Fault estimation and accommodation for virtual sensor bias fault in image-based visual servoing with eye-in-hand configuration using Particle filter

Fault Estimation and Accommodation for Virtual Sensor Bias Fault in Image-Based Visual Servoing using Particle Filter

Mien Van, Michalis Mavrovouniotis, *Member, IEEE*, Shuzi Sam Ge, *Fellow, IEEE*

***Abstract*—This study develops a fault estimation and accommodation schemes for the image-based visual servoing (IBVS) system to eliminate the effects of the faults due to the image feature extraction task, which is named as bias virtual sensor fault. First, a bias virtual sensor fault in visual servoing is declared. Then, fault accommodation scheme is developed based on a fault diagnosis (FD) observer to compensate for the effect of such kind of fault. FD task, which includes fault detection, isolation and estimation, are designed based on the means of particle filter (PF). Finally, a fault accommodation law is developed based on the information obtained from the fault estimation to compensate for the effects of the fault in the system. The proposed fault estimation and accommodation is verified through simulation and experimental studies, and the results show that the system can estimate and eliminate the unknown fault effects effectively.**

***Index Terms*—Fault diagnosis, Fault tolerant control, Image-based visual servoing, Particle filter, Robot control.**

I. Introduction

IMAGE BASED VISUAL SERVOING (IBVS), has been proven as an effective method for the robotic system guided by visual information due to its easy in implementation and high accuracy [1]. There are an increasing number of applications which can significantly benefit by using the IBVS approaches to extract position of the geometric feature/target which has inherent error. For example, robotic laser welding process guided by real-time seam tracking or edge detection (welding on the fly) [2]; or application of laser or white light scanners used for in-process or in-line 3D parts geometry inspections [3]. However, the current visual servoing approaches encounter some limitations as discussed below. In traditional visual servoing approaches, the image features are defined based on the geometric characteristic of the object such as points, ellipses, straight lines, or segments, etc, [4-5]. The control law is calculated based on the displacement of the designed image features during visual servoing. Based on this principle, the robot tracks the object correctly when the displacements of all the designed image features are correctly identified. However, this major task has sometime been failed due to the effects of the complex environment during visual servoing [5-6]. In general, the failures can be caused by: (1) image singularities [6-7], (2) the lack of field of view (FOV) of camera [8-9]: due to the visibility constraint of the camera, some features may go out of FOV of camera during visual servoing, and (3) the environment noises: due to the change of the environment such as light condition, obstacles during visual servoing, some designed image features could be occluded, or some undesired image features could be appearing. In order to avoid the image singularities, effective visual features such as polar features [6] or moment [7] have been proposed. To avoid the loss of features due to the visibility constraint, numerous published literatures have been developed to increase the FOV of camera [8-11]. The features of these methods have been thoroughly reviewed in [12]. To reduce the collision with obstacle, surface laser scanning has been developed [13]. Although these approaches can effectively avoid the failures due to the lack of FOV of camera, collision with obstacle and image singularities, the failure of the image feature extraction task due to the image noises has not been considered yet; in fact this failure scenario is usually occurred in real applications. For example, the bias fault due to image noise in seam extraction of robotic welding system, illustrated in Fig. 1: the demand of the feature extraction task in seam extraction for V-groove type is to identify the three image feature points, as shown in Fig. 1a); however, due to the similar property of the desired feature point and the noise feature point, the system extract the noise image feature point instead of the desired feature point, and thus the displacement of the designed feature will be calculated incorrectly, as shown in Fig. 1b). In order to monitor the failure due to the feature extraction task, fault diagnosis observers based on Kalman filter has been developed [12, 14]. In this approach, the nonlinear dynamic model, in which the camera velocity is defined as the input and the displacements of the feature points are defined as the output, of the visual servoing was investigated. Based on the defined dynamic model of the visual servoing, the failure of the feature extraction tasks can be considered as the virtual sensor faults [12, 14]. Then, a fault detection and isolation scheme has been established based on the Kalman filter. However, this approach has not considered the fault estimation, which is a significant task to identify the severity level of the fault. In addition, the approach has not investigated for the fault accommodation, which is desired in real applications to compensate for the effect of the fault to guarantee that the system can guarantee the desired performance even in the presence of fault.

In this paper, as a second part of the previous approach [12], we investigate a fault diagnosis scheme, which includes fault detection, isolation and estimation, for IBVS. The fault diagnosis observer is designed based on particle filter (PF). The PF is employed because it has a good capability to handle nonlinear and non-Gaussian models, as well as its robustness

and flexibility compared to Kalman filter or other filters [15-18]. After a fault is diagnosed, it is desired that the controller should be reconfigured to reduce the effect of fault [19-20]. This task is known as fault accommodation, or fault tolerant control (FTC). Generally, there are two ways to compensate for the effects of fault in the system [21]: (1) Passively, FTC is designed based on the assumption that the set of possible system faults can be predicted in advance, and a fixed control law is designed based on the predicted fault for both normal and fault operations [22]. However, the prior knowledge of the possible system fault is difficult to obtain in visual servoing system since the level of noise of the system is difficult to be known in advance. (2) Actively, namely active FTC (AFTC), the control law is adjusted based on the fault information, which is obtained from a fault diagnosis observer scheme [23-24]. The operation of the AFTC includes two stages. In the first stage, a FD observer is designed to online estimate the system faults. In the second stage, the system uses the obtained fault information to reconfigure the control law. Compared to the passive FTC, the active approach has a higher performance when the fault magnitude is correctly estimated, and thus is employed in this paper.

In summary, the contribution of this paper can be marked by the following significant points:

- Virtual sensor fault in visual servoing system is broadly reviewed.
- A FD observer is designed to detect, isolate and estimate the severity of virtual sensor fault, based on the Particle filter.
- An AFTC control law is developed to compensate for the effect of fault in the system.

The rest of this paper is organized as follows. In section II, problem formulation is stated. In section III, FD and FTC strategies based on particle filter are presented. In section IV, we verify the proposed methodologies based on simulation study. The performance of the proposed strategies is further verified through experimental study, in section V. The conclusion and future work are provided in section VI.

II. PROBLEM FORMULATION

Considering the pinhole perspective model of camera used in visual servoing system [1, 4], the projection of the 3D points $P_i = \left[x_i, y_i, z_i\right]^T$, $i = 1,...,n$ into the image plane of camera is $s_i = \left[u_i, v_i\right]^T$, $i = 1,...,n$, where [12]:

$$s_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{\lambda}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

where λ is the focal length of the camera. The relationship between the displacement of the feature point in the image plane $\dot{s}$ and the spatial velocity of camera $V_c = [\upsilon_x, \upsilon_y, \upsilon_z, \gamma_x, \gamma_y, \gamma_z]^T$ can be expressed as:

$$\dot{s}_i = L_i\left(s_i, z_i\right) V_c, \quad i = 1,...,n \tag{2}$$

where, $L_i\left(s_i, z_i\right)$ is the interaction matrix, and

$$L_i\left(s_i, z_i\right) = \begin{bmatrix} -\frac{\lambda}{z_i} & 0 & \frac{\lambda u_i}{z_i} & u_i v_i & -(1+u_i^2) & v_i \\ 0 & -\frac{\lambda}{z_i} & \frac{\lambda v_i}{z_i} & (1+v_i^2) & -u_i v_i & -u_i \end{bmatrix}$$

z is the depth of the image features and is assumed to be known [4].

