Eng Int Syst (2023) 6: 501–511

Engineering
Intelligent Systems

Visual Servo Control of Robot Arm Based on Image Features

Zhigang Ma*

In recent years, with the increasing amount of research being done in the machine vision field and other related areas, the visual servo control of the robotic arm based on image information not only strengthens the diversity of information obtained by this device, but also expands its space, cognition and adaptability. It also improves the robotic arm's ability to make precise identifications and execute fine operations. Processing image information and making good motion decisions are two of the skills that robotic arms need to master, as well as robotic arm vision. The image-based visual servo system does not require camera calibration, and can complete the motion control of the robotic arm through the image information in the camera plane. The control structure is relatively simple, so the current visual servo system has become a research focus. The online identification of the image Jacobian matrix is also studied in order to obtain more accurate Jacobian matrix values in each iteration process of the image-based visual servo control system. In this paper, two different control algorithms are proposed for visual servoing. These two algorithms are based on the Kalman filter and neural network respectively, which can omit the image depth information required in the calculation of a visual servo system and reduce the extra overhead. In response to this, this paper proposes a hybrid kernel online sequence extreme learning machine (MIXEDKOSELM) based on hybrid kernel and online sequence learning. The correction of the error in the Kalman filter algorithm greatly improves the performance of the image-based visual servo (IBVS) control system. This KF-MIXEDKOSELM-IBVS-based visual servo control method does not need camera parameters in the servo process, and is more robust against disturbance errors and noise statistical errors. The training results and test results of the MIXEDKOSELM algorithm on the abalone dataset were analyzed. It was concluded that the training results of two rbfs had MAE of 0.600 and TIMR of 11.12, and the test results had MAE of 0.579 and TIMR of 0.467. By comparing with other algorithms, it can be shown that the two rbf structures of the MIXEDKOSELM algorithm had obvious advantages in the testing process compared with other algorithms.

Keywords: vision servo; robotic arm; servo system; Kalman filter

1. INTRODUCTION

Since the birth of robots at the beginning of the last century, robots and robotics have been developing at a very fast pace. Today's robots have appeared in many aspects of people's daily lives; from aerospace equipment to household appliances, intelligent robots can be seen everywhere. Fields such as aerospace, deep-sea exploration, and field work have become a reality with the advent of robots. People no longer have to risk their lives to do certain jobs since many tasks can now be performed by robots. Robots stand guard, robots can sell tickets, etc. and have greatly liberated people from doing boring and repetitive work, allowing employees to do more meaningful tasks. Home sweepers and smart housekeepers are indispensable for office workers in today's cities. Some service robots can take good care of elderly or disabled people, in some cases eliminating potential human-safety hazards, and so on. Countless robots have appeared in many aspects of our lives. It is the urgent demand for robots that has prompted a large number of scholars to undertake robotics research [1].

Humans dream of creating robots with human-like intelligence to replace human labor. A robot is a human-like machine. Humans obtain much of their information through the eyes, and a small part is obtained by touch, hearing, taste, etc. The image Jacobian matrix can be changed to achieve the mapping relationship between robot joint angles and image feature errors. The calculation of the image Jacobian matrix is very important, and this calculation must be completed in a short time, so the representation of the image Jacobian matrix should be as simple as possible. At the same time, the calculation of the Jacobian matrix is also closely related

*Address for correspondence: Ma Zhigang, College of Artificial Intelligence, Shenzhen Polytechnic, Shenzhen 518055, Guangdong, China, Email: mzg@szpt.edu.cn

to the depth of the feature points of an image, and it is quite difficult to obtain the depth information. Therefore, using the Jacobian matrix online recognition method based on Kalman filter, this paper applies the neural-network-related algorithm to the problem of identifying the Jacobian matrix of the image, which can significantly improve the robustness of IBVS. At the same time, because the neural network used is an improved algorithm of the extreme learning machine (ELM), the real-time performance of the system can basically meet the requirements. The manipulator is controlled in real time by using the image Jacobian matrix model recognized online in order to track a target [2].

Machine vision is becoming extremely important for robots, and research in this area is increasing. For instance, Hsieh [3] proposed a stereo vision robotic arm assistance system in which the robotic arm can perform fifth-degree capture in a single instance. The proposed stereo vision-based robotic arm system enables users to manipulate objects according to the robot's ability to target objects using computer vision. Stereo vision systems compute parameters by paying attention to the true position of the instance in the coordinate system. Xie [4] developed an acupuncture-assisted robotic arm that can perform acupoint measurement and acupoint click massage. The abdominal acupuncture auxiliary robotic arm uses a single-chip microcomputer control board to control the three-axis large stepping motor arm. The reference point judgment is realized based on machine vision, and infrared auxiliary markers are used. Zhong [5] proposed a practical visual servo control using a spherical projection model. The aerial manipulator is a UAV equipped with a robotic arm, which greatly increases the degree of freedom and operational flexibility of the end effector. In order to further improve the grasping performance, a task-priority control scheme was adopted, with one main task and several sub-tasks: the control of the position and orientation of the gripper, the vertical alignment of the center of gravity, and the avoidance of joint constraints. DiPierro [6] built a single-page web application that could connect to the camera and process the video stream. Images can also be extracted and motion detected. It then uses the detected motion to control a 3D robotic arm rendered with webGL on the same page. Yang [7] reported the development of an intelligent shared control system for robotic manipulators commanded by the user's mind. From the analysis of the invoked EEG signals, a brain-computer interface was developed to infer the exact object required by the user. These results are then transferred to a shared control system which enables precise object manipulation through visual servoing.

Determining the position and size of objects by various means is important to ensuring the accurate grasping and operation of the robotic arm. Hence, Qu [8] developed a vision-based dual-arm circular motion method that addressed the problem of uncertain position when grasping objects and the phenomenon of dual-arm joint angle drift. A novel cascaded control structure was proposed that associates an adaptive neural network with kinematic redundancy optimization. Zapotezny-Anderson [9] introduced a deep learning approach whereby Deep-3DMTS can guide the end-effector of a robotic arm to improve the field of view of occluded fruits (bell peppers) when performance is comparable to that of a standard 3DMTS baseline. The final position of the end effector was within 11.4 mm of the baseline; compared to the baseline of 16.8 (average), the fruit size in the image was increased by a factor of 17.8. Qiu [10] proposed a visual servo tracking strategy for wheeled mobile robots, which can simultaneously identify unknown feature depth information during the visual servo process. By using the reference image, the desired image and the current image, the systematic error was constructed from the measurable signal obtained by decomposing the Euclidean homography. Cheng [11] was the first to implement a visual servo control on a magnetically-anchored endoscope. Designed for single-port thoracoscopic surgery, the new magnetic drive endoscope is compact and has a working space close to the chest wall for space-constrained procedures. The visual servo control allows magnetic endoscopes to automatically track surgical instruments. Tsai [12] presented the first derivation, implementation and experimental verification of visual servoing based on light field images. The derived light field image Jacobian matrix is based on a compact light field feature representation that is close to the form directly measured by light field cameras. It is necessary not only to achieve optimality in the image plane, but also to achieve optimal trajectories in space.

