the end-effector frame. The computational cost of computing manipulator Jacobian in the end effector frame is given in [46].

A comparison of the IBVS schemes for handling large orientation errors around the camera optical axis in the visual servoing literature is given in Table 10, which shows a systematic analysis of the complexity involved in the IBVS schemes for handling large orientation errors. It is evident by looking at Table 10 that the proposed SPJASM control scheme is the only method that can handle very large orientation errors up to 180 degrees without camera advance/retreat motion with a quadratic convergence rate at the lowest complexity index.

The simulation and experiment showed that our proposed scheme can efficiently achieve visual servoing tasks under large rotational errors around the camera's optical axis (up to 180 degrees) and it can also work for pure translational motions. Furthermore, it can also be used for objects placed on a tilted plane and could be effectively deployed on the real-world robotics arm for tasks comprising complex roto translational motions. The proposed controller was also compared to other established controllers. It compares the computational complexity added to the controller by using ESM and MESM methods as compared to the proposed scheme, a systematic analysis is presented in Table 10.

Additional desirable attributes of the SPJASM controllers include lesser joint velocities, consuming minimum joint energies, and a well-conditioned task Jacobian condition number (unity for the first phase during the SPJ RFM control mode). Moreover, the SPJ controllers follow a minimal trajectory in the 3D space which is contrary to the IBVS controller. SPJASM scheme provides ease of computation by using the direct user-defined part manipulator jacobian. Therefore, SPJASM controllers are the preferred control method for the IBVS of robotic arms comprising large rotational error as it is the most effective and efficient among the available controllers as most of them are discussed in this paper [15], [28], [35].

Nevertheless, we would like to point out a limitation of using a switching controller like the SPJ and other partitioned scheme which treats rotational and translational error separately, that it does not induce a straight-line motion of image points while converging in the image plane, but rather it follows a circular trajectory first and then follows the trajectory of the standard IBVS scheme.

Also switching the controller can be difficult while handshaking and it can be sensed in the image space if the gains are not tuned properly. As shown in Fig. 10 and Fig. 11, in a real robotic arm task, where switching of the controller showed a wavy velocity profile that caused a disturbance in the image feature trajectory as well. However, we figured out that it is an inherent property of the switching IBVS controllers that the switching controllers' profiles are wavy, especially while the handshaking of the two controllers. The individual gain tuning is a difficult area to manage for the two controllers, and finding a universally tuned adaptive gain controller for the two-stage switching control scheme valid under all conditions is still room for improvement in this scheme.

The IBVS schemes cause the features to first move away from the desired values and then come back towards the target positions. This behavior is normal and observable in other IBVS switching schemes [8], [9]. Please note that the switching of the controllers can be made smooth if the gains are properly tuned separately for two stages of the controller, as the switching of the controller is almost seamless as shown in Fig.8a and Fig. 5b.

VI. CONCLUSION

In this paper, we have proposed a novel solution for handling the IBVS for robotic arms with an eye-in-hand configuration where the error comprises large rotational errors around the camera's optical axis (up to 180 degrees). To achieve this task a switched part manipulator Jacobian based on a rotation first strategy inspired by humans was utilized. The proposed solution is based on the efficient second-order minimization (PMJ) approach. A directly-defined part manipulator Jacobian was introduced for the formation of partitioned task Jacobian in the visual servo control law. This eliminates the need for computing the full end effector Jacobian at each iteration.

The results showed that the SPJASM controller remained stable and successfully completed all the tasks including pure rotational, pure translational, and complex rot-translational motions with a significant 40% increase in the convergence rate, and the convergence zone was extended up to 180 degrees for rotational errors about the camera optical axis. This was not possible earlier using the Cartesian point coordinate-based IBVS schemes. This was achieved in a computationally efficient way along with a 90 % reduction in the joint velocities and energy requirements, while the task jacobian remained well-conditioned. The proposed Switched-Part Jacobian (SPJ) controller uses a minimum computational complexity as compared to other available solutions as shown in Table 10. Thus the performance of the SPJASM controller surpasses other controllers in all major traits. The proposed SPJ control can be extended to other visual servoing schemes which uses the task Jacobian approach.

