Expert Systems With Applications 72 (2017) 344–356

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Intelligent visual servoing with extreme learning machine and fuzzy logic

Tolga Yüksel

Department of Electrical and Electronics Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey

ARTICLE INFO

Article history:
Received 12 March 2016
Revised 19 August 2016
Accepted 20 October 2016
Available online 21 October 2016

Keywords:
Image-based visual servoing
Extreme learning machine
Fuzzy logic

ABSTRACT

While visual servoing (VS) provides the ability of motion using vision for robot manipulators, the approaches for a better VS have to deal with three common problems: obtaining the interaction matrix and its pseudoinverse for defined feature points, finding an appropriate gain value for the VS controller and keeping the features in the field of view (FOV) for VS permanency.

In this study, a new intelligent image-based visual servoing (IBVS) system for eye-in-hand configured robot manipulators using extreme learning machine (ELM) and fuzzy logic (FL) is proposed to solve these common problems of VS in a single system. As the first stage of the system, the pseudoinverse of the interaction matrix is approximated using trained ELMs which do not need hidden layer tuning. As the second stage, the classical IBVS controller is modified by a differential equation regarding initial velocity continuity and an appropriate gain in each loop is assigned by an FL unit to provide fast convergence within velocity limits. This unit also promotes manipulability of the manipulator to avoid singularities. As the last stage of the proposed system, regions are defined in the image plane to take precautions before feature missing. When a feature comes close to the edge of a restricted region, an FL unit is activated to obtain negative linear velocities in x and y direction which will be added to the instant velocities to drag the features towards the center of the FOV. In addition to these abilities, some VS metrics are redefined analytically to standardize the performance metric definitions of VS. To show the performance of the proposed system, simulation results of the classical and the proposed IBVS system under practical disturbances are presented for visual servoing of a Puma 560 arm. The advantages of singular matrix and joint configuration avoidance, adaptive gain with smooth gain surface, decreased convergence time within velocity limits, initial velocity continuity, FOV keeping with smooth velocity assurance, redefined VS metrics for standardization and robustness against disturbances are proved by variety of simulations. The simulation results also verify that the proposed system utilizing intelligent methods like ELM and FL is capable of dealing with common problems of VS and achieves sufficient results in terms of VS metrics.

1. Introduction

The technique of controlling a robot using the meaningful information obtained from visual feedback is named as visual servoing (VS). In VS, the camera image is used to define k feature points and the coordinates of these features in image plane form vector $\boldsymbol{s}$. Error signal vector is derived from the difference between the desired vector $\boldsymbol{s}^*$ and $\boldsymbol{s}$. This error vector is used to obtain the control signals of the end effector in terms of velocities by a VS control law. In this context, VS methods are mentioned in two main titles. Image-based visual servoing (IBVS) uses $\boldsymbol{s}$ obtained from the image directly but position-based visual servoing (PBVS) uses $\boldsymbol{s}$ obtained from 3D parameter estimations of image and robot pose (Chaumette & Hutchinson, 2006). Some hybrid approaches like 2½ and partitioned control can be found in the literature (Corke & Hutchinson, 2001; Malis, Chaumette, & Boudet, 1999). It is important to note that featureless visual servoing approaches like kernel-based or luminance-based methods are active research areas (Collewet, Marchand, & Chaumette, 2008; Kallem, Swensen, Hager, & Cowan, 2007) but feature-based methods will continue their royalty.

In contrast to PBVS, IBVS does not need any pose estimation and it is robust against depth estimation errors. These advantages make IBVS more attractive for practical applications and this study is focused on IBVS and common problems of VS. Furthermore, the configuration of the camera and the end effector should be considered before VS approaches. As a popular configuration for manipulators, mobile robots, unmanned air vehicles (UAVs) and medical applications, eye-in-hand configuration is chosen for this study. Besides these preferences, it should be noted that stereo vision

E-mail address: tolga.yuksel@bilecik.edu.tr

http://dx.doi.org/10.1016/j.eswa.2016.10.048
0957-4174/

applications of VS are becoming popular and a comparison of stereo and mono IBVS can be found in Mohebbi, Keshmiri, and Xie (2016).

There appears to be a limited analysis of three common problems of VS in the literature. The first one is obtaining the interaction matrix and its pseudoinverse. The interaction matrix (or named as image Jacobian matrix) defines the relation between the velocities of a 3D point in real world and 2D image plane. It is obtained using camera intrinsic parameters, projection of 3D points to the 2D image plane and estimated depth of a 3D point in real world. Good depth estimation approaches can be found in Chaumette and Hutchinson (2006). Although the interaction matrix can be easily obtained for different types of features with a good depth estimation, poor camera calibration and feature noise may disturb this matrix. Furthermore, the pseudoinverse of this matrix is used by VS control law and Chaumette mentioned that it may not be accessible at certain VS scenarios (Chaumette, 1998). In the scenario named as Chaumette Conundrum, the transition from starting feature points to target feature points is obtained by only rotation around camera center. This causes singular interaction matrix and the control law is offline. An approximated interaction matrix can be a good solution to this problem. Kumar et al. defined the interaction matrix with manipulator Jacobian to approximate manipulator angular positions of an eye-to-hand configured redundant manipulator with stereo cameras using a self-organizing map (SOM) but eye-in-hand configuration and IBVS are neglected in the study (Kumar & Behera, 2010). Kosmopoulos proposed a linear feature model with least square estimator as an approximator for a VS system with a six-joint manipulator to obtain illumination invariance however, the study is based on an industrial robot which follows only a single trajectory (Kosmopoulos, 2011). Sebastian et al. proposed an approximator for estimating the interaction matrix using Broyden, recursive least squares and Kalman filtering for an uncalibrated eye-to-hand configured manipulator without any starting and target feature definitions (Sebastian, Pari, Angel, & Traslosheros, 2009). Zhong et al. proposed a hybrid interaction matrix approximator employing robust Kalman filtering cooperated Elman Neural Network with delayed interaction matrix approximations and image features as inputs (Zhong, Zhong, & Peng, 2013). In this system, the robustness against noise is very limited. In their recent study, they used NN assisted Kalman filtering to improve the robustness for the interaction matrix approximation but the results of the velocities of the end effector are neglected (Zhong, Zhong, & Peng, 2015). Goncalves et al. tried to model robot velocity signals of an eye-to-hand configured manipulator without the interaction matrix using inverse fuzzy modeling with manipulator joint angles and feature variations in time (Gonçalves, Mendonça, Sousa, & Pinto, 2008). They used fuzzy modeling which is lack of self-learning mechanism and only eye-to-hand configuration is considered.

Some other approaches proposed VS systems without the interaction matrix. Miljkovic et al. proposed a hybrid IBVS system with classical IBVS and a neural network (NN) reinforcement based controller without interaction matrix approximation (Miljković, Mitić, Lazarević, & Babić, 2013). They developed this system by mapping the image space to the actuator commands directly using Q-learning and SARSA but the actuator command approximation results are not shown in the study and the velocity signals show huge chatterings. Schramm and Morel identified the VS task as a path planning problem (Schramm & Morel, 2006). They proposed an analytical solution to the path planning of feature points between two pre-recorded images without any knowledge of the interaction matrix, but their study ignored controller side of VS.

The second problem of VS is finding an appropriate gain value for the VS controller. Fast convergence is one of the main goals of VS applications but an appropriate gain should be used by the VS controller under velocity limits. Furthermore, classical IBVS controllers neglect the requirement of initial velocity continuity which means that initial velocity values should be zero for practical applications. Instead of fixed gain, adaptive gain should be deployed in VS for fast convergence. Firstly, Mansard and Chaumette proposed a hybrid IBVS-PBVS system with adaptive gain by defining a VS controller using a non-homogenous differential equation and elaborating the transient and steady state solution parts (Mansard & Chaumette, 2007). Besides this approach is only designed for smooth transition from IBVS to PBVS, convergence speed and velocity limitations of the manipulator are neglected. In another study, Kermorgant and Chaumette proposed a framework for the control of a multi-sensor manipulator which means that the features from an eye-in-hand camera, an eye-to-hand camera and the constraints of the system are assumed as the feature set (Kermorgant & Chaumette, 2014). They proposed an LQ controller with weighted error terms and they utilized adaptive gain to slow down the system in the vicinity of system constraints.

The last problem is keeping the features in the field of view (FOV). Feature loss may result in divergence and sudden velocity changes during the VS tasks. To avoid these catastrophic results, Corke et al. utilized collision avoidance methods for robot manipulators and they proposed a differentiable potential function that repels the feature points from the boundary of the image plane (Corke & Hutchinson, 2001). This function is a quadratic function depending on the shortest distance from the image point to the edge of the image plane. In the study, they mentioned that the use of a potential field may raise the issue of chattering where the feature points oscillate in and out of the potential field. Gans et al. proposed a FOV keeping system which uses a task function based on feature mean instead of goal image and they implemented the system on a mobile robot (Gans, Hu, & Nagarajan, 2011). This system is focused on mobile robots only. Mean errors in pixels are also quite high and feature motion characteristics and velocity deviations in time are neglected in the study. Chesi et al. proposed another FOV keeping system that changes linear and angular velocity control laws or implements a backward motion according to the FOV border and the feature conditions (Chesi, Hashimoto, Prattichizzo, & Vicino, 2004). This approach causes chattering in the translation and rotation velocities. It is mentioned in the study that switching between control laws causes this weakness and backward motion may cause divergence from the target features. Zhong et al. also mentioned keeping FOV in their study by using NN assisted Kalman filtering but its proof is not given clearly in their study (Zhong et al., 2013). Besides a learning controller, the controller proposed by Miljkovic al. ensures the visibility using a reward function but the moving actions according to this function are too hard to realize (Miljković et al., 2013).

In addition to these literature review, it should be noted that most of these approaches focus on a single VS problem without considering other problems but all of these problems are active in practice. These studies in the literature convincingly demonstrate the need for new approaches to these problems. In this study, a new intelligent IBVS system for eye-in-hand configured robot manipulators using extreme learning machine and fuzzy logic is proposed to solve the three common problems of VS in a single system. As an approximation solution to non-invertible and non-square interaction matrix, the first stage of the proposed system uses extreme learning machine (ELM). ELM is proposed by Huang as a “generalized” single hidden layer feedforward neural network providing a unified learning platform with a widespread type of nonlinear mappings and it can be applied in regression and multiclass classification applications directly (Huang, Zhou, Ding, & Zhang, 2012). ELM tends to have better generalization performance than other approximators such as support vector machines (SVMs), its variants and classical NNs. The proposed IBVS system will be

the first time that ELM is used in a VS system. In the proposed system, the outputs of this approximator are not the elements of the pseudoinverse of the interaction matrix but the multiplication of the interaction matrix and error with negative sign to fix the number of outputs. It is shown that ELM shows sufficient performance at learning phase of interaction matrix approximation, it avoids interaction matrix singularities efficiently and robustness against practical disturbances like feature noise and camera calibration error is obtained. Furthermore, simulation results show that the approximation errors with ELM does not affect the other stages of the proposed system.

