This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

Constrained Actor-Critic Reinforcement Learning for Deformable Object Visual Shape Servoing

Vrithik Raj Guthikonda, Rounak Bhattacharya, Ashwin P. Dani

***Abstract*—A constrained adaptive actor-critic (AAC)-based policy iteration controller for vision-based shape servoing of a deformable object is presented in this paper. A finite point-based representation of the deformable object is used, and the deformation Jacobian matrix is approximated using Fourier-series basis functions. An integral concurrent learning (ICL)-based parameter update law is designed to update the deformation model parameters. The high-dimensional state representation of the object requires large number of parameters to learn the deformation model, value function, and the optimal policy. To address this, a principal component analysis (PCA)-based dimensionality reduction method is used to compute a low dimensional representation of the object, which is then used to design the constrained AAC controller. A barrier transform (BT) method is used to transform part of the state into a constrained state, thereby ensuring that the physical constraints are satisfied. The proposed controller is extensively tested in simulations across several desired configurations, and through Monte Carlo runs on an ABB IRB120 robot platform.**

I. INTRODUCTION

The manipulation of deformable objects is a common problem in robotics and automation applications, including those in space and manufacturing. Feedback from cameras and/or depth sensors is generally used in robotic manipulation tasks involving deformable objects. The process of closed-loop control performed using camera feedback to regulate the shape of deformable objects to their desired form is termed vision-based shape servoing (VSS). Many contributions in literature have focused on the VSS problem [1]–[4], however, few have designed optimal controllers on reduced dimensional state representation while satisfying physical constraints. In this paper, a controller for the optimal VSS problem is presented using an actor-critic-based policy iteration reinforcement learning algorithm [5]–[7] applied to a reduced-dimensional state computed using principal component analysis (PCA). In addition to achieving optimal reduced dimensional VSS, a physical constraint, formulated as a constraint on the control point state, is also encoded using the barrier transform (BT) method [8], [9].

This work was supported in part by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program, in part by NSF grant no. SMA-2134367, and in part by UConn OVPR's Research Excellence Program.

V. Guthikonda, R. Bhattacharya, A. P. Dani are with the Department of Electrical and Computer Engineering at University of Connecticut, Storrs, CT 06269. Email: {vrithik.guthikonda; rounak.bhattacharya; ashwin.dani}@uconn.edu

Shape servoing controllers typically use a deformation model that describes the relationship between the velocity applied at the control point on the object and the resulting shape change. The deformation models, represented in a differential equation form using state representations based on a mesh of point coordinates, contours, B-splines, are used to develop model-based controllers. Finite element method (FEM) [10]–[12] can accurately model the object's deformation physics using mesh of connected points, but are also computationally expensive for real-time implementation. In contrast, mass-spring-damper (MSD) models [13], or geometric models [4], [14] may show better computational efficiency for real-time implementation of the model-based controllers. Using geometric modeling, contour representation of the object, parametrized using Bezier curves, is used in [15] to develop an adaptive controller for VSS. Similarly, a polynomial representation of the object is used in [14], where a deformation model-based visual servoing controller similar to Image-based Visual Servo (IBVS) control is developed.

The deformation Jacobian models can be learned using parametric function approximation techniques such as neural networks (NN), deep NN (DNN), Fourier series, or can be directly learned with no explicit parametrization. The parameters are learned offline or online using deformation data. A local linear deformation model-based control is presented in [2], [16], where the deformation Jacobian is estimated by solving a weighted least squares problem, with parameter recalibration as more deformation data becomes available. A DNN is used to learn the deformation model in [17], [18], which is then used to develop a VSS controller. In [19], [20], model inversion-based adaptive deformation Jacobian controllers, with and without state constraints, are developed using a Fourier series-based approximation, where the unknown deformation model parameters are estimated online using integral concurrent learning (ICL) method of [21]. Although all these controllers are able to achieve shape servoing to the desired configurations, the controllers are not designed to optimize a performance index and to satisfy the state constraints.

Many model-based reinforcement learning (RL) methods for continuous-time nonlinear systems have been developed for approximate optimal control, see [5]–[7], [9], [22]. For VSS problem, an optimal controller using a linear quadratic regulator (LQR) formulation is developed in [23] to achieve a desired final shape of the object along a specific deformation

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

path. A deep deterministic policy gradient (DDPG)-based model-free actor-critic algorithm is presented in [24] for shape servoing. A model-based approximate optimal control method is developed in [25], [26] for drift-free systems, in which the dynamics of deformation Jacobian-based VSS are also expressed. While applying these controllers to VSS problem, which may require high dimensional state space representation, a model order reduction (MoR) is required to achieve computational efficiency.

For VSS, the MoR problem [27] is addressed in [18], [28] using PCA, which generates a reduced state dimension before designing the shape servoing controller. In [29], a proper orthogonal decomposition (POD)-based model order reduction method is developed to project the FEM-based dynamics in reduced dimension, for FEM simulation-based control of soft robot. Within model-based RL, approximate optimal controllers for reduced order models are also designed. An efficient RL controller for high-dimensional complex systems using Koopman operators [30], where the reduced order linear model is approximated using extended dynamic mode decomposition (eDMD), are developed in [31], [32], by choosing a good set of observables. The eDMD method requires choosing a good set of observables for fitting linear system approximation of the nonlinear model [33]. The POD methods use PCA-like approach to compute dominant modes, for which the reduced order dynamics is computed using Galerkin projection of the known original nonlinear dynamics. For our paper, the deformation Jacobian model is learned online from data. PCA-based approach first captures the reduced-order dominant spatial modes, in the form of a reduced state, using variance criterion, for which the nonlinear dynamics are learned using a Fourier-basis parametrized model for designing model-based RL controller.

Compared to the existing literature on VSS [2]–[4], [14], [15], [18], [23], [28], the contribution of this paper is to develop a constrained approximate optimal controller for the VSS problem using the adaptive actor-critic (AAC) policy iteration RL for drift-free systems [26] in reduced dimensional state. Since the number of points required to accurately capture deformation behavior can be large, the state dimension of the VSS problem also becomes large. As a result, the NNs used to approximate the value function and the control policy contain a large number of parameters, making the control algorithm computationally expensive. The solution approach taken in the paper is to first reduce the state dimension of the deformable object using PCA, and then develop an AAC-RL controller using the reduced-order dynamics. By using a low-dimensional state representation, the number of parameters used to represent the deformation model is significantly reduced, which in turn lowers the computational complexity of estimating the deformation Jacobian model and the AAC controller through parameter update laws.

To avoid collisions between the robot and the environment, the state corresponding to the control point on the deformable object is constrained using a BT. The proposed controller is referred to as the PCA-BT-AAC controller. A Lyapunov stability analysis of the PCA-BT-AAC controller in the transformed space is presented, showing that the closed-loop state remains ultimately bounded. Furthermore, it is shown that the shape error between the original non-reduced state and the desired shape also remains ultimately bounded. The PCA-BT-AAC controller is validated through simulations across various cloth-folding tasks involving different desired configurations and cloths of different sizes, which may require high-dimensional state representations. Ablation studies and statistical tests are performed to evaluate the effects of different controller components, such as dimensionality reduction, constraint satisfaction and optimality. The proposed controller is also validated through Monte Carlo experiments on a cloth-folding task using object configurations estimated from real-time RGB-D camera data.

The rest of the paper is organized as follows. Section II discusses the deformation dynamics and defines the objective of the constrained optimal VSS problem. Section III shows details of the partial state constraint using BT method. Details of how to reduce state dimension before developing an RL controller is shown in Section IV. Section V discusses the mathematical and algorithmic aspects of the policy iteration RL algorithm. A Lyapunov stability analysis is carried out in Section VI, showing that the signals of closed-loop system remain bounded and that the shape servoing error converges exponentially to a bound. Sections VII and VIII provide detailed simulation and experimental results, respectively.

II. Deformation Model and Optimal VSS

In this section, the geometric representation and system dynamics of a deformable object are discussed and the optimal VSS problem is introduced. The deformable object's shape and position in 3D are represented using a set of points on the object and a set of control points, which are used to manipulate the object's shape to a desired configuration. The object deformation is defined as a change in the positions of the free moving points on the deformable object resulting from a velocity input applied at the control points. The following assumptions are made to model the deformation behavior of the object.

Assumption 1. The desired configuration is reachable using a set of selected control points.

Assumption 2. The object is deformed by applying quasi-static robot motions, which yields slower motions such that the object's inertial, rheological, and viscous effects during the motion are minimal.

Assumption 3. The deformable object is rigidly grasped at all times with no loose contact during the VSS task.

Assumption 4. The set of control points is represented using one common point.