The target of the control law in visual servoing system is to minimize the error e, which is defined as the different between the current feature point s and their goal value s^*, $\mathrm{e} = \mathrm{s} - \mathrm{s}^*$. The traditional control law is designed as

$$V_c = -\delta \hat{\mathrm{L}}_{\mathrm{s}}^{+} \mathrm{e} \tag{3}$$

where δ is a positive gain and $\hat{\mathrm{L}}_{\mathrm{s}}^{+}$ is a an approximation of pseudoinverse of $\mathrm{L}_s = [L_1, L_2, ..L_n]^T$.

Discretization is applied to (2) [12]:

$$\begin{aligned} \chi_{\mathrm{k+1}} &= \mathrm{A}\chi_{\mathrm{k}} + \mathrm{B}_k \nu_k + \Delta_k + \omega_k \\ \psi_k &= \mathrm{C}\chi_k + \zeta_k \end{aligned} \tag{4}$$

where $\chi = s = \left[u_1, v_1, ..., u_n, v_n\right]^T$ denotes the state variable of the system, $\nu = KV_c$ is the control input, where K is the sampling time, Δ_k is used to represent the system uncertainty, which is defined as the error when discretizing (2) to (4) [12], ω and ζ are the model uncertainty and measurement noise, respectively. The coefficient matrix are defined as

$$\mathrm{A} = \mathrm{I}_{(2\times n, 2\times n)}, \ \mathrm{B} = L_s = \left[L_1, L_2 ..., L_n\right]_{(2\times n, 6)}^T \tag{5}$$

$$\mathrm{C} = \mathrm{I}_{(2\times n, 2\times n)} \tag{6}$$

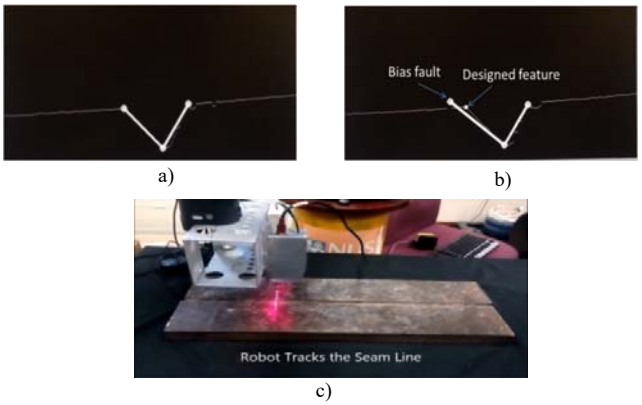
When a virtual sensor fault occurs, the true value of the designed image feature may not be determined. This means that the input signal is measured as $\overline{s}(t) = s(t) + \Delta s(t)$, where $s(t)$ is the true signal and $\Delta s(t)$ is the fault signal.

In the presence of the virtual sensor fault, the system dynamics is changed to

$$\begin{aligned} \chi_{\mathrm{k+1}} &= \mathrm{A}\chi_{\mathrm{k}} + \mathrm{B}_k \nu_k + \Delta_k + \omega_k \\ \psi_k &= \mathrm{C}(\chi_k + \theta(t)) + \zeta_k \end{aligned} \tag{7}$$

where the fault function is defined by

$$\theta(t) = \begin{cases} 0, & t \neq T_s \\ \Delta s(t), & t = T_s \end{cases} \tag{8}$$



a) b) c)

Fig. 1. Bias fault in weld seam extraction of welding robot system. a) normal operation, b) virtual sensor bias fault, c) welding robot system.

where T_S is the time that the fault occurs.

The objective of this paper is twofold: 1) design a fault diagnosis scheme based on Particle filter to detect, isolate and estimate the unknown fault $\theta(t)$, and 2) design a fault accommodation scheme such that the visual servoing can self-compensate for the effect of fault and continue reliable working with an acceptable performance even the fault existing in the system.

III. Fault Diagnosis and Fault Tolerant Control based on Particle Filter

In visual servoing, a virtual sensor fault potentially results in malfunction of the overall system. To prevent any undesired consequences, a monitoring system is necessary to detect and isolate the faults as quickly as possible.

Based on the property of the faults, fault diagnosis problem can be considered as three major tasks [19-21]: 1) fault detection: makes a binary decision whether and when any adnormal event in the monitored system happens, or if everything working well, 2) Fault isolation identifies the root of the fault, 3) Fault identification or fault estimation specifies the magnitude of the fault. In the following the Particle filter-based fault diagnosis is presented.

A. Particle Filter

Consider the dynamic system of interest is described by

$$\begin{aligned} \chi_k &= f(\chi_{k-1}, \xi_{k-1}) \\ \psi_k &= h(\chi_k, \zeta_k) \end{aligned} \tag{9}$$

where χ_k is the state variable, ψ_k is the measurement. The system and measurement noise ξ_k and ζ_k are assumed to be independent of ψ_k. However, unlike the Kalman filter, they need not be Gaussian distributed. Because the visual servoing system described in (4) might not be a Gaussian distributed system due to the uncertainty Δ, the Particle filter would be effective to approximate the system states.

From a Bayesian perspective, the problem of the state estimation is to calculate the probability density function (pdf) $p(\chi_k | \psi_{1:k})$ of the state χ_k based on the sensor data available up to time k, $\psi_{1:k} = \{\psi_1, \psi_2, ..., \psi_k\}$. Starting from the values of the initial condition $p(\chi_0 | \psi_0) \equiv p(\chi_0)$ and the pdf, $p(\chi_{k-1} | \psi_{1:k-1})$ at time $k-1$, there are two steps to update the pdf at the time k, $p(\chi_k | \psi_{1:k})$.

i) Prediction step:

$$p(\chi_k | \psi_{1:k-1}) = \int p(\chi_k | \chi_{k-1}) p(\chi_{k-1} | \psi_{1:k-1}) d\chi_{k-1} \tag{10}$$

ii) Update step:

$$p(\chi_k | \psi_{1:k}) = \frac{p(\psi_k | \chi_k) p(\chi_k | \psi_{1:k-1})}{p(\psi_k | \psi_{1:k-1})} \tag{11}$$

where $p(\psi_k | \psi_{1:k-1})$ is a normalizing factor that depend on the pdf $p(\psi_k | \chi_k)$.

In theory, the Bayesian filter can estimate the true state variable by using the above two recursive steps (10) and (11). However, the approach can only give the optimal solution if the system can satisfy two assumptions: the noises are Gausian distribution and the system is linear. However, these assumptions are not usually satisfied in real applications. To overcome the limitation, Particle filter, which is an approximation method of Bayesian filter, has been proposed. The PF approximate the pdf using a set of N particles, $\{\chi_k^i, \rho_k^i\}$, where χ_k^i presents the ith particle and ρ_k^i presents its associated weights.

In literature, many algorithms have been developed for Particle filter. In this paper, we use sequential importance resampling (SIR) [15] due to its efficient and simple in implementation. In the following, the structure of the SIR algorithm is presented.

SIR Algorithm [15].

1) For i=1,…, N, a new particle χ_k^i is generated based on the pdf $p\left(\chi_k \mid \chi_{k-1}^i\right)$ and the corresponding weight is computed as $\rho_k^i = p\left(\psi_k \mid \chi_k^i\right)$.

2) Compute the sum of weights $\Omega_w = \sum_{i=1}^{N} \rho_k^i$ and then normalize the particle weights: $\rho_k^i = \Omega_w^{-1} \rho_k^i$.

3) Do the sampling process:

3.1. Start from $c_0 = 0$, construct the cumulative sum of weights (CSW) by computing $c_i = c_{i-1} + \rho_k^i$ for $i = 1, ..., N$.

