Electrical Engineering (2023) 105:3263–3281
https://doi.org/10.1007/s00202-023-01875-7

ORIGINAL PAPER

Adaptive intelligent vision-based control of a flexible-link manipulator

Umesh Kumar Sahu[1] · Dipti Patra[2] · Bidyadhar Subudhi[3]

Received: 30 January 2023 / Accepted: 16 May 2023 / Published online: 12 June 2023

Abstract

Present space robots such as planetary robots and flexible robots have structural flexibility in their arms and joints that leads to an error in the tip positioning owing to tip deflection. The flexible-link manipulator (FLM) is a non-collocated system that has unstable and inaccurate system performance. Thus, tip-tracking of FLM possesses difficult control challenges. The purpose of this study is to design adaptive intelligent tip-tracking control strategy for FLMs to deal with this control challenges of FLM. A vision sensor is utilized in conjunction with a traditional mechanical sensor to directly measure tip-position in order to address the aforementioned problem. Image-based visual servoing (IBVS), one of several visual servoing control techniques, is more efficient. However, the IBVS scheme faces numerous difficulties that impair the system's performance in real-time applications, including singularities in the interaction matrix, local minima in trajectory, visibility issues. To address the issues with the IBVS scheme, a novel adaptive intelligent IBVS (AI-IBVS) controller for tip-tracking control of a two-link flexible manipulator (TLFM) is designed in this study. In particular, this paper addresses the IBVS issues along-with retention of visual features in the field-of-view (FOV). First, in order to retain object within the camera FOV, an intelligent controller with off-policy reinforcement learning (RL) is proposed. Second, a composite controller for TLFM is developed to combine RL controller and IBVS controller. The simulation has been conducted to examine the effectiveness and robustness of the proposed controller. The obtained results show that the AI-IBVS controller developed here possesses the capabilities of self-learning and decision-making for robust tip-tracking control of TLFM. Further, a comparison with other similar approach is presented.

Keywords Flexible-link manipulator · Image-based visual servoing · Reinforcement learning · Robot vision · Tip-tracking control

1 Introduction

Nowadays, the satellite, aerospace, and space industries use lightweight flexible robots, planetary robots, and space robots. Due to its lightweight, lower overall cost, low energy consumption during transportation, larger payload handling capacity, increased maneuverability, and faster operational speed, the flexible-link manipulator (FLM) has many advantages. However, compared to a rigid manipulator, the structural flexibility of FLM arms and joints causes inaccuracy in tip positioning [1]. Over the past four decades, research on FLM control has been active. The control of flexible-link manipulators (FLMs) is well reviewed in [2, 3]. Because FLM is nonlinear and non-collocated, it acts as a non-minimum phase system. Additionally, model truncation and errors are evident, which affects system stability and also leads to the inaccurate tip-tracking performance.

The primary cause of the non-collocation in FLM is the placement of the sensor and actuator in different locations. The majority of the literature uses a standard mechanical sensor, such as a accelerometer, encoder, strain gauge to measure tip position information. However, occasionally electromagnetic interference causes these sensors to perform poorly in the difficult environment and give a noisy response. Since the

✉ Umesh Kumar Sahu
umesh.sahu@manipal.edu

Dipti Patra
dpatra@nitrkl.ac.in

Bidyadhar Subudhi
bidyadhar@iitgoa.ac.in

[1] Department of Mechatronics, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

[2] Department of Electrical Engineering, National Institute of Technology, Rourkela, Odisha 769008, India

[3] School of Electrical Sciences, Indian Institute of Technology Goa, Ponda, Goa 403401, India

tip point information is measured indirectly by these mechanical sensors, a model is required to relate the information to the tip deflection. Moreover, wave propagation along the beam causes the end-effector response to occur a little bit later than a control input. To address this issue, sensor and actuator averaging method were developed in [4]. However, the use of multiple sensors and actuators increases the weight of the flexible manipulator. Instead of using mechanical sensors, optical sensors can also be utilized for the measurement of tip point information, but they are very susceptible to noise. These challenges, which yield an indirect estimate of tip point deflection, are overcome by the vision sensor. Research in flexible manipulator high-performance control using visual servoing (VS) has grown recently. VS in FLM can significantly increase the accuracy of the tip point information.

The eye-in-hand configuration (camera placed in tip, just observing target object) is taken into consideration in this work because it does not take kinematics into account when determining positioning accuracy. Based on the error, there are four visual servoing strategies. It has been established that image-based visual servoing (IBVS), which is more competent than other VS techniques, is one of the preferable strategy for controlling FLMs. Additionally, IBVS removes inaccuracies caused on by sensor modeling and is adaptable to errors in camera calibration. However, the IBVS scheme faces numerous difficulties that impair the system's performance in real-time applications, including singularities in the interaction matrix, local minima in trajectory, visibility issues.

Singularity and local minima in IBVS are caused by improper pairings of visual features that impair FLM's ability to monitor tips. Recent studies reveal that IBVS faces two significant difficulties: (1) choosing visual features to avoid singularities in the interaction matrix and (2) designing a control scheme using those chosen visual features such that FLM track the target trajectory with the least amount of tracking error. Designing and choosing appropriate visual features for IBVS is a challenging task. In [5], the shifted moment-based visual feature is used to address the IBVS approach's issues with singularity in the interaction matrix and local minima in trajectories. The work described in [5] demonstrated robustness with a field-of-view (FOV) limitation, i.e., when the object is partially occluded out of the FOV.

Usually, measured visual features are used as control input for IBVS to compute the controller output. However, due to disruption during movement, objects may occasionally depart the camera's FOV. Keeping the visual characteristics in the camera's field of view becomes difficult in this case. Additionally, the stability and performance of the system are directly impacted by the visual features' visibility. However, the work presented in [5] may fail if the object is fully out of the FOV. Given the success of the image moment-based visual serving control scheme in many robotic applications, in this work to address the visibility issue of IBVS, we expand the approach to design and build an adaptive IBVS controller based on image moment for robust tip-tracking control of TLFM.

Many approaches have been reported to prevent the aforesaid visibility issue of IBVS, for example, potential field [6], navigation function [7], path planning [8]. Also, the visibility issue of IBVS is addressed by employing a pan-tilt camera [9], odometry with vision system [10] and specific visual features [11]. The methods described in [6–11] lack the self-learning and online decision-making capabilities, rendering them unsuitable for real-time applications (i.e., they cannot automatically adapt to changing control tasks). Also, these approaches cannot guarantee that all visual features remain in the FOV [12]. Therefore, a machine learning solution is necessary to solve the aforesaid issue of IBVS. In the realm of robotics, reinforcement learning (RL) [13] is a well-known method for increasing flexibility to changing control tasks and environments and for enhancing self-learning and decision-making capabilities. RL in robotics is applied for control of flexible aircraft wing [14], TLFM [15], SLFM [16] and in many other applications. The algorithm in [15] employs the method of on-policy learning. In the design of proposed intelligent controller, the off-policy learning method is used, as it is model-free, data efficient and faster as compared to the on-policy learning method [17]. In order to keep objects in the FOV of the camera, an intelligent controller with off-policy reinforcement learning is proposed in this study.

In this line of research, similar studies that combine both RL and VS for mobile robot are presented in [18–32]. For VS-based control of a 7-DOF redundant robot manipulator to reach the target position, a self-organizing map (SOM) network-based learning algorithm has been given in [28]. In [29], an interesting method for controlling a mobile robot manipulator by fusing RL and IBVS is described. In this work, off-line training with traditional Q-learning is adopted for robust grasping of spherical object. An improvement over [29] is presented in [30], in which neural network RL (NN-RL) and IBVS is used for control of robot manipulator. To enable online learning and flexibility with changing control tasks, the NN-RL algorithm is applied into a hybrid control system in [30]. In [31], a model-free RL strategy is introduced for the robotic grasping of unknown objects. In [32], the learning outcome of a generative model is directly used in real-time application. Also, asymmetric actor-critic and variational auto-encoder-based RL algorithm are designed to achieve the desired target. However, results on integration of RL and IBVS for tip-tracking control of the TLFM have not been reported yet in the literature, which motivates us to make an effort in this paper. Therefore, in this work, off-policy RL controller is integrated with IBVS controller for accurate and robust tip-tracking control of TLFM is developed.

The objective of this paper is to develop a vision-based tip-tracking control of TLFM, with a view of developing a novel adaptive intelligent IBVS controller. It consists of following contributions.

- An intelligent controller with off-policy reinforcement learning (RL) is developed to guarantee that the object remains within the camera FOV for accurate tip-tracking control of TLFM.
- An adaptive intelligent IBVS (AI-IBVS) controller is implemented into the composite controller to enable the ability of self-learning and decision-making for robust tip-tracking control of TLFM.

The remaining sections of the paper are structured as follows. The preliminary TLFM dynamics and the robust tip-tracking control (RTTC) problem formulation are presented in Sect. 2. The solution to the RTTC problem is presented in Sect. 3, in which the basics of RL (Sect. 3.1) are presented followed by the design of actor-critic-based off-policy RL controller (Sect. 3.2) and the new two-time scale IBVS control scheme (Sect. 3.3). Section 4 presents the development of the proposed adaptive intelligent IBVS controller. In Sect. 5, the training procedure (Sect. 5.1) is presented and analyzed the tip-tracking performance (Sect. 5.2) with symmetrical and non-symmetrical objects to validate the proposed hybrid (AI-IBVS) controller using simulation studies. Also, a brief theoretical comparison is given in Sect. 5.3. The conclusion and scope of further work is given in Sect. 6. Appendix A and Appendix B are included to support the theoretical and simulation studies of the work.

2 Preliminaries and problem formulation

2.1 Dynamics of TLFM

The dynamics of TLFM is given by [5]

$$M(\theta_i, \delta_i)\begin{bmatrix}\ddot{\theta}_i\\ \ddot{\delta}_i\end{bmatrix} + \begin{bmatrix}c_1(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i)\\ c_2(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i)\end{bmatrix} + K\begin{bmatrix}0\\ \delta_i\end{bmatrix} + D\begin{bmatrix}0\\ \dot{\delta}_i\end{bmatrix} = \begin{bmatrix}\tau_i\\ 0\end{bmatrix} \tag{1}$$

The matrices M, c_1, c_2, K, and D in (1) are, respectively, a positive definite symmetric inertia matrix, Coriolis and centrifugal force vectors, stiffness matrix, and damping matrix. The detailed theoretical TLFM model conversion and a comprehensive explanation of matrices of (1) are given in Appendix A.

In state space form, the dynamics of TLFM (1) can be expressed as

$$\begin{aligned}\dot{x}(t) &= f_i(x(t)) + g_i(x(t))u_i(t)\\ y(t) &= l(x(t))\end{aligned} \tag{2}$$

where $x(t) \in \Re^{2n}$ represents the state vector, $y(t) \in \Re^m$ represents the output vector (or tip position), $u(t) \in \Re^n$ denotes the control input, $f_i(x(t)) \in \Re^n$ is the drift dynamics of TLFM, $g_i(x(t)) \in \Re^{n\times m}$ is the input dynamics and $l(x(t))$ is the output dynamics. A comprehensive explanation of matrices of (2) is provided in Appendix A.

Assumption 1 The system (2) has the following properties:

1. $f(.) = 0$, when the variable $x(t)$ is equal to zero;
2. $f(.) + g(.)u_i(t)$ is Lipschitz continuous to all $x(t)$ and (2) is controllable/stabilizable.
3. $\mid f(x(t+T)) \mid - \mid f(x(t)) \mid \leq b_f \mid x(t+T) - x(t) \mid$ where $T = \Delta t$ is the sampling period and b_f is a constant.
4. $\mid g(x(t)) \mid \leq b_g$, i.e., $g(x(t))$ is bounded by a constant b_g.