The innovation of this paper is that in EEMD-RVFL-IBVS, the inverse of the initialized Jacobian matrix and the combination of EEMD-RVFL are used to replace the inverse image Jacobian matrix. This method considers not only the coordinates of the feature points in the camera plane, but also the error of the image feature points. In fact, the dimension of the input vector is expanded so that the input vector has more information. The advantage of this is that it can better fit the image Jacobian matrix.

2. ROBOTIC ARM VISUAL SERVO

2.1 Composition and Classification of Visual Servo

In the related research field of visual servo manipulator control, there are many ways to classify it, but the standards are not the same. According to the current widely-recognized classification, the categories are: classification according to the installation position and number of cameras; classification according to the type of visual feedback information; and classification according to whether the internal and external parameters of the camera have been determined [13].

In the visual servo system, the camera acts as the eyes of the robot. It transmits the observed data to the computer by observing the outside, and then generates useful information by means of computer processing. In cases where visual information needs to be very precise, CCD industrial cameras are generally used as visual sensors. Under normal circumstances, the scene's information generally includes the position, attitude of the target or the point, line and area features of the target [14]. The number of cameras to be used and their placement is an important issue because the greater the number of cameras, the more will be the amount

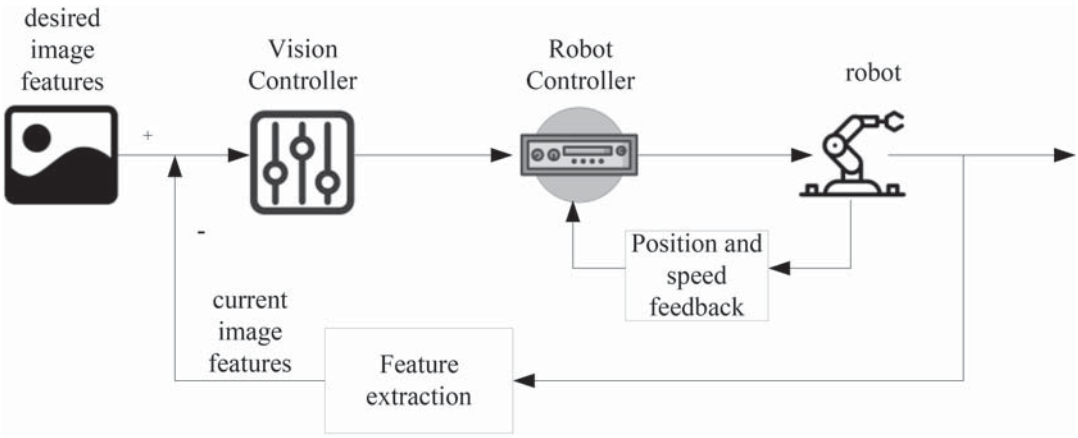


Figure 1 IBVS control structure.

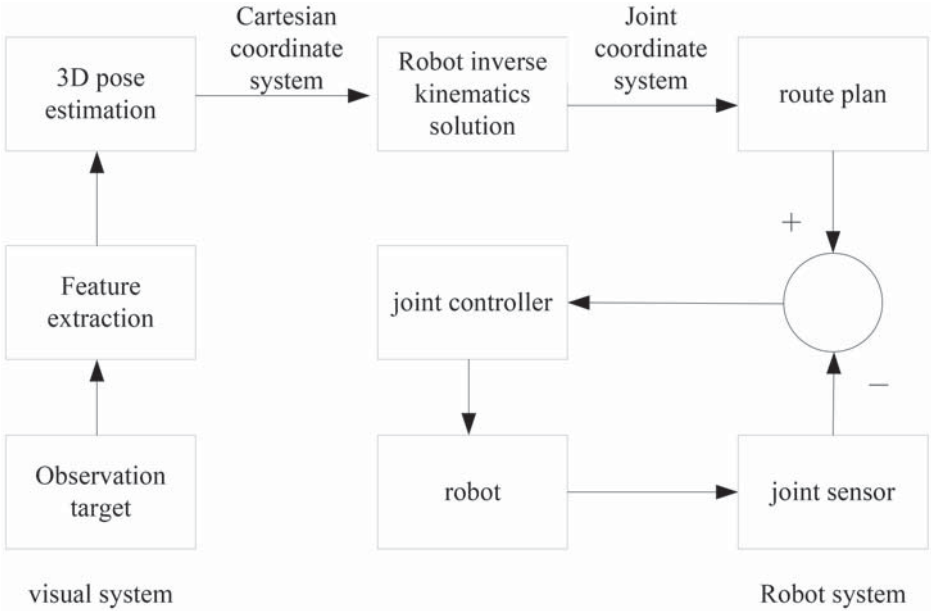


Figure 2 The look-then-move principle.

of information obtained. If the camera is placed properly, a very good shooting effect can be obtained. Conversely, if the camera is placed incorrectly or the information provided by the camera is insufficient, the visual servo will fail.

A system with more than two cameras in a visual servo system is generally referred to as a "multi-eye" visual servo system. The multi-eye visual servo system can observe different parts of the target according to the needs, so it can obtain enough rich information. However, it has greater computational complexity, the control system design is difficult and the cost is high, so it cannot be applied to industrial practice [15]. Therefore, in practical applications, people usually choose a simpler and lower-cost monocular system or a binocular system. The IBVS control structure is shown in Figure 1.

It can be seen from Figure 1 that the eye-in-hand visual servo system can achieve precise positioning and tracking of characteristic targets. However, it is worth mentioning that this configuration can be applied only to scenes with small workspaces, and the application field is relatively limited. On the other hand, the camera in this configuration cannot observe the robotic arm itself. So, when the robotic arm moves, accidents such as collisions are likely to occur, resulting in damage to the robotic arm [16]. In addition, this configuration is highly susceptible to calibration and kinematic errors [17]. Conversely, a configuration that separates the camera and robot and places them separately is called a "fixed-eye" system. With this configuration, the entire work scene can be comprehensively observed and sufficient environmental information can be obtained. However, there is a disadvantage that must be pointed out: the camera movement is likely to occlude the feature target and make the visual servoing fail.