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

A. Deformation Dynamics

Consider the following system representing the change in the 3D positions of points on the deformable object as a function of the velocity applied at a point on the object

$$\dot{s} = J_s(s, \theta)v \tag{1}$$

where $s(t) = [s_1^T, ..., s_n^T]^T \in \mathbb{R}^n$ are feature points representing the deformable object, and $v(t) \in \mathbb{R}^m$ represents the control input. Such a model can be derived from global deformation model presented in [34]. The unknown deformation Jacobian $J_s(s, \theta) \in \mathbb{R}^{n\times m}$ is approximated using a Fourier series-based regression, expressed as

$$J_s(s, \theta) = [\theta_1 Y_1(s), \theta_2 Y_2(s), \ldots, \theta_n Y_n(s)]^T \tag{2}$$

where $Y_i(s) \in \mathbb{R}^{d\times 1}$ represents a set of basis functions and $\theta_i \in \mathbb{R}^{m\times d}$ represents corresponding parameters. Each basis function $Y_i(s)$ is constructed as follows

$$Y_i(s) = \frac{1}{\sqrt{d}}\left[\cos\left(\frac{\omega_1^T s}{\tau}\right), \sin\left(\frac{\omega_2^T s}{\tau}\right), \ldots, \cos\left(\frac{\omega_{d-1}^T s}{\tau}\right), \ldots \sin\left(\frac{\omega_d^T s}{\tau}\right)\right]^T \tag{3}$$

where $\omega_k \in \mathbb{R}^n$ denotes the frequencies with $k = \{1, ..., d\}$, d is the number of basis functions, and $\tau \in \mathbb{R}$ is a constant selected based on the size of the basis functions. To facilitate VSS controller development, the system dynamics (1) are written in the following linear-in-parameter form

$$\dot{s} = \mathcal{Y}(s, v)\bar{\theta} \tag{4}$$

where $\mathcal{Y}(s, v) = \bar{Y}^T(s) \otimes v^T \in \mathbb{R}^{n\times nmd}$ and $\bar{\theta} = \left[\theta_{11}^T, \theta_{12}^T, \ldots, \theta_{21}^T \ldots \theta_{nd}^T\right]^T \in \mathbb{R}^{nmd}$, such that θ_{ij} represents the i^{th} matrix's j^{th} column, and $\bar{Y}(s) \in \mathbb{R}^{nd\times n}$ is constructed such that $\bar{Y}(s) = Y_i(s) \otimes I_{n\times n}$.

B. Constrained Optimal VSS Objective

Given the object shape representation and deformation model, the constrained optimal VSS objective is to design an (approximate) optimal control $v(t)$ such that the object's shape is regulated to a desired shape by minimizing a performance index and keeping the states within a prescribed bound. In addition, the computation required for the controller implementation should scale to various sizes of objects from small to large, i.e., from low-to-high dimensional state spaces. To achieve these objectives, an AAC-based policy iteration RL controller is designed to achieve the optimal VSS objective, a BT method is used to keep the states within specified bounds, and a PCA-based dimensionality reduction method is used to obtain a set of features with low dimensional state space to which the constrained AAC method is applied. For the control design, the shape regulation error $\tilde{s}(t) \in \mathbb{R}^n$ is defined as

$$\tilde{s}(t) \triangleq s(t) - s_d \tag{5}$$

where $s_d \in \mathbb{R}^n$ are the features of the deformable object at the desired configuration. The parameter estimation error $\tilde{\bar{\theta}} \in \mathbb{R}^{nmd}$ is defined as

$$\tilde{\bar{\theta}} \triangleq \bar{\theta} - \hat{\bar{\theta}}. \tag{6}$$

where $\hat{\bar{\theta}} \in \mathbb{R}^{nmd}$ is a model parameter estimate. Taking the time derivative of (5) and using (1) yields the following open-loop dynamics

$$\dot{\tilde{s}}(t) = J_s(s, \theta)v = J_s(\tilde{s}, s_d, \theta)v \tag{7}$$

III. Constrained Model using Barrier Transformation

In this section, details of the BT are provided to formulate a partially constrained state and corresponding system dynamics. Only the states corresponding to the points on the object grasped by the robot are constrained. To constrain part of the state, the error $\tilde{s}(t)$ is represented as $\tilde{s}(t) = [\tilde{s}_{uc}(t), \tilde{s}_c(t)]$ where $\tilde{s}_{uc}(t) \in \mathbb{R}^{n-l}$ denotes the unconstrained error and $\tilde{s}_c(t) \in \mathbb{R}^l$ denotes constrained error states. Let $b : \mathbb{R} \to \mathbb{R}$ be a BT defined as

$$b(\tilde{s}_{c_i}) = \ln \frac{A_i\left(a_i - \tilde{s}_{c_i}\right)}{a_i\left(A_i - \tilde{s}_{c_i}\right)}, \quad \forall i = 1, 2, \ldots, l \tag{8}$$

where a_i and A_i are bounds corresponding to the i^{th} error state such that $a_i < 0 < A_i$. Consider the BT-based constrained state, $e_c(t) \in \mathbb{R}^l$, whose components are computed as

$$e_{c_i} = b\left(\tilde{s}_{c_i}\right). \tag{9}$$

The error dynamics of e_{c_i} are obtained by taking the time derivative of (9) and applying the chain rule, yielding

$$\dot{e}_{c_i} = B_i\left(e_{c_i}\right)\left(J_s(b^{-1}_{(a_i, A_i)}\left(e_{c_i}\right) + s_d, \theta)v\right)_i \tag{10}$$

where $B_i(e_{c_i}) = \left(\frac{d(b^{-1}(e_{c_i}))}{de_{c_i}}\right)^{-1}$, and the inverse of (8), defined on the interval (a_i, A_i), is given by

$$\tilde{s}_{ci} = b^{-1}(e_{c_i}) = a_i A_i \frac{e^{e_{c_i}} - 1}{a_i e^{e_{c_i}} - A_i}. \tag{11}$$

The complete error, including the transformed error state, can be written as $e(t) = [\tilde{s}_{uc}(t), e_c(t)]^T \in \mathcal{E} \subset \mathbb{R}^n$. Taking the time derivative of (5) and using (1) and (10) yield the following error dynamics

$$\dot{e} = B(e_c)J_s(s, \theta)v = J(e, s_d, \theta)v \tag{12}$$

where $B(e_c) = \operatorname{diag}\left[I_{[n-l\times n-l]}, B_1(e_{c_1}), \ldots, B_l(e_{c_l})\right] \in \mathbb{R}^{n\times n}$, which using (4) can also be expressed in the following parametrized form

$$\dot{e} = B(e_c)\mathcal{Y}(s, v)\bar{\theta}. \tag{13}$$

Remark 1. Constraints on a subset of the state are required for the VSS task because, during manipulation, the robot end-effector holding the deformable object may collide with the environment.

A following lemma is presented to show that the solutions of the original system (7) and the transformed system (12) are related through the BT.

Lemma 1. If $\Phi\left(t, e(t_0), v_e(e(t))\right)$ is a complete Caratheodory solution to (12), starting from the initial condition $e(t_0) = (\tilde{s}_{uc}(t_0), e_c(t_0))$, under the feedback policy $v_e(e,t)$ and $\Lambda\left(t, \tilde{s}(t_0), v(\tilde{s}(t))\right)$ is a Caratheodory solution to (7), starting from the initial condition $\tilde{s}(t_0)$, under the feedback policy $v(\tilde{s}, t)$, defined as $v(\tilde{s}, t) = v_e(e,t)$, then $\Lambda\left(\cdot, \tilde{s}(t_0), v\right)$ is complete and $\Lambda\left(t, \tilde{s}(t_0), v\right) = b^{-1}\left(\Phi\left(t, b\left(\tilde{s}(t_0)\right), v_e\right)\right)$ for all $t \in \mathbb{R}_{\geq 0}$.

Proof. Proof of the lemma is provided in the Appendix. □

IV. Dimensionality reduction method

In this section, a PCA-based dimensionality reduction method is described. The dynamics of the reduced feature state are used in the optimal VSS control design. The state vector $p \in \mathbb{R}^k$, which represents the deformable object's configuration, consists of a large number of points and is reduced to a lower-dimensional state $s \in \mathbb{R}^n$ using PCA. To determine the reduced state, $n_p > k$ different configurations of the deformable object are collected, and the mean state vector $\bar{p}$ is computed as $\bar{p} = \sum_{i=1}^{n_p} p_i/n_p$. The mean-centered data matrix is then constructed as

$$P_c = \begin{bmatrix} p_1 - \bar{p} & p_2 - \bar{p} & \cdots & p_{n_p} - \bar{p} \end{bmatrix} \in \mathbb{R}^{k \times n_p} \tag{14}$$

where the columns of $P_c \sim \mathcal{D}$. The covariance matrix $C = \frac{1}{(n_p-1)} P_c P_c^T$ is computed and Singular Value Decomposition (SVD) of C is performed, resulting in $C = U\Sigma V^T$, where $U \in \mathbb{R}^{k\times k}$, $V \in \mathbb{R}^{k\times k}$ are orthonormal matrices whose columns contain the left and right singular vectors, respectively, and $\Sigma \in \mathbb{R}^{k\times k}$ is a diagonal matrix containing singular values $\sigma_i \ \forall i = \{1, ..., k\}$. Using the first n columns of U denoted by $U^{(n)} \in \mathbb{R}^{k\times n}$, the k-dimensional point vector is projected onto the lower-dimensional feature vector $s \in \mathbb{R}^n$ as follows

$$s = (U^{(n)})^T (p - \bar{p}). \tag{15}$$

The value of n is selected by thresholding the explained variance Υ, defined as, $\Upsilon(n) = \frac{\sum_{j=1}^{n} \sigma_j}{\sum_{j=1}^{k} \sigma_j}$, where σ_j denotes jth singular value and, $\Upsilon \in [0, 1]$. Since PCA calculates features associated with an orthonormal basis, the resulting features are linearly independent. For controlling m DoF, at least the same number of independent features are required.