Lemma 1 *If $f(x(t))$ is Lipschitz and $f(.) = 0$ (Assumption (1)), which is a typical assumption to ensure that the solution $x(t)$ of the system (2) is unique for any finite initial condition, then Assumption (3) in Assumption 1 is satisfied for the system (2). On the other hand, some physical systems do meet this condition even though Assumption (4) is not appropriate for the considered nonlinear system (TLFM).*

2.2 Problem formulation

The aim is to create control input $u(t)$ for a system (2) such that state of the system $x(t)$ shall track a desired trajectory $x_d(t)$ and stabilize the TLFM (by controlling link vibration). The tracking error is described as

$$e(t) = x(t) - x_d(t). \tag{3}$$

The control input $u(t)$ for the robust tip-tracking control (RTTC) problem can be expressed as

$$u(t) = \begin{cases} u_{rl}(t) & \text{if object is out of FOV}\\ u_{sp}(t) & \text{if object is in desirable/safe area}\end{cases} \tag{4}$$

where $u(t)$ denotes the TLFM's behavior policy that has to be modified. To bring the object within the FOV, the RL control input $u_{rl}(t)$ is used to correct the tip position of the TLFM. To accomplish the visual servoing operation, IBVS control input $u_{sp}(t)$ is used.

The formulation of the RTTC problem can be split into two subproblems for TLFM when taking into account the overall dynamics of the system (2).

Problem 1 The control input is intended to correct the position of the TLFM's tip for the system (2) in order to maintain the object's FOV. Consider the following cost function

$$J(e(t), u_{rl}(t)) = \int_{t}^{\infty} e^{-\frac{\tau-t}{\psi}} [e(\tau)^T Q_1 e(\tau) + u^T{}_f(\tau) R_1 u_{rl}(\tau)] \mathrm{d}\tau \tag{5}$$

where $R_1 = R_1^T > 0$ and $Q_1 \geq 0$ are positive-definite function, and $0 < \psi \leq 1$ describes the constants used to discount future costs.

The Hamilton–Jacobi–Bellman (HJB) equation related to (5) can be used to determine the input $u_{rl}(t)$.

$$\frac{\partial J(e(t), u_{rl}(t))}{\partial u_{rl}(t)} = 0 \tag{6}$$

Remark 1 It is not possible to encode input constraints into the optimization problem by employing a non-quadratic performance function since only the feedback part of the control input $u_{rl}(t)$ is acquired by minimizing the cost function (5).

Remark 2 Note that singular perturbation (SP) approach [33] uses the gap between the fast and slow variables to separate overall dynamics into two reduced order system. In [5], presents decomposition of TLFM dynamic model into two-time scale by singular perturbation approach (slow and fast subsystems).

Problem 2 The control input $u_{sp}(t)$ for the system (2) is intended to, (i) ensure perfect tracking and, (ii) account for link vibration (for system stabilization). $u_{sp}(t)$ control input can be written as

$$u_{sp}(t) = u_f(t) + u_s(t) \tag{7}$$

where $u_f(t)$ and $u_s(t)$ are control input for fast and slow subsystem, respectively.

Remark 3 The RTTC problem for the slow subsystem is to realize the tracking performance of $x(t)$ to the desired trajectory $x_d(t)$ with minimum tracking error. The desired trajectory $x_d(t)$ can be achieved if $e(t) \to 0$.

Therefore, a new formulation that provides both control inputs concurrently needs to be created. Due to RL's greater ability to address the RTTC problem without necessitating in-depth understanding of system dynamics, it has been successfully used in a variety of practical applications.

3 Solution to the robust tip-tracking control problem

In this section, two controllers for Problems 1 and 2 are designed. An actor-critic-based off-policy reinforcement learning controller is developed and new two-time scale IBVS controller [5] are utilized to deal with Problems 1 and 2, respectively. The proposed composite controller is termed as adaptive intelligent IBVS (AI-IBVS) controller.

3.1 Reinforcement learning

In RL, action-value methods have three major limitations that cause problems in real-time application and their convergence. First, their target policies are deterministic, where as many problems have stochastic optimal policies. Second, for larger action space, it is very difficult to find the greedy action with respect to action-value function. Third, a small variation in the action-value function results in major deviations in the policy that causes convergence issue for some real-time applications [34].

To overcome the limitations of action-valued methods, actor-critic methods are utilized. The on-policy actor-critic policy gradient algorithm is successfully used for learning in continuous action spaces in many robotics applications [35]. The on-policy actor-critic algorithm does not take advantages of off-policy learning. Off-policy algorithms make it possible to follow and collect data from behavior policy while learning a target policy. However, off-policy actor-critic algorithms are advantageous for real-time applications than action-value methods as well as off-policy actor-critic algorithms, because it presents the policy, as a results the policy can be stochastic and used large action space [34].

The memory structure of actor-critical techniques is independent, allowing them to present the policy without regard to any value function. The actor is called as policy structure, because it is used to update the control policy. The critic is called the estimated value function, because it is used to criticize the actions made by the actor.

In recent years, neural networks (NNs) have been widely employed for the control design of uncertain nonlinear systems since NNs have a good ability to approximate with less system knowledge. This ability of NN helps to cop-up with nonlinearity and uncertainty present in the TLFM. Therefore, NNs are used for approximation in the present work. The proposed RL controller comprises of two NNs: actor NN for generating control input by estimating the uncertain parameter or system information, and critic NN for approximating the cost function. For a continuous function $f(Z) : \mathbb{R}^k \to \mathbb{R}$, following NN is applied

$$f(Z) = WS(Z) \tag{8}$$

where $Z = [Z_1, Z_2, Z_3, \dots, Z_k] \in \Omega\mathbb{R}^k$ is the input vector, $W = [w_1, w_2, w_3, \dots, w_l] \in \Omega\mathbb{R}^l$ is the weight vector with NN node number $l > 1$. $S(Z) = [S_1(Z), S_2(Z), S_3(Z), \dots, S_l(Z)]$ in which $S_i(Z)$ uses Gaussian function. It has been established that NN is capable of estimating any continuous function over a compact set $\Omega_z \subset \mathbb{R}^k$ to any desired precision as

$$f(Z) = \varepsilon_b + W^* S(Z), \quad \forall Z \in \Omega_z \tag{9}$$

where ε_b is the bounded estimation error and W^* is the ideal constant weight.

3.1.1 Off-policy RL algorithm

In order to develop off-policy algorithm, augmented system and value function need to be constructed. To determine tracking error defined in (3), desired trajectory is assumed as

$$\dot{x}_d(t) = h_d(x_d(t)) \tag{10}$$

where $x_d(t) \in \mathbb{R}^n$. Taking into account $e(t)$ (3) and $x_d(t)$ (10), an augmented closed loop system can be constructed as

$$\begin{aligned} \dot{X}(t) = \begin{bmatrix} \dot{e}(t) \\ \dot{x}_d(t) \end{bmatrix} &= \begin{bmatrix} f_i(x_d(t) + e(t)) - h_d(x_d(t)) \\ h_d(x_d(t)) \end{bmatrix} \\ &+ \begin{bmatrix} g_i(e(t) + x_d(t)) \\ 0 \end{bmatrix} \\ &= F_i(X(t)) + G_i(X(t)) u_{rl}(t) \end{aligned} \tag{11}$$

where, the augmented states are

$$X(t) = \begin{bmatrix} e(t) \\ x_d(t) \end{bmatrix} \tag{12}$$

The value function in terms of the states of the augmented system thus produces

$$\begin{aligned} V(X(t)) &= \int_t^{\infty} e^{-\frac{\tau - t}{\psi}} r(X(t), u_{rl}(t)) \\ &= (X^T(\tau) Q_T X(\tau)) + u_{rl}(\tau) R_T u_{rl}^T(\tau)) \end{aligned} \tag{13}$$

where $Q_T \geq 0$ and $R_T \geq 0$ are positive-definite function.

The augmented system dynamics (11) is expressed as the off-policy RL algorithm.

$$\begin{aligned} \dot{X}(t) = F_i(X(t)) &+ G_i(X(t)) u_{rl}(t) \\ &+ G_i(X(t))(u_{rl}(t) + u_j(t)) \end{aligned} \tag{14}$$

where $u_j(t)$ denotes the policy that needs to be updated. In contrast, the behavior policy $u_{rl}(t)$ is the one that is actually applied to the dynamics of the system to produce the data for learning.

Differentiating value function along with the dynamics (14) and using $u_{j+1}(t) = -0.5 R_T^T G^T(x) \left(\frac{\partial V_j(X(t))}{\partial X(t)} \right)$

$$\begin{aligned} V_j &= \left(\frac{\partial V_j(X(t))}{\partial X(t)} \right)^T \big(F_i + G_i u_j(t) \big) \\ &+ \left(\frac{\partial V_j(X(t))}{\partial X(t)} \right) G_i(u_{rl}(t) - u_j(t)) \\ &= -Q_T(X) - u_j^T R_T u_j - 2u_{j+1}^T R_T (u_{rl}(t) - u_j(t)) \end{aligned} \tag{15}$$

Integrating both sides of (15) yields the off-policy RL Bellman equation

$$\begin{aligned} & e^{-\frac{\tau}{\psi}} V_j(X(t+T)) - V_{j+1}(X(t)) \\ &= \int_t^{t+T} e^{-\frac{\tau}{\psi}} \Big((Q_T(X(t)) - u_j^T R_T u_j \\ & -2u_{j+1}^T R_T (u_{rl}(t) - u_j(t)) \Big) d\tau \end{aligned} \tag{16}$$

Equation (16) is also known as off-policy Bellman equation, that yields the following off-policy RL algorithm.

Algorithm 1 Off-Policy RL Algorithm to Find the Solution of HJB

```
1: procedure
2: Given admissible policy $u_0$
3: for $j = 0, 1, 2...$ given $u_j$, solve for the value $V_j$ and $u_{j+1}$ using
   off-policy Bellman equation
       $e^{-\frac{\tau}{\psi}} V_j(X(t+T)) - V_{j+1}(X(t)) =$
           $\int_t^{t+T} e^{-\frac{\tau}{\psi}} ((Q_T(X(t))$
       $-u_j^T R_T u_j - 2u_{j+1}^T R_T (u_{rl}(t) - u_j(t))\Big) d\tau$
   on convergence, set $V_{j+1} = V_j$,
4: Go to 3.
5: end procedure
```

The design of actor-critic structure is utilized to approximately the various function and control policy in order to build off-policy RL Algorithm 1. Design of actor-critic structure is given in Sect. 3.2.

3.2 Design of actor-critic-based off-policy reinforcement learning controller

Problem 1 is resolved by developing an actor-critic-based off-policy reinforcement learning controller. The structure of off-policy RL controller is depicted in Fig. 1.

In Fig. 1, actor is used to update the desired control policy to minimize the cost function, critic is used to approximate

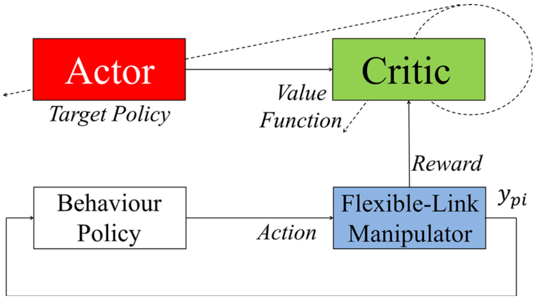


Fig. 1 Off-policy RL controller for adaptive tip-tracking control of TLFM

the reward function/current state information and cost function, behavior policy is used to select/generate the action data/control input while learning about target policy for TLFM. The estimated/target policy is unrelated to policy that is evaluated and improved.

3.2.1 Design of critic NN

As the cost function (5) describes, the approximate error of cost function can be expressed as

$$\gamma(t) = \dot{\hat{J}}(e(t), u(t)) - \tfrac{1}{\psi}\hat{J}(e(t), u(t)) + \phi(t) \tag{17}$$

where $\phi(t)$ represent the instant cost function. As the constant $\psi \rightarrow \infty$, the approximate error of the cost function can be represented as

$$\begin{aligned}\gamma(t) &= \dot{\hat{J}}(e(t), u(t)) + \phi(t)\\ &= \nabla\dot{\hat{J}}(e(t), u(t))\dot{Z}_c + \phi(t)\end{aligned} \tag{18}$$

where $Z_c = y(t) = z_1 = x(t) - x_d(t) = e(t)$ and ∇ is the gradient of Z_c. Equation (18) is also known as Bellman equation.