A continuous velocity controller with adaptive gain approach is proposed as the second stage of the proposed IBVS system. The controller uses a differential equation similar to Mansard and Chaumette (2007) with a fuzzy logic (FL) unit modification to assign appropriate gain in each loop. This provides smooth gain adaptation and fast convergence within velocity limits. Furthermore, all VS systems should start with zero velocity to obtain a smooth starting motion in the realization of the system. This opportunity comes up with this proposed system. Besides these superiorities, it is clear that velocity signal profiles affect the position and orientation profiles of a manipulator while it is following a generated path. Yoshikawa mentioned that these profiles should be examined not only to show the motion ability but also to avoid manipulator singularities and he proposed a quantitative index of this ability as manipulability (Yoshikawa, 1985). According to these impacts, the inputs of FL unit are chosen as error, error derivative which define the dynamics of an IBVS system and manipulability as the motion ability index. The membership functions of FL are defined as gaussian functions and the rulebase is tuned to obtain a smooth adaptive gain surface and increased manipulability. Here, it should be noted that the proposed system assumes the VS controller as a kinematic controller. It neglects robot dynamics model and focus on the effects of VS actions as in Chesi and Hung (2007), Chesi et al. (2004), Collewet et al. (2008), Corke and Hutchinson (2001), Kosmopoulos (2011) and Mezouar and Chaumette (2002). VS controllers including robot dynamics and control can be found in Lizarralde, Leite, Hsu, and Costa (2013), Nasisi and Carelli (2003), Wang, Jiang, Chen, and Liu (2012).

The proposed system also utilizes FL for FOV keeping as the third stage. Firstly, image FOV regions are defined to take precautions before feature missing. Then, instead of backward countermotion as in Chesi et al. (2004) and analytical potential functions as in Corke and Hutchinson (2001) and Gans et al. (2011), a negative motion in the counter direction of the approached FOV border is defined using FL to direct the features through the image center smoothly.

In this study, it is also aimed to give a clear sight to evaluation metrics of VS. These metrics should be defined analytically instead of just evaluating velocity and feature trajectory graphs of VS. These metrics are convergence time, error costs, trajectory lengths, feature trajectory length, curvatures of the trajectories.

On the whole, the main contributions of this study can be referred as follows. Firstly, according to author's knowledge, this is the first time that a single IBVS system solves three common problems of VS. Secondly, this is the first time that ELM is used in a VS system. It shows high approximation, robustness performance and provides interaction matrix singularity avoidance. Thirdly, control law and FL based smooth adaptive gain in the study increases convergence speed with low velocity profiles that can be a significant step in the realization of VS systems. Again, initial velocity continuity is essential in the realization and this opportunity comes with this system. Lastly, this is the first time that a VS system utilizes an intelligent unit, FL for keeping FOV. FL units assist achievements of the system and provide easier integration of user experience to the system. Furthermore, defining VS metrics analytically will provide standardization in the field of VS. Simulation results prove these capabilities of the proposed system under practical disturbances.

The layout of the paper is organized as follows. In the following section, the details of the proposed intelligent IBVS system are given. The evaluation metrics for VS systems are summarized in Section 3. The proposed IBVS system is simulated for an eye-in-hand configured Puma560 manipulator and 4 feature points. The simulation results of classical IBVS and the proposed IBVS system for ideal and under practical disturbance cases are shown in Section 4. In the last section, conclusions of the study and future goals are finally given.

2. The proposed intelligent IBVS system

This section provides a detailed presentation of the proposed intelligent IBVS system. Here, it should be noted that the following descriptions aim an IBVS system with point features for eye-in-hand configured robot manipulators. The block diagram of the proposed system is shown in Fig. 1.

The description starts with image plane projection. Let's assume that an eye-in-hand camera configured robot manipulator is viewing k feature points in 3D. The coordinates of these feature points in camera image plane are given as

$$s_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^T, \quad \forall i \in \{1 \dots k\} \tag{1}$$

where u_i, v_i are coordinates in u-v image plane. These vectors form a matrix to define the behavior of the fixed-motionless feature points

$$\boldsymbol{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_k \end{bmatrix}^T \in \mathbb{R}^{2\times k} \tag{2}$$

with $\boldsymbol{s}^*$ as the desired fixed feature points matrix in the same dimensions. The main goal of all VS methods is to converge error to zero

$$\boldsymbol{e}(t) = \boldsymbol{s}(t) - \boldsymbol{s}^* \tag{3}$$

After these notations, velocity of a point is defined in real world in terms of linear velocity $\boldsymbol{v}$ and angular velocity $\boldsymbol{\Omega}$

$$\dot{\boldsymbol{P}}_i = -\boldsymbol{\Omega} \times \boldsymbol{P}_i - \boldsymbol{v} \tag{4}$$

where $\dot{\boldsymbol{P}}_i, \boldsymbol{P}_i \in \mathbb{R}^6$ define point velocity vector and point position vector in 3D, respectively. The projection of linear and angular velocities $\boldsymbol{v}$ in (4) to the image plane is defined using s_i, focal length of the camera λ_f and depth of the feature Z

$$\begin{aligned} \boldsymbol{\upsilon} &= \begin{bmatrix} v_x & v_y & v_z & w_x & w_y & w_z \end{bmatrix}^T, \begin{bmatrix} \dot{u}_i \\ \dot{v}_i \end{bmatrix} \\ &= \begin{bmatrix} -\frac{\lambda_f}{Z} & 0 & \frac{u_i}{Z} & \frac{u_i.v_i}{\lambda_f} & -\frac{{\lambda_f}^2+{u_i}^2}{\lambda_f} & v_i \\ 0 & -\frac{\lambda_f}{Z} & \frac{v_i}{Z} & \frac{{\lambda_f}^2+{v_i}^2}{\lambda_f} & -\frac{u_i.v_i}{\lambda_f} & u_i \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \end{aligned} \tag{5}$$

The details of this projection can be found in many VS reviews (Chaumette & Hutchinson, 2006; Chaumette & Hutchinson, 2008; Corke, 2011) . This projection is generalized as

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{L}_{\boldsymbol{s}}.\boldsymbol{\upsilon} \tag{6}$$

where $\boldsymbol{L}_{\boldsymbol{s}} \in \mathbb{R}^{2k\times 6}$ is the interaction matrix (or named as image Jacobian). It is hard to acquire the real depth of a feature so its estimated value which is always assumed as the depth at $\boldsymbol{s}^*$ is used and $\boldsymbol{L}_{\boldsymbol{s}}$ becomes $\hat{\boldsymbol{L}}_{\boldsymbol{s}}$ as the estimated interaction matrix. Depth estimation approaches can be found in Chaumette and Hutchinson (2006).

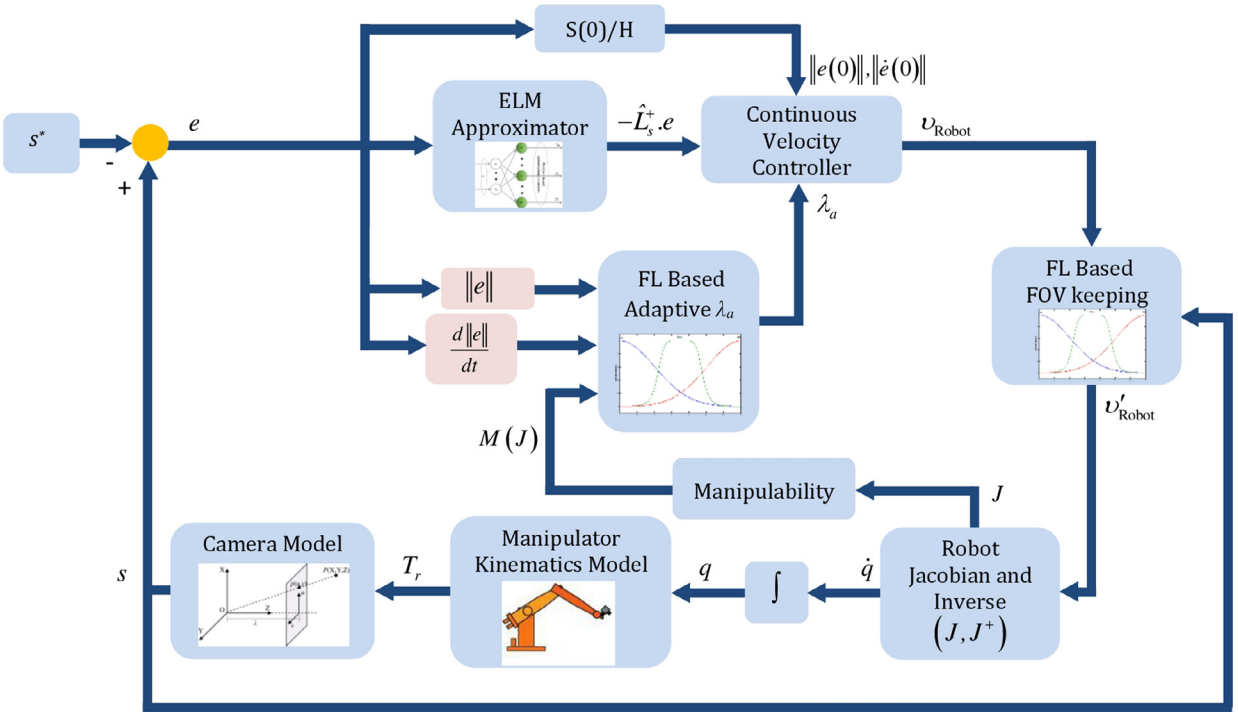

Fig. 1. The block diagram of the proposed IBVS system.