The "see first, then move" method is a control method that separates the vision of the robotic arm and the motion of the robotic arm into two independent processes, as shown in Figure 2.

It can be seen from Figure 2 that in the control process, the visual camera first detects the target and obtains the target information. Then, according to this information, it selects the representative target features, which is conducive to the identification of the target. Finally, it calculates the three-dimensional pose of the target according to the calibrated

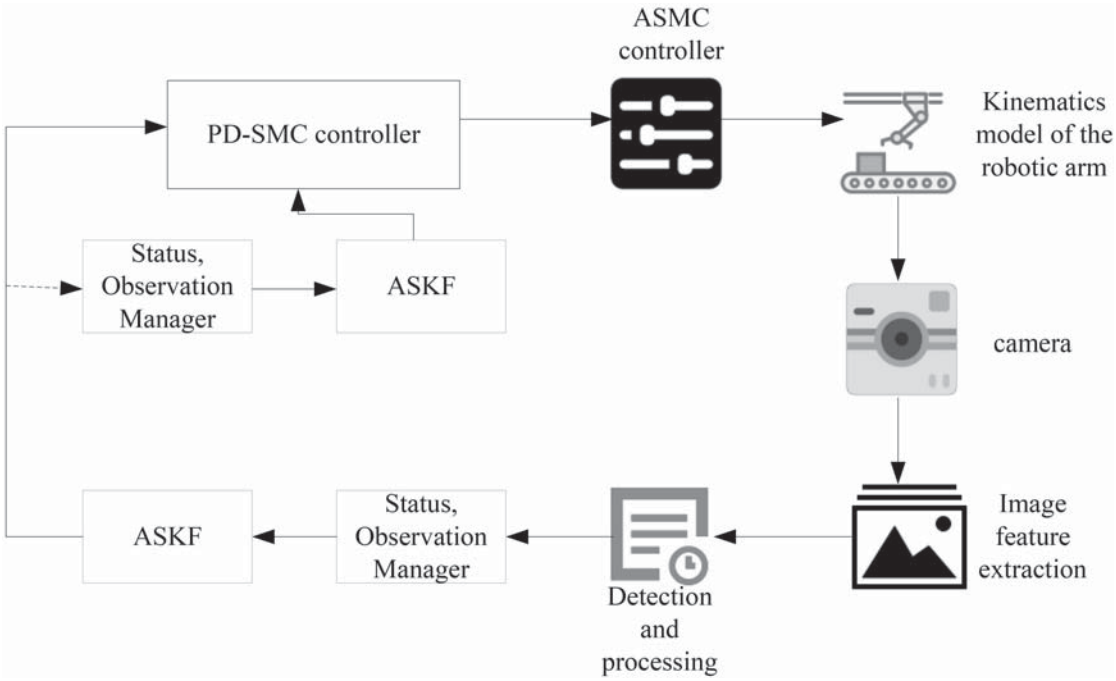


Figure 3 Visual servo double closed-loop control algorithm structure.

camera and the characteristics of the target. So far, this is the task that the entire vision system is responsible for in one movement of the robotic arm [18]. It performs path planning and controls the movement of the robotic arm. During this movement, the vision system no longer detects the target, which is equivalent to open-loop control.

This control method links the vision system of the robotic arm with the motion system. The amount of control is calculated after external information has been acquired through vision, and the changes in visual information during the movement are fed back in time to update the control amount in time [19]. Therefore, this control method can detect whether there is an error in the visual system or an error in the movement process, thereby ensuring the accuracy of the entire control process. In addition, the visual control system is no longer limited to relying only on accurate camera calibration, and the mapping relationship can be updated during the movement process. In this way, the adaptability of the robotic arm to the environment is improved, and the autonomous control capability of the robotic arm is greatly improved, which can eliminate the need for manual operations to a great extent. Therefore, this control method has been widely used in practical situations. However, the "look-and-move" control method also has problems. The most critical problem is how to use the image features to establish a relationship with the motion control of the robotic arm. In addition, since real-time feedback is required during the movement, this relationship requires real-time, updated calculations. How to ensure the speed and accuracy of the update process is also one of the problems [20]. Therefore, the later research on visual servoing is often based on the "look-and-move" control method, and the algorithm research and optimization are carried out on each link, especially in terms of the relationship between vision and motion.

2.2 Double Closed-Loop Control Method for Visual Servoing Considering Image Occlusion/Interference Filtering

In this section, the visual servoing of the robotic arm based on the eye-in-hand structure adopts the look-and-move structure to track and control the image features. The control structure is shown in Figure 3.

It can be seen from Figure 3 that for the controller, a double-loop structure including a vision controller and a joint controller is adopted. The vision controller is set in the external control loop through the hand-eye camera according to the image-based visual servo method. As an inner loop controller, an adaptive sliding mode controller for joint angular position control is derived. By means of the ASMC algorithm, the internal joint controller improves the robustness of the tracking of the robotic arm.

First, it is determined whether occlusion/interference occurs according to the image feature information captured by the camera. Using the position information of the image features, the ASTKF algorithm deals with the noise and interference, and data closer to the real image is obtained and introduced into the control law [21].

Second, each element of the image Jacobian matrix is taken as input, and the system state is estimated online to obtain the characteristic Jacobian matrix, which mitigates its vulnerability to external noise and uncertainty.