3.2. Give $i = 1$ and generate a starting point τ_1 from the uniform distribution $U\left[0, N^{-1}\right]$.

3.3. For $j = 1, ..., N$

- *Make $\tau_j = \tau_1 + N^{-1}(j-1)$.*
- *While $\tau_j > c_i$ make $i = i + 1$.*

- *Assign* $\chi_k^j = \chi_k^i$.
- *Assign* $\rho_k^j = N^{-1}$

B. Fault Detection and Isolation Based On Particle Filter

The PF estimate output for the dynamic model described in (7) is:

$$\hat{\psi}=\mathrm{C}\hat{\chi} \tag{12}$$

where $\hat{\chi}$ is the PF state estimation outputs, which are determined as the output of SIR algorithm applied for the system (7).

In fault diagnosis task, it is important to choose the effective residual, which can be easily distinguished between normal condition and fault condition when the system is being changed from the normal operation to the fault operation, and the corresponding threshold. In this paper, the error e, which is defined as in (13), is chosen as the residual.

$$\mathrm{e} = \psi - \hat{\psi} \tag{13}$$

In normal operation, the Particle filter state tends to approximate the state variable of the system. Thus, from (7) and (12), the residual e tends to approximate the system uncertainties and noises, $\mathrm{e} \approx \vartheta + \Delta$, where $\vartheta \approx \omega + \zeta$.

Assumption 1: the system uncertainties and noises are bounded by $\|\vartheta + \Delta\| \le e_{Th}$, where e_{Th} is a known constant.

The assumption 1 is reasonable in real application because the noise value is usually bounded by a constant. In practice, the bound value of the system uncertainty and noise are usually obtained by experiments. Since $\mathrm{e} \approx \vartheta + \Delta$ when the system in normal operation, the bound value of $\vartheta + \Delta$ can be estimated based on the bound value of the error e, defined as in (13). In this paper, we employ this method. The procedure to obtain the bound value is performed offline, and is as follows. First, a desired image is obtained by moving the robot to the target position. Then, starting from the arbitrary position, but guarantee that the object be within the FOV of camera, command the robot to track the object using the control law (3). Obtain the residual e when the system in normal operation. As shown in Figs. 5, 8 and 11 (will be discussed latter), the residual e will converge to close zero with small variation due to the noise and uncertainty. The bound value e_{Th} is chosen such that it is bigger than the peak of the variation. After obtaining the e_{Th}, this value is used for online monitoring of the proposed fault diagnosis. Since $\|e\| \approx \|\vartheta + \Delta\| \le e_{Th}$, to distinguish between the effects of the system uncertainty and the virtual sensor fault, e_{Th} is selected as the threshold. Fault decision is made when the residual, ($\|e\|$), surpass its corresponding threshold e_{Th}.

From (1), the change of the state variables u or v can be represented by the change of s. Thus, in order to facilitate in fault detection and isolation of a feature point, the residuals of two state variables u and v should be represented by s, as follows:

$$e_{s_i} = \sqrt{e_{u_i}^2 + e_{v_i}^2} \tag{14}$$

where e_{u_i} and e_{v_i} represent the PF estimation error of the state variables u_i and v_i of the feature point i, respectively, and e_{s_i} is used to represent the PF estimation error of the feature point i.

Then, the decision rule is defined as

$$r_i = \begin{cases} 0 & if\ e_{s_i} \le Th_i \\ 1 & if\ e_{s_i} > Th_i \end{cases} \tag{15}$$

where $Th_i = \sqrt{e_{Th_{u_i}}^2 + e_{Th_{v_i}}^2}$, where $\|\vartheta + \Delta\|_u \le e_{Th_u}$ and $\|\vartheta + \Delta\|_v \le e_{Th_v}$, is a chosen threshold.

The robustness property of the fault diagnosis system is guaranteed and can be explained as follows: when the system in normal operation, the residual is approximated as $e_{s_i} \approx \vartheta + \Delta$, and based on the assumption 1, the residual is always smaller than the chosen threshold value, $e_{s_i} \le Th_i$ and $r_i = 0$. However, when a fault occurs, the residual is approximated as $e_{s_i} \approx \vartheta + \Delta + \theta$. This residual signal will overshoots the threshold value $e_{s_i} > Th_i$ and $r_i = 1$, the fault decision will be made. The fault detection and isolation rules are defined as in Table I.

Remark 1: Table I is also used to define the multiple faults working condition. For example, when faults occur in the

TABLE I
FAULT-SIGNATURE TABLE

Fault	r_1	r_2	r_3	r_4	…	r_n
None	0	0	0	0	…	0
Sen. 1	1	0	0	0	…	0
Sen. 2	0	1	0	0	…	0
Sen. 3	0	0	1	0	…	0
Sen. 4	0	0	0	1	…	0
…	…	…	…	…	…	
Sen. n	0	0	0	0	…	1

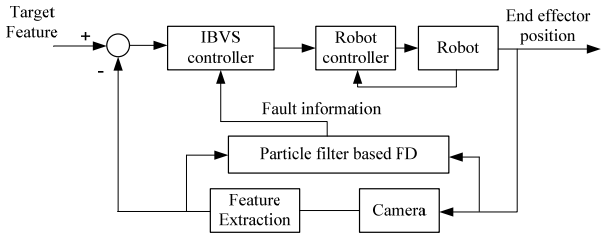


Fig. 2. Fault diagnosis and fault tolerant control scheme for visual servoing system.

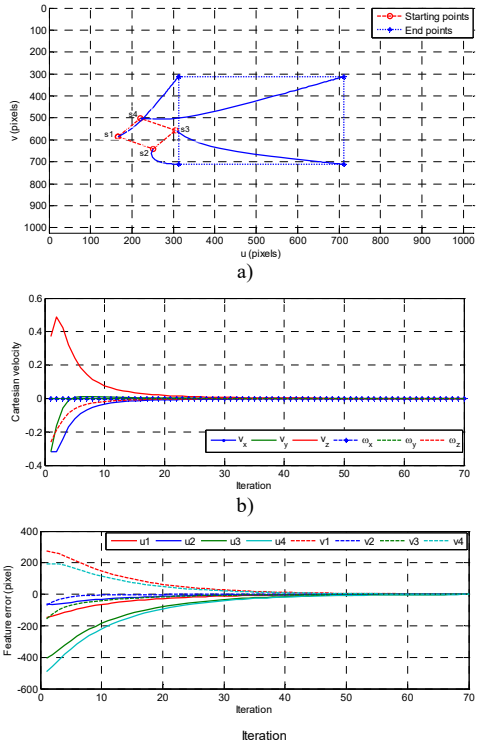


Fig. 3. Tracking performance of visual servoing when the system in normal operation. a) Image space, b) control inputs (noted that $\omega_X \equiv \omega_Y$ in the figure), c) image error.

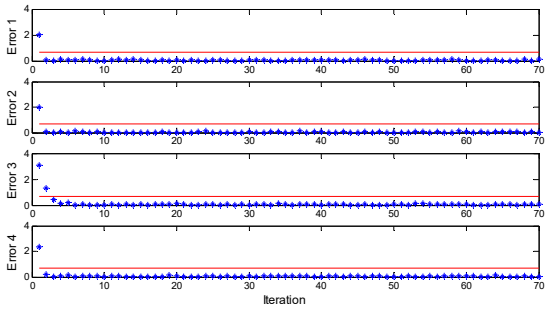


Fig. 4. Residual values and the selected thresholds when the system in normal operation.

sensors 1, 2 and 3 at the same time, the corresponding residuals are $r_1 = 1$, $r_2 = 1$ and $r_3 = 1$.