V. Constrained Optimal Shape Servoing

In this section, the constrained optimal VSS problem is formulated using a policy iteration RL algorithm. First, an expression for the optimal controller is provided, followed by details of the adaptive actor-critic architecture, in which the critic and actor are approximated using NNs. The parameter update laws for the deformation model, as well as the update laws for the actor and critic NNs, are then derived.

A. Optimal Controller

To design an AAC-based optimal controller using the transformed dynamics in the reduced state space, the performance index is defined in terms of the optimal value function as follows

$$V^*(e(t)) = \min_{u(\tau)\in\Theta(\mathcal{E})} \int_t^\infty r(e,u)d\tau \tag{16}$$

where $\Theta(\mathcal{E})$ denotes the set of admissible control policies. Assuming that $V^*(e)$ is continuously differentiable and satisfies $V^*(0) = 0$, it can be shown that $V^*(e)$ satisfies the Hamilton-Jacobi-Bellman (HJB) equation. The local cost $r \in \mathbb{R}$ is defined as $r(e,u) = Q(e) + u^T R u$, where $Q(e) \in \mathbb{R}$ and $R \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices. Using (16), the optimal control policy $u^* \in \mathbb{R}^m$ is given by

$$u^*(e) = -\frac{1}{2} R^{-1} J^T(e, s_d, \theta) V_e^{*T} \tag{17}$$

where $V_e^* = \frac{\partial V^*}{\partial e}$. The optimal Hamiltonian of the system is defined as

$$H\left(e, u^*, V_e^*\right) = V_e^* J(e, s_d, \theta) u^* + r(e, u^*) = 0 \tag{18}$$

B. Adaptive Actor-Critic based Approximate Optimal Control

Since the HJB equation is a partial differential equation, computing its analytical solution is challenging when the parameters of the system dynamics θ are unknown. Therefore, an actor-critic NN structure is employed to approximate the value function and the control policy [5]. The optimal value function and optimal control policy are represented by NNs as follows

$$V^*(e(t)) = W_c^T \phi(e) + \epsilon_c(e) \tag{19}$$

$$u^*(e) = -\frac{1}{2} R^{-1} J^T(e, s_d, \theta) \left(\phi'(e)^T W_c + \epsilon_c'(e)^T\right) \tag{20}$$

where $W_c \in \mathbb{R}^{n_c\times 1}$ is the critic NN weight, $\phi : \mathbb{R}^n \to \mathbb{R}^{n_c}$ are the basis functions, $\phi'(e) \in \mathbb{R}^{\frac{n(n+1)}{2}\times n}$ is the Jacobian of ϕ with respect to $e(t)$, $\epsilon_c(e) \in \mathbb{R}$ is the function reconstruction error. The basis functions are selected to be polynomial functions $\phi(e)$ of dimension $n_c = \frac{n(n+1)}{2}$, defined as

$$\phi(e) = [e_1^2, \ldots, e_n^2, e_1 e_2, \ldots, e_1 e_n, e_2 e_3, \ldots, e_2 e_n, \ldots, e_{n-1} e_n]^T \tag{21}$$

In implementable form, the approximated value function $\hat{V}(e) \in \mathbb{R}$ and the approximated optimal control law $\hat{u}(e) \in \mathbb{R}^m$, are represented by

$$\hat{V}(e(t)) = \hat{W}_c^T \phi(e) \tag{22}$$

$$\hat{u}(e) = -\frac{1}{2} R^{-1} J^T(e, s_d, \hat{\theta}) \left(\phi'(e)^T \hat{W}_a\right) \tag{23}$$

where $\hat{W}_c \in \mathbb{R}^{n_c\times 1}$ and $\hat{W}_a \in \mathbb{R}^{n_a\times 1}$ are the estimated critic and actor weights, respectively, and $\hat{\theta}(t) \in \mathbb{R}^{nmd}$ are the model parameter estimates. To derive the actor and critic

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

weight update laws, a residual term $\delta \in \mathbb{R}$, referred to as the Bellman error, is computed using the difference between the optimal and approximated Hamiltonian, $H(e, \hat{u}, \hat{V}_e)$ as follows

$$\delta = H\left(e, \hat{u}, \hat{V}_e\right) - H\left(e, u^*, V_e^*\right) = \hat{V}_e J(e, s_d, \hat{\theta})\hat{u} + r(e, \hat{u}) \tag{24}$$

The actor and critic NN weights are updated using weight update laws designed to minimize the Bellman error, which can be expressed in measurable form as follows

$$\delta = \hat{W}_c^T \phi'(e) J(e, s_d, \hat{\theta})\hat{u} + r(e, \hat{u}) \tag{25}$$

C. Critic and Actor NN Weight Update Laws

Noting that δ depends on $\hat{W}_c$ and $\hat{W}_a$, the following least-squares based update laws are used. Similar to the developments in [6], [22], [26], the least-squares update law for the critic NN weights is given by

$$\dot{\hat{W}}_c = \text{proj}\left(-\gamma_c \Gamma \frac{w}{1 + c_1 w^T \Gamma w} \delta\right) \tag{26}$$

where $c_1 \in \mathbb{R}$ and $\gamma_c \in \mathbb{R}$ are constant gains, and $w \in \mathbb{R}^{n_c}$ is defined as follows

$$w = \phi'(e) J(e, s_d, \hat{\theta})\hat{u}, \tag{27}$$

and $\Gamma(t) = \left(\int_0^t w(\tau) w(\tau)^T d\tau\right)^{-1} \in \mathbb{R}^{n_c \times n_c}$ is a symmetric estimation gain matrix computed using

$$\dot{\Gamma} = -\gamma_c \Gamma \frac{w w^T}{1 + c_1 w^T \Gamma w} \Gamma, \quad \Gamma(0) = c_2 I \tag{28}$$

for a positive constant c_2.

A gradient-based update law for the actor NN weights is designed as follows

$$\dot{\hat{W}}_a = \text{proj}\left(-\frac{\gamma_a \hat{J}_\phi (\hat{W}_a - \hat{W}_c)\delta}{\sqrt{1 + w^T w}} - \gamma_{a2}\left(\hat{W}_a - \hat{W}_c\right)\right) \tag{29}$$

where γ_a, γ_{a2} are constant gains and $\hat{J}_\phi = \phi' \hat{J} R^{-1} \hat{J}^T \phi'^T$.

D. Model Parameter Update Law

The update law for model parameter estimate $\hat{\theta}(t)$ is designed using an ICL-based approach [21] and is given by

$$\begin{aligned}\dot{\hat{\theta}}(t) =& \text{proj}\Big(\gamma_\theta \mathcal{Y}^T B(e_c)^T \phi'^T \hat{W}_c \\ &+ K_{cl} \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T (s(t_i) - s(t_i - \triangle t) - \bar{\mathcal{Y}}_j \hat{\theta})\Big)\end{aligned} \tag{30}$$

where $\text{proj}(\cdot)$ denotes a projection operator [35], $\gamma_\theta \in \mathbb{R}$, $K_{cl} \in \mathbb{R}^{nmd \times nmd}$ is a constant positive definite diagonal gain matrix, Δt is the integration window and $\bar{\mathcal{Y}}_i$ is the i^{th} $\mathcal{Y}$ matrix from the history stack integrated over Δt, defined as

$$\bar{\mathcal{Y}}(t) = \int_{max\{t - \triangle t, 0\}}^{t} \mathcal{Y}(s(\tau), v(\tau)) d\tau. \tag{31}$$

The history stack, denoted by $\left\{s\left(t_j\right), s\left(t_j - \Delta t\right), \bar{\mathcal{Y}}_i\left(t_j\right)\right\}_{j=1}^{N}$, is recorded at an increasing sequence of time instants $\{t_j\}_{j=1}^{N}$. A singular value maximization procedure from [19], [21] is used to update the history stack.