Finally, the inner loop control adopts the ASMC algorithm to control the position error of the manipulator joints. The robot arm is driven from the initial position to the desired position by the PD-SMC algorithm, and the outer loop control of the visual servo system is performed. Therefore, the occlusion/interference prediction and the online estimation of the image Jacobian matrix are achieved by ASTKF using the

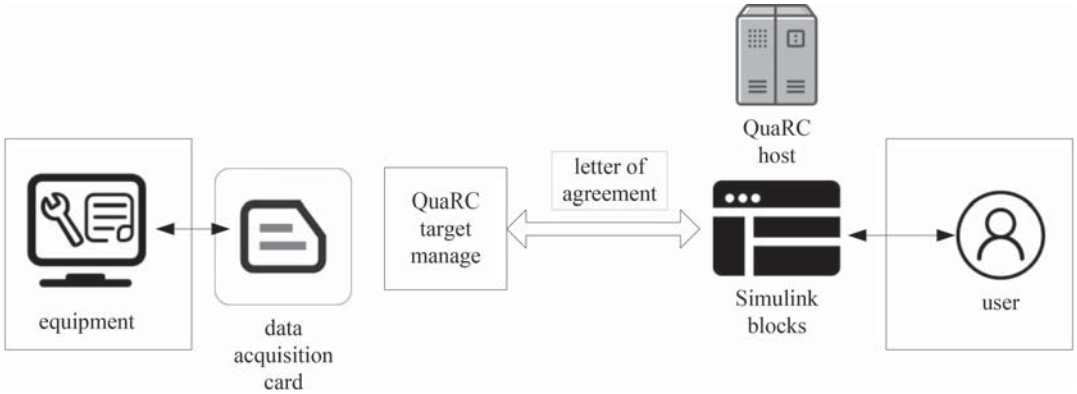


Figure 4 Execution flowchart of QuaRC2.3.

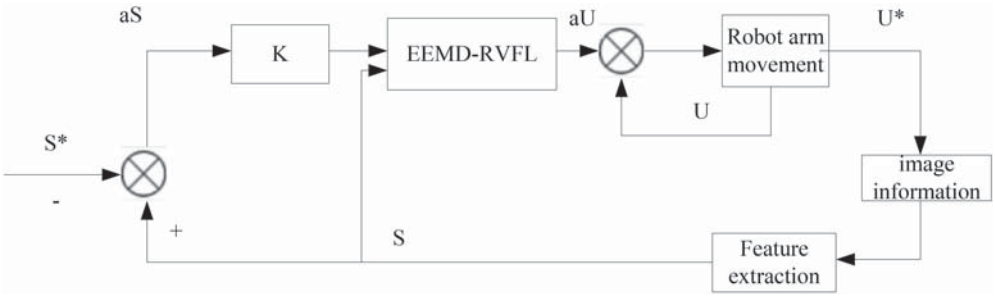


Figure 5 EEMD-RVFL-IBVS system structure.

image feature information, and a double closed-loop control strategy is proposed to realize the tracking control of the visual servo system of the manipulator [22].

2.3 Experimental Platform

The execution flow of QuaRC2.3 of the 6-DOF DENSO open-architecture robot system produced by the company is shown in Figure 4.

As shown in Figure 4, QuaRC2.3 is a multi-functional rapid control development environment developed by Quanser, which seamlessly integrates Simulink. The Quanser open architecture control module features six amplifiers with built-in FF (feedforward) and PID (proportional, integral, derivative) controllers. The controller runs on each motor at a rate of 1kHz. The QUARC2.3 block set has access to all the gains of the built-in controller, and these blocks also have direct access to the amplifier current commands in the block. Users can either tune the built-in controller gain of the QUARC2.3 interface or design their own controller in the Simulink environment and directly command the amplifier current in a completely open architecture [23]. In fully open-architecture mode, a user-defined current is sent directly to the motor, allowing the operator to design a stable feedback system.

2.4 Calibration-Free Visual Servo System Based on eemd-rvfl

Empirical Mode Decomposition (EMD) is a new adaptive signal time-frequency processing method, which is suitable for the analysis and processing of nonlinear and non-stationary signals. It is a major breakthrough in Fourier transform-based linear and steady-state spectral analysis.

The signal is decomposed by EEMD, and then an EEMD-RVFL algorithm is formed by using the prediction function of the RVFL network. In the process of identifying the Jacobian matrix of the robot arm with this algorithm online, the EEMD-RVFL-IBVS algorithm is formed. Since EEMD can decompose the signal containing noise, it can better filter the noise in IBVS. The IBVS structure based on EEMD-RVFL is shown in Figure 5.

In Figure 5, S^* and S represent the expected coordinates of the spatial feature points on the camera plane and the current image feature coordinates, respectively. aS represents the image feature error during each iteration. aU represents the change amount of the manipulator joint in the current iteration process, and U and U^* represent the current joint angle of the manipulator and the joint angle of the previous moment. The value of K is obtained with Formula (1).

$$K = aS \cdot aU^{-1} \tag{1}$$

In this model, the mapping relationship between the changes of the joints of the manipulator and the changes of the image features in each iteration process is shown in Formula (2).

$$aU = f_{EMMD-RVFL}(S, K^{-1} \cdot aS) \tag{2}$$

This visual servo control method does not need to solve the inverse of the matrix every time during the iterative process, avoiding the singularity problem. At the same time, it is not necessary to iteratively update J every time, and the system

structure is relatively simple. Moreover, EEMD-RVFL has a certain filtering effect, so the mapping relationship of this image Jacobian matrix performs well in IBVS.

First, the purpose of IBVS control is to control the end effector of the manipulator from the current pose to the desired pose by minimizing the difference between the current feature and the desired feature of the image. That is, to minimize the error $k_s(i)$, the error is defined as follows:

$$k_s(i) = S(p_t(i), a) - S^* \tag{3}$$

In the IBVS method, the linear mapping relationship between the image feature vector and the camera speed is as Formula (4):

$$\dot{S} = L_s \alpha \tag{4}$$

In Formula (4), $L_k \in R^{n\times 6}$ is called the interaction matrix with S, and $\dot{S}$ is the time derivative of the image feature. α is the speed of the camera.

$$k(i) = L_k \alpha \tag{5}$$

Of these,

$$\alpha = -\beta L_k^+ K \tag{6}$$

In a real visual servo system, it is actually impossible to really know L_k or L_k^+. Therefore, one needs to obtain an approximation or estimate of L_k^+ for one of these two matrices, then:

$$\alpha = -\beta \hat{L}_k^+ k = -\beta \hat{L}_k^+ (S - S^*) \tag{7}$$

It is assumed that the robotic arm has m DOF, and that the camera is fixed on the robotic arm end effector with the same pose as the end effector. The joint angle of the manipulator is $d = [d_a, \ldots d_m]^T$, then the relationship between the joint speed $d^* = [d_2^*, \ldots d_m^*]^T$ and the end effector speed α is as follows:

$$\alpha = J(d)d^* \tag{8}$$

The relationship between the rate of change of the image feature error and the joint speed of the manipulator is as Formula (9):

$$S^* = J_d \cdot d^* \tag{9}$$

According to formulas (7), (8) and (9), the joint speed controller of the manipulator can be redefined as:

$$d^* = \beta J_d^+ k \tag{10}$$

From Formula (9), it can be known that:

$$J(d) = \left[\frac{\partial S}{\partial d}\right] \tag{11}$$

The precise calculation of the image Jacobian matrix is a key issue in IBVS control. The Kalman filter is an optimal linear state estimation algorithm in the environment of independent white Gaussian noise. The two algorithms are:

$$Y_{n+1/n} = EY_{n/n} + M_n \tag{12}$$

$$Z_{n+1} = H_{n+1} Y_{n+1/n} + V_{n+1} \tag{13}$$

$Y_{n/n}$ is the state vector of the robotic system, so:

$$Y_{n/n} = [j_{11}, j_{12}, \ldots, j_{tm}]^T_{(t\bullet m)\times 1} \tag{14}$$

In the observation model Formula (13), $Z_{n+1} \in R^t$ is the observation quantity of the system at the current moment, which is mainly given by the following formula:

$$Z_{n+1} = S_{n+1} - S_n = Jd \cdot q(n) \tag{15}$$

From this, the observation matrix H_{n+1} can be obtained with:

$$H_{n+1} = \begin{bmatrix} d^*(n)^N & 0 \\ 0 & d^*(n)^N \end{bmatrix}_{m\times(t*m)} \tag{16}$$

According to the state model Formula (12) and the observation model Formula (13), and according to the Kalman-Bacy filtering algorithm, the following recursive estimation is established:

Prediction steps:

$$Y_{n+1/n} = EY_{n/n} \tag{17}$$

$$P_{n+1/n} = H_n P_{n/n} H_n + D_n \tag{18}$$

Update steps:

$$K_{n+1} = P_{n+1/n} H_{n+1}^N (H_{n+1} P_{n+1/n} H_{n+1}^N + R_{n+1})^{-1} \tag{19}$$

$$Y_{n+1/n+1} = Y_{n+1/n} + K_{n+1}(Z_{n+1} - H_{n+1} Y_{n+1/n}) \tag{20}$$

$$P_{n+1/n+1} = (E - K_{n+1} H_{n+1}) P_{n+1/n} \tag{21}$$

Of these, D_n and R_{n+1} are the process noise and observation noise covariance matrix, respectively. From the prediction step and the update step, it can be seen that the simple KF-based image Jacobian matrix online identification algorithm has obvious defects, which requires the process noise and observation noise to be Gaussian white noise. However, in the real environment, the KF algorithm is very sensitive to the statistical characteristics of the noise generated in the robot operation, and the noise introduced by the vision sensor is not simple Gaussian white noise.

3. EXPERIMENTS RELATED TO VISUAL SERVOING OF ROBOTIC ARMS

3.1 Hybrid Visual Servo

Image-based visual servoing takes the image feature error as the system reference input, so it is called "2D visual servoing". This method acts directly on the image feature space, so the control algorithm can ensure that the image features converge to the desired position with a specific trajectory, but it cannot guarantee the optimal trajectory in the Cartesian space. A typical problem is the camera rollback phenomenon. The position-based visual servo control takes the Cartesian space velocity as the system reference input, and the control acts on the working space of the manipulator instead of the image feature space. In the initial stage of some specific visual servoing tasks, the movement trajectory of the end of the robotic arm in a certain direction is first away from the target position for a distance, and then approaching the target position, which is called the "camera back-off phenomenon", as shown in Figure 6.

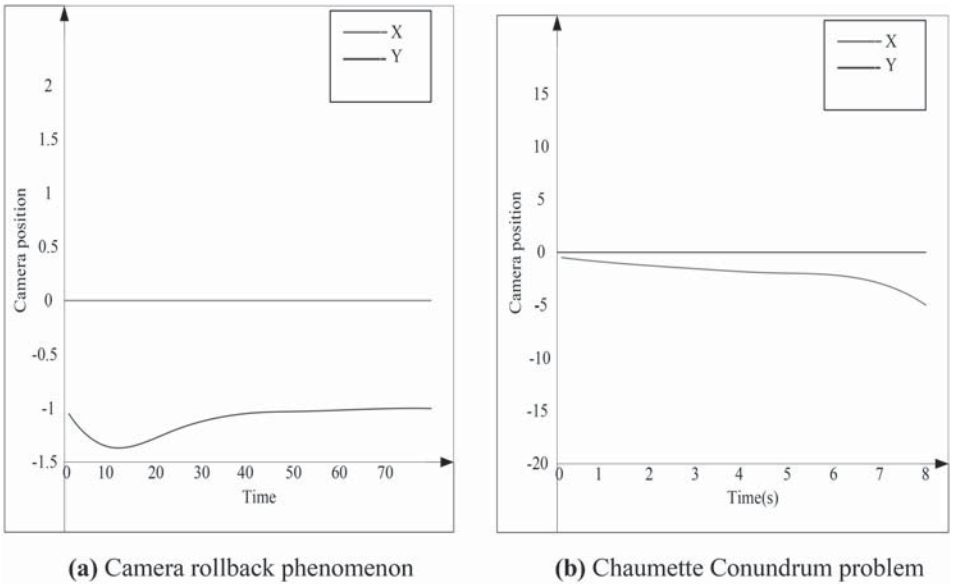


Figure 6 Camera rollback phenomenon and the Chaumette.