C. Fault Estimation

In the previous section, the analyses show that the PF approximates the image feature states with a very small error when the system in normal operation, $e_s \approx \vartheta + \Delta$. However, when a fault occurs at the time T_s, the estimation error e_s tends to approximate the fault component, $e_s \approx \vartheta + \Delta + \theta$. Because $\vartheta + \Delta$ is usually much smaller compared to θ, the estimation error approximate the fault magnitude, $e_s \approx \theta$. Thus, the fault magnitude can be approximated as

$$\Delta s(t)_{t=T_s} = e_{t=T_s} \tag{16}$$

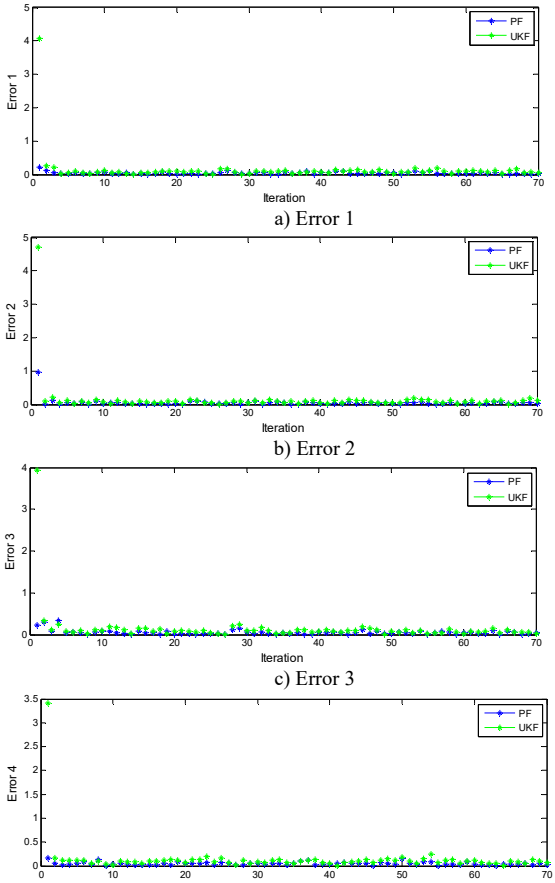


Fig. 5. Comparison between particle filter (PF) and unscented Kalman filter (UKF).

However, T_s is the unknown time, it only can be predicted by using the information from the fault detection and isolation scheme. If we denote T_d as the time that the fault is detected, the fault magnitude can be estimated as

$$\Delta s(t)_{t=T_s} = e_{t=T_d} \tag{17}$$

where $e_{t=T_d}$ denotes the PF estimation error at the time T_d. In practice, if the fault detection and isolation scheme is working well, we will have $T_d \equiv T_s$.

D. Fault Tolerant Control

After a fault is diagnosed, it is desired that the fault should be compensated to reduce its effects on the system. In the traditional visual servoing, the system is controlled by the conventional law (3). The system performance is satisfied only when it operates in normal condition, wherein the system get the feedback of the correct image feature input s. However, when a fault occurs, the fault feature value $\bar{s}(t) = s(t) + \Delta s(t)$ is used as the input signal to the controller that will generate incorrect control input calculated and decrease the tracking performance consequently. In order to increase the system performance, the correct image feature value $s(t)$ at the time

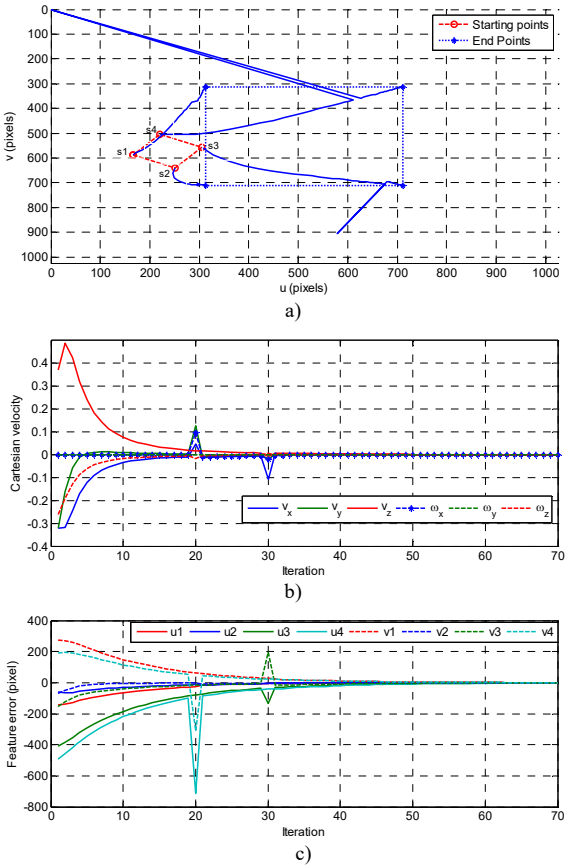


Fig. 6. Tracking performance of visual servoing when the faults occurred in feature points 4 and 3 without FTC. a) Image space, b) control inputs, c) image error.

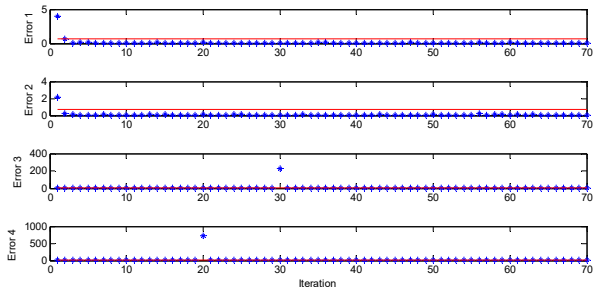


Fig. 7. Residual values when the faults in the points 4 and 3 occur.

T_s should be reconstructed and fed back to the controller instead of the fault value $\bar{s}(t)$. The correct feature value can be simply calculated as $s(t) = \bar{s}(t) - \Delta s(t)$. However, since the correct fault value $\Delta s(t)$ can not be calculated, its estimation value obtained from the fault estimation scheme (17) is used instead. Then, the estimation value of the correct image feature at the time T_s, $\hat{s}(t)$, can be obtained as:

$$\hat{s}(t) = \bar{s}(t) - \mathrm{e}_{t=T_d} \tag{18}$$

Afterward, when a fault is detected, to reduce its effect on the system, the controller is reconfigured as

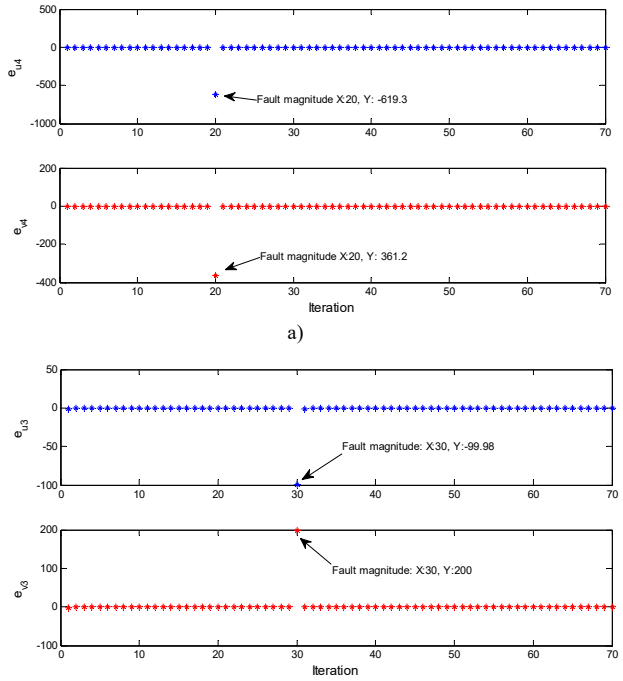


Fig. 8. PF state estimation errors when the faults existed in feature points 4 and 3 without FTC. a) PF estimation error for feature point 4, b) PF estimation error for feature point 3.

$$V_c = -\lambda \hat{\mathrm{L}}_{\mathrm{s}}^{+}(\bar{s}(t) - \mathrm{e}_{t=T_d} - \mathrm{s}^{*}) \tag{19}$$

Finally, the whole FTC law for the visual servoing system is designed as:

$$V_c = \begin{cases} -\lambda \hat{\mathrm{L}}_{\mathrm{s}}^{+}(s - \mathrm{s}^{*}) & t \neq T_d \\ -\lambda \hat{\mathrm{L}}_{\mathrm{s}}^{+}(\bar{s}(t) - \mathrm{e} - \mathrm{s}^{*}) & t = T_d \end{cases} \tag{20}$$

The overall FD and FTC scheme developed in this paper is illustrated in Fig. 2.