Remark 2. By using a reduced state vector to implement the AAC policy iteration controller, the number of parameters in critic network decreases from $\frac{k(k+1)}{2}$ to $\frac{n(n+1)}{2}$. Similarly, the number of parameters in the actor network decreases from $\frac{k(k+1)}{2}$ to $\frac{n(n+1)}{2}$, and the number of parameters required to learn the deformation model decreases from kmd to nmd. Since $n << k$, this reduction is substantial.

Algorithm 1 Constrained Optimal VSS

1: Collect n_p object configurations and perform PCA using procedure in Section IV.
2: Initialize parameters $\hat{\theta}$, $\hat{W}_c$, $\hat{W}_a$ and store s_d
3: **while** $\|e(t)\| \geq \epsilon_T$, for each iteration i **do**
4: Compute a reduced state s using PCA
5: Append the BT state using (9) to the state vector
6: Compute shape regulation error $e(t)$ using (5)
7: Compute regressor $Y_i(s)$ using (3)
8: Compute $B(e_c)\mathcal{Y}(s, v)$ (10), construct $J(e, s_d, \hat{\theta})$
9: Compute value and policy basis function $\phi(e)$, $\phi'(e)$
10: Apply $\hat{u}$ computed using (23)
11: Update ICL history stack using Singular Value Maximization
12: Update $\hat{\theta}$ using (30)
13: Compute r, δ using (25)
14: Update $\hat{W}_c$, $\hat{W}_a$, $\hat{V}$ using (26), (29), (16)
15: **end while**

VI. Stability Analysis

For the controller presented in Section V, the error system dynamics, which includes shape regulation error $e(t)$, the actor and critic weight estimation errors $\tilde{W}_a(t)$ and $\tilde{W}_c(t)$ and the model parameter estimation error $\tilde{\theta}(t)$, are first derived. A Lyapunov-based stability analysis is then carried out to investigate the stability of the closed-loop system.

A. Error Dynamics

To derive the error dynamics of the critic and actor weights, the following expression for δ is obtained using (24)

$$\begin{aligned}\delta =& \hat{W}_c^T \phi'(e) J(e, s_d, \hat{\theta})\hat{u} + \hat{u}^T R \hat{u} - W_c^T \phi'(e) J(e, s_d, \theta) u^* \\ &- u^{*T} R u^* - \epsilon_c' J(e, s_d, \theta) u^*.\end{aligned} \tag{32}$$

Using $\hat{a} R \hat{a} - a^{*T} R a^* = \tilde{a}^T R \tilde{a} - 2\tilde{a}^T R a^*$ for $-u^{*T} R u^* + \hat{u}^T R \hat{u}$, δ can be written as

$$\begin{aligned}\delta =& -\tilde{W}_c^T w - W_c^T \phi' \tilde{J} \tilde{u} - \epsilon_c' J u^* + \frac{1}{4} \tilde{W}_a^T J_\phi \tilde{W}_a \\ &- \frac{1}{2} \tilde{W}_a^T J_\phi W_c + \frac{1}{4} \hat{W}_a^T \tilde{J}_\phi \hat{W}_a - \frac{1}{2} \hat{W}_a^T \tilde{J}_{a\phi} \hat{W}_a \\ &- \frac{1}{4} \epsilon_c' J_r \epsilon_c'^T - \frac{1}{2} \epsilon_c'^T J_r \phi'^T W_c\end{aligned} \tag{33}$$

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

where $\tilde{u} = u^* - \hat{u}$, $\tilde{J} = J - \hat{J}$, $J_\phi = \phi' J R^{-1} J^T \phi'^T$, $J_r = J R^{-1} J^T$, $\tilde{J}_{a\phi} = \phi' J R^{-1} \tilde{J}^T \phi'^T$, $\tilde{J}_\phi = \phi' \tilde{J} R^{-1} \tilde{J}^T \phi'^T$.

a) Actor NN weight error: Consider the actor NN weight estimation error, defined as $\tilde{W}_a = W_a - \hat{W}_a$. Taking the time derivative of $\tilde{W}_a(t)$ and substituting (29) yield

$$\begin{aligned}\dot{\tilde{W}}_a =& \frac{\gamma_a \hat{J}_\phi (\hat{W}_a - \hat{W}_c)\delta}{\sqrt{1 + w^T w}} + \gamma_{a2}\left(\hat{W}_a - \hat{W}_c\right) \\ =& C_1 \hat{J}_\phi \left(\hat{W}_a - \hat{W}_c\right)\delta - \gamma_{a2}\tilde{W}_a + \gamma_{a2}\tilde{W}_c \end{aligned} \tag{34}$$

where $C_1 = \frac{\gamma_a}{\sqrt{1+w^T w}}$.

b) Critic NN weight error: Consider the critic NN weight estimation error $\tilde{W}_c = W_c - \hat{W}_c$. Taking the time derivative of $\tilde{W}_c(t)$ and substituting (26) and (33) yield

$$\begin{aligned}\dot{\tilde{W}}_c =& -\gamma_c \Gamma \xi \xi^T \tilde{W}_c + \gamma_c \Omega \Big(-W_c^T \phi' \tilde{J} \tilde{u} - \epsilon_c' J u^* \\ &+ \frac{1}{4}\tilde{W}_a^T J_\phi \tilde{W}_a - \frac{1}{2}\tilde{W}_a^T J_\phi W_c + \frac{1}{4}\hat{W}_a^T \tilde{J}_\phi \hat{W}_a \\ &- \frac{1}{2}\hat{W}_a^T \tilde{J}_{a\phi} \hat{W}_a - \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T - \frac{1}{2}\epsilon_c'^T J_r \phi'^T W_c\Big)\end{aligned} \tag{35}$$

where $\xi = \frac{w}{\sqrt{1+c_1 w^T \Gamma w}}$, and $\Omega = \Gamma \frac{w}{1 + c_1 w^T \Gamma w}$.

Assumption 5. The normalized critic regressor $\xi = \frac{w}{\sqrt{1+c_1 w^T \Gamma w}}$ is bounded and persistently exciting (PE), i.e.,

$$\mu_a I \le \int_{t_0}^{t_0+T} \xi(\tau)\xi^T(\tau) d\tau \le \mu_b I, \quad \forall t_0 > 0 \tag{36}$$

where μ_a, μ_b, T are positive constants [6].

Under Assumption (5), the nominal system defined by the first term of (35) is globally exponentially stable. By the converse Lyapunov Theorem, there exists a Lyapunov function $V_{wc}\left(t, \tilde{W}_c\right)$ with the following properties

$$\begin{aligned}\gamma_1 \left\|\tilde{W}_c\right\|^2 \le V_{wc}\left(t, \tilde{W}_c\right) \le \gamma_2 \left\|\tilde{W}_c\right\|^2 \\ \frac{\partial V_{wc}}{\partial t} + \frac{\partial V_{wc}}{\partial \tilde{W}_c}\left(-\gamma_c \Gamma \xi \xi^T \tilde{W}_c\right) \le -\eta_1 \left\|\tilde{W}_c\right\|^2 \\ \left\|\frac{\partial V_{wc}}{\partial \tilde{W}_c}\right\| \le \bar{\gamma} \left\|\tilde{W}_c\right\|\end{aligned} \tag{37}$$

where $\gamma_1, \gamma_2, \eta_1, \bar{\gamma} \in \mathbb{R}^+$.

c) Model parameter error: The model parameter estimation error dynamics $\tilde{\bar{\theta}}(t)$ are derived using $s(t) - s(t - \triangle t) = \bar{\mathcal{Y}}(t)\bar{\theta}$ [19] and (30), yielding

$$\dot{\tilde{\bar{\theta}}} = -\gamma_\theta \mathcal{Y}^T B(e_c)^T \phi'^T \hat{W}_c - K_{cl} \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i \tilde{\bar{\theta}} \tag{38}$$

Assumption 6. The system in (1) is sufficiently exciting over a finite time interval. Specifically, $\exists \delta > 0$, $\exists T > \Delta t$: $\forall t \ge T$, $\lambda_{\min}\left\{\sum_{i=1}^N \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i\right\} > \delta$, where $\lambda_{\min}$ denotes the minimum eigenvalue of the matrix.