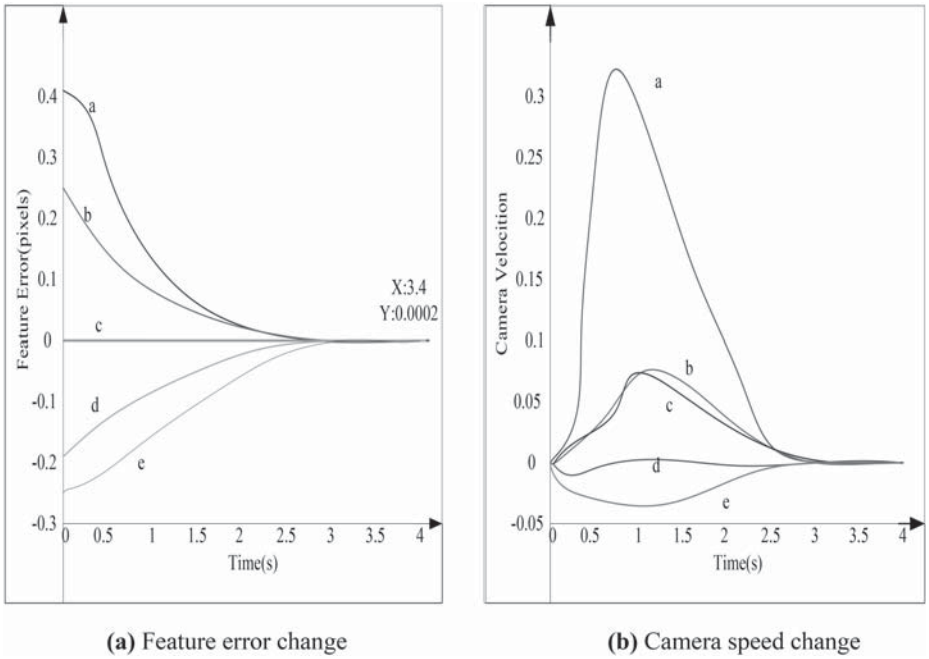


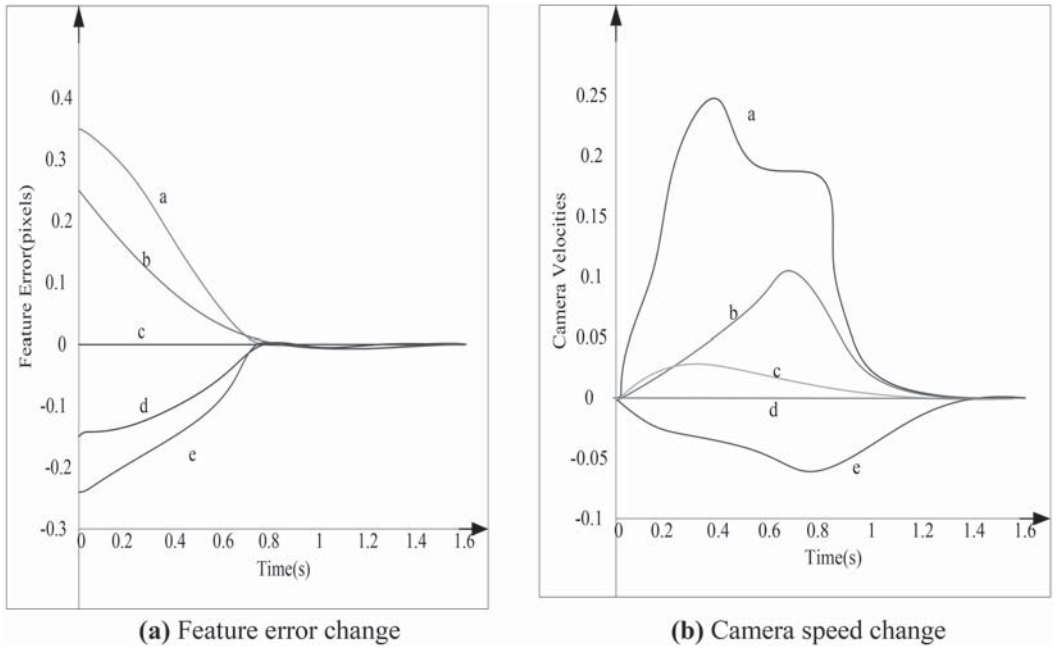
Figure 7 Image-based viper850 visual servo simulation.

It can be seen from Figure 6 that for small amplitude movements, this phenomenon is not obvious. However, this phenomenon is particularly serious for large-scale motion, especially when there is a rotational motion around the X-axis. It can be seen from Figure 6(a) that Chaumette Conundrum requires the feature point of the image plane to rotate 180 degrees around the center point. In theory, this task can be accomplished as long as the camera rotates 180 degrees around the X axis. However, in the actual image-based visual servoing system, according to the proportional control law, the image feature points would converge to the desired feature points along the gradient direction, that is, the direction of the blue line arrow. When the feature points coincide at the center, the image Jacobian matrix appears singular, resulting in visual servo failure. It can be seen from Figure 6 (b) that the camera would retreat to infinity along the Y-axis direction.

3.2 Visual Servo Control Simulation of viper850 Manipulator Based on Image

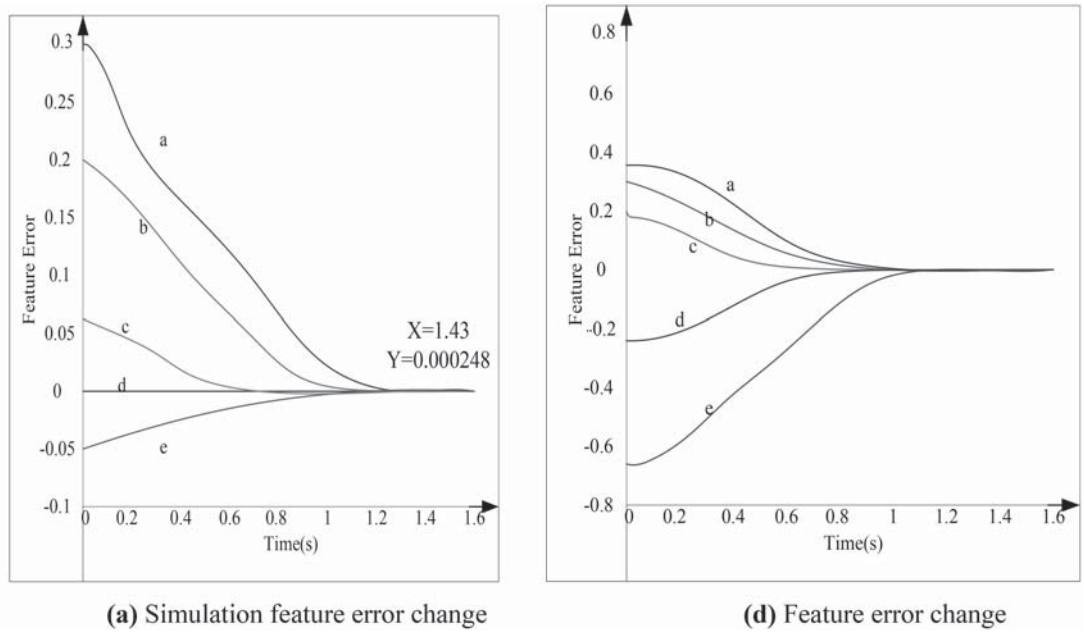
The Viper850 visual servo simulation of the image is shown in Figure 7.

Figure 7 depicts the visual servo simulation effect diagram of the Viper850 manipulator based on the image when the scale factor is 0.5. Figure 7(a) is the change curve of the image features, and Figure 7(b) is the change curve of the camera



(a) Feature error change **(b)** Camera speed change

Figure 8 Simulation of visual servo control based on adaptive gain.