IV. Simulation Study

In this section, the performances of the visual servoing system with and without the FTC are simulated to show the performance of the FD and FTC schemes. The target used in this simulation is masked by four feature points. The image resolution is 1000x1000 pixel. The sampling time is 5 frame-per-seconds (fps). The number of particles is set as *N*=500; this value is chosen based on the trial and error validation through several experiments. Eight internal states, $u_1, v_1, u_2, v_2, u_3, v_3$ and u_4, v_4 are approximated by the PF. Starting from the initial camera location, where the four points can be seen by dashed lines in the image space in Fig. 3a), the target of the visual servoing system is to locate the camera at the position such that the four points can be seen by dot-dashed lines in the image space in Fig. 3a). In order to compare the tracking performance of the system between normal operation, fault operation without FTC and fault operation with FTC, the visual servoing is modeled in three

TABLE II
COMPARISON BETWEEN UNSCENTED KALMAN FILTER (UKF) AND PARTICLE FILTERS (PFS)

Method		Point 1		Point 2		Point 3		Point 4		Computation time
		RMSE	VAR	RMSE	VAR	RMSE	VAR	RMSE	VAR	
UKF		0.0903	0.0420	0.0892	0.0425	0.0896	0.0515	0.0971	0.0438	0.003149
PF	N=200	0.0922	0.0595	0.1065	0.1061	0.0995	0.0528	0.1072	0.1250	0.002373
	N=500	0.0450	0.0256	0.0478	0.0250	0.0450	0.0312	0.0496	0.0289	0.007080

different working conditions. In the first case, the visual servoing system is modeled to operate in normal condition. In the second case, the failures of features points 3 and 4 are generated to the system without FTC. In the third case, the FTC control law is employed to reduce the effects of faults generated in the second case. In addition, multiple faults condition is also reported to further verify the performance of fault estimation and compensation of the system.

A. Visual Servoing System in Normal Operation

Considering the operation of the system in normal operation, as shown in Fig. 3, the PF approximates the nonlinear visual servoing system with a small error due to the uncertainties and noise, $\vartheta + \Delta$, as shown in Fig. 4. We can see from Fig. 3 that the camera tracks the object very well. From Fig. 4, the PF estimation errors converge close to zero very fast (after a few iterations). To distinguish between the effects of the uncertainties and faults such that the system can correctly distinguish between the normal operation and fault operation, the threshold values Th are selected as the red line, shown in Fig. 4. The fault decision is made whenever the residual surpasses the corresponding threshold value.

To further evaluate the performance of the PF to approximate the system states, we simulate the system with the measurement noise $\delta = 5$. In addition, we compare the performance of the PF, with different number of particles used, with unscented kalman filter (UKF), which is a well-known nonlinear filter and widely applied in real applications. The results are shown in Fig. 5. For easy in comparison, the root mean square error (RMSE) and standard deviation (STD) and the computation time of these methods are also reported in Table II. From the results we can see that both the UKF and PF filters approximate the system states very well. On the other hand, with the lower number particles (N=200), the performance of the PF is worse than the UKF. When the number of particle is larger (N=500), the performance of the PF is better than the UKF. However, there is a trade off in the approximation capability and computation time of the PF. The higher number of particles the better approximation performance but higher computation time, and vice versa.

B. Visual Servoing System with Assumed Virtual Sensor Faults

In order to show the effects of virtual bias fault in the visual servoing system and to verify the performance of the developed FD and FTC schemes, we generate a bias fault to the visual servoing system. Particularly, we assume that the feature point 4 is disappear (occluded) at the iteration 20; it means the extracted value of feature point 4 at the iteration 20 is $[u_4 = 0, v_4 = 0]_{t=20}$, as shown in Fig. 6a). This fault can be represented by a bias fault with magnitude

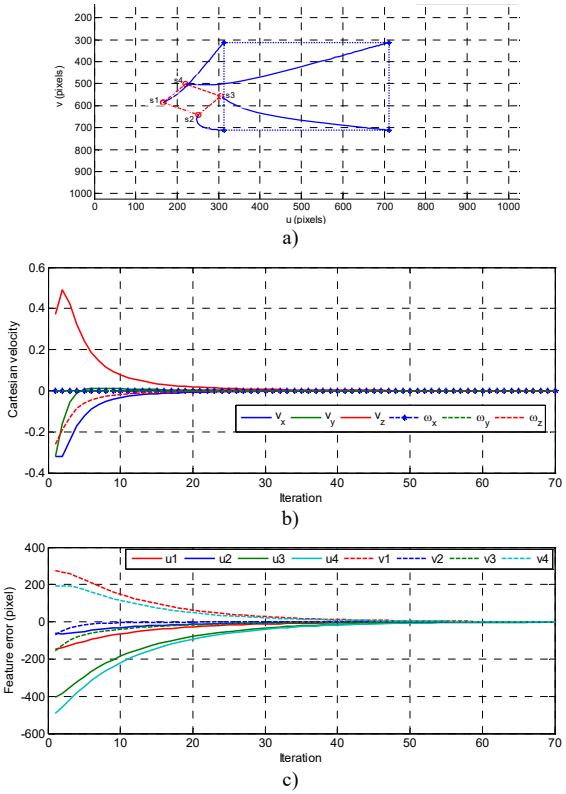


Fig. 9. Tracking performance of visual servoing when the faults existed in feature points 4 and 3 with FTC. a) Image space, b) control inputs (noted that $\omega_x \equiv \omega_y$ in the figure), c) Image error.

$\Delta s_4 = [-u_4, -v_4]_{t=20} = [-619.42, 361.17]$. And, another bias fault is assumed to exist on the feature point 3 at the iteration 30 with the magnitude $\Delta s_3 = [-100, 200]$. Figure 6 illustrates the varying of the system performance when the system changes from the normal operation to fault operation. Comparison results between Fig. 3 and Fig. 6 show that the motion of the camera is incorrect if the displacement of a feature is incorrectly extracted. In particular, due to the effect of the virtual sensor faults, the corresponding velocity control input is discontinuous at the iterations 20 and 30, as shown in Fig. 6b), that will make the visual servoing system unstable. Due to the effect of the fault, the convergence of the PF is broken, as shown in Fig. 7. The residual of the feature point 4 overshoots the threshold at the iteration 20, indicating that the fault has been existed in the virtual sensor 4 and the residual of feature point 3 overshoots the corresponding threshold at the iteration 30, indicating that the fault has been existed in the virtual sensor 3 at the iteration 30. Thus, in this experiment, the system has detected and isolated the faults successfully.