For the stability result presented in the following Theorem, the following bounds are defined

$$\begin{aligned}\|\frac{1}{4}\epsilon_c' J_r \epsilon_c'^T + \frac{1}{4}W_c^T J_\phi W_c + \frac{1}{2}\epsilon_c' J_r \phi'^T W_c \\ -\frac{1}{2}\epsilon_c' J_{ra} \phi'^T \hat{W}_a - \frac{1}{2}W_c^T \hat{J}_\phi \hat{W}_a + \tilde{W}_c^T \phi' \tilde{J} \hat{u}\| \le \kappa_1\end{aligned} \tag{39}$$

$$\begin{aligned}\|\frac{1}{2}W_c^T J_\phi\| \le \kappa_2, \quad \|\frac{1}{4}\tilde{W}_a^T J_\phi \tilde{W}_a\| \le \kappa_3, \\ \overline{w} = \left\|\hat{J}_\phi(\hat{W}_a - \hat{W}_c)\right\|\end{aligned} \tag{40}$$

$$\begin{aligned}\| -W_c^T \phi' \tilde{J} \tilde{u} - \epsilon_c' J u^* - \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T - \frac{1}{2}\epsilon_c'^T J_r \phi'^T W_c \\ + \frac{1}{4}\hat{W}_a^T \tilde{J}_\phi \hat{W}_a - \frac{1}{2}\hat{W}_a^T \tilde{J}_{a\phi} \hat{W}_a\| \le \kappa_4\end{aligned} \tag{41}$$

where $\kappa_1, \kappa_2, \kappa_3, \kappa_4$ and $\bar{w}$ are constants and $J_{ra} = J R^{-1} \hat{J}^T$.

Theorem 1. Given that Assumptions 1-6 hold and the following sufficient condition is satisfied

$$\gamma_{a2} - \gamma_a \kappa_2 \overline{w} > 0 \tag{42}$$

the adaptive actor-critic controller in (23), along with the model parameter update law in (30) and the critic and actor weight update laws in (26)-(28) and (29), guarantees that the signals $e(t), \tilde{s}, \tilde{\bar{\theta}}(t)$, $\tilde{W}_a(t)$, and $\tilde{W}_c(t)$ are uniformly ultimately bounded.

Proof. Consider a positive definite continuously differentiable Lyapunov function $V : \mathcal{E} \times \mathbb{R}^{n_c} \times \mathbb{R}^{n_a} \times \mathbb{R}^{nmd} \times [0, \infty) \to \mathbb{R}^+$

$$\begin{aligned}V(e, \tilde{W}_c, \tilde{W}_a, \tilde{\bar{\theta}}, t) = V^*(t) + V_{wc}\left(t, \tilde{W}_c\right) \\ + \frac{1}{2}\tilde{W}_a^T \tilde{W}_a + \frac{1}{2\gamma_\theta}\tilde{\bar{\theta}}^T \tilde{\bar{\theta}}\end{aligned} \tag{43}$$

where $V^*(t)$ is the optimal value function, V_{wc} is the Lyapunov function defined in (37), and $\gamma_\theta \in \mathbb{R}^+$. Since the optimal value function $V^*(t)$ is continuously differentiable and positive definite, there exist class-$\mathcal{K}$ functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ such that $\alpha_1(\|e\|) \le V^*(t) \le \alpha_2(\|e\|), \forall e \in \mathcal{B}_e \subset \mathcal{E}$. Let's define $z(t) = \left[e(t)^T, \tilde{W}_c(t)^T, \tilde{W}_a(t)^T, \tilde{\bar{\theta}}(t)^T\right]^T \in \mathbb{R}^{n+n_c+n_a+nmd}$. Based on the bounds on $V^*(t)$, the following bounds can be derived

$$\alpha_3(\|z\|) \le V\left(e, \tilde{W}_c, \tilde{W}_a, \tilde{\bar{\theta}}, t\right) \le \alpha_4(\|z\|) \quad \forall e \in \mathcal{B}_e \tag{44}$$

where $\alpha_3(\cdot)$ and $\alpha_4(\cdot)$ are class-$\mathcal{K}$ functions. Taking the time derivative of (43), adding and subtracting $V_e^* J(s_d, e, \theta)u^*$, using $V_e^* J(s_d, e, \theta) = -2u^{*T} R$ and $V_e^* J(s_d, e, \theta)u^* = -Q(e) - u^{*T} R u^*$, and substituting u^* and $\hat{u}$ from (20) and

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

(23), respectively, and $\dot{\tilde{\bar{\theta}}}$ from (38), the following expression for $\dot{V}$ is obtained

$$\dot{V}_z = -Q(e) + \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T + \frac{1}{4}W_c^T J_\phi W_c + \frac{1}{2}\epsilon_c' J_r \phi'^T W_c - \frac{1}{2}W_c^T \phi' J R^{-1} \hat{J}^T \phi'^T \hat{W}_a - \frac{1}{2}\epsilon_c' J R^{-1} \hat{J}^T \phi'^T \hat{W}_a + \dot{V}_{wc} + \tilde{W}_a^T \dot{\hat{W}}_a - \tilde{\bar{\theta}}^T \mathcal{Y}^T B(e_c)^T \phi'^T \hat{W}_c - k_{cl} \tilde{\bar{\theta}}^T \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i \tilde{\bar{\theta}}$$

Adding and subtracting $W_c^T \phi' \hat{J} \hat{u}$ and $\hat{W}_c^T \phi' \tilde{J} \hat{u}$, and using $-\frac{1}{2} W_c^T \phi'^T J R^{-1} \hat{J}^T \phi'^T \hat{W}_a = W_c^T \phi'^T J \hat{u}$ the following expression is obtained.

$$\begin{aligned}\dot{V} =& - Q(e) + \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T + \frac{1}{4}W_c^T J_\phi W_c + \frac{1}{2}\epsilon_c' J_r \phi'^T W_c \\ &- \frac{1}{2}\epsilon_c' J R^{-1} \hat{J}^T \phi'^T \hat{W}_a + \tilde{W}_c^T \phi' \tilde{J} \hat{u} + \hat{W}_c \phi' \tilde{J} \hat{u} \\ &+ W_c^T \phi' \hat{J} \hat{u} + \dot{V}_{wc} - \tilde{W}_a^T \dot{\hat{W}}_a - \tilde{\bar{\theta}}^T \mathcal{Y}^T B(e_c)^T \phi'^T \hat{W}_c \\ &- k_{cl} \tilde{\bar{\theta}}^T \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i \tilde{\bar{\theta}}\end{aligned} \tag{45}$$

Using $\hat{W}_c^T \phi' \tilde{J} \hat{u} = \hat{W}_c^T \phi' B(e_c) \mathcal{Y} \tilde{\bar{\theta}}$ and canceling the resulting terms yields

$$\begin{aligned}\dot{V} =& - Q(e) + \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T + \frac{1}{4}W_c^T J_\phi W_c + \frac{1}{2}\epsilon_c' J_r \phi'^T W_c \\ &- \frac{1}{2}\epsilon_c' J R^{-1} \hat{J}^T \phi'^T \hat{W}_a + \tilde{W}_c^T \phi' \tilde{J} \hat{u} + W_c^T \phi' \hat{J} \hat{u} \\ &+ \dot{V}_{wc} - \tilde{W}_a^T \dot{\hat{W}}_a - k_{cl} \tilde{\bar{\theta}}^T \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i \tilde{\bar{\theta}}\end{aligned} \tag{46}$$

The expression for $\dot{V}$ can be further simplified by substituting (34), using the bounds on Lyapunov function V_{wc} from (37), and utilizing $W_c^T \phi' \hat{J} \hat{u} = -\frac{1}{2} W_c^T \hat{J}_\phi W_c + \frac{1}{2} W_c^T \hat{J}_\phi \tilde{W}_a$

$$\begin{aligned}\dot{V} \le& -Q(e) - \eta_1 \left\|\tilde{W}_c\right\|^2 - \gamma_{a2} \|\tilde{W}_a\|^2 + \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T \\ &+ \frac{1}{4}W_c^T J_\phi W_c + \frac{1}{2}\epsilon_c' J_r \phi'^T W_c - \frac{1}{2}\epsilon_c' J_{ra} \phi'^T \hat{W}_a \\ &+ \tilde{W}_c^T \phi' \tilde{J} \hat{u} - \frac{1}{2} W_c^T \hat{J}_\phi \hat{W}_a + \gamma_c \Omega \bar{\gamma} \left\|\tilde{W}_c\right\| \Big(-W_c^T \phi' \tilde{J} \tilde{u} \\ &- \epsilon_c' J u^* + \frac{1}{4}\tilde{W}_a^T J_\phi \tilde{W}_a - \frac{1}{2}\tilde{W}_a^T J_\phi W_c + \frac{1}{4}\hat{W}_a^T \tilde{J}_\phi \hat{W}_a \\ &- \frac{1}{2}\hat{W}_a^T \tilde{J}_{a\phi} \hat{W}_a - \frac{1}{4}\epsilon_c' J_r \epsilon_c'^T - \frac{1}{2}\epsilon_c'^T J_r \phi'^T W_c\Big) \\ &+ \gamma_{a2} \tilde{W}_a^T \tilde{W}_c + C_1 \tilde{W}_a^T \left(\hat{J}_\phi \left(\hat{W}_a - \hat{W}_c\right)\right) \delta \\ &- k_{cl} \tilde{\bar{\theta}}^T \sum_{i=1}^{N} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i \tilde{\bar{\theta}}\end{aligned} \tag{47}$$