Therefore, it is concluded that the proposed SPJASM scheme provides a visual feature error constraint-free visual servoing experience, yet it is effective, efficient, and computationally inexpensive. Considering the desirable properties of the SPJ approach, it is recommended to be used for visual servoing in confined spaces, aerial robotics, and for assistive and Wheelchair mounted robotic arms. In the future, SPJ controllers will be tested with an adaptive online learning system using deep learning and markerless visual servoing and, it will be further extended to find applications in medical robotics, especially in minimally invasive surgical cases.

ACKNOWLEDGMENT

The authors would like to thank the support extended in the completion of this work by Markus Rickert for

his support with Robotics-Library® and also would like to thank Dr. Wenpeng Gao of the Biomedical Department, HIT, China, for the experimental setup and Khadija Zubair for providing valuable feedback in the preparation of this manuscript.

REFERENCES

[1] M. M. A. de Graaf, S. Ben Allouch, and J. A. G. M. van Dijk, "Why would i use this in my home? A model of domestic social robot acceptance," *Hum.-Comput. Interact.*, vol. 34, no. 2, pp. 115–173, Mar. 2019, doi: 10.1080/07370024.2017.1312406.

[2] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, "A survey of robots in healthcare," *Technologies*, vol. 9, no. 1, p. 8, Jan. 2021, doi: 10.3390/technologies9010008.

[3] S. Hutchinson and F. Chaumette, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[4] A. Palla, G. Meoni, L. Fanucci, and A. Frigerio, "Position based visual servoing control of a wheelchair mounter robotic arm using parallel tracking and mapping of task objects," *ICST Trans. Ambient Syst.*, vol. 4, no. 13, May 2017, Art. no. 152545, doi: 10.4108/eai.17-5-2017.152545.

[5] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 5, pp. 967–983, Oct. 2011, doi: 10.1109/TMECH.2010.2063710.

[6] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Auto. Robots*, vol. 42, no. 5, pp. 957–975, Jun. 2018, doi: 10.1007/s10514-017-9677-2.

[7] A. Singh, V. Kalaichelvi, and R. Karthikeyan, "A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks," *Cogent Eng.*, vol. 9, no. 1, Dec. 2022, Art. no. 2050020, doi: 10.1080/23311916.2022.2050020.

[8] H. W. Ka, "Three dimensional computer vision-based alternative control method for assistive robotic manipulator," *Int. J. Adv. Robot. Autom.*, vol. 1, no. 1, pp. 1–6, Feb. 2016, doi: 10.15226/2473-3032/1/1/00104.

[9] L.-V. Bruno-Siciliano and L. Sciavicco, *Robotics: Modeling, Planning, and Control*, vol. 16, no. 4. London, U.K.: Springer-Verlag, 2009, doi: 10.1007/978-1-84628-642-1.

[10] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 12, pp. 2633–2651, Dec. 2016, doi: 10.1109/TVCG.2015.2513408.

[11] C.-S. Chung, H. Wang, and R. A. Cooper, "Autonomous function of wheelchair-mounted robotic manipulators to perform daily activities," in *Proc. IEEE 13th Int. Conf. Rehabil. Robot. (ICORR)*, Jun. 2013, pp. 1–6, doi: 10.1109/ICORR.2013.6650378.

[12] H. Jiang, J. P. Wachs, and B. S. Duerstock, "Integrated vision-based robotic arm interface for operators with upper limb mobility impairments," in *Proc. IEEE 13th Int. Conf. Rehabil. Robot. (ICORR)*, no. 1, Jun. 2013, pp. 1–6, doi: 10.1109/ICORR.2013.6650447.

[13] O. Bourquardez, R. Mahony, T. Hamel, and F. Chaumette, "Stability and performance of image based visual servo control using first order spherical image moments," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Oct. 2006, pp. 4304–4309, doi: 10.1109/IROS.2006.281963.

[14] F. Chaumette, "Potential problems of unstability and divergence in image-based and position-based visual servoing," in *Proc. Eur. Control Conf. (ECC)*, 1999, pp. 4549–4554, doi: 10.23919/ecc.1999.7100052.

[15] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2004, pp. 1843–1848, doi: 10.1109/ROBOT.2004.1308092.