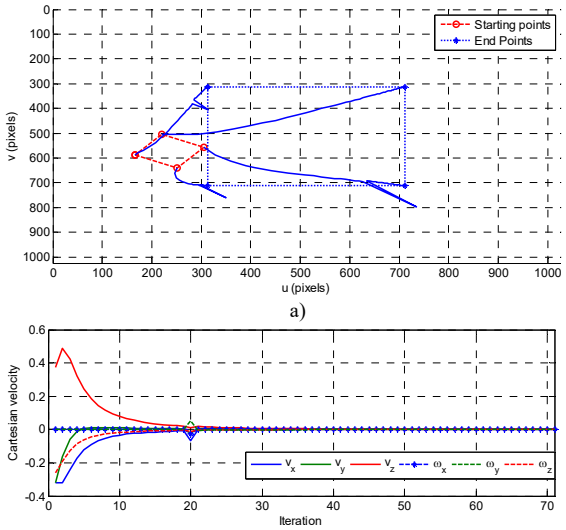


Fig. 10. Tracking performance of experimented visual servoing when the faults in the points 1, 2 and 3 occur. a) Image space, b) control inputs.

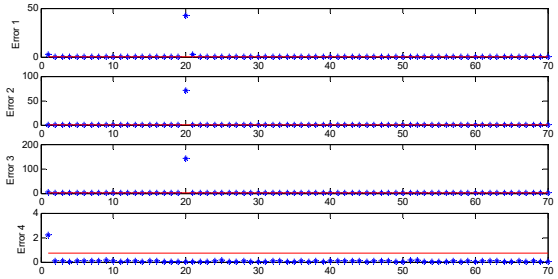


Fig. 11. Residual values when the faults in the points 1, 2 and 3 occur.

Now, we consider the fault estimation performance. At the iterations 20 and 30, where the existed faults in the feature point 4 and the feature point 3 have been successfully detected and isolated, the fault estimations were then computed by the formulation defined in (17) that gave $\hat{\Delta}s_4 = \mathrm{e}_{t=T_d=20} = [-619.3, -361.2]$ for the virtual sensor 4 and $\hat{\Delta}s_3 = \mathrm{e}_{t=T_d=30} = [-99.98, 200]$ for the virtual sensor 3, as shown in Fig. 7. The estimated fault values, $\hat{\Delta}s_4$ and $\hat{\Delta}s_3$, are very close to the generated fault values, Δs_4 and Δs_3. Thus, we can conclude that the faults have been estimated correctly.

C. Visual Servoing System With Fault Tolerant Control

As shown in Fig. 5b), the effects of fault generate the discontinuous control input, which will make the incorrect camera motion. To reduce the effect of fault, the developed FTC law in (20) is employed based on the feedback information of the estimated fault obtained in the Fig. 7. The performance of the developed FTC for the visual servoing is shown in Fig. 8. Comparison results between the Fig. 8 and Figs. 2 and 5 show that the FTC system compensates the effects of the fault very well. The system performance of the visual servoing under FTC input is comparable with the normal operation case. The velocity control input of the system with FTC is continuous, as shown in Fig. 8b). Thus,

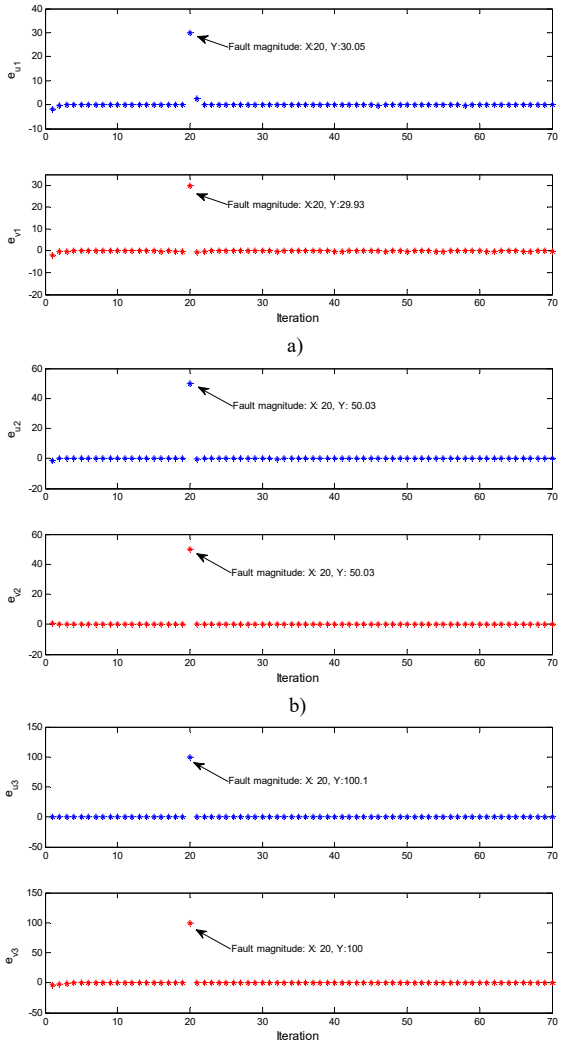


Fig. 12. PF state estimation errors when the faults in the points 1, 2 and 3 occur. a) PF estimation error for feature point 1, b) PF estimation error for feature point 2, and c) PF estimation error for feature point 3.

we can conclude that the fault has been accommodated successfully.

D. Visual Servoing System with Multiple Fault Condition and Fault Tolerant Control

In order to further show the performance of the proposed method to handle the multiple faults, we simulate the system with a severity fault condition, $\Delta s_1 = [30, 30]$, $\Delta s_2 = [50, 50]$ and $\Delta s_3 = [100, 100]$. The three faults are assumed to be occurred at the same time at the iteration 20. The response of the system under the effect of the faults is shown in Fig. 10. The residual generated by PF is shown in Fig. 11. From Fig. 11, it is obvious to see that the faults occurred in the points 1, 2 and 3 have been detected and isolated correctly. The fault estimation and FTC performance of the system are shown in Figs. 12 and 13, respectively. It can be seen from the Figs. 12 and 13 that the faults have been estimated and compensated effectively.

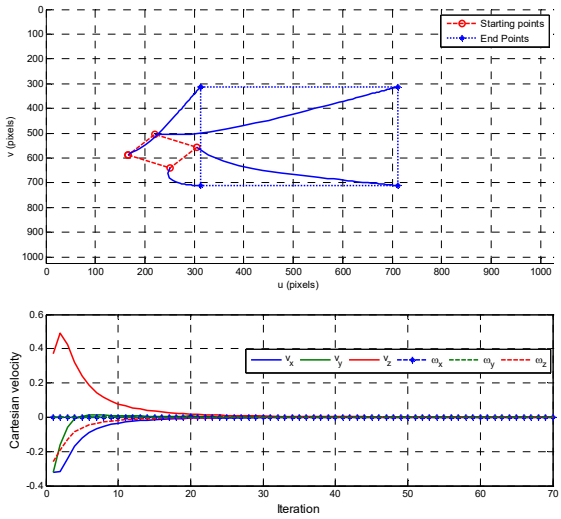


Fig. 13. FTC performance for the visual servoing system when the faults in the points 1, 2 and 3 occur.

Remark 2: If we consider the sensor fault only and assume that the actuator is always healthy, the proposed fault diagnosis and accommodation can handle for the heavy sensor fault case, where all the sensors are failed at the same time. However, if we consider both the actuator and sensor faults, the control system will isolate the wrong actuator fault instead of sensor faults when all the sensors are failed at the same time [12]. Fortunately, this situation is rarely occurring in real application.

Remark 3: To easily verify the performance of the proposed fault estimation, the noise components has not been considered in the fault estimation part. However, the fault diagnosis system is still working well if there is a noise existing in the system, as shown in the Fig. 5. For the bigger of the noise system, the threshold should be chosen as the big value to guarantee the robustness, as analyzed in section IIIB.