Utilizing the bounds in (39)-(41), using $|C_1| \le \gamma_a$, and completing the squares, $\dot{V}$ can be upper bounded as

$$\begin{aligned}\dot{V} \le& -Q(e) - (1-\theta_1)\,\eta_1 \left\|\tilde{W}_c\right\|^2 \\ &- (1-\theta_1)\left(\gamma_{a2} - \gamma_c \kappa_2 \bar{w}\right) \left\|\tilde{W}_a\right\|^2 - \sigma_1 \|\tilde{\bar{\theta}}\|^2 \\ &+ \kappa_1 + \left(C_1\left(\kappa_4 + \kappa_3\right)\bar{w}\right) \left\|\tilde{W}_a\right\| + \left(\bar{\gamma}(\kappa_4 + \kappa_3)\right) \left\|\tilde{W}_c\right\|\end{aligned} \tag{48}$$

where $\eta_2 = \gamma_{a2} - \gamma_a \kappa_2 \bar{w}, \sigma_1 = k_{cl} \lambda_{\min}\left(\sum_{i=1}^{m} \bar{\mathcal{Y}}_i^T \bar{\mathcal{Y}}_i\right), 1-\theta_1 > 0$. The final bound on $\dot{V}$ can be derived as

$$\begin{aligned}\dot{V} \le& -Q(e) - (1-\theta_1-\theta_2)\,\eta_1 \left\|\tilde{W}_c\right\|^2 \\ &- (1-\theta_1-\theta_2)\,\eta_2 \left\|\tilde{W}_a\right\|^2 - \sigma_1 \|\tilde{\bar{\theta}}\|^2 + \kappa_1 \\ &+ \frac{\left(\bar{\gamma}(\kappa_4+\kappa_3)\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_1} + \frac{\left(C_1\left(\kappa_4+\kappa_3\right)\bar{w}\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_2}\end{aligned} \tag{49}$$

where $1-\theta_1-\theta_2 > 0$. Since $Q(e)$ is positive definite, Lemma 4.3 of [36] can be invoked to derive

$$\begin{aligned}\alpha_5(\|z\|) \le& \; Q + (1-\theta_1-\theta_2)\,\eta_1 \left\|\tilde{W}_c\right\|^2 + \sigma_1 \|\tilde{\bar{\theta}}\|^2 \\ &+ (1-\theta_1-\theta_2)\,\eta_2 \left\|\tilde{W}_a\right\|^2 \le \alpha_6(\|z\|)\end{aligned} \tag{50}$$

where $\alpha_5(\cdot)$ and $\alpha_6(\cdot)$ are class-$\mathcal{K}$ functions. Using (50), the expression can be upper bounded as

$$\begin{aligned}\dot{V} \le& -\alpha_5(\|z\|) + \kappa_1 + \frac{\left(\bar{\gamma}(\kappa_4+\kappa_3)\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_1} \\ &+ \frac{\left(C_1\left(\kappa_4+\kappa_3\right)\bar{w}\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_2}\end{aligned} \tag{51}$$

which proves that $\dot{V} \le 0$, whenever $z(t)$ is outside the compact set

$$\begin{aligned}\bar{\Omega} = \Bigg\{ z : \|z\| \le \alpha_5^{-1} \Bigg(& \kappa_1 + \frac{\left(\bar{\gamma}(\kappa_4+\kappa_3)\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_1} \\ &+ \frac{\left(C_1\left(\kappa_4+\kappa_3\right)\bar{w}\right)^2}{4\left(1-\theta_1-\theta_2\right)\eta_2}\Bigg)\Bigg\}\end{aligned} \tag{52}$$

Invoking Theorem 4.18 of [36], it follows that $\|z\|$ is uniformly ultimately bounded (UUB). Consequently, the BT-VSS error satisfies $\|e(t)\| \le \epsilon_e$ for some $\epsilon_e \in \mathbb{R}^+$. Since the BT in (8) is smooth and invertible, the reduced-dimensional error also satisfies $\|\tilde{s}\| \le \epsilon_s$, for some $\epsilon_s \in \mathbb{R}^+$ [37]. □

Corollary 1. The bound on error $\mathbb{E}\left[\|p_c - p_{dc}\|\right]$, where $p_c(t), p_{dc} \sim \mathcal{D}$ denote the mean-centered current state and desired states, respectively, decreases as the number of PCA features increases, provided that the PCA-BT-AAC controller guarantees $\|s - s_d\| \le \epsilon_s$.

Proof. Let $\|p - p_d\| = \|p_c - p_{dc}\|$ denote the norm of the total configuration error of the object, which can be written as

$$p_c - p_{dc} = U^{(n)} U^{(n)T}\left(p_c - p_{dc}\right) + U_p\left(p_c - p_{dc}\right)$$

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

where $U_p = I - U^{(n)}U^{(n)T}$. Using (5), (15), and the Pythagorean theorem yields

$$\|p_c - p_{dc}\|^2 = \|U^{(n)}\tilde{s}\|^2 + \|U_p\left(p_c - p_{dc}\right)\|^2 \tag{53}$$

Based on the development shown in Appendix B

$$\mathbb{E}[\|U_p\left(p_c - p_{dc}\right)\|^2] = \sum_{i=n+1}^{k} \sigma_i + \epsilon_{pd} \tag{54}$$

where $\epsilon_{pd} = \|U_p p_{dc}\|^2$, and σ_i denote the ith singular value of the PCA data matrix C computed in Section IV. Using the deterministic bound on $\|\tilde{s}\|$ gives $\mathbb{E}[\|U^{(n)}\tilde{s}\|^2] = \mathbb{E}[\|\tilde{s}\|^2] \leq \epsilon_s^2$. Taking the expectation of (53) yields

$$\mathbb{E}[\|p_c - p_{dc}\|^2] \leq \epsilon_s^2 + \sum_{i=n+1}^{k} \sigma_i + \epsilon_{pd} \tag{55}$$

Using Jensen's inequality $\mathbb{E}[\|p_c - p_{dc}\|]^2 \leq \mathbb{E}[\|p_c - p_{dc}\|^2]$, which leads to the following bound

$$\mathbb{E}[\|p_c - p_{dc}\|] \leq \left(\epsilon_s^2 + \sum_{i=n+1}^{k} \sigma_i + \epsilon_{pd}\right)^{1/2} \tag{56}$$

Thus, the shape error of the full state vector $p_c(t) - p_{dc}$ remains bounded. □

Remark 3. The ultimate bound on $\|z\|$ can be reduced by appropriately choosing the gains γ_{a2}, γ_c, and by increasing the number of neurons in the NN, thereby reducing the function approximation errors of the actor and critic NNs.

VII. Simulations

A. Simulation Setup

The PCA-BT-AAC algorithm is evaluated on a cloth object in the NVIDIA Flex simulator. The controller is implemented in C++ and executed on an Alienware Intel-i7 desktop with 32-GB RAM and an NVIDIA GeForce RTX 2080 GPU. A subset of points is sampled symmetrically from the edges and central region of the cloth, based on its size, to accurately capture the shape and deformation. One corner of the cloth is fixed to a surface, while a rigid cube is attached to another corner to simulate a firm grasp by a robot end-effector. The control velocity is applied to the rigid cube at 60Hz.

To compute the reduced feature set using PCA, predefined constant velocity motions are applied to the cloth in all directions and $n_p = 700$ object configuration vectors $p \in \mathbb{R}^{27}$ are stored in a data matrix P. For the deformation-Jacobian model, a total of $d = 10$ Fourier basis functions are used. The weights $\omega_k \in \mathbb{R}^n$, for $k = \{1, \ldots, d\}$ are initialized according to $\omega_k \sim \mathcal{N}(0, I_n)$, and $\tau = 1$. The initial critic and actor NN weight estimates $\hat{W}_c(t_0)$, $\hat{W}_a(t_0)$, and model parameter estimates $\hat{\bar{\theta}}(t_0)$ are selected as $3I_{n_c}$, $4I_{n_a}$ and $4I_{nmd}$, respectively.

For all the simulations, the ICL integration window and history stack size are selected as $\Delta t = 0.5$s and $N = 70 \geq dm$, respectively. The ICL history stack is initialized by collecting the velocities applied to the rigid cube and the corresponding cloth configurations represented by the reduced feature set $s(t)$.