Before velocity control law design, two essential assumptions should be given:

Assumption 1. It is assumed that there is no transformation between the end effector frame ($\mathcal{F}_e$) and the camera frame ($\mathcal{F}_c$) with eye-in-hand configuration.

$$\mathcal{F}_c = T_e.\mathcal{F}_e,\ T_e = I^{4\times4}\ (I\text{: Identitymatrix}) \tag{7}$$

Assumption 2. All feature points are collinear and all depths are the same for all features. The features obey the collinearity rule defined in Tahri and Chaumette (2005) as A, B and C are constants

$$\frac{1}{Z} = Ax_i + By_i + C \tag{8}$$

Classical IBVS proposes a kinematic velocity controller with the Moore-Penrose pseudoinverse of the estimated interaction matrix $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$, the error vector $\boldsymbol{e}$ and a fixed gain λ to decrease the error exponentially

$$\left.\begin{array}{l}\dot{\boldsymbol{s}} = \boldsymbol{L}_s.\boldsymbol{\upsilon} \\ \dot{\boldsymbol{e}} + \lambda.\boldsymbol{e} = 0\end{array}\right\} \Rightarrow \boldsymbol{\upsilon} = -\lambda.\hat{\boldsymbol{L}}_s^{+}.\boldsymbol{e} \tag{9}$$

The performance of this velocity controller depends on $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$ and λ. It is not easy to use analytical $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$ because of configurations which may cause singularities as mentioned in Chaumette (1998) and errors in camera calibration parameters or feature noise. Then, an intelligent function approximator seems to be a promising solution to this problem. Single hidden layer feedforward neural networks (SLFNs), SVMs, FL and hybrid structures as adaptive neuro-fuzzy inference systems (ANFIS) are best-known approximators in the literature but they all have drawbacks. The hidden layer parameters of SLFNs and the user defined parameters of SVMs should be tuned wisely according to input-output mapping (Huang et al., 2012). Furthermore, these approximators suffer from slow learning speed because of slow gradient-based learning algorithms and iteratively tuned multi-hidden-layer of neurons. The performance of fuzzy function approximation is completely user dependent. ANFIS is also user dependent, parameters of its layers should be tuned wisely and it needs more time than SLFNs for regression (Jang & Sun, 1997). As a member of NN family, ELM generalizes SLFNs whose hidden layer doesn't need tuning and it brings fast learning. Also, it doesn't use multi-layers of neurons and different learning algorithms. The output function of ELM as one output node case for generalized SLFNs is Huang et al. (2012)

$$f_{ELM}^{L}(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i.h_i(x) = \boldsymbol{\beta}.\boldsymbol{h}(x) \tag{10}$$

where $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_L] \in \mathbb{R}^L$ is the vector of the output weights between the hidden layer of L nodes and the output node, $h_i(.)$ is the activation function and $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}) \quad h_2(\boldsymbol{x}) \quad \dots \quad h_L(\boldsymbol{x})]$ is the output vector of the hidden layer with respect to the input $\boldsymbol{x} \in \mathbb{R}^N$. Actually, $\boldsymbol{h}(\boldsymbol{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden-layer feature space (ELM feature space) as shown in Fig. 2 (Huang, 2015).

The activation functions of the green hidden layer nodes in Fig. 2 can be defined as sigmoid, sine or hard limiter functions. Traditional learning algorithms only tend to reach the smallest training error between output and target $\boldsymbol{T} = [\boldsymbol{t}_1, \dots, \boldsymbol{t}_L]^T$ but ELM aims to minimize the training error as well as the norm of the output weights

$$\left\|\hat{\boldsymbol{\beta}}.\boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{T}\right\| = \min_{\boldsymbol{\beta}} \left(\left\|\boldsymbol{\beta}.\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{T}\right\|\right) \tag{11}$$

where $\hat{\boldsymbol{\beta}}$ is the minimum norm least square solution of $\boldsymbol{\beta}.\boldsymbol{H} = \boldsymbol{T}$ with $\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}.\boldsymbol{T}$. $\boldsymbol{H}^{+}$ is Moore-Penrose pseudoinverse as in (9). In Huang et al. (2012), it is mentioned that this pseudoinverse can be found using orthogonal projection method, orthogonalization method, iterative method, and singular value decomposition (SVD). Note that this mapping also contains input weights $\boldsymbol{w}_j = [w_j^1 \quad w_j^2 \quad \dots \quad w_j^L] \in \mathbb{R}^L,\ \forall j \in \{1 \dots d\}$ connecting the jth hidden node, the input nodes and the bias of the jth hidden node b_j. ELM is proposed by Guang et al. and they summarized the learning

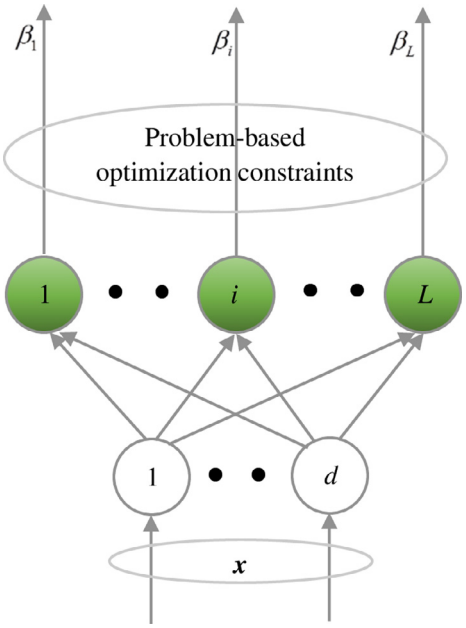

Fig. 2. ELM feature mapping with fully connected random hidden nodes.

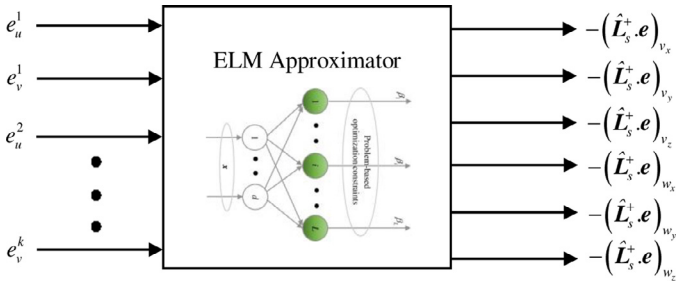

Fig. 3. ELM input-output mapping for interaction matrix approximation.

steps in their paper announcing ELM as follows (Huang, Zhu, & Siew, 2006):

Given a training set $\mathbf{X} = \{(\mathbf{x}_i)|\mathbf{x}_i \in \mathbb{R}^n, i = 1...N\}$, activation function $\mathbf{g}(\mathbf{x})$ and hidden node number N,

Step 1: Randomly assign input weight $\boldsymbol{w}_i$ and bias b_i, $i = 1...N$
Step 2: Calculate the hidden layer output matrix $\mathbf{H}$
Step 3: Calculate the output weight β, $\beta = \mathbf{H}^+.\mathbf{T}$ where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$

The studies on ELM tested its performance for popular function approximation data sets like sinc, Abalone, Delta ailerons from UCI Machine Learning Repository and the evaluation can be found in Huang et al. (2006, 2012). In Fig. 3, interaction matrix approximation in the proposed system is shown in details. The inputs of the approximator are the elements of the feature error vector in (3). Due to dimension dependency of $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+$ matrix to the number of feature points as given in (5) and (6), the outputs of ELM are chosen as each element of $-\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+.\boldsymbol{e}$ instead of $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+$. This approach provides fixed number of outputs which are independent of the number of feature points. Here, it should be noted that all inputs of ELM are normalized in order to use activation function more effectively.

Implementations of IBVS kinematic velocity controller approaches based on (9) neglect the fact that fast convergence within velocity limits should be provided for a VS system in order to increase the labor force in a manufacturing line. From a practical point of view, this shortcoming may result in low product numbers in a day. Furthermore, these classical controllers neglect another fact that a VS system which controls the motions of a manipulator should consider the effects of initial velocity value. Only studies in Kermorgant and Chaumette (2014) and Mansard and Chaumette (2007) refer these practical issues. Instead of (9), Mansard and Chaumette proposed a non-homogenous first order differential equation in Mansard and Chaumette (2007)

$$\dot{\boldsymbol{e}} = -\lambda.\boldsymbol{e} + \boldsymbol{\rho}(t) \tag{12}$$

where $\boldsymbol{\rho}(t)$ is a function of time. This function has to be chosen wisely to ensure the continuity and it has to be a transient function which will be equal to 0 at $t \to \infty$. This function is chosen as

$$\boldsymbol{\rho}(t) = \left(\|\dot{\boldsymbol{e}}(0)\| + \lambda.\|\boldsymbol{e}(0)\|\right).e^{-\gamma.t} \tag{13}$$

Here, γ defines the time constant of the transient response and it is chosen as $\gamma = 10.\lambda$ experimentally in Mansard and Chaumette (2007). By utilizing this approach, the proposed IBVS controller is defined using (9), (12) and (13)

$$\boldsymbol{v} = -\lambda.\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+.\boldsymbol{e} + \left(\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+\|\dot{\boldsymbol{e}}(0)\| + \lambda.\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+.\|\boldsymbol{e}(0)\|\right).e^{-\gamma.t} \tag{14}$$

Mansard and Chaumette proposed this controller for smooth transition between different VS approaches but convergence speed is neglected in their study. Fixed λ in (9) and (14) determines convergence time and velocity profile. If it is converted to adaptive, the system may show more efficient velocity profiles with shorter convergence time. The adaptive gain λ_a should be dependent on $\|\boldsymbol{e}\|$ and $d\|\boldsymbol{e}\|/dt$ which determine velocity profiles. As well as fast convergence with low velocity profiles, another term is considered in this proposed system while planning the velocity profiles. Path planning is deployed in robotic systems to avoid joint constraints and collusion. It requires trajectory generation to assign velocity profiles. In a VS application with trajectory generation, the trajectory is generated first. Then, the camera is steered to the desired orientation by using a VS controller as in Chesi and Hung (2007) and Mezouar and Chaumette (2002) but this task needs homography-based motion planning. In this study, this motion planning strategy is neglected but a constraint on manipulator motions, manipulability is considered. Manipulability is a global measure of velocity limits, the ability to change position-orientation of an end effector at an arbitrary trajectory point and it also provides joint singularity awareness (Yoshikawa, 1985). These properties make manipulability an important metric for path planning and it can be practical as the proposed system plans the motion in the image plane. As $\boldsymbol{J}(\boldsymbol{q})$ is the manipulator Jacobian at the joint configuration $\boldsymbol{q} \in \mathbb{R}^n$ for an n-joint manipulator, manipulability is defined as

$$M = \sqrt{\det\left(\boldsymbol{J}(\boldsymbol{q}).\boldsymbol{J}(\boldsymbol{q})^T\right)} \tag{15}$$

In this study, this term is also evaluated for λ_a to perform maximum manipulability while obtaining fast convergence with low velocity profiles. It is more convenient to use an intelligent unit for λ_a not only to obtain a smooth variation but also to include user experience. As shown in Fig. 1, an FL unit is deployed to obtain adaptive gain λ_a. To define linguistic rules of FL, the effects of $\|\boldsymbol{e}\|$, $d\|\boldsymbol{e}\|/dt$ and M on IBVS characteristics are considered. The effects of $\|\boldsymbol{e}\|$ and $d\|\boldsymbol{e}\|/dt$ on the IBVS controller are like classical PD controllers and the rules can be defined as in Jang and Sun (1997). In addition to these rules, λ_a should be high if M is high. This means that the manipulator is far from singularity and M allows high velocities. These linguistic inferences are transferred to the FL rulebase.