Critic weight (W_c) *update* Critic weight update law can be designed as

$$\dot{\hat{W}}_c = -l_c\tfrac{\partial E_c}{\partial W_c} \tag{19}$$

where E_c is the square Bellman error [17], i.e., defined as

$$E_c = \tfrac{1}{2}\gamma^T(t)\gamma(t) \tag{20}$$

Substituting (20) in (19), one obtains

$$\begin{aligned}\dot{\hat{W}}_c &= -l_c\gamma(t)\frac{\partial\gamma(t)}{\partial W_c}\\ &= -l_c\gamma(t)\frac{\partial[\dot{\hat{J}}(e(t), u(t)) - \frac{1}{\psi}\hat{J}(e(t), u(t)) + \phi(t)]}{\partial W_c}\\ &= -l_c\gamma(t)\left[-\frac{1}{\psi}\frac{\partial\hat{J}}{\partial W_c} + \frac{\partial}{\partial W_c}\left(\frac{\partial\hat{J}}{\partial Z_c}\right)\right]\\ &= -l_c(\phi(t) + W_c^T\wedge)\wedge\end{aligned} \tag{21}$$

where $l_c > 0$, which represents the learning rate of critic NN and $\wedge = -(S_c/\psi) + \nabla S_c\dot{Z}_c$.

3.2.2 Design of actor NN

The dynamics of TLFM (1) can be rewritten as

$$M_{11}\ddot{\theta} + M_{12}\ddot{\delta} + c_{11}\dot{\theta} + c_{12}\dot{\delta} = \tau \tag{22}$$

$$M_{21}\ddot{\theta} + M_{22}\ddot{\delta} + c_{21}\dot{\theta} + c_{22}\dot{\delta} + K\delta + D\dot{\delta} = 0 \tag{23}$$

From (23), one obtains

$$\ddot{\delta} = -M_{22}^{-1}[M_{21}\ddot{\theta} + c_{21}\dot{\theta} + c_{22}\dot{\delta} + K\delta + D\dot{\delta}] \tag{24}$$

Substituting (24) into (22) gives

$$\begin{aligned}&(M_{11} - M_{12}M_{22}^{-1}M_{21})\ddot{\theta} + (c_{11} - M_{12}M_{22}^{-1}c_{21})\dot{\theta}\\ &\quad + (c_{12} - M_{12}M_{22}^{-1}c_{22} - M_{12}M_{22}^{-1}D)\dot{\delta}\\ &\quad - M_{12}M_{22}^{-1}K\delta = \tau\end{aligned} \tag{25}$$

Equation (25) can be expressed as

$$P\ddot{\theta} + Q\dot{\theta} + S = \tau \tag{26}$$

The dynamic of TLFM (26) can be rewritten by considering $x_1(t) = \theta$, and $x_2(t) = \dot{\theta}$ as

$$\begin{cases}\dot{x}_1(t) = x_2(t)\\ \dot{x}_2(t) = P^{-1}(\tau - (Qx_1(t) + S)) = P^{-1}\tau + x_3(t)\end{cases} \tag{27}$$

where $x_3(t) = -P^{-1}(Qx_1(t) + S)$.

To achieve the control objective, the tracking error variables $e_1(t)$ and $e_2(t)$ are defined as

$$\begin{aligned}e_1(t) &= x_1(t) - x_{1d}(t)\\ e_2(t) &= x_2(t) - \alpha_1(t)\end{aligned} \tag{28}$$

where $x_{1d}(t)$ is the control input and $\alpha_1(t)$ is a virtual backstepping control variable to $e_1(t)$.

Using (27), derivative of (28) can be written as

$$\begin{aligned}\dot{e}_1(t) &= e_2(t) + \alpha_1(t) - \dot{x}_{1d}(t)\\ \dot{e}_2(t) &= P^{-1}\tau + x_3(t) - \dot{\alpha}_1(t).\end{aligned} \tag{29}$$

Virtual control variable is selected as $\alpha_1(t) = \dot{x}_{1d}(t) - k_1 e_1(t)$, where $k_1 > 0$ is the constant design parameter. From (29), $\dot{e}_1(t)$ can be presented as

$$\dot{e}_1(t) = e_2(t) - k_1 e_1(t). \tag{30}$$

Define a candidate Lyapunov function $V_1 = \frac{1}{2}e_1^2(t)$. Its time-related derivative can be expressed as

$$\begin{aligned}\dot{V}_1 &= e_1(t)\dot{e}_1(t) = [e_2(t) - k_1 e_1(t)]e_1(t)\\ &= -k_1 e_1^2(t) + e_2(t)e_1(t).\end{aligned} \tag{31}$$

To realize $e_2(t) \to 0$, we define candidate Lyapunov function $V_2 = V_1 + \frac{1}{2}e_2^2(t)$. Its derivative with respect to time can be written as

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + e_2(t)\dot{e}_2(t)\\ &= -k_1 e_1^2(t) + e_2(t)[e_1(t) + P^{-1}\tau + x_3(t) - \dot{\alpha}_1(t)].\end{aligned} \tag{32}$$

To realize $\dot{V}_2 < 0$, we choose

$$e_1(t) + P^{-1}\tau + x_3(t) - \dot{\alpha}_1(t) = -k_2 e_2(t) \tag{33}$$

where $k_2 > 0$ is the constant design parameter. Then (32) can be expressed as

$$\dot{V}_2 = -k_1 e_1^2(t) - k_2 e_2^2(t) \tag{34}$$

From (33), the desired control law can be designed as

$$u_{rl}(t) = P[\dot{\alpha}_1(t) - k_2 e_2(t) - e_1(t) - x_3(t)] \tag{35}$$

However, to realize the control law (35), modeling information $x_3(t)$ are needed, which are difficult in practical engineering. In order to estimate the unknown information, actor NN must be introduced.

So, control law $u_{rl}(t)$ can be redefined as

$$u_{rl}(t) = P[\dot{\alpha}_1(t) - k_2 e_2(t) - e_1(t) - \hat{W}_a^T S_a(Z_a)] \tag{36}$$

where $\hat{W}_a = W_a^* + \tilde{W}_a$ is the neural weight estimation and $Z_a = [x_1(t), x_2(t), x_{1d}(t), \dot{x}_{1d}(t)]^T$. W_a^* and $\tilde{W}_a$ are the ideal and instant neural weights, respectively.

The instant estimation error is expressed as

$$\varepsilon_a = \tilde{W}_a^T S_a(Z_a) \tag{37}$$

Then, the actor NN error e_a can be designed as

$$e_a(t) = \varepsilon_a + \kappa_I[\hat{J}(e(t), u(t)) - J_d(t)] \tag{38}$$

where κ_I is a positive constant and $J_d(t) \in \Re^{N+1}$ is the desired cost.

Actor weight (W_a) *update* Actor weight update law can be designed as

$$\dot{\hat{W}}_a = -l_a \tfrac{\partial E_a}{\partial W_a} \tag{39}$$

where $E_a = \frac{1}{2}{e_a}^T(t)e_a(t)$.

Substituting (38) in (39), we get

$$\begin{aligned}\dot{\hat{W}}_a &= -l_a \frac{\partial E_a}{\partial e_a}\frac{\partial e_a}{\partial \varepsilon_a}\frac{\partial \varepsilon_a}{\partial W_a}\\ &= -l_a(\varepsilon_a + \kappa_I \hat{J}(e(t), u(t)))S_a\end{aligned} \tag{40}$$

where l_a is the actor NN's learning rate. As ε_a is unavailable, we can redefine update law as

$$\dot{\hat{W}}_a = -l_a(\hat{W}_a^T S_a(Z_a) + \kappa_I \hat{J}(e(t), u(t)))S_a \tag{41}$$

3.2.3 Stability analysis

Define a candidate Lyapunov function V_c as

$$V_c = \frac{1}{2}\tilde{W}_c^T \tilde{W}_c \tag{42}$$

Taking the time derivative of (42), and substitute (21) into (42), we have

$$\begin{aligned}\dot{V}_c &= \tilde{W}_c^T \dot{\tilde{W}}_c = \tilde{W}_c^T \dot{\hat{W}}_c\\ &= -l_c \tilde{W}_c^T(\phi(t) + W_c^T \Lambda)\Lambda\end{aligned} \tag{43}$$

As $\gamma(t) \to 0$, Eq. (18) will become

$$\phi(t) = -\nabla \dot{\hat{J}}(e(t), u(t))\dot{Z}_c = -\nabla \dot{\hat{J}}\dot{e}(t) \tag{44}$$

Substituting $\phi(t)$ from (44) to (43), one obtains

$$\begin{aligned}\dot{V}_c &= -l_c \tilde{W}_c^T(-\nabla \dot{\hat{J}}\dot{e}(t) + W_c^T \Lambda)\Lambda\\ &\le l_c \tilde{W}_c^T \nabla \dot{\hat{J}}\dot{e}(t)\Lambda - l_c \tilde{W}_c^T W_c^T \Lambda^T \Lambda\end{aligned} \tag{45}$$

This means that when tracking error $e(t)$ will be zero, $\dot{V}_c$ will be negative definite, i.e., $\dot{V}_c \le 0$ that will ensure the stability.

The following lemma can be used to demonstrate the closed loop system's boundedness.

Lemma 2 [16] *Candidate Lyapunov function $V_r(t)$ is bounded if the initial condition $V_r(0)$ is bounded, $V_r(0) \geq 0$ is continuous and the following equation satisfies*

$$\dot{V}_r(t) \leq -\kappa V_r(t) + \lambda \tag{46}$$

where λ and κ are both positive constant.

Define a candidate Lyapunov function as

$$V_r = \frac{1}{2}e_1^T e_1 + \frac{1}{2}e_2^T P e_2 + \frac{1}{2}\tilde{W}_c^T \tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \tilde{W}_a \tag{47}$$

Its time-derivative can be expressed as

$$\dot{V}_r = e_1^T \dot{e}_1 + e_2^T P \dot{e}_2 + \tilde{W}_c^T \dot{\tilde{W}}_c + \tilde{W}_a^T \dot{\tilde{W}}_a \tag{48}$$

Substituting (41) into (48), one obtains

$$\begin{aligned}\dot{V}_r &= -e_1^T k_1 e_1 - e_2^T k_2 e_2 + e_2^T \left(\tilde{W}_a^T S_a - \varepsilon_a\right) \\ &\quad - l_c \tilde{W}_c^T (-W_c^T \Lambda + \varepsilon_c)\Lambda \\ &\quad - l_a \tilde{W}_a^T S_a \left(\tilde{W}_a^T S(Z_a) + \kappa_I \hat{J}(e(t), u(t))\right)\end{aligned} \tag{49}$$

As $\hat{J}(e(t), u(t)) = W_c^T S_c(Z_c) + \tilde{W}_c^T S_c(Z_c)$, one obtains

$$\begin{aligned}&\hat{J}(e(t), u(t))^T \hat{J}(e(t), u(t)) \\ &\quad \leq 2(W_c^T S_c)^T W_c^T S_c + 2(\tilde{W}_c^T S_c)^T \tilde{W}_c^T S_c\end{aligned} \tag{50}$$

Substituting (50) into (49), one obtains

$$\dot{V}_r \leq -\kappa V_r + B_r \tag{51}$$

where,

$$\begin{aligned}\kappa = \min\Bigg(&\lambda_{\min}(k_1), \frac{l_a - 1}{2} b_s^2, \\ &\lambda_{\min}(k_2 - I), \frac{l_c b_\Lambda^2 - 2 l_a \kappa_I^2 \|S_c\|^2}{2}\Bigg)\end{aligned} \tag{52}$$

$$\begin{aligned}B_r &= \frac{l_a}{2}\|W_a\|^2 \|S_a\|^2 + l_a \kappa_I^2 \|S_c\|^2 \|W_c\|^2 \\ &\quad + \frac{1}{2}\|\varepsilon_a\|^2 + \frac{1}{2}\left\|\varepsilon_{c,\max}\right\|^2\end{aligned} \tag{53}$$

where I represents an identity matrix, B_r is positive constant, $b_\Lambda \leq \|\Lambda\|$ and $b_s \leq \|S_a\|$. Further following condition must satisfy to ensure $\kappa > 0$.

$$\begin{aligned}&\lambda_{\min}(k_1) > 0,\ \lambda_{\min}(k_2 - I) > 0, \\ &\frac{l_a - 1}{2} > 0,\ \frac{l_c b_\Lambda^2 - 2 l_a \kappa_I^2 \|S_c\|^2}{2} > 0\end{aligned} \tag{54}$$

As per Lemma 2, $V_r(t)$ is bounded. Now, by using the subsequent theorem, the RL controller's boundedness is established.