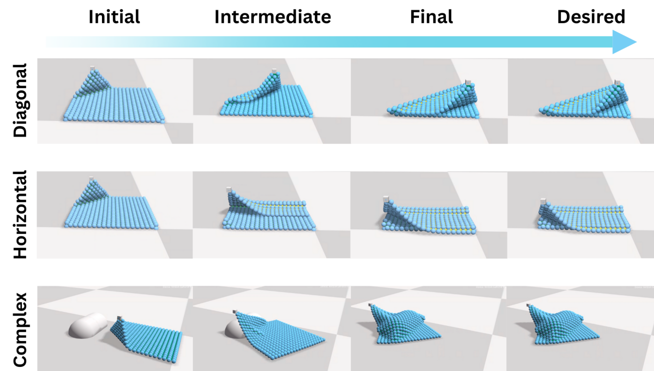


Figure 1: Simulation of VSS (a) Diagonal cloth folding, (b) Horizontal cloth folding, (c) Cloth manipulation to a complex desired shape

B. Simulation 1: Shape Servoing on Various Desired Configurations

In this simulation, the PCA-BT-AAC controller is implemented for three different desired configurations, folds across horizontal and diagonal axes and a complex configuration. The purpose of this simulation is to test whether the PCA-BT-AAC controller can achieve different desired configurations. One simulation is conducted per desired configuration. The results of the VSS implementation are shown in Fig. 1 as a sequence of starting, intermediate and final configurations. For the horizontal and diagonal fold configurations, one end of the cloth is fixed and the other end of the cloth is manipulated using a robot. For the complex configuration, the cloth has no points fixed to the surface and requires a pulling motion while ensuring the control point's safety.

The PCA-BT-AAC controller is implemented using the quadratic local cost $r(e, u) = e^T Qe + u^T Ru$, where $\mathrm{Q} = 1\mathrm{I}_{n\times n}$, $\mathrm{R} = 50\mathrm{I}_{n\times n}$, and $n = \dim(s)$. The lower and upper bounds for BT are chosen as $a = -0.04$ and $A = 0.1$ such that the control point always stays above the highest point of the object. The controller parameters for PCA-BT-AAC are selected to be $\gamma_\theta = 0.5$, $\mathrm{K}_{cl} = 500$, $\gamma_a = 0.001$, $\gamma_{a2} = 0.1$, $\gamma_c = 1$, $\mathrm{c}_1 = 3$, and $\mathrm{c}_2 = 50$. The VSS is able to successfully achieve the cloth's desired configuration in all three cases without violating any constraints.

C. Simulation 2: Shape Servoing on Same Object Size with Different Reduced State Dimension

This simulation is conducted to evaluate the controller's performance as the number of retained PCA features varies while the cloth size remains fixed. Increasing the number of PCA features increases the computational complexity of the algorithm. Therefore, the trade-off between dimensionality reduction, which reduces computational complexity, and controller's performance, as measured by the steady state

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

Table I: Performance metrics for varying PCA state.

Υ	dim of s	Mean RMSE (m) $\|p-p_d\|$		Residual error (m) $\|(I-U^{(n)}U^{(n)T})(p-p_d)\|$		Total # of parameters	Computation time per iteration (ms)
		Diagonal	Horizontal	Diagonal	Horizontal		
0.89	3	0.1474	0.1629	0.0349	0.0272	132	1.8863
0.93	4	0.1300	0.2086	0.0248	0.0215	170	1.9279
0.96	5	0.1150	0.1753	0.0092	0.0145	210	1.9371
0.976	6	0.1228	0.1692	0.0090	0.0121	252	1.9554
0.98	7	0.1159	0.1685	0.0074	0.0094	296	1.9852
No PCA	27	0.1034	0.1532	0	0	900	3.205

error, is studied. A cloth of size $0.3048 \times 0.3048 m^2$ is used to perform 50 simulation trials for each of the diagonal and horizontal fold desired configurations using different number of PCA features. In each simulation trial, the initial configuration is varied by uniformly sampling the control point's position, and the same initial configuration is used across all PCA feature dimensions. The reduced sets of PCA features are computed from the cloth configuration data matrix for $n = \{3, 4, 5, 6, 7\}$ retained features, corresponding to different levels of explained variance.

The following controller parameters are chosen for all simulations. The local cost is defined using $Q = 1I_{n\times n}$, and $R = 50I_{n\times n}$. The controller parameters are selected as $\gamma_\theta = 0.5, \gamma_c = 1, \gamma_a = 0.001, \gamma_{a2} = 0.1, k_{cl} = 50, c_1 = 3$, and $c_2 = 50$. The lower and upper bounds on the control point height are selected as $a = -0.1$, and $A = 0.15$, respectively.

The results of Table I show that as the PCA state dimension increases, the residual shape error decreases because a large proportion of the variance is explained, thereby improving the expressiveness of the reduced-order shape representation. However, this improvement is not reflected in the mean RMSE, which remains largely unchanged once the dominant modes have been captured. This is expected because the ultimate bound given in (52) depends on several factors including, the NN weights, state dimension, model parameters, and the NN gains. Reducing the PCA dimension decreases the total number of parameters that must be updated online by the controller (Remark 2), thereby decreasing the computation time as show in Table I. The last row shows the controller's performance when it is applied directly to the full-dimensional state, $p \in \mathbb{R}^{27}$, without PCA. The mean RMSE obtained using the reduced-dimensional states is of the same order as that of the full-dimensional state, demonstrating the advantage of PCA in reducing computational cost while maintaining similar controller performance.

D. Simulation 3: Shape Servoing on Varying Object Size

To evaluate the effectiveness of dimensionality reduction for different cloth sizes, one cloth folding simulation trial is conducted for each of four cloth sizes and for each of two desired configurations: folds along the diagonal and horizontal axes. The results demonstrate how the algorithm scales to objects of different sizes when a similar number of PCA feature is retained, i.e., when algorithms with comparable computational complexity are used. A total of $p = \{9, 17, 29, 41\}$ points are sampled symmetrically from the edges and central region of the square cloths with areas of 0.0929, 0.3716, 0.8361 and 1.4864 m^2, respectively. The number of retained PCA features is selected to achieve 95% explained variance, which results in 5 or 6 retained features, corresponding to a total of 210 and 252 total parameters, respectively.

Table II: Performance metrics for varying cloth sizes.

Cloth size (m^2)	dim of p	dim of s	Normalized steady-state error		T_s (s)	
			Diagonal	Horizontal	Diagonal	Horizontal
0.3048^2	9	5	0.3056	0.2336	8.2	10.3
0.6096^2	17	5	0.3059	0.0896	14.1	20.2
0.9144^2	29	6	0.2053	0.1529	42.4	45.2
1.2192^2	41	6	0.2052	0.2139	27.8	25.0

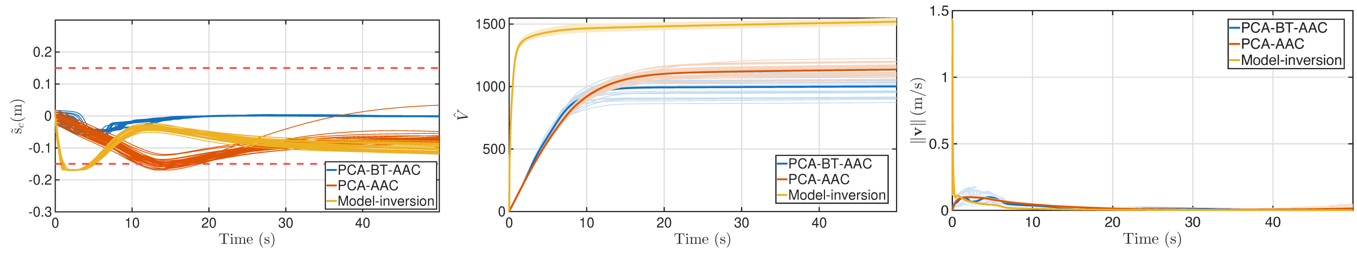
The PCA-BT-AAC controller is implemented using $\mathrm{Q} = 1\mathrm{I}_{n\times n}$, and $\mathrm{R} = 50\mathrm{I}_{n\times n}$, where $n = \dim(s)$. The controller parameters are selected as $\gamma_\theta = 0.7$, $\gamma_a = 0.001$, $\gamma_{a2} = 0.1$, $\gamma_c = 1$, $\mathrm{c}_1 = 3$, $\mathrm{c}_2 = 50$, and the bounds for BT are chosen as $\mathrm{a} = -0.3$ and $\mathrm{A} = 0.5$ to ensure that the control point remains above the table surface. For ICL, the gains are selected as $\mathrm{K}_{cl} = 25$ for the first two cloth sizes, and $\mathrm{K}_{cl} = 50$, and $\mathrm{K}_{cl} = 100$ for other two cloth sizes.

The controller is evaluated for two desired configurations using two performance metrics, steady-state shape error normalized with area of the cloth, and the convergence time T_s. The convergence time is computed as time taken to reduce the error to within 95% of the initial error. The results are reported in Table II and show that, with the reduced feature set, the final shape servoing error remains bounded and is of the same order of magnitude for all cloth sizes. The time required to achieve the desired shape also remains within a reasonable threshold. These results indicate that the algorithm scales to larger object sizes while maintaining similar computational complexity, as indicated by the number of parameters required to implement the algorithm.