(a) Simulation feature error change **(d)** Feature error change

Figure 9 Visual servo rendering of viper850 robotic arm based on hybrid visual servoing.

shooting speed. The manipulator converges to the desired position at 3.40s. The image-based Viper850 visual servo simulation shown in Figure 7 considers the introduction of an adaptive gain algorithm for the above-mentioned possible robotic arm tremors. A larger scale factor is used at the beginning of the control, and this is gradually decreased as the manipulator gets closer to the desired position. A typical approach is to make the scale factor decay exponentially with time.

The visual servo control simulation is shown in Figure 8.

Figure 8 is a data analysis diagram of the simulation effect of visual servo control based on adaptive gain. Figure 8(a) is an image feature (pixel) map, and Figure 8(b) is a graph of the camera speed curve. The manipulator converges to the desired position in 1.66s, which is faster than with fixed gains.

3.3 Viper850 Robot Arm Hybrid Vision Servo Simulation

The visual servo effect of the Viper850 robotic arm of the hybrid visual servo is shown in Figure 9.

Figure 9(a) is a hybrid visual servo simulation of a robotic arm using extended image features. Figure 9(b) is the control effect diagram of the Viper850 robotic arm. The algorithm can make the manipulator converge to the desired position in 1.43s.

Table 1 UCI dataset.

data set	Number of samples	input feature dimension
abalone	4000	7
fried	25000	8
wine	1600	10
machine_cpu	300	9
puma8NH	8000	7

Table 2 Results of various ELM algorithms on the abalone dataset.

algorithm	activation function	training results		test results	
		MAE	TIMR(s)	MAE	TIMR(s)
ELM	sig($L = 50$)	0.0579	0.0016	0.0579	0.0003
	sin($L = 50$)	0.0619	0.0013	0.0629	0.0003
OSRLM	sig($L = 20$)	0.0587	0.0553	0.0599	0.0008
	sin($L = 50$)	0.0601	0.0511	0.0611	0.0005
MXEDKOSELM	rbf + rbf	0.0600	11.12	0.0579	0.0467
	rbf + poly	0.0609	13.01	0.0611	0.0304

Table 3 Results of various ELM algorithms on the Fried dataset.

algorithm	activation function	training results		test results	
		MAE	TIMR(s)	MAE	TIMR(s)
ELM	sig($L = 50$)	0.0629	0.0911	0.0619	0.0086
	sin($L = 50$)	0.0978	0.0889	0.1269	0.0081
OSRLM	sig($L = 20$)	0.0621	1.0921	0.0637	0.0087
	sin($L = 50$)	0.0964	1.1062	0.0981	0.0093
MXEDKOSELM	rbf + rbf	0.0626	75.75	0.0379	0.6670
	rbf + poly	0.0674	93.12	0.0660	0.7289

3.4 MIXEDKOSELM Algorithm Performance

First, in order to test the stability and accuracy of the MIXEDKOSELM algorithm proposed in this chapter, the MIXEDKOSELM algorithm is compared with several existing common ELM algorithms. The UCI dataset is shown in Table 1.

It can be seen from Table 1 that for testing the performance of various algorithms, 70% of the samples were used as the training set, and 30% were used as the test set. Also, each algorithm was tested 20 times on each dataset, and the average of the 20 results was used to determine the final training accuracy. The three types of errors are defined as shown in the following formulas:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|x_t - \hat{x}_t| \tag{22}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2} \tag{23}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{x_t - \hat{x}_t}{x_t}\right| \times 100\% \tag{24}$$

For the comparison between the MIXEDKOSELM algorithm and other more common ELM algorithms, the objects being compared were ELM and OSELM. Then, different algorithms were trained with different datasets respectively. The results are shown in Table 2.

For the purpose of comparison, Table 2 shows the training results and test results obtained after testing various algorithms on the abalone dataset. The two results are compared in terms of MAE and TIME respectively. It can be seen from Table 2 that ELM still maintains the minimum value in terms of training time and test time. Since ELM randomly generates input layer weights and hidden layer biases, it has a great advantage in regard to time. However, in terms of accuracy, the proposed MIXEDKOSELM algorithm performs well, especially during the testing process. Hence, the structure of the two rbf cores has obvious advantages. In addition to the slowness in terms of time, the other three values are lower, indicating that the MIXEDKOSELM algorithm has better fitting and scalability, as shown in Table 3.

Table 3 presents the results of the three algorithms used on the Fried dataset. Similar to Table 2, the training results and the test results are given, but unlike Table 2, the comparison between the KELM and MIXEDKELM algorithms is not given. This is because the Fried dataset is too large, and KELM needs to calculate the value of the kernel function. At this time, the multiplication operation of two matrices of size 30400*30400 needs to be performed, and the memory of the computer used in the experiment is only 8Gb, which cannot be directly calculated. Therefore, this experimental result is

Table 4 Comparison results of three ELM algorithms on the visual servo dataset.

algorithm	activation function	training results		test results	
		MAE	TIMR(s)	MAE	TIMR(s)
ELM	sig($L = 50$)	0.0161	0.0024	0.0990	0.0005
	sin($L = 50$)	0.0175	0.0022	0.0481	0.0005
OSRLM	sig($L = 20$)	0.0161	0.0370	0.0568	0.0012
	sin($L = 50$)	0.0415	0.0299	0.0671	0.0004
MXEDKOSELM	rbf + rbf	0.0399	6.1211	0.0482	0.2471
	rbf + poly	0.0616	9.9096	0.0611	0.3011

not compared with that of the simple KELM algorithm. It can be seen from Table 3 that the MIXEDKOSELM algorithm has good performance on the Fried dataset. The algorithms are very stable in terms of accuracy. Although the performance in terms of time is not excellent, the system performs better in terms of accuracy.

3.5 Comparison of MIXEDKOSELM Algorithm and Other ELM Algorithms on Visual Servo Datasets

To test the applicability of the proposed algorithm in IBVS, 1000 sets of data are collected during the Kalman filtering process. The input is a vector consisting of the Kalman filter gain error, state estimation error and observation error, and the output is a vector representation of the Jacobian matrix. In this process, the input dimension is 440 and the output dimension is 48. Several algorithms are compared with the MIXEDKOSELM algorithm. The results are shown in Table 4.