V. Experimental Study

In order to show the tracking performance of the system with FD and FTC, a lab experimental setup is developed as shown in Fig. 14. The Baxter industrial robot [25] is used to do experiment. The Baxter is a new generation industrial robot and has been widely using in industrial application and research. The Baxter has two independent arms and each has seven degree-of-freedom (DOF). Each arm has been attached with an eye-in-hand configuration. In this study, we used the left-arm and left-hand camera to do experiments. The object to be tracked used in paper includes four feature points, as shown in Fig. 17. The camera capturing rate is 30 fps (frame/s), and the image information is sent into the host Linux PC to processing. The camera of the Baxter robot has 640x400 pixels resolution and has an effective focal length of 1.2 mm.

Faults are introduced in the virtual sensor by changing the displacements of the image feature points at an arbitrary time. In the following, we present the performance of the visual servoing system without FTC and with FTC when the system in normal and fault operations.

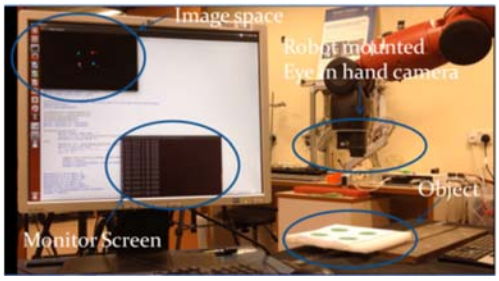


Fig. 14. Experiment setup of eye-in-hand visual servoing.

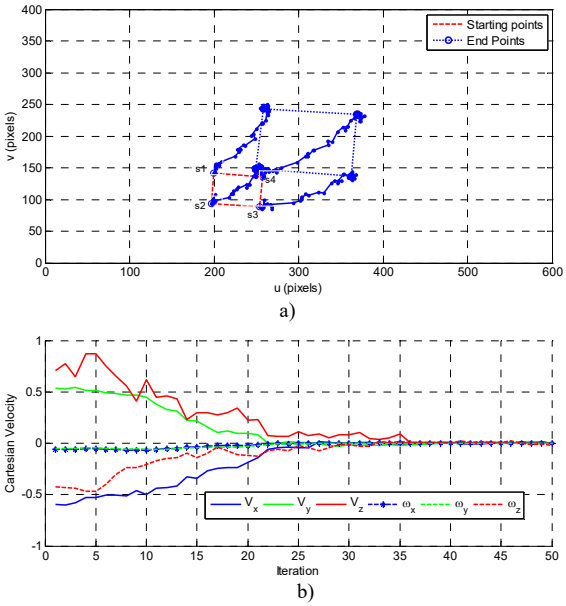


Fig. 15. Tracking performance of experimented visual servoing when the system in normal operation. a) Image space, b) control inputs.

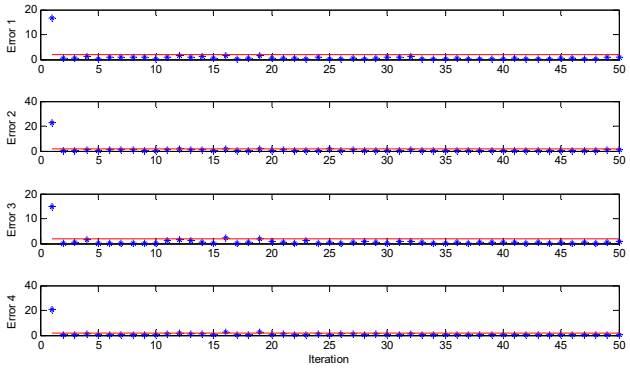


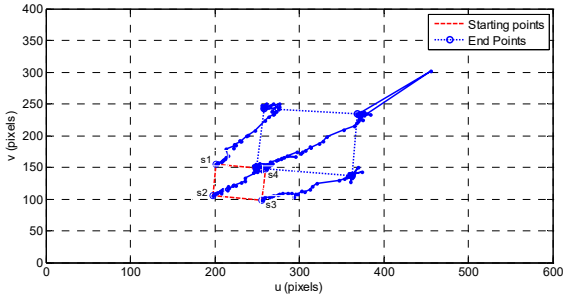
Fig. 16. Residual values of experimented visual servoing and the selected threshold values.

A. Visual Servoing in Normal Operation

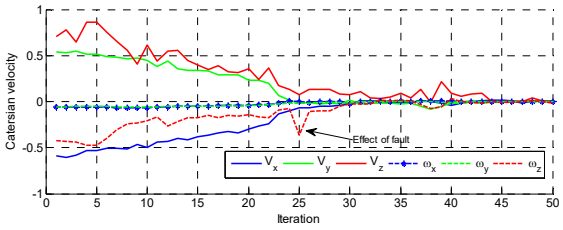
In fault-free working condition, the visual servoing system tracks the object very well, as shown in Fig. 15. The PF estimation errors, which are used as the residuals in this paper, are shown in Fig. 16. The Fig. 16 shows that the PF estimation errors are quickly convergent after a few iterations. As analyzed in section IIIB, the residual value obtained when the system in fault-free operation is the uncertainty and noise components, $\vartheta + \Delta$, of the system. Thus, to avoid any

a) b)

Fig. 17. Image feature in a) normal extraction, b) bias virtual sensor fault.



a)



b)

Fig. 18. Tracking performance of visual servoing when the fault existed in the virtual sensor (feature point 4) without FTC. a) Image space, b) control inputs.

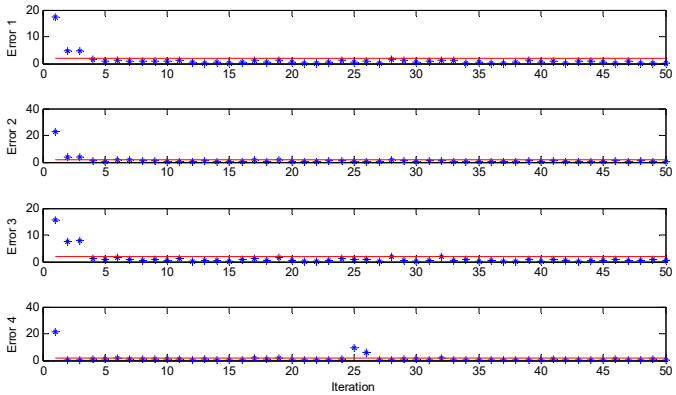
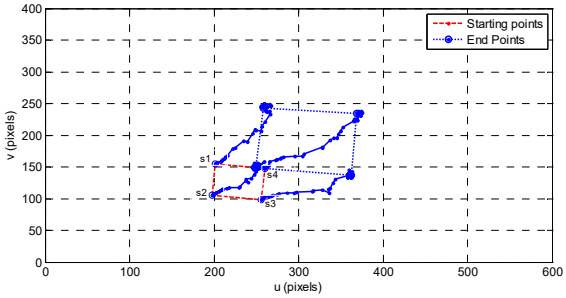


Fig. 19. PF estimation error when the experimented visual servoing sytem in normal operation.

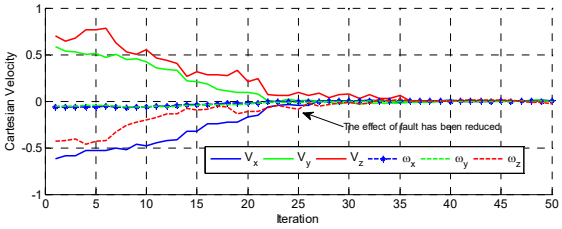
incorrect fault decision due to the effect of uncertainties and noises, the threshold values Th are selected to be bigger than the bound value of $\vartheta + \Delta$. The selected thresholds are the red lines shown in Fig. 16.