The output of an FL unit is based on FL unit type, input membership function types, aggregation type of membership functions, rulebase and defuzzification type. In this study, FL unit type for λ_a is Mamdani which uses fuzzy membership functions for output functions. Aggregation type is maximum and defuzzification

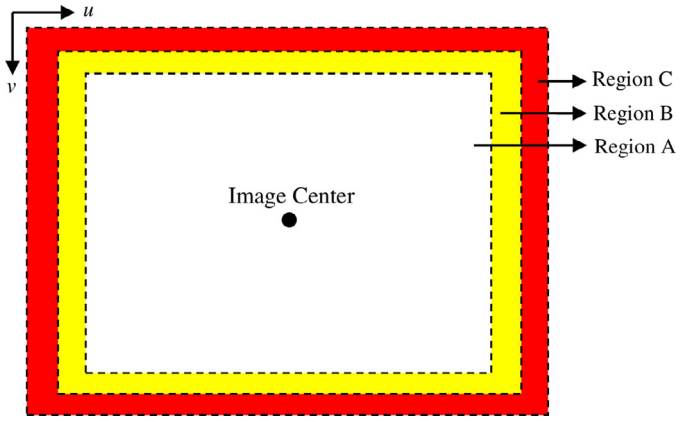

Fig. 4. Region illustration for keeping FOV.

type is centroid of area (COA) which is equal to the weighted average of the centroids of output membership functions ($\mu_i(\lambda_a)$) with weighting factors (ω_i) of input membership functions and ith rule

$$\omega_i = \max\left(\mu_j(\|e\|), \mu_k(d\|e\|/dt), \mu_m(\|M\|)\right), \ \forall j,k,m \in \{1 \dots z\}$$
$$\lambda_a = \frac{\sum_{i=1}^{n} \omega_i \int \mu_i(\lambda_a).\lambda_a.d\lambda_a}{\sum_{i=1}^{n} \omega_i \int \mu_i(\lambda_a).d\lambda_a} \tag{16}$$

After velocity control law, keeping features in FOV capability is considered. To keep the features in FOV, a method is to move the camera backwards in $-Z$ direction to direct features through image center (Chesi et al., 2004; Corke & Hutchinson, 2001). This approach can be effective for robot manipulators but two cases should be kept in mind. This approach may result in an increase of convergence time by suspending the desired goal position. Additionally, when this approach is used for a UAV application, it will increase the distance between the target and the UAV. Besides these cases, the motion in $-Z$ direction should not cause chattering in the velocities as in Chesi et al. (2004). Instead of backward motion, a negative linear motion in the counter direction of the approached image FOV border will keep features in sight. This negative motion can be in the form of a nonlinear quadratic function as in Corke and Hutchinson (2001) but to integrate user experience to the system, FL based approach is chosen in this study instead of an analytical approach.

It is hard to defeat feature missing when the features are on the border of the image. To prevent this, regions along the image boundary should be defined as in Corke and Hutchinson (2001). In this study, three image FOV regions according to u-v coordinates in image plane are defined as shown in Fig. 4.

In Fig. 4, Region A is assumed as the safe region where no response is required. Region B is the yellow region where FOV risk is possible and necessary precautions by FOV keeping approach should be taken. Region C is the red region and FOV keeping approach will drag features in the counter direction with full throttle. The consequent velocities after this FOV keeping approach are given as $\boldsymbol{v'} = [v'_x \quad v'_y \quad v_z \quad w_x \quad w_y \quad w_z]^T$ and

$$v'_x = v_x - v_x^{FOV}$$
$$v'_y = v_y - v_y^{FOV} \tag{17}$$

where v_x^{FOV} and v_y^{FOV} are counter velocities. These velocities should be a function of feature coordinates and region boundaries. Two FL units are used to define these velocities under the following conditions

$$s'_i = \{s_i \in \boldsymbol{s} | \min\max\{\boldsymbol{s}\}\}, \forall i \in \{1, \dots, k\} \tag{18}$$

$$\begin{aligned} s'_i &= (u_i, v_i) \in \text{RegionA} \Rightarrow v_x^{FOV} = 0, v_y^{FOV} = 0 \\ s'_i &= (u_i, v_i) \in \text{RegionB} \Rightarrow v_x^{FOV} = \pm FL_x^{FOV}(u_i, v_i), v_y^{FOV} = \pm FL_y^{FOV}(u_i, v_i) \\ s'_i &= (u_i, v_i) \in \text{RegionC} \Rightarrow v_x^{FOV} = \pm v_{x\_\max}^{FOV}, v_y^{FOV} = \pm v_{y\_\max}^{FOV} \end{aligned} \tag{19}$$

where $\pm$ signs in (19) vary according to s'_i approaching to horizontal and vertical region borders of the image. Here, it should be noted that $v_{x\_\max}^{FOV}$, $v_{y\_\max}^{FOV}$ should be equal to the velocity limits of the manipulator. Again, FL unit type used for FOV keeping is Mamdani, the aggregation type is maximum and the defuzzification type is COA.

After $\boldsymbol{v'}$ is obtained in Fig. 1, joint velocities are obtained using robot Jacobian

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^+ . \boldsymbol{v'} \tag{20}$$

where $\boldsymbol{J}(\boldsymbol{q})^+$ is pseudoinverse of robot manipulator Jacobian. Joint variables $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are obtained with numerical integration and differentiation. By using kinematic model of the robot manipulator, position and orientation of the end effector are obtained. At the end of the proposed system in Fig. 1, end-effector transformation matrix $\boldsymbol{T_r}$ is obtained by robot kinematic model and this matrix is applied to camera model to acquire instant image and feature points.

3. VS evaluation metrics

It is very hard to evaluate a newly proposed VS system. Most studies on VS emphasize only converging error to zero as a success but from a practical point of view, if these systems will be used in real-time applications, some metrics should be defined to quantitatively measure the performance of these systems under certain conditions. For this purpose, only two studies mention remarkable metrics in their own way. Gans et al. chose five different VS methods and they posited a set of performance metrics that measure the performance of these methods for a specific task. Although they aimed standardization, they only defined error and number of iterations (Gans, Hutchinson, & Corke, 2003). Chesi and Hung focused on constrained optimal visual servoing with homography-based path planning and they mentioned remarkable metrics such as spanned image area, trajectory length and curvature using their own terms (Chesi & Hung, 2007). In addition to these metrics, Jabani-Sharifi et al. investigated PBVS and IBVS according to system stability, robustness, sensitivity and dynamic performance in the Cartesian and image spaces but they presented only their observations in their study (Janabi-Sharifi, Deng, & Wilson, 2011)

In this study, VS evaluation metrics are defined analytically as follows to set the pace.

(1) *Convergence time*: As the first and main goal of a VS system, error convergence is essential. For fixed step systems, the time instant of convergence is assumed as

$$t_c = \arg\max_{0 \le t \le \infty} (\|\boldsymbol{e}(t)\| < e_{thr}) \tag{21}$$

where e_{thr} is the threshold for error convergence and it can be defined according to accuracy requirements.

(2) *Camera trajectory length*: It is the length of the camera trajectory, at the same time end-effector trajectory according to (7) in 3D space. In Chesi and Hung (2007), only error and maximum the path is defined in terms of desired rotational and translational feature vector and the camera trajectory length is

defined in terms of desired translational feature vector. Based on this formulation, all path should be planned in this format. Instead of this formulation, curve length formulation in vector calculus can be useful if the trajectory is assumed as a curve (Colley, 2012). The tangent vector of camera trajectory in each step is needed for the camera trajectory length and the coordinates of the trajectory steps are obtained from $\boldsymbol{T_r}$. Let $\dot{\boldsymbol{d}}_{\boldsymbol{r}} \in \mathbb{R}^3$ be the tangent vector of camera trajectory, then the trajectory length l_c is

$$\boldsymbol{T_r} = \begin{bmatrix} \boldsymbol{R_r} & \boldsymbol{d_r} \\ 0 & 1 \end{bmatrix} \tag{22}$$

$$l_c = \int_0^{t_c} \left\| \dot{\boldsymbol{d}}_{\boldsymbol{r}} \right\| .dt \tag{23}$$

(3) *Error costs*: Error is only evaluated as convergence error in Gans et al. (2003). From the point of control theory, the controller can be evaluated according to few performance metrics such as overshoot, settling time in the response or according to entire response in terms of error integrals called as error costs (Aström & Hagglund, 2006). These costs are related to the behavior of error in time and they are significiant performance metrics in realization and parameter tuning. In this study, these cost functions are chosen as error metrics in three forms

$$e_{IAE} = \int_0^{t_c} \|\boldsymbol{e}(t)\| .dt \tag{24}$$

$$e_{ISE} = \int_0^{t_c} \|\boldsymbol{e}(t)\|^2 .dt \tag{25}$$

$$e_{ITAE} = \int_0^{t_c} t.\|\boldsymbol{e}(t)\| .dt \tag{26}$$

where e_{IAE} is an aggregate measure of the error in time, e_{ISE} is a measure which penalizes large errors more than smaller ones in time and e_{ITAE} is a measure which weights errors according to time.

(4) *Curvatures of camera and feature trajectories*: As mentioned in Chesi and Hung (2007) and Tahri and Chaumette (2005), camera trajectory of an IBVS system should be a straight line. Furthermore, as in Chaumette and Hutchinson (2006), feature trajectories in image plane should also be straight for FOV keeping. Considering these recommendations, curvatures of these trajectories are assumed as metrics of straightness. Curvature of the camera trajectory is defined in terms of translational feature vector as camera trajectory in Chesi and Hung (2007). In this study, curvature is defined using tangent vector as in Colley (2012)

$$\boldsymbol{\Psi} = \frac{\dot{\boldsymbol{d}}_{\boldsymbol{r}}}{\left\| \dot{\boldsymbol{d}}_{\boldsymbol{r}} \right\|}$$
$$\kappa = \left\| \frac{d\boldsymbol{\Psi}}{dl_c} \right\| \tag{27}$$

where $\boldsymbol{\Psi}$ is unit tangent vector and κ is the curvature.