[16] Z. Arif and Y. Fu, "Mix frame visual servo control framework for autonomous assistive robotic arms," *Sensors*, vol. 22, no. 2, p. 642, Jan. 2022, doi: 10.3390/s22020642.

[17] H. Jiang, T. Zhang, J. P. Wachs, and B. S. Duerstock, "Enhanced control of a wheelchair-mounted robotic manipulator using 3-D vision and multimodal interaction," *Comput. Vis. Image Understand.*, vol. 149, pp. 21–31, Aug. 2016, doi: 10.1016/j.cviu.2016.03.015.

[18] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999, doi: 10.1109/70.760345.

[19] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 507–515, 2001, doi: 10.1109/70.954764.

[20] N. R. Gans and S. A. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 530–540, Jun. 2007, doi: 10.1109/TRO.2007.895067.

[21] O. Kermorgant and F. Chaumette, "Combining IBVS and PBVS to ensure the visibility constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Sep. 2011, pp. 2849–2854, doi: 10.1109/iros.2011.6094589.

[22] M. Iwatsuki and N. Okiyama, "Rotation-oriented visual servoing based on cylindrical coordinates," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 4198–4203, doi: 10.1109/robot.2002.1014411.

[23] M. Iwatsuki and N. Okiyama, "A new formulation of visual servoing based on cylindrical coordinate system," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 266–273, Apr. 2005, doi: 10.1109/TRO.2004.837242.

[24] P. I. Corke, F. Spindler, and F. Chaumette, "Combining Cartesian and polar coordinates in IBVS," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 5962–5967, doi: 10.1109/IROS.2009.5354569.

[25] G. Ye, W. Li, H. Wan, and H. Lou, "Novel two-stage hybrid IBVS controller combining Cartesian and polar based methods," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2015, pp. 397–402, doi: 10.1109/ICMA.2015.7237518.

[26] A. Ghasemi, P. Li, and W.-F. Xie, "Adaptive switch image-based visual servoing for industrial robots," *Int. J. Control Automat. Syst.*, vol. 18, no. 5, pp. 1324–1334, 2019, doi: 10.1007/s12555-018-0753-y.

[27] A. Ghasemi and W.-F. Xie, "Adaptive image-based visual servoing of 6 DOF robots using switch approach," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2018, pp. 1210–1215, doi: 10.1109/ICInfA.2018.8812440.

[28] A. Taherian, A. H. Mazinan, and M. Aliyari-Shoorehdeli, "Image-based visual servoing improvement through utilization of adaptive control gain and pseudo-inverse of the weighted mean of the Jacobians," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106580, doi: 10.1016/j.compeleceng.2020.106580.

[29] P. Y. Oh and P. K. Allen, "Visual servoing by partitioning degrees of freedom," *IEEE Trans. Robot. Autom.*, vol. 17, no. 1, pp. 1–17, Feb. 2001, doi: 10.1109/70.917078.

[30] D. Katz, J. Kenney, and O. Brock, "How can robots succeed in unstructured environments?" in *Proc. Robot. Sci. Syst. Workshop Robot Manipulation*, 2008, pp. 1–6.

[31] D. Xu, J. Lu, P. Wang, Z. Zhang, and Z. Liang, "Partially decoupled image-based visual servoing using different sensitive features," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2233–2243, Aug. 2017, doi: 10.1109/TSMC.2016.2641951.

[32] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004, doi: 10.1109/TRO.2004.829463.

[33] F. Chaumette, "Visual servoing and visual tracking," in *Handbook of Robotics*, O. Siciliano and B. Khatib, Eds. Berlin, Germany: Springer, 2008, pp. 563–583.

[34] S. Trinh, F. Spindler, E. Marchand, and F. Chaumette, "A modular framework for model-based visual tracking using edge, texture and depth features," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Oct. 2018, pp. 89–96, doi: 10.1109/IROS.2018.8594003.

[35] O. Tahri and Y. Mezouar, "On visual servoing based on efficient second order minimization," *Robot. Auto. Syst.*, vol. 58, no. 5, pp. 712–719, May 2010, doi: 10.1016/j.robot.2009.11.003.

[36] A. Atawnih, D. Papageorgiou, and Z. Doulgeri, "Kinematic control of redundant robots with guaranteed joint limit avoidance," *Robot. Auto. Syst.*, vol. 79, pp. 122–131, May 2016, doi: 10.1016/j.robot.2016.01.006.