Theorem 1 *Consider the TLFM, with the proposed RL controller, the system parameters $e_1(t)$, $e_2(t)$, $\tilde{W}_c$ and $\tilde{W}_a$ are bounded, since the initial conditions are bounded. Also, the parameters $e_1(t)$, $e_2(t)$, $\tilde{W}_c$ and $\tilde{W}_a$ will eventually remain within the compact set Ω_{e_1}, Ω_{e_2}, $\Omega_{\tilde{W}_c}$ and $\Omega_{\tilde{W}_a}$, respectively, which are defined as*

$$\begin{aligned}\Omega_{e_1} &= \left\{e_1 \in \mathbb{R}^{N+1} \mid \|e_1\| \sqrt{2V_r(0) + B_r/\kappa}\right\} \\ \Omega_{e_2} &= \left\{e_2 \in \mathbb{R}^{N+1} \mid \|e_2\| \sqrt{\frac{2V_r(0) + B_r/\kappa}{\lambda_{\min}(P)}}\right\} \\ \Omega_{\tilde{W}_c} &= \left\{\tilde{W}_c \in \mathbb{R}^{N+1} \mid \left\|\tilde{W}_c\right\| \sqrt{2V_r(0) + B_r/\kappa}\right\} \\ \Omega_{\tilde{W}_a} &= \left\{\tilde{W}_a \in \mathbb{R}^{N+1} \mid \left\|\tilde{W}_a\right\| \sqrt{2V_r(0) + B_r/\kappa}\right\}\end{aligned} \tag{55}$$

Proof In (51), multiply $e^{\kappa t}$ yields

$$\frac{d(V_r e^{\kappa t})}{dt} \leq B_r e^{\kappa t} \tag{56}$$

From (56), one obtains

$$V_r \leq (V_r(0) - B_r/\kappa)\, e^{-\kappa t} + B_r/\kappa \leq V_r(0) + B_r/\kappa \tag{57}$$

From (47) and (57), it can be observed that

$$\begin{aligned}&e_1^T e_1 \leq 2(V_r(0) + B_r/\kappa) \\ &e_2^T P e_2 \leq 2(V_r(0) + B_r/\kappa) \\ &\tilde{W}_c^T \tilde{W}_c \leq 2(V_r(0) + B_r/\kappa) \\ &\tilde{W}_a^T \tilde{W}_a \leq 2(V_r(0) + B_r/\kappa)\end{aligned} \tag{58}$$

Then, one can obtain

$$\begin{aligned}&\frac{1}{2}\|e_1\|^2 \leq (V_r(0) + B_r/\kappa) \\ &\frac{1}{2}\|e_2\|^2 \leq \frac{(V_r(0) + B_r/\kappa)}{\lambda_{\min}(P)} \\ &\frac{1}{2}\left\|\tilde{W}_c\right\|^2 \leq (V_r(0) + B_r/\kappa) \\ &\frac{1}{2}\left\|\tilde{W}_a\right\|^2 \leq (V_r(0) + B_r/\kappa)\end{aligned} \tag{59}$$

3.3 Design of new two-time scale IBVS controller

A new two-time scale IBVS control scheme [5] is utilized in order to address Problem 2. The goal of the new two-time scale IBVS control scheme is to ensure tracking and stabilize the system in order to fulfil the visual servoing task (to damp out the vibration).

3.3.1 Model decomposition by two-time scale perturbation method

According to the SP technique, the design of a feedback control system for an under-actuated system can be divided into two subsystems: a fast subsystem for compensating tip deflection/vibration and a slow subsystem for measuring and controlling tip position. The state variable of the TLFM dynamic model (1) can be expressed using SP theory as

$$\begin{aligned} x_1 &= \theta_i = \bar{x}_1 + O(\varepsilon_s) \\ x_2 &= \dot{\theta}_i = \bar{x}_2 + O(\varepsilon_s) \\ z_1 &= K\delta_i = \bar{z}_1 + \eta_1 + O(\varepsilon_s) \\ z_2 &= \varepsilon_s K\dot{\delta}_i = \bar{z}_2 + \eta_2 + O(\varepsilon_s) \end{aligned} \tag{60}$$

where $\varepsilon_s = \frac{1}{\sqrt{k}}$ is the SP parameter with the common stiffness coefficient scale factor, and the overbars indicate the slow part of each variable. The fast parts of the variables z_1 and z_2 are η_1 and η_2, respectively.

The slow subsystem is described as

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{x}_2 \\ \dot{\bar{x}}_2 &= M_{rr}^{-1}(\bar{x}_1,\ 0)[-c_1(\bar{x}_1,\ \bar{x}_2) + \bar{u}_s] \end{aligned} \tag{61}$$

The fast subsystem can be expressed as

$$\begin{aligned} \bar{z}_1 &= -\hat{H}_{ff}^{-1}(\bar{x}_1,\ 0)\hat{H}_{rf}(\bar{x}_1,\ 0)[c_1(\bar{x}_1,\ \bar{x}_2) - \bar{u}_f] \\ &\quad - c_2(\bar{x}_1,\ \bar{x}_2) \\ \bar{z}_2 &= 0 \end{aligned} \tag{62}$$

In terms of η_1 and η_2, the fast subsystem can be defined as

$$\begin{aligned} \frac{d\eta_1}{dT} &= \eta_2 \\ \frac{d\eta_1}{dT} &= \hat{H}_{rf}(\bar{x}_1, 0)(u_{sp} - \bar{u}_{sp}) - \hat{H}_{ff}^{-1}(\bar{x}_1, 0)\eta_1 \end{aligned} \tag{63}$$

where $H = M^{-1}$, $T = \frac{t}{\varepsilon_s}$ is the fast time scale, u_f and u_s are the fast and slow control signal, respectively.

With respect to (61) and (63), the slow and fast components of the tip position variables and the deflection variables change, respectively. Consequently, using the composite control theory, the TLFM's control input can be written as

$$u = u_f(\bar{x}_1, \eta_1, \eta_2) + \bar{u}_s(\bar{x}_1, \bar{x}_2) \tag{64}$$

where $\bar{u}_f$ and u_s are the fast and slow control inputs, respectively. $u_f(\bar{x}_1, 0, 0) = 0$, i.e., fast control signal is not needed during trajectory tracking with slow subsystem (61).

3.3.2 Slow subsystem controller

Shifted moment-based IBVS is used to create the $u_s(t)$ for the slow subsystem. Two moment-based visual features are required to control the 2-DOF of TLFM, according to [36]. To adjust the 2-DOF of the TLFM and decrease the sensitivity of the data noise, a low order shifted moment-based visual feature is applied. These are three polynomials that were calculated using shifted moments. Here are the polynomials of orders 2 and 3 that were constructed from shifted moments [37].

$$\begin{aligned} I_{s1} &= \mu_{20}^s\mu_{02}^s - \mu_{11}^s\mu_{11}^s; \\ I_{s2} &= -\mu_{30}^s\mu_{12}^s + \mu_{21}^s\mu_{21}^s - \mu_{03}^s\mu_{21}^s + \mu_{12}^s\mu_{12}^s; \\ I_{s3} &= 3\mu_{30}^s\mu_{12}^s + \mu_{30}^s\mu_{30}^s + 3\mu_{03}^s\mu_{21}^s + \mu_{03}^s\mu_{03}^s \end{aligned} \tag{65}$$

Features that are invariant to scaling, rotation, and translation include

$$\begin{aligned} & r_{s1} = \frac{I_{s2}}{I_{s1}^{8/10}};\ r_{s2} = \frac{I_{s3}}{I_{s1}^{8/10}};\ r_{s3} = \frac{I_{s3}}{I_{s2}}; \\ & r_{s4} = \frac{I_{s3}}{m_{00}^5};\ r_{s5} = \frac{I_{s2}}{m_{00}^5};\ r_{s6} = \frac{I_{s1}}{m_{00}^4}. \end{aligned} \tag{66}$$

By integrating three different types of moment invariants (invariant to translation, to the 2D rotation and to scale), two visual features with shifted moments are chosen from two invariants from (65) and (66). The L_θ^s interaction matrix for the two shifted moment-based visual features that regulate the 2-DOF of the TLFM can be represented as

$$L_{\mu_{ij}^s} = [\, L_{\theta_1}^s\ L_{\theta_2}^s \,] \tag{67}$$

where,

$$\begin{aligned} L_{\theta_1}^s &= (i + j + 3)\mu_{i,j+1}^s \\ &\quad + (i + 2j + 3)y_o\mu_{ij}^s + jx_o\mu_{i-1,j+1}^s \\ L_{\theta_2}^s &= -(i + j + 3)\mu_{i,j+1}^s \\ &\quad - (2i + j + 3)x_o\mu_{ij}^s - qy_o\mu_{i+1,j-1}^s \end{aligned} \tag{68}$$

From a binary or a segmented image, the analytical form of the interaction matrix corresponding to every moment can be calculated.

The purpose of a shifted moment-based IBVS controller is to ensure that the real visual feature approaches the desired visual feature asymptotically. For the slow subsystem, the control input is designed for guaranteed accurate/perfect tracking. It is designed using IBVS approach.

$$u_s(t) = -kL_s^{-1}[\dot{x}_d(t) - f(x_d(t))] \tag{69}$$

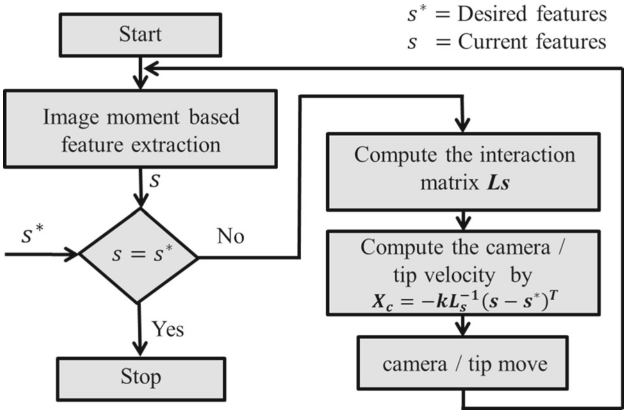

s^* = Desired features
s = Current features
Start
Image moment based feature extraction
s
s^*
$s = s^*$
No
Yes
Stop
Compute the interaction matrix $\boldsymbol{Ls}$
Compute the camera / tip velocity by $\boldsymbol{X_c = -kL_s^{-1}(s - s^*)^T}$
camera / tip move

Fig. 2 IBVS flow control algorithm of TLFM

Equation (69) can be derived in the similar fashion as adopted in [5]. In (69), $L_s = L_{\mu^s_{ij}}$ is the interaction matrices related to shifted moment (67) of the tip with respect to the position variables [5].

The interaction matrices for the shifted tip moment (67) with regard to the position variables are represented by $L_s = L_{\mu^s_{ij}}$ in Eq. (69) [5].

To achieve the objective of shifted moment-based IBVS controller, the formulation of problem is described in the following steps:

1. Initially, pre-processed captured image-based features based on shifted moments are extracted.
2. Interaction matrix is estimated from features, which are extracted from shifted moments in previous step.
3. Camera/tip velocity or acceleration for robot controller to be calculated from estimated interaction matrix related to visual features.
4. Then camera/tip is move to reach desired position unless and until error of image features is minimized. When the features align with the desired ones, the visual servoing work is finished.