4. Simulation results

The simulations are performed to compare the performance of the classical IBVS system and the proposed intelligent IBVS system according to evaluation metrics mentioned in the previous section. These systems are simulated using MATLAB *Simulink, Robotics Toolbox, Machine Vision Toolbox* (Corke, 2011), *Fuzzy Logic Toolbox* and *ELM* codes from Huang et al. (2006). The camera is assumed as attached to the end effector with no transformation as in (7). The manipulator is chosen as a six-DOF Puma560 arm and the parameters of the model of this manipulator can be found in Armstrong, Khatib, and Burdick (1986). The resolution of the camera is 1024×1024, the coordinates of the principal point is (512,512), the focal length is 8 mm and the video rate (and the control loop) of the system is 33 Hz. The feature points in Cartesian coordinates are defined as $\boldsymbol{P}^*$ using 4 fixed collinear points of a square with 0.5 m. side length. The center of these points should collide with the principal point for $\boldsymbol{s}^* \in \mathbb{R}^{2\times 4}$. $\boldsymbol{P}^*$ and $\boldsymbol{s}^*$ in matrix form for these points are given as

$$\boldsymbol{P}^* = \begin{bmatrix} 0.25 & 0.25 & -0.25 & -0.25 \\ -0.25 & 0.25 & 0.25 & -0.25 \\ 2.5 & 2.5 & 2.5 & 2.5 \end{bmatrix}$$
$$\boldsymbol{s}^* = \begin{bmatrix} 612 & 412 & 412 & 612 \\ 412 & 412 & 612 & 612 \end{bmatrix} \tag{28}$$

The estimated depth value for $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$ in (9) is assumed as 2 m. Again, it should be noted that this estimation may affect the performance of the system as mentioned in Chaumette and Hutchinson (2006).

In the following subsections, three different cases are assumed to make a better comparison of classical IBVS and the proposed intelligent IBVS system. As the first case, the results of the proposed intelligent IBVS system and classical IBVS system for the VS task defined as reaching $\boldsymbol{s}^*$ are given and compared. In the second case, image FOV regions A, B, C are defined, $\boldsymbol{s}^*$ is assumed as located near the edge of Region A and FOV keeping ability of the proposed system is shown. In the last case, feature noise and camera calibration errors are added to the first case to show the robustness of the proposed IBVS system against practical VS disturbances.

For the first and third cases, initial joint angle vector of the robot manipulator which ensures that all features are in sight is given as

$$\boldsymbol{q}_0 = \begin{bmatrix} 0 & \pi/4 & \pi & \pi/10 & \pi/4 & -\pi/4 \end{bmatrix} \text{rad.} \tag{29}$$

4.1. Case 1: classical IBVS system vs. proposed intelligent IBVS system

In the first case, a classical IBVS system and the proposed intelligent IBVS system are tested to compare the performances of each system according to feature and end effector trajectories, velocities of the end effector and errors. Classical IBVS assumes that $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$ is obtained with high accuracy and gain λ is fixed through the VS task. $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}$ is obtained using 2 m. estimated depth and λ is 0.5 for classical IBVS. On the other hand, the proposed intelligent IBVS system uses 6 ELMs with single output for each element of $-\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}.\boldsymbol{e}$. The type of activation functions for these ELMs are sigmoid and 20 hidden neurons are used for each ELM. ELMs are trained using the training data set of input-output signals shown in Fig. 3. The training data set is obtained from classical IBVS system with 17 different initial camera-end effector pose, 2 m. depth estimation and fixed $\lambda = 0.5$ gain. 85% data is used for training and 15% for testing. 17 different initial camera-end effector poses are chosen in order to start the feature points in all quadrants of the image plane according to the image center and sufficient number of training pairs are obtained for adequate approximation. As an example of training measures of ELMs, the training time of the ELM approximating the first output $-(\hat{\boldsymbol{L}}_{\boldsymbol{s}}^{+}.\boldsymbol{e})_{v_x}$ is 0.0312 s. and RMSE is 0.0052. As shown in Table 1, the other ELMs for other outputs show similar training results . The training time is also an indicator of convenience of ELM for online learning when the robot starts with a different initial end effector pose. It should be noted that this capability is not mentioned in the other studies. RMSE rate is also good enough for interaction matrix approximation.

The controller of the proposed IBVS system in (14) uses FL for defining λ_a. All inputs and output are limited to obtain a limited λ_a surface. The membership functions of this FL unit are gaussian

Table 1
Training performances of ELMs.

ELM index (output)	Convergence time (s.)	RMSE
ELM 1 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{v_x})$	0.0312	0.0052
ELM 2 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{v_y})$	0.0311	0.0048
ELM 3 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{v_z})$	0.0328	0.0065
ELM 4 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{w_x})$	0.0341	0.0054
ELM 5 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{w_y})$	0.0319	0.0049
ELM 6 $(-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{w_z})$	0.033	0.0057

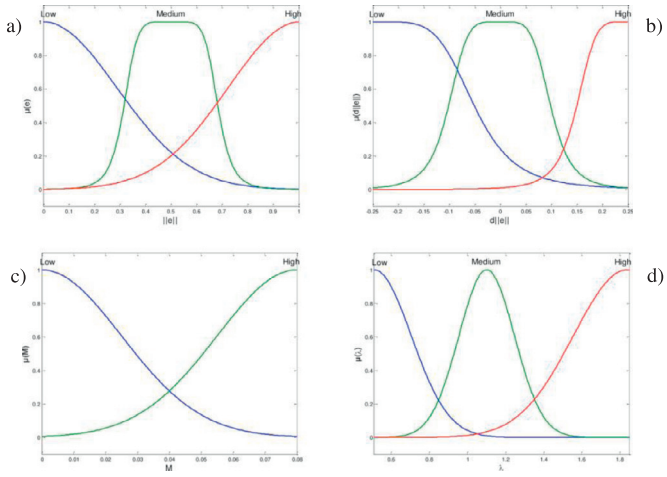

Fig. 5. Membership functions of FL for adaptive gain. (a) $\|\boldsymbol{e}\|$, (b) $d\|\boldsymbol{e}\|/dt$, (c) M, (d) λ_a.

Table 2
FL rulebase for adaptive gain.

$\|\boldsymbol{e}\|$	$d\|\boldsymbol{e}\|/dt$	M	λ_a
L	L	L	H
L	L	H	H
L	M	L	H
L	M	H	H
L	H	L	M
L	H	H	H
M	L	L	M
M	L	H	H
M	M	L	M
M	M	H	M
M	H	L	L
M	H	H	M
H	L	L	M
H	L	H	M
H	M	L	M
H	M	H	M
H	H	L	L
H	H	H	L

L: Low M: Medium H: High

and generalized bell functions as shown in Fig. 5. $\|\boldsymbol{e}\|$ and $d\|\boldsymbol{e}\|/dt$ inputs of FL are normalized according to maximum and minimum values and three membership functions are defined for these inputs, Low-Medium-High. M input is used directly and two membership functions are defined for this input, Low-High. Three membership functions are also defined for λ_a output, Low-Medium-High. The rulebase is given in Table 2 and it should be noted that the rulebase is adjusted according to experiences gained from trials. All the rules use AND operator for rule definitions in Table 2.

As mentioned in (14), γ defines time constant of the transient respone and it is chosen as 2 to obtain a faster response within velocity limits.

The results of each system for Case 1 are shown in Fig. 6. The feature trajectories are shown for both IBVS systems with red circles as starting feature points and blue circles as finishing feature points in Fig. 6(a-b). The same circles are also used for end effector trajectories in Fig. 6(c-d). The feature trajectories of classical IBVS are linear as it should be (Chaumette & Hutchinson, 2006). It is a consequence of a good $\hat{\boldsymbol{L}}_s^+$ estimation but it causes high initial velocities as mentioned in Chaumette and Hutchinson (2006). It is shown in Fig. 6(e).

The feature trajectories of the proposed IBVS system is slightly curvilinear as shown in Fig. 6(b) as a consequance of interaction matrix approximation. The initial velocities for the proposed system are 0 as shown in Fig. 6(f). This outcome is a big positive in practical applications of VS.

The end effector trajectories are smooth for both systems as shown in Fig. 6(c-d) but l_c of each system are not the same. It is 0.6782 m. for classical IBVS system and 0.6193 m. for the proposed IBVS system. κ of the classical IBVS is 0.0073 and it is 0.0153 for the proposed IBVS system. These results show that the proposed IBVS system reaches the desired feature point in a shorter but a rough path. t_c results also promote the proposed IBVS system. The scores of t_c for the classical system and the proposed system are 10.65 s. and 8.79 s., respectively. The proposed system converges more rapidly and it demonstrates another big positive against classical IBVS for real-time applications.

The errors are decreasing with exponential functions of time for both systems as shown in Fig. 6(g-h) but the error costs are different. e_{thr} in (21) is assumed as 3 pixels for Case 1. e_{IAE}, e_{ISE} and e_{ITAE} of the classical system and the proposed system are $3.91e^{+04}$, $1.19e^{+07}$, $7.64e^{+04}$ and $3.12e^{+04}$, $1.1e^{+07}$, $4.03e^{+04}$, respectively. Firstly, lower e_{IAE} means that the proposed system attends to limit error oscilations. Lower e_{ISE} proves that the proposed system tries to eliminate large errors more rapidly and supports convergence speed. Lastly, lower e_{ITAE} demonstrates that the proposed system endevaours error oscilations in all time period of convergence. Again, these results also promote the proposed IBVS system.

The outputs of the intelligent units of the proposed system should be examined to show the performance of the system. The training times in Table 1 promote the proposed system and as an illustration, the approximation of the ELM for $-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{v_x}$ is shown in Fig. 7(a). In Fig. 7(a), it is clear that the first ELM achieved $-(\hat{\boldsymbol{L}}_s^+.\boldsymbol{e})_{v_x}$ approximation successfully with small error. Please note that the errors in approximations are tolerated by the proposed system but it caused curvilinear feature trajectories in Fig. 6(b). As an impact of this study, it should be noted that interaction matrix approximation with training data of different initial robot poses and defined feature targets with obtained velocity signals are given in details in an interaction matrix approximation approach for the first time in the literature.

FL output for adaptive gain λ_a is shown in Fig. 7(b). λ_a in Fig. 7(b) varies in time according to input membership functions in Fig. 5(a-c), output membership functions in Fig. 5(d) and rules in Table 1. While λ_a is fixed 0.5 in classical IBVS, it starts from 1.1 for the proposed system and it is fixed to 1.6 after convergence. For the classical IBVS system, the limits of the end effector velocities are assigned as ± 0.5 (m/s, rad/s) according to the peak value of v_Z in Fig. 6(e) as the largest velocity and the velocities of the proposed system stay within the limits as shown in Fig. 6(f). This is also a consequance of initial velocity continuity. The convergence time and the velocity signals of the proposed system prove fast convergence capability of the proposed system within velocity limits.