B. Visual Servoing System under Virtual Sensor Fault without Fault Tolerant Control

In the presence of a bias fault, the controller system read a noise feature value instead of the true designed feature, as an example shown in Fig. 1. To simulate the bias fault, we change the displacement of the feature point 4 at the iteration



a)



b)

Fig. 20. Tracking performance of the experimented visual servoing system when the fault existed in the feature point 4 with FTC. a) Image space, b) control inputs.

25, as illustrated in Fig. 17. Fig. 18 shows the transition of the tracking performance when the system changing from normal operation to fault operation. From Fig. 18b), due to the presence of fault at the iteration 25, the computed control velocity input is changed suddenly. Due to the effects of the fault, the convergence of the PF estimation error is broken, as shown in Fig. 19. In the Fig. 19, the residual of the feature point 4 overshoots the corresponding threshold at the iteration 25, and thus the fault decision is made.

C. Visual Servoing with Fault Tolerant Control Under Virtual Sensor Fault

The generated bias virtual sensor fault generates the discontinuous control input as shown in Fig. 18b). To tackle this problem, the developed FTC law in (20) is employed. Fig. 20 shows the results of the fault compensation. By comparing Fig. 20b) with 18b), we can see that the effects of the fault in the computed control input are much reduced. From this, we can conclude that the fault has been estimated precisely and its effects on the visual servoing system have been correctly compensated by the developed FTC law.

VI. CONCLUSION

This paper reviews the failure scenarios of the feature extraction task in visual servoing system, namely virtual sensor fault. Then, the PF-based FD is developed to monitor the presence of the fault. An AFTC is developed based on the estimated fault information. Both the designed FD and FTC schemes have a simple structure and easily implement in real application. Simulation and experimental results verify that the presence of the failures due to the feature extraction task can be detected accurately and its effects can be compensated effectively.

According to [12], the failure of visual servoing system could be caused by incorrect robot motion. Fault estimation and

accommodation for the failures of incorrect robot motion will be investigated in our future work.

REFERENCES

[1] M. Keshmiri, W. F. Xie, and A. Mohebbi, “Augmented image-based visual servoing of a manipulator using acceleration command,” *IEEE Trans. Ind. Electron.,* vol. 61, no. 10, pp. 5444-5452, 2014.

[2] D. Ceglarek, M. Colledani, J. Vancza, D. Y. Kim, C. Marine, M. Kogel-Hollacher, A. Mistry, L. Bolognese, “Rapid Deployment of Remote Laser Welding Processes in Automotive Assembly Systems,” *Annals of the CIRP*, vol. 64, no. 1, pp. 389-394, 2015.

[3] Y. Ding, P. Kim, D. Ceglarek, J. Jin, “Optimal Sensor Distribution for Variation Diagnosis in Multi-station Manufacturing Processes,” *IEEE Trans. Robot. Auto.*, vol. 19, no. 4, pp. 543-556, 2003.

[4] A. Hajiloo, M. Keshmini, W. F. Xie, T. T. Wang, “Robust online model predictive control for a constrained image based visual servoing,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2242-2250, 2016.

[5] W. F. Xie, Z. Li, X. W. Tu, C. Perron, “Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 520-529, 2009.

[6] P. Corke, F. Spindler, F. Chaumetter, "Combining Cartesian and polar coordinates in IBVS," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., (IROS’09)*, Oct. 2009, pp. 5962-5967.

[7] J. Wang, H. Cho, “Micropeg and hole alignment using image moments based visual servoing method,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 1286-1294, 2008.

[8] N. G. Aracil, E. Malis, R. A. Santonja, C. P. Vidal, “Continuous visual servoing despite the changes of visibility in image features,” *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713-723, 2004.

[9] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, “Keeping features in the field of view in Eye-In-Hand visual servoing: A switch approach,” *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 908-913, 2004.

[10] M. Kazemi, K. K. Gupta, M. Mehrandezh, “Randomized kinodynamic planning for robust visual servoing,” *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 195-200, 2010.

[11] M. Baumann, S. Leonard, E. A. Croft, J. J. Little, “Path planning for improved visibility using a probabilistic road map,” *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1197-1211, 2013.

[12] M. Van, D. Wu, S. S. Ge, H. Ren, "Fault Diagnosis in Image-Based Visual Servoing with Eye-in-Hand Configuration Using Kalman Filter," *IEEE Trans. Ind. Inform.*, vol.PP, no.99, pp.1-1, doi: 10.1109/TII.2016.2590338, 2016.

[13] L. C. B. Preciado, O. Y. Sergiyenko, J. C. R. Quinonex, X. Garcia, V. V. Tyrase, M. R. Lopez, D. H. Balbuena, P. Mercorelli, M. Podrygalo, A. Gurko, I. Tabakova, O. Starostenko, “Optical 3D laser measurement system for navigation of autonomous mobile robot,” *Optics and Lasers in Engineering*, vol. 54, pp. 159-169, 2014.

[14] M. Van, D. Wu, S. S. Ge, H. Ren, “Condition monitoring for image based visual servoing using Kalman Filter,” *Advances in Visual Computing*, pp. 842-850, 2015.

[15] M. S. Arulampalam, S. Maskell, N. Gordon, ans T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 12, pp. 174-188, 2012.

[16] T. Wei, Y. Huang, C. L. Philip Chen, “Adaptive sensor fault detection and identification using particle filter algorithms,” *IEEE Trans. Syst. Man Cyber.*, vol. 39, no. 2, pp. 201-213, 2009.

[17] N. Widynski, M. Mignotte, “A multiscale particle filter framework for contour detection,” *IEEE Trans. Pattern. Analy. Machine. Intel.*, vol. 36, no. 10, pp. 1922-1935, 2014.

[18] B. Zhao, R. Skjetne, M. Blanke, and F. Dukan, "Particle filter for fault diagnosis and robust navigation of underwater robot," *IEEE Trans. Control Syst. Tech.*, vol. 22, no. 6, pp. 2399-2407, 2014.

[19] S. Laghrouche, J. Liu, F. S. Ahmed, M. Harmouche and M. Wack, "Adaptive Second-Order Sliding Mode Observer-Based Fault Reconstruction for PEM Fuel Cell Air-Feed System," *IEEE Trans. Control Syst. Tech.*, vol. 23, no. 3, pp. 1098-1109, 2015.

[20] J. Liu, W. Luo, X. Yang and L. Wu, "Robust Model-Based Fault Diagnosis for PEM Fuel Cell Air-Feed System," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3261-3270, 2016.

[21] Z. Gao, C. Cecati, S. X. Ding, "A survey of fault diagnosis and fault-tolerant control techniques Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757-3767, 2015.

[22] M. Van, S. S. Ge, H. Ren, "Robust Fault-Tolerant Control for a Class of Second-Order Nonlinear Systems Using an Adaptive Third-Order Sliding Mode Control," *IEEE Trans. Systems, Man, and Cybern.: Systems* , vol.PP, no.99, pp.1-8 doi: 10.1109/TSMC.2016.2557220, 2016.

[23] M. Van, H. J. Kang, K. S. Shin, “Robust fault tolerant control for uncertain robot manipulators based on adaptive quasi-continuous high-order sliding mode and neural network,” *Part C: Journal of Mechanical Engineering Scienece*, doi:10.1177/0954406214544311, 2015.

[24] M. Van, S. S. Ge, H. Ren, "Finite Time Fault Tolerant Control for Robot Manipulators Using Time Delay Estimation and Continuous Nonsingular Fast Terminal Sliding Mode Control," *IEEE Trans. Cybern.*, vol.PP, no.99, pp.1-13, doi: 10.1109/TCYB.2016.2555307, 2016.

[25] C. Fitzgerald, "Developing baxter," in *Proc. Int. Conf. technol. Practical Robot Appl.*, Woburn MA, USA, 2013, pp. 1-6.