Figure 2 shows the IBVS flow control algorithm, in which s^* is the desired image features and s is the current value of image features.

For a closed-loop system (61), it is necessary to construct an IBVS-based shifted moment control strategy so that the output trajectory closely tracks the reference output trajectory. As stated in [38], slow control input is planned as

$$\bar{u}_s(\bar{x}_1, \bar{x}_2) = c_1(\bar{x}_1, \bar{x}_2) + M_{rr}(\bar{x}_1)v \tag{70}$$

3.3.3 Fast subsystem controller

Here, the fast subsystem of the TLFM is controlled by the LQR controller. A state observer is typically required in fast controllers to estimate the immeasurable modal coordinates. The best option for closed-loop system stability and robustness against time delay is a Kalman filter based on a fast model that contains the first three modes and a fast feedback that dampens the first mode only [38].

For the fast subsystem, consider the following cost function

$$J = \int_0^\infty x^T (Q_2 + K^T R_2 K) x \mathrm{d}t \tag{71}$$

where Q_2 and R_2 are positive definite symmetric matrices, $K_f = [K_1, K_2]$ is the feedback gain. After minimizing the cost function (71), the fast subsystem control input is represented by

$$u_f(t) = -R^{-1} B^T P x(t) \tag{72}$$

Equation (72) can be derived in the similar fashion as adopted in [5].

The new two-time scale IBVS control law $u_{sp}(t)$ is derived from (70) and (72) to solve Problem 2.

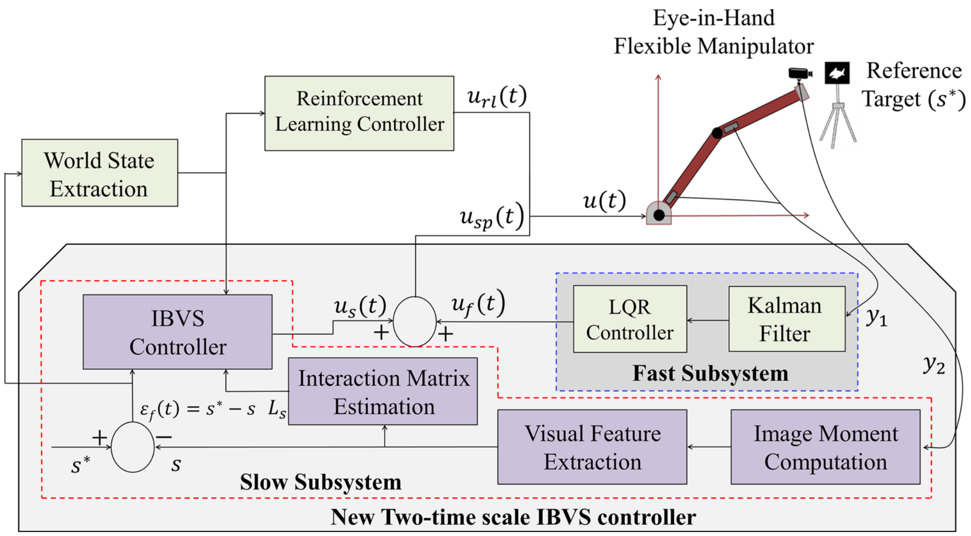
$$u_{sp}(t) = c_1(\bar{x}_1,\ \bar{x}_2) + \tau_f(\bar{x}_1,\ \eta_1,\ \eta_2) + M_{rr}(\bar{x}_1)v \tag{73}$$

4 Proposed adaptive intelligent IBVS controller for TLFM

The new two-time scale IBVS controller presented in Sect. 3.3 is a summary of work presented in [5] has the following practical problem: (1) the proposed controller cannot guarantee the retention of visual features within the camera FOV, (2) increased input torque results from increased controller gain, which causes the visual feature to move out of the FOV more quickly, resulting in system instability and inaccurate system performance. In this section, the design of a novel adaptive intelligent IBVS (AI-IBVS) Controller for robust tip-tracking control of TLFM is presented in order to address the visibility issue of the proposed new two-time scale IBVS controller.

The proposed AI-IBVS controller design is depicted in Fig. 3; it is discussed in Sect. 3. To increase the reliability of vision-based tip-tracking control of the TLFM, RL-based adaptive intelligent IBVS controller is built. The position of the tip is corrected by the proposed RL controller (36) and the new two-time scale IBVS controller (73). The proposed RL controller brings the visual feature on the FOV by choosing the best control input, while the new two-time scale IBVS controller moves the tip of the TLFM in the direction of the reference target.

Fig. 3 Proposed adaptive intelligent IBVS control scheme



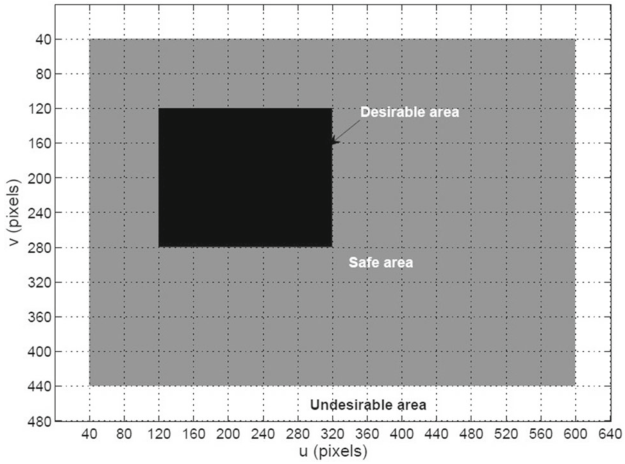


Fig. 4 FOV for visual feature

In particular, the controller will employ the AI-IBVS controller to learn and choose the best control input $u(t)$ for the robot under the current state. The TLFM's RL controller will receive the optimal control input to direct the visual features into a desirable or safe region of the image plane. The reward is used to update the actor-critic weight of the action under the world state after the TLFM takes action. The reward is computed based on the updated position of the visual features on the image plane.

The image plane in Fig. 4 is arranged as a discrete grid with 40 pixels per cell that is 16 × 12. It is divided into three areas: desirable, safe, and undesirable. If the image features is present in the desired/safe region, a new two-time scale IBVS controller is employed. If not, an RL controller is employed. As a result, the proposed AI-IBVS controller ensures the presence of visual features inside the FOV.

When a vision sensor captures an image, it is simple to translate the location of the visual features on the image plane into coordinates in the grid world using the formulation below:

$$X = round(r/40); \quad Y = round(c/40). \tag{74}$$

where $r = 0, 1, \ldots, 639$ and $c = 0, 1, \ldots, 479$ are the pixel coordinates of the visual feature point on the image plane, $X = 0, 1, \ldots, 15$ and $Y = 0, 1, \ldots 11$ are the corresponding coordinates in the grid world.

For each state, the RL controller is only expected to take two actions. The default value of w_x or w_y is 2 degrees per second for the tip/camera rotational velocity. Therefore, one of these actions is used in each stage or iteration depending on the location of the visual feature in the image.

Algorithm 2 AI-IBVS Algorithm

```
Moment-based feature computation.
Compute current feature coordinate $(X, Y)$ in the grid from (74)
if $(X, Y) \in$ the undesirable/out of FOV area then
    repeat
        tip move $w_x$ or $w_y$ in each state
        reward value computation from (75)
        update critic weights (21)
        update actor weights (41)
        generate action data (control input)
          $u_{rl}(t)$ is computed from (36)
    until $(X, Y) \in$ in the desirable/safe area
else if $(X, Y) \in$ in the desirable/safe area then
    repeat
        interaction matrix estimation
        estimation of error vector
        $u_{sp}(t)$ is computed from (73)
    until visual servoing task is achieved.
end if
```

The environment will reward the TLFM after it takes an action. Based on the placement of visual features, the reward

Table 1 Physical parameter values of TLFM

Parameter	Link-1	Link-2
Mass of link	$m_1 = 0.15268$ kg	$m_2 = 0.0535$ kg
Link length	$l_1 = 0.201$ m	$l_2 = 0.2$ m
Armature resistance	$R_{m1} = 11.5\ \Omega$	$R_{m2} = 2.32\ \Omega$
Armature inductance	$L_{m1} = 3.16$ mH	$L_{m2} = 0.24$ mH
Gear ratio	$K_{g1} = 100$	$K_{g2} = 50$
Torque constant	$K_{t1} = 0.0119$ Nm/A	$K_{t2} = 0.0234$ Nm/A
Back-EMF constant	$K_{m1} = 0.119$ V s/rad	$K_{m2} = 0.0234$ V s/rad
Torsional stiffness	$K_{s1} = 22$ Nm/rad	$K_{s2} = 2.5$ Nm/rad
Young's modulus	$E_1 = 2.0684 \times 10^{11}$ N/m^2	$E_2 = 2.0684 \times 10^{11}$ N/m^2
Rotor MI*	$Jm_1 = 6.28 \times 10^{-6}$ kg m^2	$Jm_2 = 1.03 \times 10^{-6}$ kg m^2
Drive MI*	$J_1 = 7.361 \times 10^{-4}$ kg m^2	$J_2 = 44.55 \times 10^{-6}$ kg m^2
Link MI*	$I_1 = 0.17043$ kg m^2	$I_2 = 0.0064387$ kg m^2

*MI, moment of inertia

value is calculated using the relation shown below.

$$reward = \begin{cases} +100, if(X, Y) \in \text{the desirable area} \\ -40, if(X, Y) \in \text{is out of FOV} \\ -20, if(X, Y) \in \text{the undesirable area} \\ 0,\ \ if(X, Y) \in \text{the safe area} \end{cases} \tag{75}$$

where (X, Y) is the new coordinate of the grid world in the image plane, after the TLFM takes action.

It is obvious from (75) that the reinforcement signal rewards actions that keep visual features inside the FOV by forcing them into the desirable part of the image plane and punishes them when they are in the undesirable area. To accomplish the TLFM's vision-based tip positioning task, the AI-IBVS Algorithm 2 is used.

5 Results and discussion

In this section, performance of proposed AI-IBVS controller is analyzed by simulation studies. The proposed controller is evaluated using machine vision toolbox for MATLAB [39]. The physical TLFM parameters taken into account for simulation studies are listed in Table 1. Tasks-1 and task-2 in this study are referred to as tip positioning with symmetrical and non-symmetrical objects, respectively.

5.1 Training procedure

The critic NN and actor NN are set as fully connected NNs with a hidden layer, an input layer, and an output layer in the actor-critic-based off-policy RL controller. Given that the size of the feature column is five, the input layer has six neurons. Two neurons in the output layer correspond to each state's two RL controller actions. There are six neurons in the hidden layer. The learning rates, i.e., l_c of critic NN is set as 0.6 and l_a of actor NN is set as 0.9.

Six activation functions are present in the hidden layer and two activation functions are present in the output layer for the actor and critic NNs. The actor and critic NN is utilized, which employs the backpropagation algorithm, a hyperbolic tangent (nonlinear) activation function for the hidden layer, and a liner activation function for the output layer. The hyperbolic tangent activation function is differentiable; therefore, it can be easily employed in backpropagation (derivative-based) learning algorithm. The output of actor and critic network is RL control input $u_{rl}(t)$ for TLFM. The RL control input of hub-2 for task-1 and task-2 are shown in Figs. 5 and 6, respectively.

5.2 Tip-tracking performance

The effectiveness of the proposed controller is evaluated for two distinct object shapes: the symmetrical object (rectangle) and the non-symmetrical object (whale). The object in the initial position of the visual servoing task is not in the FOV. In this work, the TLFM uses the AI-IBVS controller to perform the tip-tracking task for both objects with small undesirable areas. The undesirable region is described as

$$80\ >\ r >\ 560\quad \text{or}\quad 80\ > c\ > 400 \tag{76}$$

Figure 7 depicts the unwanted area, which is the outer part of the white bounding box; the remaining space is thought to be safe and desirable.