In Fig. 8, manipulability of each system are shown. It can be seen that the proposed system increases manipulability of the IBVS system. This proves that the proposed system draws away the manipulator farther from singular poses and thus the system is capable of performing VS tasks with reduced manipulator velocity limits.

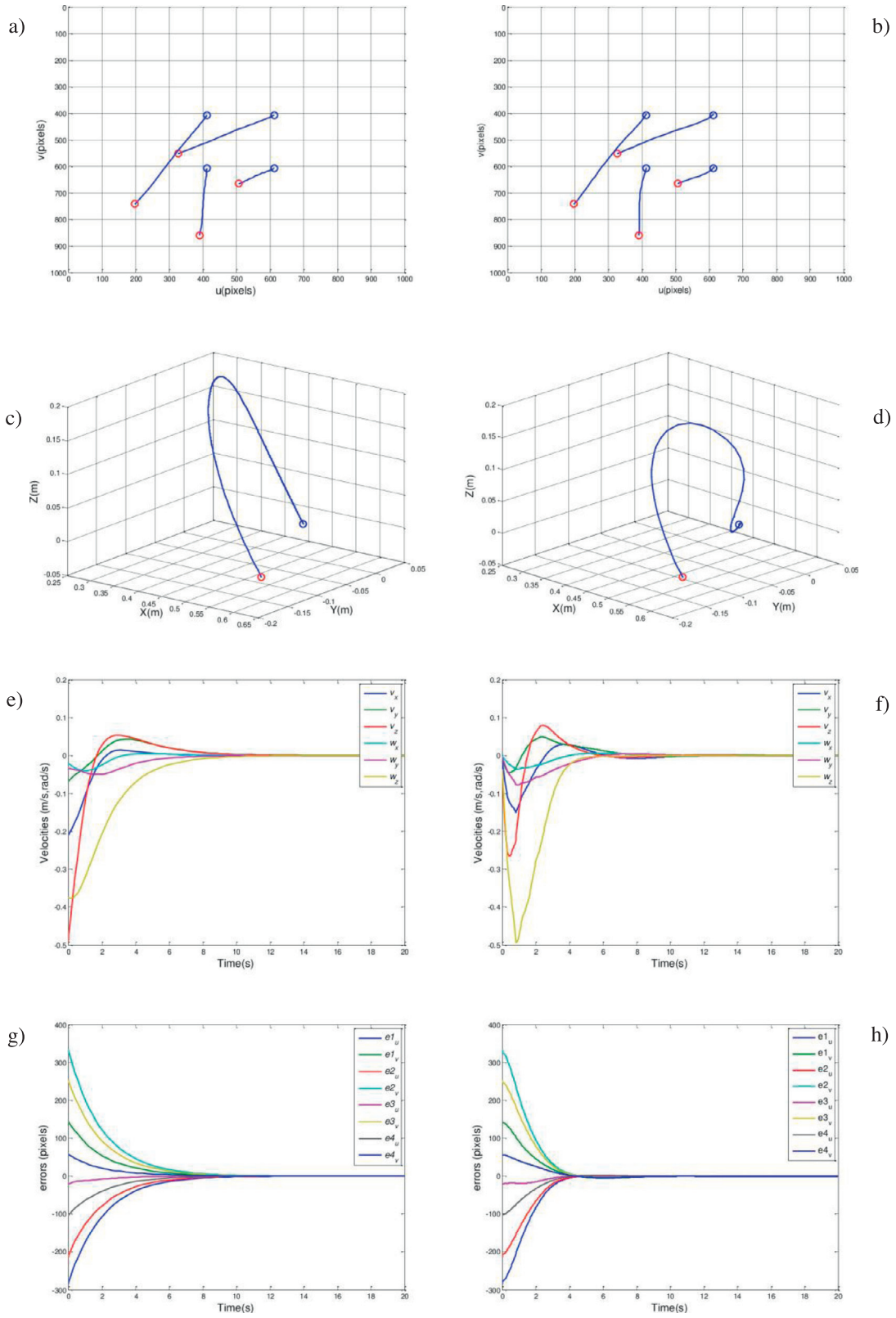

Fig. 6. Result for Case 1 as 1st column for classical IBVS system and 2nd column for the proposed IBVS system. (a-b) Feature trajectories, (c-d) end effector trajectories, (e-f) velocities of the end effector, (g-h) the feature errors.

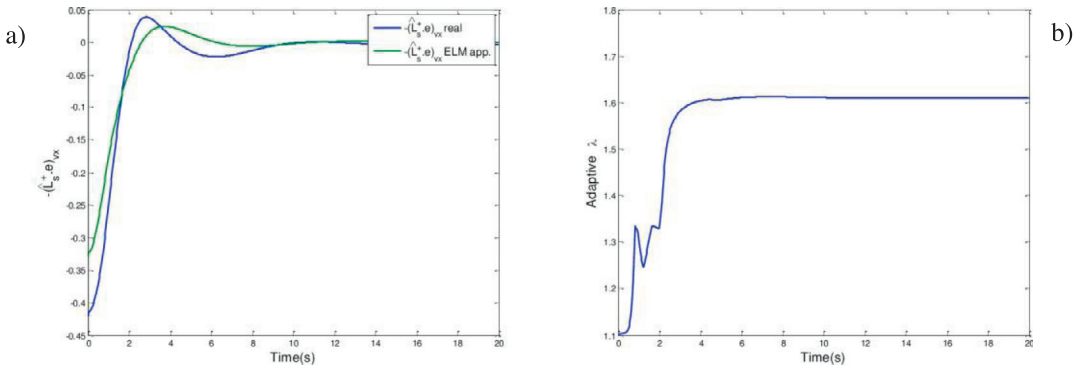

Fig. 7. Intelligent units' outputs of the proposed system for Case 1.

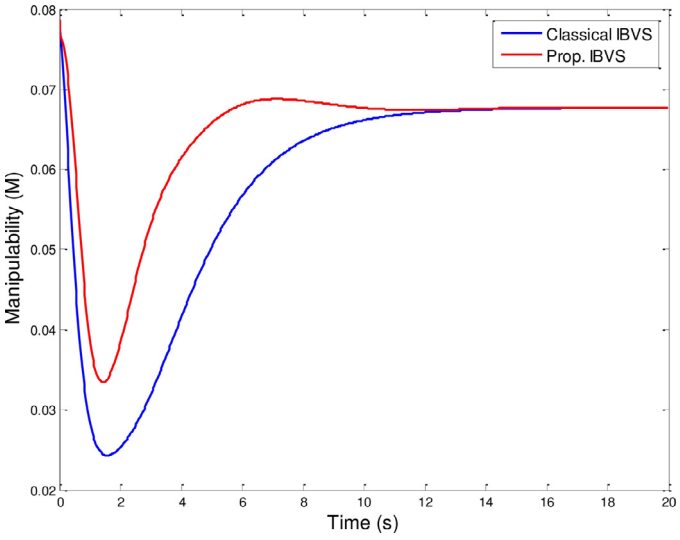

Fig. 8. Manipulability of the classical and the proposed system.

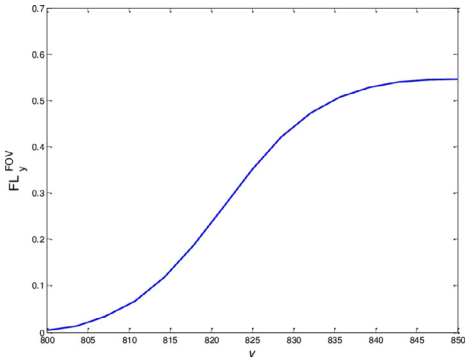

Fig. 9. The surface for $\pm FL_y^{FOV}(u_i, v_i)$.

4.2. Case 2: FOV keeping

While performing a VS task, underestimations or overestimations of depth or calibration errors may cause the features to leave FOV. To show FOV keeping ability of the proposed system, the same 4 fixed collinear points $\boldsymbol{P}^*$ with another $\boldsymbol{s}^*_{FOV}$ and initial joint angle vector $\boldsymbol{q}_0^{FOV}$ are considered as Case 2

$$\boldsymbol{s}^*_{FOV} = \begin{bmatrix} 177 & 33 & 240 & 366 \\ 460 & 653 & 771 & 580 \end{bmatrix}$$
$$\boldsymbol{q}_0^{FOV} = \begin{bmatrix} \pi/10 & 3\pi/10 & 9\pi/10 & 3\pi/25 & 3\pi/10 & -5.7\pi/10 \end{bmatrix} \text{rad.} \quad (30)$$

The region borders in Fig. 4 are assigned with 50 pixel thickness for Region B and 174 pixel thickness for Region C as in Fig. 10(a-b). $\pm v^{FOV}_{x\_\max}$ and $\pm v^{FOV}_{y\_\max}$ in (19) are ± 0.55 m/s. FL units of $\pm FL_x^{FOV}(u_i, v_i)$ and $\pm FL_y^{FOV}(u_i, v_i)$ assume (u_i), (v_i) as inputs in (19), respectively. These FL units are formed using same two gaussian membership functions.

The rulebases are based on 4 rules. Instead of these rulebases, the surface for $\pm FL_y^{FOV}(u_i, v_i)$ is shown in Fig. 9 as an illustration. It is clear that both FL units provide same surfaces and these surfaces provide smooth transition from restricted regions to safe FOV region.

In Fig. 10(a-b), feature trajectories both without and with FOV keeping are shown. The features cross Region B and C without FOV keeping as shown in Fig. 10(a). The counter velocity in $+y$ direction provides that the features are directed through Region A as they are aimed to cross Region B in Fig. 10(b). A feature attempts to enter Region B at 2.52 s. and the velocity v'_y changes its profile after this time instant as shown in Fig. 10(d). It is also obvious that this negative velocity affects other velocities in Fig. 10(d). Similar changes in the error profiles can be seen in Fig. 10(e). It should be noted that the proposed system provides FOV keeping with smooth transition between image regions, ensures continous velocity signals without any sudden changes. Furthermore, it is easy to implement negative velocity and this doesn't move the features away from the target. Lastly, FOV keeping is ensured using an intelligent unit which provides rapid adaptation for all cases with user experience. This is also a contribution of the study to VS approaches.