[37] J. Liu, Y. Tong, Z. Ju, and Y. Liu, "Novel method of obstacle avoidance planning for redundant sliding manipulators," *IEEE Access*, vol. 8, pp. 78608–78621, 2020, doi: 10.1109/ACCESS.2020.2990555.

[38] W. Bolton, "Process controllers," in *Instrumentation and Control Systems*, 2nd ed. London, U.K.: Newnes, 2015, pp. 101–124.

[39] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 244–257, Feb. 2014, doi: 10.1109/TRO.2013.2281560.

[40] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Oct. 2016, pp. 4193–4198, doi: 10.1109/IROS.2016.7759617.

[41] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[42] É. Marchand, F. Spindler, and F. Chaumette, "ViSP: A generic software platform for visual servoing," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Nov. 2011.
[43] M. Zhong, Y. Zhang, X. Yang, Y. Yao, J. Guo, Y. Wang, and Y. Liu, "Assistive grasping based on laser-point detection with application to wheelchair-mounted robotic arms," *Sensors*, vol. 19, no. 2, p. 303, Jan. 2019, doi: 10.3390/s19020303.
[44] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, "Robust online model predictive control for a constrained image-based visual servoing," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2242–2250, Apr. 2016, doi: 10.1109/TIE.2015.2510505.
[45] S. Benhimane and E. Malis, "Homography-based 2D visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2006, pp. 2397–2402, doi: 10.1109/ROBOT.2006.1642061.
[46] D. E. Orin and W. W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *Int. J. Robot. Res.*, vol. 3, no. 4, pp. 66–75, Dec. 1984, doi: 10.1177/027836498400300404.

ZUBAIR ARIF was born in Karachi, Pakistan. He received the B.E. degree in mechanical engineering and the M.E. degree in industrial engineering from the NED University of Engineering & Technology, Karachi, Pakistan, in 2007 and 2012, respectively.

He has been with the National Space Agency of Pakistan, since 2008, as a Senior Research Engineer. Currently, he is a Ph.D. Research Scholar with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology (HIT), China. He is a certified professional in industrial automation and has received several pieces of training in countries, including Germany, Japan, and, China. He has published two papers in refereed journals and refereed conference proceedings. His research interests includes robotics, vision-based control, space robotics, intelligent systems, and automation and controls.

Prof. Arif has received several awards during his services in Pakistan. He has been awarded the scholarship from Asia Pacific Space Cooperation Organization (APSCO) and China's Scholarship Council for his current Ph.D. Program in China.

YILI FU (Member, IEEE) received the Ph.D. degree in mechatronics and automation from the Harbin Institute of Technology (HIT), Harbin, China, in 1996. He worked with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, from 1997 to 2000. Since February 2000, he has been with HIT. He is currently a Professor with the Robotics Institute. He has published over 100 papers in refereed journals and refereed conference proceedings. His research interests include collaborative robots, medical robots, mobile robots, humanoid robots, and machine intelligence.

He has received numerous research awards from international conferences in robotics and automation and government agencies. He serves as an Editor in seven journals, such as the *International Journal of Humanoid Robotics*, *Robotics*, and *Biomimetic*. He serves as the general chair, program chair, and organizing chair at over ten IEEE conferences.

MUHAMMAD KASHIF SIDDIQUI was born in Lahore, Pakistan. He received the B.E. degree in space electronics engineering from the Institute of Space Technology, Pakistan, in 2010, and the M.E. degree in automation engineering from Aalto University, Finland, in 2015.

He has been with the National Space Agency of Pakistan, as a Senior Research Engineer. Currently, he is a Ph.D. Research Scholar with the Fluid Flow Control and Automation Department, Harbin Institute of Technology (HIT), China. His current research interests include fluid power control, robotics, and automation.

Prof. Siddique has received the Asia Pacific Space Cooperation Organization (APSCO) and China Scholarship Council's Scholarships for his current Ph.D. program in China.

FUHAI ZHANG received the Ph.D. degree in mechatronic engineering from the Harbin Institute of Technology, Harbin, China, in 2010.

He is currently an Associate Professor with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His current research interests include rehabilitation robotics and space robotics.