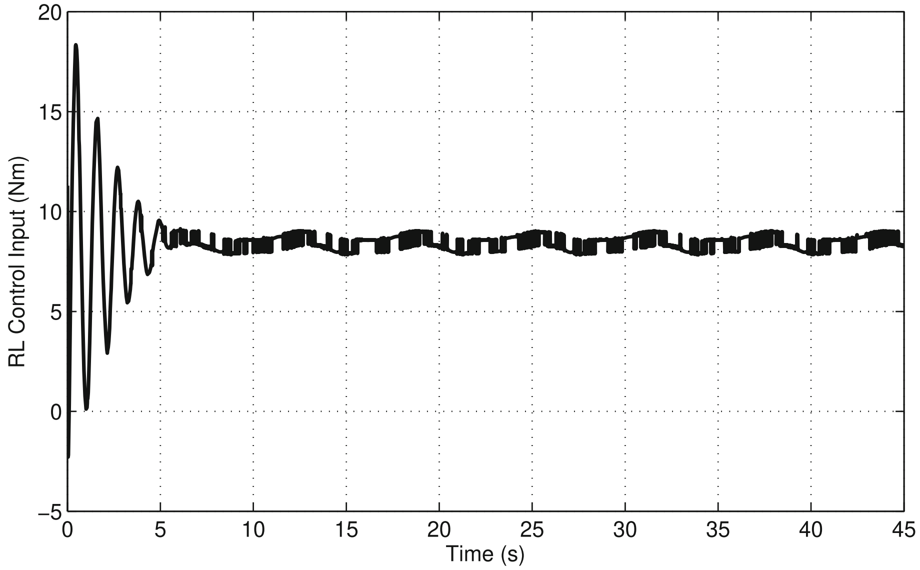


Fig. 5 RL control input of hub-2 for task-1

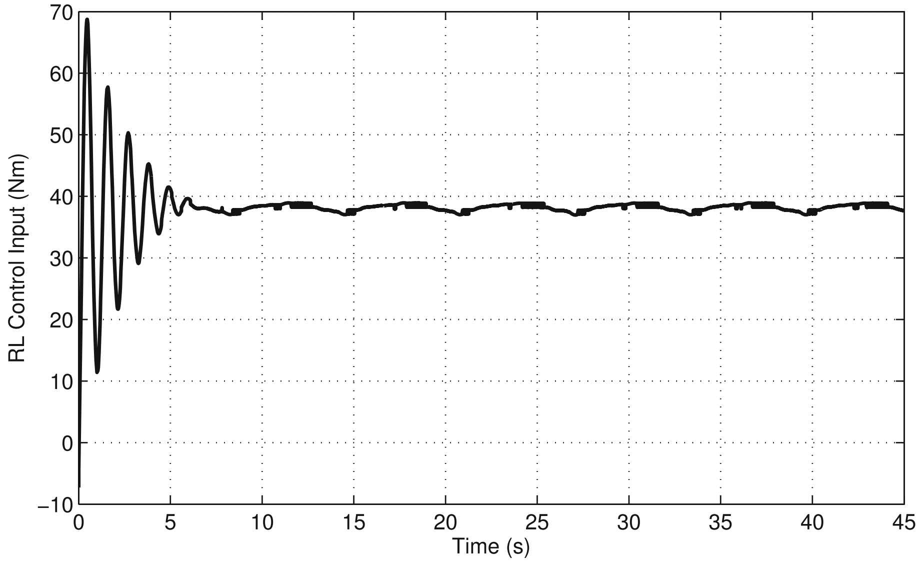


Fig. 6 RL control input of hub-2 for task-2

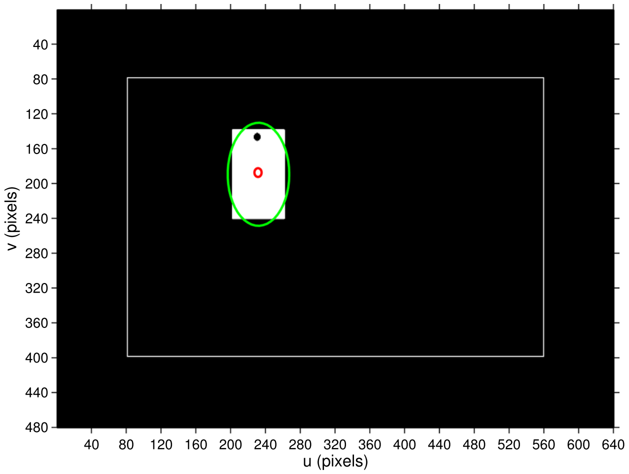


Fig. 7 Task-1's desired position

5.2.1 Tip-tracking performance for task-1

Figures 7 and 8 depict the task-1's desired location and initial position, respectively. Because the object centroid on the image is initially in an undesirable location, specifically at (608, 224), RL controller is employed to correct the TLFM position. The history of pixel coordinates for a visual feature is shown in Fig. 9. As seen in Fig. 9, the RL controller only takes six steps to put the visual feature inside the image plane's safe area, or within FOV.

A new two-time scale IBVS controller becomes active to finish the visual servoing task once the object enters the FOV. With the invariants r_{s5} and r_{s6} that are acquired from (66), the interaction matrix (67) is computed for the required position. Table 2 gives the initial and expected values of selected image features. Observed condition number is 2.49, which is

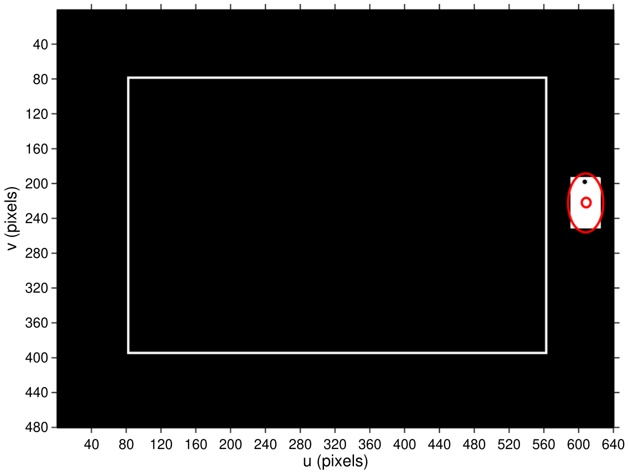


Fig. 8 Task-1's initial position

satisfactory. The image feature errors are shown in Fig. 10. As seen in Fig. 10, task-1's feature errors converge to zero after 62 s.

5.2.2 Tip-tracking performance for task-2

Figures 11 and 12 show the desired position and initial position of task-2, respectively.

Because the object centroid on the image is initially in an undesirable location, specifically at (585, 220), RL controller is chosen to adjust the TLFM position. The history of pixel coordinates for a visual feature is shown in Fig. 13. As seen in Fig. 13, the RL controller only takes five steps to put the visual feature into the image plane's safe area, or within FOV.

A new two-time scale IBVS controller becomes active to accomplish the visual servoing task whenever the object enters the FOV. With the invariants r_{s4} and r_{s6} that are derived from (66), the interaction matrix (67) is computed for the required position. Table 2 gives the initial and desired values of selected image features. It is seen that the condition number is 3.89, which is satisfactory. The image feature errors are shown in Fig. 14. As can be seen in Fig. 14, for task-2, the feature errors converge to zero after 42 s.

The task-1 and task-2 results indicate that the AI-IBVS controller is able to quickly correct the tip position of the TLFM when the visual feature is in an undesirable area or outside of FOV, allowing the visual feature to move through a significant distance as quickly as possible into the safe area to complete the visual servoing task.

In addition, the detailed study on coordinate vector relative to coordinate frame is included in Appendix B, in which the position and orientation (pose) of the object coordinate frames with respect to the base coordinate frame are highlighted.

5.3 Comparison

In this work, the important difference between the proposed control scheme as compared to other schemes [29–32] is presented as follows. First, the control scheme in [29–32] is not intended for flexible manipulators. Second, in order to prevent joint damage, it is not advised for a robot manipulator

Fig. 9 Visual features pixel coordinates of task-1

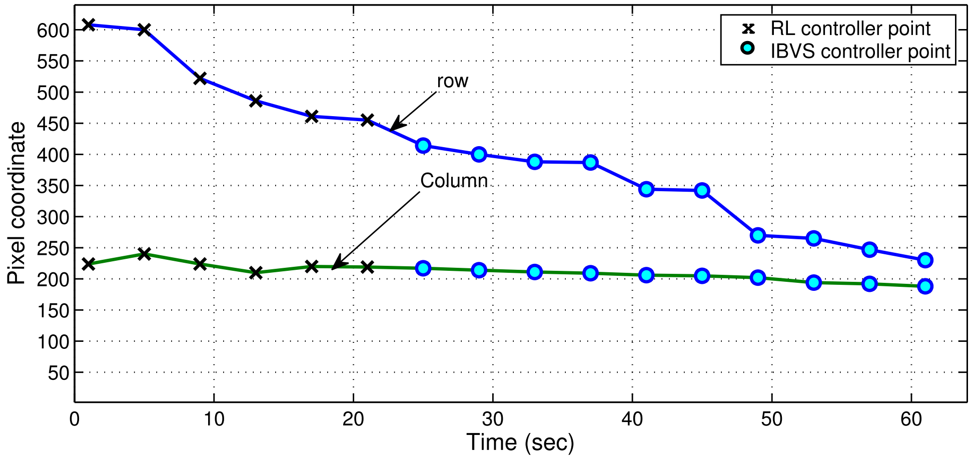


Table 2 The initial and desired value of image features for IBVS controller

Visual feature	Task-1 Desired value	Initial value	Feature error	Task-2 Desired value	Initial value	Feature error
r_{s5}/r_{s4} *	– 0.027	0.213	0.240	0.318	0.151	– 0.167
r_{s6}	0.0684	0.0524	– 0.016	0.082	0.563	0.481

*r_{s5} and r_{s4} are selected for task-1 and task-2, respectively

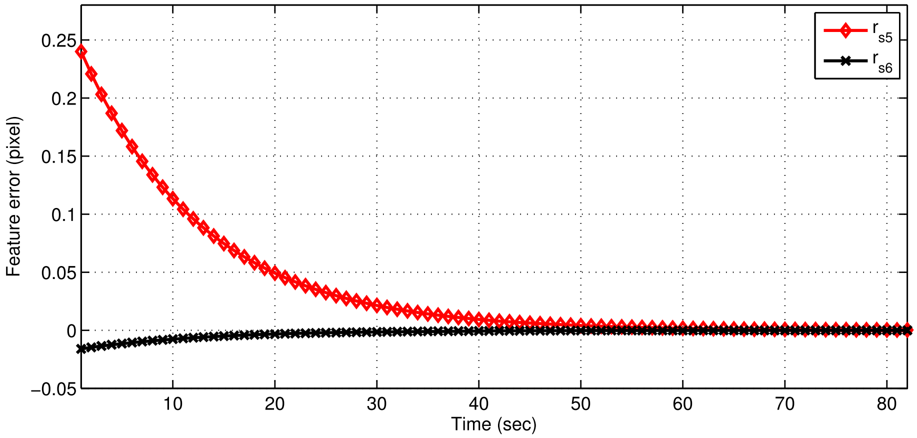


Fig. 10 Task-1’s Feature error

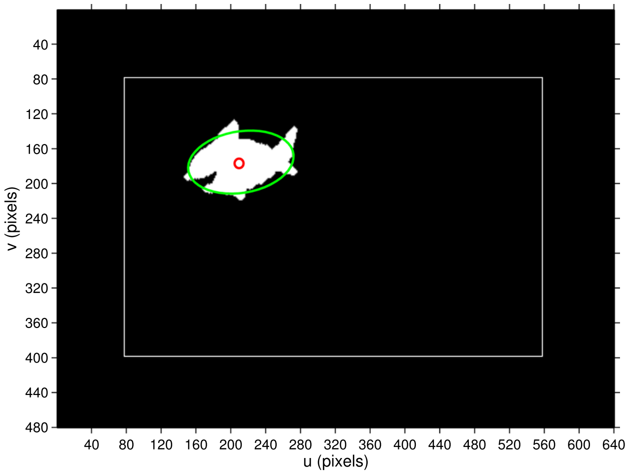


Fig. 11 Task-2’s desired position

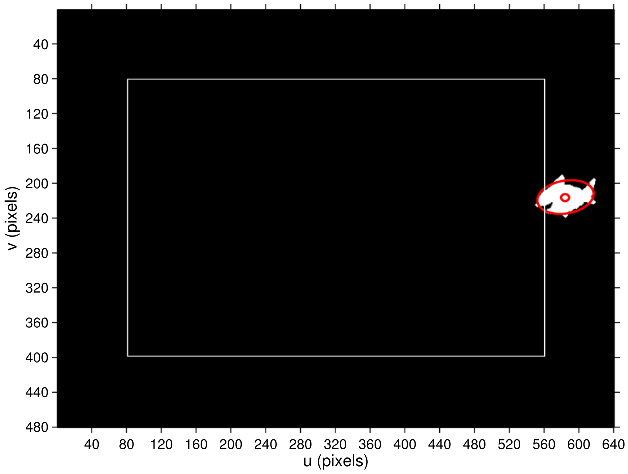


Fig. 12 Task-2’s initial position

to transition between two controllers in the hybrid scheme presented in [29]. Third, in [29, 31], a typical Q-learning algorithm with offline training is implemented in the hybrid system, while in [30], two RL algorithms with NN are separately constructed and in [32], asymmetric actor-critic and variational auto-encoder-based RL algorithm are designed, making the control scheme complex.