4.3. Case 3: robustness against disturbances

All real-time VS systems may encounter practical disturbances like bad camera calibration and feature noise as mentioned in Lizarralde et al. (2013), Tahri and Chaumette (2005), Zhong et al. (2015). Bad camera calibration affects not only the calibrated image taken from the camera but also the interaction matrix in (5). Furthermore, noise in the features may cause oscillations when the feature errors are small as the system is close to convergence. Hence, it is quite important to show the robustness of a newly proposed IBVS system against these disturbances.

As the last case, disturbances are added to camera intrinsic parameters (15% on the focal length and 20 pixels on the coordinates of the principal point) and all feature points are disturbed by uniformly distributed random noise with a standart deviation of 1 pixels (min = −1, max = 1). $\boldsymbol{s}^*$ and $\boldsymbol{q}_0$ are chosen as in Case 1. The results for Case 3 are shown in Fig. 11.

Firstly, the results of classical IBVS is not shown as it diverges to infinity. The feature trajectories and errors with bad calibrated

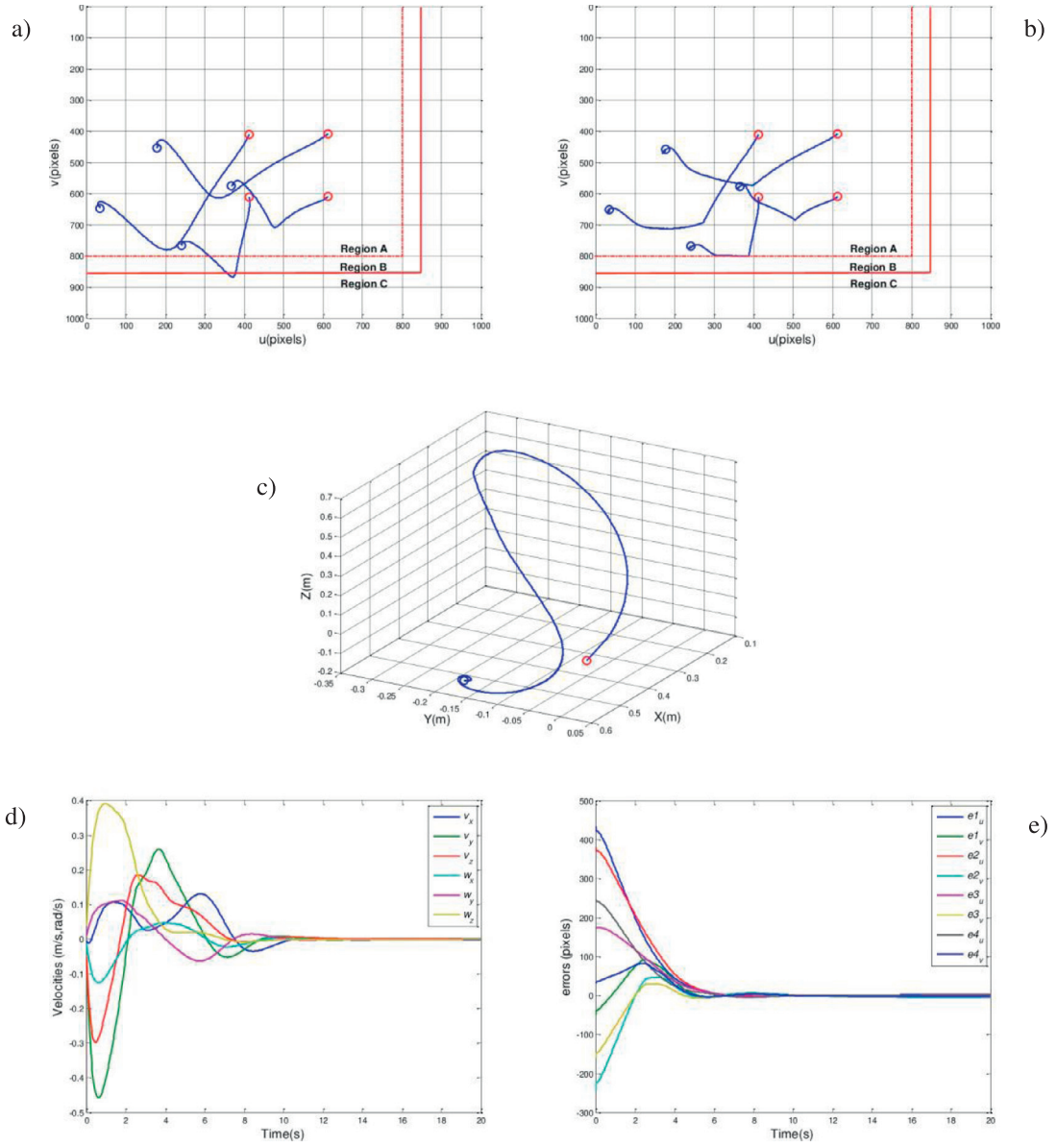

Fig. 10. Result for Case 2 for the proposed IBVS system. (a) Feature trajectories without FOV keeping, (b) feature trajectories with FOV keeping, (c) end effector trajectory, (d) velocities of the end effector, (e) the feature errors.

camera and noise in Fig. 11(a)–(d) shows that the proposed IBVS system converges $\boldsymbol{s}^*$ and shows robustness against these practical disturbances. The feature trajectories in Fig. 11(a) with red circles as starting feature points and blue circles as finishing feature points are curvilinear as a result of interaction matrix appromation by ELMs. ELM approximators show their approximation abilities, achieve convergence and robustness. Noise cause oscilations at the end of the trajectories after convergence in Fig. 11(a-b) but the proposed system copes with this effect and doesn't diverge.

A comparison of VS metric results of Case 1 and Case 3 for the proposed system is not reliable because of bad camera calibration and feature noise disturbances added to the system. e_{thr} in (21) is changed to 5 pixels as each feature contains uniformly distributed random noise. As shown in Fig. 11(b), it is clear that the end effector follows a different trajectory from Fig. 6(d) for Case 3. l_c is 0.4059 m. and κ is 0.0059. The convergence time for Case 3 is 9.57 s. which takes longer than Case 1. This is a consequence of noise added to the features.

Again, it is clear that errors in Figs. 6(h) and 11(d) show different characteristics. e_{IAE}, e_{ISE} and e_{ITAE} of the proposed system for Case 3 are 3.5182e + 04, 1.0817e + 07 and 5.7424e + 04, respectively. These results are close to the results in Case 1 and these promote that the proposed system retains its limiting abilities against error oscilations in all time period of convergence under disturbances.

The proposed IBVS system copes with noise easily however, the noise causes sudden changes in$\|\boldsymbol{e}\|$, $d\|\boldsymbol{e}\|/dt$ and this causes gain oscillations as shown in Fig. 12. As a consequence, the velocities of the end effector show oscillations and characteristics of noise. From a practical point of view, this may cause oscillations in end effector trajectory as shown in Fig. 11(b) but it should be noted that limited bandwidth of manipulators will filter these oscillations in realization. The proposed system agrees with the velocity limit rule defined in Case 1 as shown in Fig. 11(c). The trajectory in Fig. 11(b) changes manipulability characteristic of the system and the manipulability for Case 3 is shown in Fig. 13. $\boldsymbol{T}_r^*$ is same both cases and M converges to 0.0676 as in Case 1. These results also

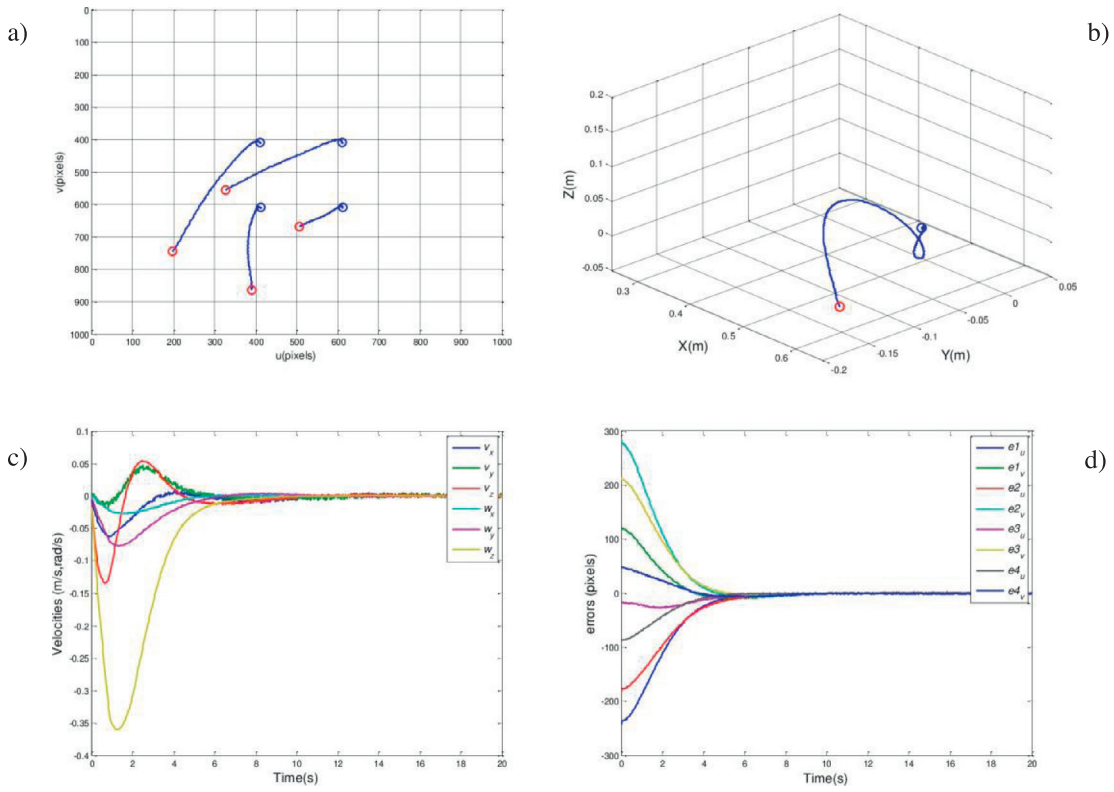

Fig. 11. Result for Case 3 for the proposed IBVS system. (a) Feature trajectories, (b) end effector trajectory, (c) velocities of the end effector, (d) the feature errors.

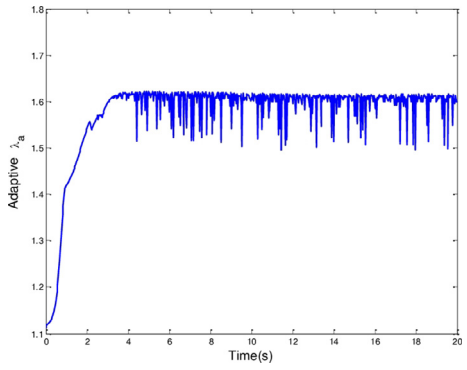

Fig. 12. The FL output as adaptive gain λ_a for Case 3.