It can be seen from Table 4 that the data sets collected during the Kalman filtering process perform differently in terms of the three comparison algorithms mentioned above. The MIXEDKOSELM proposed in this paper outperforms other algorithms in the testing process, especially in the configuration of two rbf kernel functions. Therefore, the MIXEDKOSELM algorithm in the subsequent KF-MIXEDKOSELM-IBVS algorithm adopts this configuration.

4. CONCLUSIONS

In today's society, robots play a role that cannot be underestimated. More and more intelligent robots appear in daily life, liberating people from tasks that are either or boring. Vision is the most important means by which robots perceive the real world, so the research of robotic arms based on image features is a very hot topic. This paper focused on the study of a visual servo system based on image features, and proposed two visual control algorithms, which are based on Kalman filter and neural network respectively. Due to the introduction of Kalman filtering and neural network methods, the calculation of depth information in IBVS can be omitted, reducing equipment and additional overhead. The KF-MIXEDKOSELM-IBVS was formed using MIXEDKOSELM in the Kalman filtering process. This algorithm can compensate for the error caused by the Kalman filter. It can be seen from the experimental results that this IBVS control algorithm has good robustness, and the experimental accuracy is greatly improved. In this paper, the neural network was used to fit the image Jacobian matrix, and the RVFL algorithm was used to train the neural network. Before the input of RVFL, the method of ensemble empirical mode decomposition was added to decompose the complex signal containing noise information into several sub-signals, and then these signals were used as the input of RVFL to train the neural network.

For a monocular visual servo system, the acquisition of depth information is far more difficult than the camera parameters that can be estimated by a one-time algorithm, and its real-time requirements are greater. In the visual servoing process, the depth information was set as a constant in this paper. When the difference between the constant and the actual depth information is small, the influence on the servo system is small; on the contrary, if the difference is large, this causes a large lag in the vision system. In addition, depth information can be estimated online by means of machine vision technology, or through the image features and motion information of the robotic arm. The acquisition of depth information is a problem requiring in-depth discussion in the field of robotic arm visual servoing.

REFERENCES

1. Q. Liang, YJ. Li, DL. Chen, S. Serikawa, M. Guizani, ZH. Lv. A survey on 5G/6G, AI, and robotics. Computers & Electrical Engineering 95 (2021), 107372.
2. PF. Shan. Image segmentation method based on K-mean algorithm. EURASIP Journal on Image and Video Processing (2018): 81 (2018), DOI: 10.1186/s13640-018-0322-6.
3. YZ. Hsieh, SS, Lin. Robotic arm assistance system based on simple stereo matching and q-learning optimization. IEEE Sensors Journal 20(18) (2020), 10945–10954.
4. YG. Xie, GJ. Liu, JY. Lan, BB. Gao. Abdominal needle assisted robotic arm motion control and trajectory planning. Journal of Computer Science 31(2) (2020), 298–311.
5. H. Zhong, Z. Miao, Y. Wang, J. Mao, L. Li, H. Zhang. A practical visual servo control for aerial manipulation using a spherical projection model. IEEE Transactions on Industrial Electronics 67(12) (2020), 10564–10574.
6. DP. Massimo. Toy vision-guided 3d robotic arm in Javascript. Computing in Science & Engineering 20(1) (2018), 43–49.
7. C. Yang, H. Wu, Z. Li, H. Wei, W. Ning, CY. Su. Mind control of a robotic arm with visual fusion technology. IEEE Transactions on Industrial Informatics 14(9): (2018), 3822–3830.

8. J. Qu, F. Zhang, Y. Fu, G. Li, S. Guo. Adaptive neural network visual servoing of dual-arm robot for cyclic motion. Industrial Robot 44(2): (2017), 210–221.
9. P. Zapotezny-Anderson, C. Lehnert. Towards active robotic vision in agriculture: a deep learning approach to visual servoing in occluded and unstructured protected cropping environments. IFAC-PapersOnLine 52(30) (2019), 120–125.
10. Y. Qiu, B. Li, W. Shi, Y. Chen. Concurrent-learning-based visual servo tracking and scene identification of mobile robots. Assembly Automation 39(3): (2019), 460–468.
11. T. Cheng, W. Li, SH. Ng. Visual servo control of a novel magnetic actuated endoscope for uniportal video assisted thoracic surgery. IEEE Robotics and Automation Letters 4(3): (2019), 3098–3105.
12. D. Tsai, DG. Dansereau, T. Peynot, P. Corke. Image-Based visual servoing with light field cameras. IEEE Robotics and Automation Letters 2(2): (2017), 912–919.
13. P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, C. Silvestre. Landing of a quadrotor on a moving target using dynamic image-based visual servo control. IEEE Transactions on Robotics 32(6) (2017), 1524–1535.
14. X. Chen, B. Zhao, Y. Wang, X. Gao. Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm. Journal of neural engineering 16(2): (2019), 026012.1–026012.11.
15. G. Silveira. On intensity-based nonmetric visual servoing. IEEE Transactions on Robotics 30(4) (2017),1019–1026.
16. XF. Li, J. Wei, HS. Jiao. Real-time tracking algorithm for aerial vehicles using improved convolutional neural network and transfer learning. IEEE Transactions on Intelligent Transportation System 23(3) (2022), 2296–2305.
17. JD. Brown, CE O'Brien, SC. Leung, KR. Dumon, DI. Lee, KJ. Kuchenbecker. Using contact forces and robot arm accelerations to automatically rate surgeon skill at peg transfer. IEEE Transactions on Biomedical Engineering 64(9) (2017), 2263–2275.
18. C, Bousquet-Jette, S. Achiche, D. Beaini, LK. Cio, C. Leblond-Menard, M. Raison. Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot. Engineering Applications of Artificial Intelligence 63(Aug.) (2017), 33–44.
19. D. Banerjee, K. Yu, G. Aggarwal. Robotic arm based 3d reconstruction test automation. IEEE Access 6(1) (2018), 7206–7213.
20. M. Reynolds. Vision system speeds syringe filling. Packaging World 25(6) (2018), 52–53.
21. G. Silveira. On intensity-based nonmetric visual servoing. IEEE Transactions on Robotics 30(4) (2017), 1019–1026.
22. H. Fu, CH. Yan, A. Cheung, HC. Man, KY. Chiu. Robotic-Arm assistance simplifies hip arthrodesis conversion to total hip arthroplasty. Arthroplasty Today 6(4) (2020), 877–887.
23. I. Daniyan, K. Mpofu, B. Ramatsetse, A. Adeodu. Design and simulation of a robotic arm for manufacturing operations in the railcar industry. Procedia Manufacturing 51(8) (2020), 67–72.