The proposed AI-IBVS controller possesses the capabilities of self-learning and decision-making and provides a balanced performance to complete the visual servoing task similar to [29–32].

6 Conclusion

In this work, an adaptive intelligent IBVS (AI-IBVS) controller for two-link flexible manipulator (TLFM) is developed. The challenges with IBVS and the retention of visual details in the FOV are specifically covered in this work. A wise selection of shifted moment-based visual features has been made in the new two-time scale IBVS controller to address the problems of singularity and local minima in IBVS. Therefore, in order to retain the object within camera FOV, an intelligent controller with reinforcement learning (RL) is proposed here. Moreover, a composite controller for TLFM is developed to combine RL controller and IBVS controller. Simulation have been performed to investigate the performance and robustness of the proposed controller. The results demonstrated that the proposed controller can successfully complete the visual servoing task by quickly correcting the tip position to bring the object within FOV. The proposed control scheme will be implemented and adapted in the real-time flexible manipulator in future studies.

Author contributions All authors contributed to the study’s conception and design. Material preparation, data collection, and analysis were performed by Umesh Kumar Sahu, Dipti Patra, and Bidyadhar Subudhi. The first draft of the manuscript was written by Umesh Kumar Sahu and extended by Dipti Patra and Bidyadhar Subudhi. Umesh Kumar Sahu contributed simulations evaluation. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

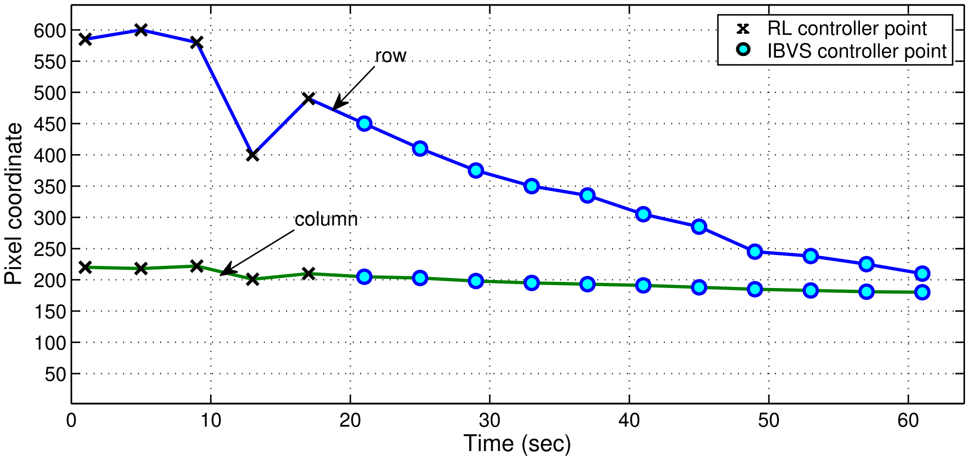


Fig. 13 Visual features pixel coordinates of task-2

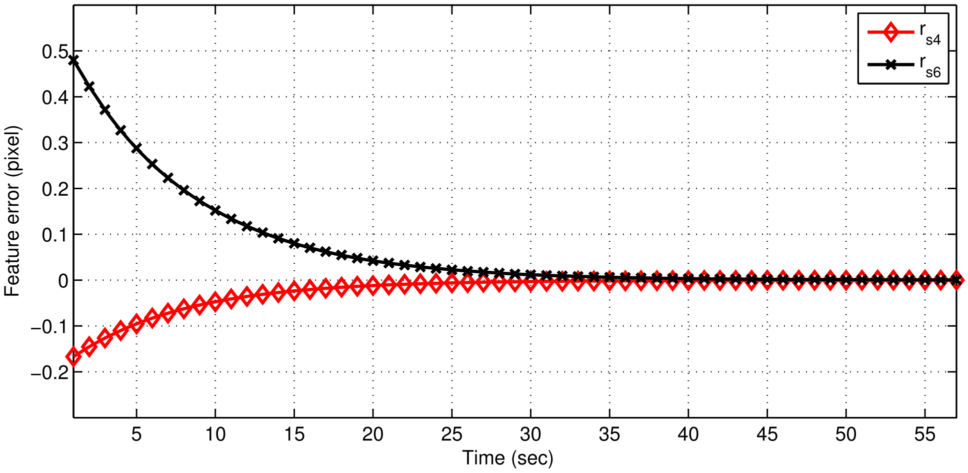


Fig. 14 Task-2's feature error

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal No funds, grants, or other support was received.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Appendix A Dynamics of TLFM

The dynamics of FLM is a distributed parameter system owing to the distributed link flexure. Due to distributed link flexure, the positioning and tracking of the tip in case of a TLFM are very difficult. In this case, it is assumed that motion of the TLFM in the horizontal plane, the links have uniform material properties and have constant cross-sectional area [40]. The schematic diagram of TLFM with a tip mounted camera is shown in Fig. 15, where $X_b O_b Y_b$ is the fixed coordinate frame with the joint of link-1 located at world coordinate $X_w O_w Y_w$. $X_2 O_2 Y_2$ and $\hat{X}_b \hat{O}_b \hat{Y}_b$ are the rigid and flexible body moving coordinate frame, respectively, of ith link and is fixed at the joint between link-1 and link-2. τ_i represents the applied torque of ith link, θ_i represents the joint angle of ith joint, and $y_i(l_i, t)$ denotes the deflection along ith link.

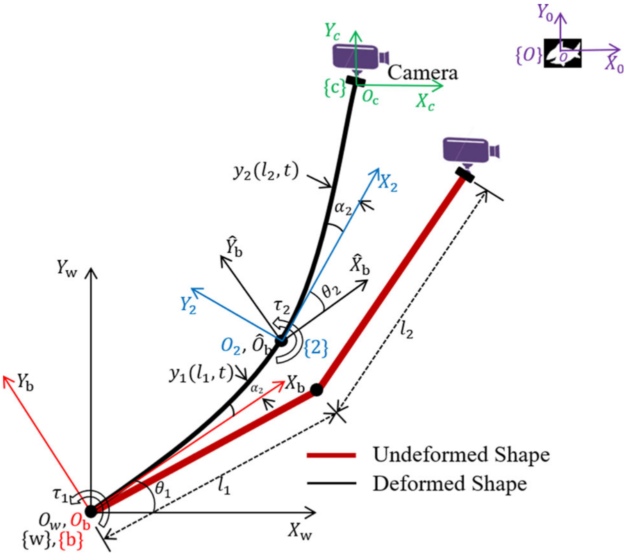


Fig. 15 Schematic diagram of an eye-in-hand camera configured TLFM

The complete system behaves as a non-minimum phase system, when the tip position is taken as the output. The actual output vector y_{pi} is considered as the output for the ith link. Hence, the redefined output can be written as

$$y_{pi} = \theta_i + \left(\frac{y_i(l_i,t)}{l_i}\right) \tag{A1}$$

where l_i is the length of ith link.

The dynamics of flexible links are derived as Euler–Bernoulli beams with deformation $y_i(l_i, t)$ for ith link satisfying the link partial differential equation

$$(EI)_i \frac{\partial^4 y_i(l_i,t)}{\partial l_i{}^4} + \rho_i \frac{\partial^2 y_i(l_i,t)}{\partial t_i{}^2} = 0 \tag{A2}$$

where ρ_i and $(EI)_i$ represent the density and flexural rigidity of the ith link, respectively.

The finite-dimensional expression for $y_i(l_i, t)$ can be presented using the AMM [1] as

$$y_i(l_i, t) = \sum_{j=1}^{n} \varphi_{ij}(l_i)\delta_{ij}(t) \tag{A3}$$

where φ_{ij} and δ_{ij} denote jth mode shape and modal coordinate of the ith link, respectively, and n is the number of assumed modes.

The dynamics of TLFM is derived by using the energy principle and the Lagrangian formulation technique along with AMM. The total Lagrangian (L) can be defined as

$$\frac{d}{dt}\frac{\partial L}{\partial q_i} - \frac{\partial L}{\partial q_i} = \tau_i \tag{A4}$$

where q_i is the ith generalized coordinates, i.e., $q_i = [\theta_i \;\; \dot{\theta}_i \;\; \delta_i \;\; \dot{\delta}_i]$. In (A4), total Lagrangian (L) value is substituted, i.e., difference of total kinetic energy and total potential energy of the TLFM and solve for the q_i generalized coordinates. The dynamics of TLFM is expressed in (1). The details of the matrices and vectors of (1) are

$$\begin{aligned} M(\theta_i, \delta_i) &= \begin{bmatrix} M_{rr}(\theta_i, \delta_i) & M_{rf}(\theta_i, \delta_i) \\ M_{fr}(\theta_i, \delta_i) & M_{ff}(\theta_i, \delta_i) \end{bmatrix} \\ &= \begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \end{aligned} \tag{A5}$$

$$\begin{aligned} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} &= \begin{bmatrix} c_{rr}(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) & c_{rf}(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) \\ c_{fr}(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) & c_{ff}(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) \end{bmatrix} \\ &= \begin{bmatrix} c_{rr} & c_{rf} \\ c_{fr} & c_{ff} \end{bmatrix} \end{aligned} \tag{A6}$$

$$K = diag\{0, 0, k_{11}, k_{12}, k_{21}, k_{22}\} \tag{A7}$$

$$\theta_i = \begin{bmatrix} \theta_1 \; \theta_2 \end{bmatrix}^T \tag{A8}$$

$$\delta_i = [\delta_1 \;\; \delta_2]^T = [\delta_{11} \;\; \delta_{12} \;\; \delta_{21} \;\; \delta_{22}]^T \tag{A9}$$

where M_{rr} and M_{ff} describe the positive definite submatrix related to rigid and flexible variable, respectively. $M_{rf} = M_{fr}$ representing coupling between the rigid and the flexible displacement variable. $k_{ij} = \omega_{ij}^2 m_i$ with ω_{ij} is natural frequency of jth mode and ith link, and m_i is the mass of ith link. The damping matrix, $D = diag\{d_{ij}\}$ for jth mode of ith link. θ_i and $\dot{\theta}_i$ are the joint angle and velocity of the ith joint, respectively. δ_i and $\dot{\delta}_i$ are the modal displacement and velocity for the ith link, respectively. τ_i is the actual applied torque for the ith link.