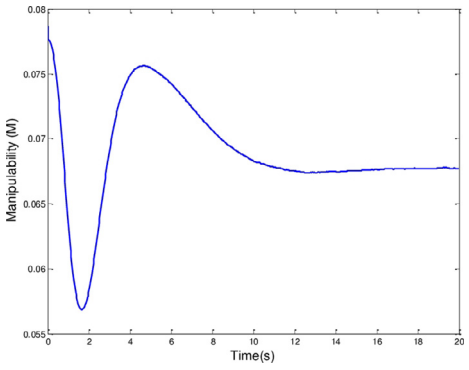

Fig. 13. Manipulability for Case 3.

prove the robustness of the proposed system under practical disturbances.

5. Conclusion

The studies on problems of visual servoing as bad camera calibration and noise, gain tuning and FOV keeping usually focus on each problem one by one but a VS system should cope with all these problems while achieving better performance in terms of performance metrics. In this paper, an intelligent IBVS system solving these common problems in a single system is proposed. Additionally, the study is focused on IBVS systems with eye-in-hand configuration that is the most popular VS system.

As an SLFN unit without hidden layer tuning, ELM is deployed in the proposed system to avoid singularities of the pseudoinverse of the interaction matrix and to obtain robustness against practical disturbances like bad camera calibration and image errors. The other approaches in the literature which focus on interaction matrix approximation using intelligent methods are very constrainted because of very limited learning capabilities, undefined feature targets, neglected velocity signals and velocity signals with huge chattering. The simulations show that the proposed system is independent of all of these contraints.

A differential equation used as velocity control law with initial velocity continuity in the literature is modified by FL tuned gain using error, error derivative and manipulability as inputs. This tuned differential equation ensures initial velocity continuity, smooth adaptive gain and better converge performance within velocity limits with increased manipulability providing joint singularity avoidance. These capabilities are also the superiorities of the proposed system against other approaches while designing a VS kinematic controller.

An FL based FOV keeping unit is added the velocity control law. Image regions for precautions of feature loss are defined and this unit guided the features in restricted regions through the safe region by defining linear velocities in the counter direction

of approached image borders. An intelligent unit is used for FOV keeping for the first time and it provides smooth velocity signals and transitions between borders. User experience is involved in FL which will provide rapid adaptation for all cases encountered. Furthermore, negative velocity realization is easy to implement for practical applications and it doesn't cause divergence from the target features.

Besides solving the problems of VS, the gap in the field of performance metrics for evaluating new VS approaches is filled with this study by defining old and new metrics analytically in terms of VS. Besides these metrics, it should noted that the type of the VS application directly affects the evaluation of a VS system in practical applications. While a pick and place robot application may prefer lower convergence time for increasing the number of finished tasks, an assembly robot application may prefer less errors. Therefore, these metrics should be evaluated wisely according to the type of the application. Simulation results in this study promote the proposed system for each metric and the achievements of the proposed IBVS system are demonstrated in terms of VS metrics. Furthermore, robustness of the proposed system under practical disturbances are also proved by simulation results. This proves that intelligent methods in VS are capable of dealing with disturbances and the proposed system can be applicable in all kinds of robotic applications.

While analytical methods are still dominant in the field of VS, an intelligent IBVS system is proposed in this study. The author believes that the proposed system is a good demonstration of intelligent methods deployed in VS and the study will motivate expert systems professionals for proposing new approaches in VS using new intelligent methods.

In this study, only simulation results are presented but the following study will focus on realization, firstly. Furthermore, the proposed system is just applicable to point features. The industrial applications of VS tend to use shape features so other feature types such as image moments are becoming more popular and the system will be modified according to these features as a future work. Furthermore, the proposed system is applicable to only manipulator systems. The underactuated robotic systems like mobile robots or UAVs need more attention while performing VS tasks. The future studies will focus on the adaptation of the proposed system to these robotic systems.

References

Armstrong, B., Khatib, O., & Burdick, J. (1986). The explicit dynamic model and inertial parameters of the PUMA 560 arm. In *Proceedings. 1986 IEEE international conference on robotics and automation: Vol. 3* (pp. 510–518). http://doi.org/10.1109/ROBOT.1986.1087644.

Aström, K. J., & Hagglund, T. (2006). *Advanced PID control*. ISA Publishing.

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In *Lecture notes in control and information sciences* (pp. 66–78). Springer-Verlag.

Chaumette, F., & Hutchinson, S. (2006). Visual servo control. I. Basic approaches. *IEEE Robotics and Automation Magazine, 13*(4), 82–90. http://doi.org/10.1109/MRA.2006.250573.

Chaumette, F., & Hutchinson, S. (2008). Visual servoing and visual tracking. *Handbook of Robotics http://doi.org/10.1007/978-3-540-30301-5_25*.

Chesi, G., Hashimoto, K., Prattichizzo, D., & Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: A switching approach. *IEEE Transactions on Robotics, 20*(5), 908–913. http://doi.org/10.1109/TRO.2004.829456.

Chesi, G., & Hung, Y. S. (2007). Global path-planning for constrained and optimal visual servoing. *IEEE Transactions on Robotics, 23*(5), 1050–1060.

Collewet, C., Marchand, E., & Chaumette, F. (2008). Visual servoing set free from image processing. In *Proceedings - IEEE international conference on robotics and automation* (pp. 81–86). http://doi.org/10.1109/ROBOT.2008.4543190.

Colley, S. J. (2012). *Vector calculus*. Pearson.

Corke, P. I. (2011). *Robotics, vision & control: Fundamental algorithms in matlab*. Springer.

Corke, P. I., & Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation, 17*(4), 507–515. http://doi.org/10.1109/70.954764.

Gans, N., Hutchinson, S., & Corke, P. I. (2003). Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *International Journal of Robotics Research, 22*(10-11), 955–981. http://doi.org/10.1177/027836490302210011.

Gans, N. R., Hu, G., & Nagarajan, K. (2011). Keeping multiple moving targets in the field of view of a mobile camera. *IEEE Transactions on Robotics, 27*(4), 822–828.

Gonçalves, P. J., Mendonça, L. F., Sousa, J. M. C., & Pinto, J. R. C. (2008). Uncalibrated eye-to-hand visual servoing using inverse fuzzy models. *IEEE Transactions on Fuzzy Systems, 16*(2), 341–353.

Huang, G. (2015). What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation, 7*, 1–13. (April) http://doi.org/10.1007/s12559-015-9333-0.

Huang, G., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 42*(2), 513–529.

Huang, G., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine : Theory and applications. *Neurocomputing, 70*, 489–501. http://doi.org/10.1016/j.neucom.2005.12.126.

Janabi-Sharifi, F., Deng, L., & Wilson, W. J. (2011). Comparison of basic visual servoing methods. *IEEE/ASME Transactions on Mechatronics, 16*(5), 967–983. http://doi.org/10.1109/TMECH.2010.2063710.

Jang, S. R., & Sun, C. T. (1997). *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. Prentice-Hall.

Kallem, V., Swensen, J. P., Hager, G. D., & Cowan, N. J. (2007). Kernel-based visual servoing. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1975–1980).

Kermorgant, O., & Chaumette, F. (2014). Dealing with constraints in sensor-based robot control. *IEEE Transactions on Robotics, 30*(1), 244–257.

Kosmopoulos, D. I. (2011). Robust Jacobian matrix estimation for image-based visual servoing. *Robotics and Computer-Integrated Manufacturing, 27*(1), 82–87. http://doi.org/10.1016/j.rcim.2010.06.013.

Kumar, P. P., & Behera, L. (2010). Visual servoing of redundant manipulator with Jacobian matrix estimation using self-organizing map. *Robotics and Autonomous Systems, 58*(8), 978–990. http://doi.org/10.1016/j.robot.2010.04.001.

Lizarralde, F., Leite, A. C., Hsu, L., & Costa, R. R. (2013). Adaptive visual servoing scheme free of image velocity measurement for uncertain robot manipulators. *Automatica, 49*(5), 1304–1309. http://doi.org/10.1016/j.automatica.2013.01.047.

Malis, E., Chaumette, F., & Boudet, S. (1999). 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation, 15*(2), 238–250. http://doi.org/10.1109/70.760345.

Mansard, N., & Chaumette, F. (2007). Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics, 23*(1), 60–72. http://doi.org/10.1109/TRO.2006.889487.

Mezouar, Y., & Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation, 18*(4), 534–549.

Miljković, Z., Mitić, M., Lazarević, M., & Babić, B. (2013). Neural network reinforcement learning for visual control of robot manipulators. *Expert Systems with Applications, 40*, 1721–1736. http://doi.org/10.1016/j.eswa.2012.09.010.

Mohebbi, A., Keshmiri, M., & Xie, W. F. (2016). A comparative study on eye-in-hand image-based visual servoing : Stereo vs . Mono. *Journal of Integrated Design and Process Science, 19*(3), 25–54.

Nasisi, O., & Carelli, R. (2003). Adaptive servo visual robot control. *Robotics and Autonomous Systems, 43*, 51–78. http://doi.org/10.1016/S0921-8890(02)00370-6.

Schramm, F., & Morel, G. (2006). Ensuring visibility in calibration-free path planning for image-based visual servoing. *Robotics, IEEE Transactions on, 22*(4), 848–854. http://doi.org/10.1109/TRO.2006.878955.

Sebastian, J. M., Pari, L., Angel, L., & Traslosheros, A. (2009). Uncalibrated visual servoing using the fundamental matrix. *Robotics and Autonomous Systems, 57*(1), 1–10. http://doi.org/10.1016/j.robot.2008.04.002.

Tahri, O., & Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics, 21*(6), 1116–1127. http://doi.org/10.1109/TRO.2005.853500.

Wang, H., Jiang, M., Chen, W., & Liu, Y. (2012). Visual servoing of robots with uncalibrated robot and camera parameters. *Mechatronics, 22*(6), 661–668. http://doi.org/10.1016/j.mechatronics.2011.05.007.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *The International Journal of Robotics Research, 4*(2), 3–9.

Zhong, X., Zhong, X., & Peng, X. (2013). Robust Kalman filtering cooperated Elman neural network Learning for vision-sensing-based robotic manipulation with global stability. *Sensors, 13*, 13464–13486. http://doi.org/10.3390/s131013464.

Zhong, X., Zhong, X., & Peng, X. (2015). Robots visual servo control with features constraint employing Kalman-neural-network filtering scheme. *Neurocomputing, 151*, 268–277. http://doi.org/10.1016/j.neucom.2014.09.043.