The matrices and vectors of state space model of TLFM presented in (2) are

$$x(t) = [\theta_i \;\; \dot{\theta}_i \;\; \delta_i \;\; \dot{\delta}_i\,]^T$$

$$u_i(t) = [\,\tau_i \;\; 0\,]^T$$

$$\begin{aligned} f_i(x(t)) = M(\theta_i, \delta_i)^{-1} \Bigg(&- \begin{bmatrix} c_1(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) \\ c_2(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i) \end{bmatrix} \\ &- K \begin{bmatrix} 0 \\ \delta_i \end{bmatrix} - D \begin{bmatrix} 0 \\ \dot{\delta}_i \end{bmatrix} \Bigg) \end{aligned}$$

$$g_i(x(t)) = M(\theta_i, \delta_i)^{-1}$$

Appendix B Pose of Coordinate Frames

With reference to Fig. 15, the object coordinate frame $\{o\}$ can be described by coordinate vectors relative to either frame $\{w\}$, $\{b\}$, $\{2\}$, $\{c\}$ or $\{o\}$ is shown in Fig. 16, where $\{w\}$, $\{b\}$, $\{2\}$, $\{c\}$, $\{o\}$ are the coordinate frame of world, joint-1 base, joint-2 base, camera, object, respectively. In Fig 16, solid and dashed vector represent the known and unknown pose, respectively.

ξ_o^b represents the position and orientation of an object coordinate frame (is known as its pose) $\{o\}$ with respect to

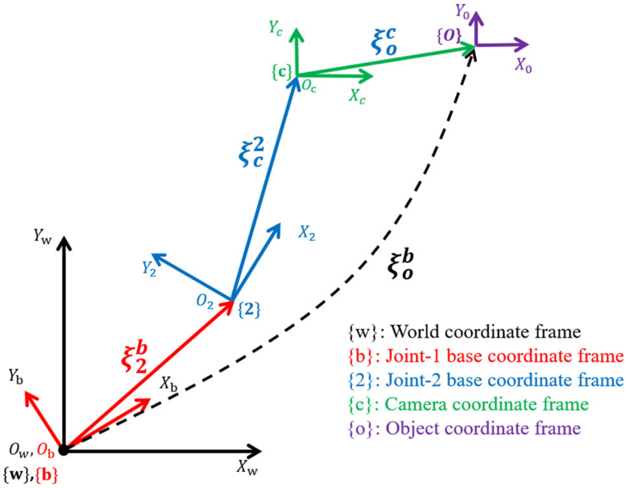

Y_c
{c}
O_c
X_c
ξ_o^c
{O}
Y_o
X_o
ξ_c^2
Y_w
X_2
Y_2
O_2
{2}
ξ_o^b
ξ_2^b
Y_b
X_b
O_w, O_b
{w},{b}
X_w
{w}: World coordinate frame
{b}: Joint-1 base coordinate frame
{2}: Joint-2 base coordinate frame
{c}: Camera coordinate frame
{o}: Object coordinate frame

Fig. 16 Coordinate vectors relative to either frames

base coordinate frame $\{b\}$. In the pose representation, the superscript denotes the reference coordinate frame and the subscript denotes the frame being described [39]. The pose of $\{o\}$ relative to $\{b\}$ can be expressed as

$$\xi_0^b = \xi_2^b \oplus \xi_c^2 \oplus \xi_o^c \tag{B10}$$

where $\oplus$ is used to indicate composition of relative poses. For TLFM, the pose of the object relative to base coordinate is expressed as

$$\xi_o^b = (R(\theta_1)T_x(l_1)R(\theta_2)T_x(l_2)) \oplus \xi_o^c \tag{B11}$$

where $R(\cdot)$ and $T(\cdot)$ represent the rotational and transnational motion of coordinate frame.

References

1. Subudhi B, Pradhan SK (2016) A flexible robotic control experiment for teaching nonlinear adaptive control. Int J Electric Eng Educ 53(4):341–356. https://doi.org/10.1177/0020720916631159
2. Lochan K, Roy BK, Subudhi B (2016) A review on two-link flexible manipulators. Ann Rev Control 42:346–367. https://doi.org/10.1016/j.arcontrol.2016.09.019
3. Sayahkarajy M, Mohamed Z, Faudzi AAM (2016) Review of modelling and control of flexible-link manipulators. Proc Inst Mech Eng Part I J Syst Control Eng 230(8):861–873. https://doi.org/10.1177/0959651816642099
4. Weng MC, Lu X, Trumper DL (2002) Vibration control of flexible beams using sensor averaging and actuator averaging methods. IEEE Trans Control Syst Technol 10(4):568–577. https://doi.org/10.1109/TCST.2002.1014676
5. Sahu U, Patra D, Subudhi B (2020) Vision based tip position tracking control of two-link flexible manipulator. IET Cyber Syst Robot 2(2):53–66. https://doi.org/10.1049/iet-csr.2019.0035
6. Corke PI, Hutchinson SA (2001) A new partitioned approach to image-based visual servo control. IEEE Trans Robot Autom 17(4):507–515. https://doi.org/10.1109/70.954764
7. Cowan NJ, Weingarten JD, Koditschek DE (2002) Visual servoing via navigation functions. IEEE Trans Robot Autom 18(4):521–533. https://doi.org/10.1109/TRA.2002.802202
8. Chesi G, Shen T (2012) Conferring robustness to path-planning for image-based control. IEEE Trans Control Syst Technol 20(4):950–959. https://doi.org/10.1109/TCST.2011.2157346
9. Nierobisch T, Fischer W, Hoffmann F (2006) Large view visual servoing of a mobile robot with a pan-tilt camera. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems. IEEE, Beijing, pp 3307–3312. https://doi.org/10.1109/IROS.2006.282503
10. Huang X, Houshangi N (2011) Lane following system for a mobile robot using information from vision and odometry. In: Proceedings of 24th Canadian conference on electrical and computer engineering. IEEE, Niagara Falls, pp 1009–1013. https://doi.org/10.1109/CCECE.2011.6030612
11. Remazeilles A, Chaumette F (2007) Image-based robot navigation from an image memory. Robot Auton Syst 55(4):345–356. https://doi.org/10.1016/j.robot.2006.10.002
12. Chesi G, Hashimoto K, Prattichizzo D, Idea AM (2004) Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Trans Robot 20(5):908–913. https://doi.org/10.1109/TRO.2004.829456
13. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction, 2nd edn. MIT Press, Cambridge
14. Kumar M, Rajagopal K, Balakrishnan SN, Nguyen NT (2014) Reinforcement learning based controller synthesis for flexible aircraft wings. IEEE/CAA J Autom Sin 1(4):435–448. https://doi.org/10.1109/JAS.2014.7004670
15. Pradhan SK, Subudhi B (2012) Real-time adaptive control of a flexible manipulator using reinforcement learning. IEEE Trans Autom Sci Eng 9(2):237–249. https://doi.org/10.1109/TASE.2012.2189004
16. Ouyang Y, He W, Li X (2017) Reinforcement learning control of a single-link flexible robotic manipulator. IET Control Theory Appl 11(9):1426–1433. https://doi.org/10.1049/iet-cta.2016.1540
17. Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL (2018) Optimal and autonomous control using reinforcement learning: a survey. IEEE Trans Neural Netw Learn Syst 29(6):1–21. https://doi.org/10.1109/TNNLS.2017.2773458
18. Asada M, Noda S, Tawaratsumida S, Hosoda K (1994) Vision-based behavior acquisition for a shooting robot by using a reinforcement learning. In: Proceedings of the workshop on visual behaviors. IEEE, Nagoya, pp 112–118. https://doi.org/10.1109/VL.1994.365601
19. Hosoda K, Asada M, Noda S, Tawaratsumida S (1996) Purposive behavior acquisition for a real robot by vision-based reinforcement learning. Mach Learn 23(2–3):279–303. https://doi.org/10.1023/A:1018237008823
20. Prabhu SM, Garg DP (1998) Fuzzy-logic-based reinforcement learning of admittance control for automated robotic manufacturing. Eng Appl Artif Intell 11(1):7–23. https://doi.org/10.1016/S0952-1976(97)00057-2
21. Takahashi Y, Takeda M, Asada M (1999) Continuous valued Q-learning for vision-guided behavior acquisition. In: Proceeding of the IEEE international conference on multisensor fusion and integra tion for intelligent systems. IEEE, Taipei, pp 255–260. https://doi.org/10.1063/1.97470
22. Distante C, Anglani A, Taurisano F (2000) Target reaching by using visual information and Q-learning controllers. Auton Robots 9(1):41–50. https://doi.org/10.1023/A:1008972101435
23. Gaskett C, Fletcher L, Zelinsky A (2000) Reinforcement learning for a vision based mobile robot. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems. IEEE, Takamatsu, pp 403–409. https://doi.org/10.1109/IROS.2000.894638

24. Gaskett C (2002) Q-Learning for robot control. PhD thesis, The Australian National University
25. Busquets D, De Mantaras RL, Sierra C, Dietterich TG (2002) Reinforcement learning for landmark-based robot navigation. In: Proceedings of first international joint conference on autonomous agents and multiagent systems. ACM, Bologna, pp 841–842. https://doi.org/10.1145/544862.544938
26. Hafner R, Riedmiller M (2003) Reinforcement learning on an omnidirectional mobile robot. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. IEEE, Las Vegas, pp 418–423. https://doi.org/10.1109/IROS.2003.1250665
27. Martinez-Marin T, Duckett T (2005) Fast reinforcement learning for vision-guided mobile robots. In: Proceedings of IEEE international conference on robotics and automation. IEEE, Barcelona, pp 4170–4175. https://doi.org/10.1109/ROBOT.2005.1570760
28. Kar I, Behera L (2010) Visual motor control of a 7 DOF robot manipulator using a fuzzy SOM network. Intell Serv Robot 3(1):49–60. https://doi.org/10.1007/s11370-009-0058-3
29. Wang Y, Lang H, De Silva CW (2010) A hybrid visual servo controller for robust grasping by wheeled mobile robots. IEEE/ASME Trans Mechatron 15(5):757–769. https://doi.org/10.1109/TMECH.2009.2034740
30. Miljkovic Z, Mitic M, Lazarevic M, Babic B (2013) Neural network Reinforcement Learning for visual control of robot manipulators. Expert Syst Appl 40(5):1721–1736. https://doi.org/10.1016/j.eswa.2012.09.010
31. Al-Shanoon A, Lang H, Wang Y, Zhang Y, Hong W (2021) Learn to grasp unknown objects in robotic manipulation. Intell Serv Robot 14(4):571–582. https://doi.org/10.1007/s11370-021-00380-9
32. Kim S, Jo HJ, Song JB (2022) Object manipulation system based on image-based reinforcement learning. Intell Serv Robot 15(2):171–177. https://doi.org/10.1007/s11370-021-00402-6
33. Karimi HR, Yazdanpanah MJ (2006) A new modeling approach to single-link flexible manipulator using singular perturbation method. Electr Eng 88(5):375–382. https://doi.org/10.1007/s00202-005-0302-6
34. Degris T, White M, Sutton RS (2012) Off-policy actor-critic. In: Proceedings of the 29th international conference on machine learning, Scotland, pp 457–464
35. Peters J, Schaal S (2008) Natural actor-critic. Neurocomputing 71(7–9):1180–1190. https://doi.org/10.1016/j.neucom.2007.11.026
36. Chaumette F (2004) Image moments: a general and useful set of features for visual servoing. IEEE Trans Robot 20(4):713–723. https://doi.org/10.1109/TRO.2004.829463
37. Tahri O, Tamtsia Y, Mezouar Y (2015) Visual servoing based on shifted moments. IEEE Trans Robot 31(3):798–804. https://doi.org/10.1109/TRO.2015.2412771
38. Bascetta L, Rocco P (2006) Two-time scale visual servoing of eye-in-hand flexible manipulators. IEEE Transection Robot 22(4):818–830. https://doi.org/10.1109/TRO.2006.878946
39. Corke P (2007) MATLAB toolboxes: robotics and division for students and teachers. IEEE Robot Autom Mag 14(4):16–17. https://doi.org/10.1109/M-RA.2007.912004
40. Sahu UK, Patra D (2016) Observer based backstepping method for tip tracking control of 2-dof serial flexible link manipulator. In: Proceedings of the international conference 2016 IEEE region 10 conference (TENCON). IEEE, Singapore, pp 3567–3572. https://doi.org/10.1109/TENCON.2016.7848721

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.