E. Simulation 4: Controller Performance Comparison for Shape Servoing

The simulation is conducted to compare the performance of the PCA-BT-AAC controller with that of two other controllers: a model-inversion-based controller, which is designed neither to minimize a performance criterion nor to enforce state constraints, and the PCA-AAC controller,

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045



(a) (b) (c)

Figure 2: Performance comparison for randomly generated initial and desired configuration of PCA-BT-AAC, PCA-AAC, and a model-inversion controllers: (a) Control point height error, (b) Evolution of the value function, (c) End-effector velocity norm

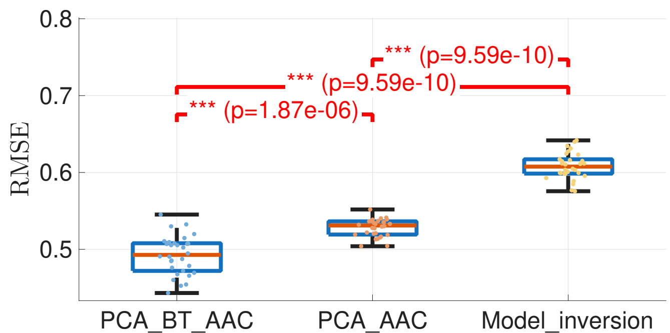


Figure 3: RMSE of (a) PCA-BT-AAC (b) PCA-AAC (C) Model-inversion controller and p-values computed using one-way repeated measures ANOVA, where statistically significant differences ($p < 0.05$) are denoted by ***

which is an optimal controller without state constraints. The controllers are compared in terms of their optimality with respect to a performance criterion and their ability to maintain part of the state within bounds. A $0.6096 \times 0.6096\mathrm{m}^2$ cloth is used to conduct 30 simulations with randomly generated initial configurations and a randomly generated reachable desired configuration. To generate a random but feasible desired configuration, a sequence of user-defined velocities is applied to the grasped point from a randomly chosen initial configuration. The grasped point is then further displaced by a vector uniformly sampled from a ball of radius 3cm to generate random initial configurations. For each simulation, the same randomly generated initial position of the grasped point is used for all controllers, resulting in similar initial cloth configurations across the controllers.

The parameters for the PCA-BT-AAC controller are selected as $\mathrm{Q} = 1\mathrm{I}_{6\times 6}$, $\mathrm{R} = 20\mathrm{I}_{6\times 6}$, $\gamma_\theta = 0.5$, $\mathrm{K}_{cl} = 10$, $\gamma_a = 0.001$, $\gamma_{a2} = 0.1$, $\gamma_c = 1$, $\mathrm{c}_1 = 3$, and $\mathrm{c}_2 = 50$. The BT bounds are chosen as $\mathrm{a} = -0.175$ and $\mathrm{A} = 0.25$. For the PCA-AAC controller, the same gains as those used for the PCA-BT-AAC controller are used. For the model-inversion based controller, the gain is selected as $\alpha_p = 0.1$, and the gains for model adaptation using ICL are selected to be the same as those used for the PCA-BT-AAC controller.

The performance comparison of the controllers is shown in Figs. 2a, 2b, 2c, and 3. As shown in Fig. 2a, only the PCA-BT-AAC controller ensures that the control-point position remains within the bounds. Fig. 2c shows the control velocities generated by the three controllers. The controllers with PCA-AAC components exhibit smaller peak control velocities because they minimize the performance criterion that includes control velocities. From Fig. 2b, it is observed that the final value of the value function, computed in the reduced state dimension, is smallest for the PCA-BT-AAC controller compared to the other two controllers. This shows that the PCA-BT-AAC controller is optimal with respect to the selected cost. From the results shown in Fig. 3, it can be observed that the PCA-BT-AAC controller achieves the lowest RMSE, followed by PCA-AAC controller and model-inversion based controller. The statistical significance of the RMSE values across multiple simulations is evaluated using a one-way repeated-measures analysis of variance (ANOVA) test, with the p-values shown in Fig. 3. Since the pairwise p-values are less than 0.05, the ANOVA test indicates that there is a statistically significant difference among the RMSEs of the three controllers.

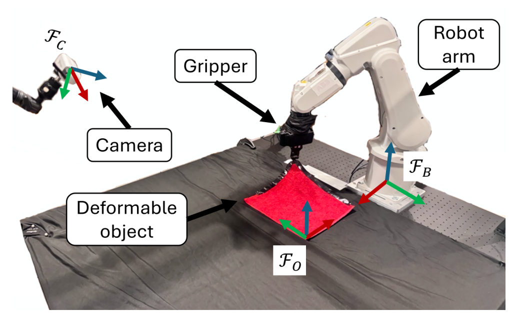


Figure 4: Experimental setup for visual shape servoing. Red denotes X, Green denotes Y and Blue denotes Z axes.

VIII. Experiment

In this section, the experimental results of the PCA-BT-AAC controller applied to a cloth folding task are presented. The experimental setup, shown in Fig. 4, consists of an Intel RealSense D435i camera and an ABB IRB-120

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

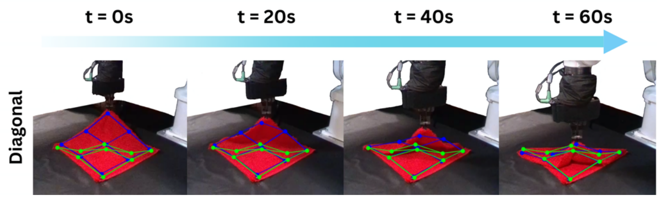


Figure 5: Experimental results of VSS showing initial, intermediate, and final (desired) configurations of the cloth, where green points show the desired configuration and blue points show the current configuration of the cloth

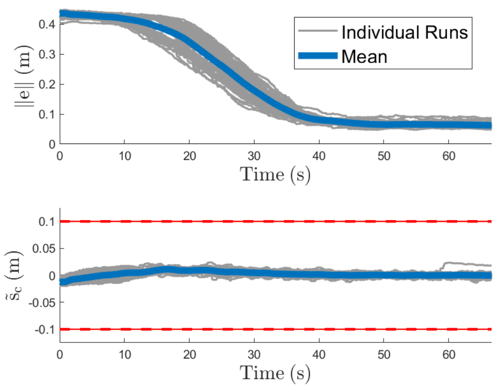


Figure 6: Results for the cloth folding experiment: (a) Shape error norm (b) Control point height error

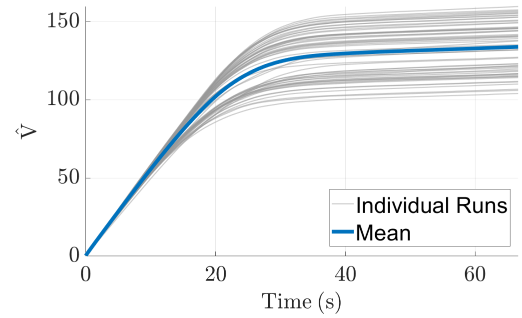


Figure 7: Evolution of the Value function for cloth folding experiment.

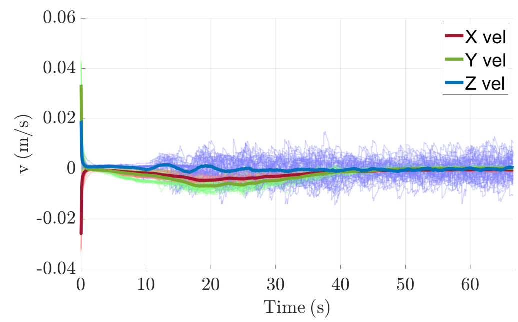


Figure 8: End-effector velocity for cloth folding task.

robot equipped with a Schunk gripper to manipulate the cloth. Let $\mathcal{F}_C$, $\mathcal{F}_B$, and $\mathcal{F}_O$ denote the reference frames attached to the camera, robot base, and object, respectively. A $0.3048 \times 0.3048\text{m}^2$ cloth, placed horizontally on a table, is used as the deformable object. One end of the cloth is fixed, while the other end is firmly grasped by the robot gripper. A shape estimation algorithm called Constrained Deformable Coherent Point Drift (CDCPD) [38] is used to estimate the positions of points on the cloth in $\mathcal{F}_C$, which are then transformed to the $\mathcal{F}_O$ frame using ${}^{O}T_C$. Shape estimation algorithms such as those presented in [39], [40] can also be used for shape tracking purposes. For the experiment, a cloth of the same size as the one in the simulation is used. The reduced feature set computed from the simulation data is used in the experiment. The ICL history stack is initialized using simulation data collected from a cloth with the same dimensions and similar material properties. The evolution of the diagonal cloth-folding task is shown in Fig. 5 with the controller operating at 30 fps. A total of 60 diagonal folding experiments are conducted, each starting from a similar initial configuration and ending at the same desired final configuration. In each experiment, the cloth configuration estimation algorithm introduces noise due to several environmental factors such as lighting conditions, intrinsic and extrinsic calibration errors, and sensor noise. The PCA-BT-AAC controller is implemented using the following parameters: $\text{R} = 50\text{I}_{6\times6}$ and $\text{Q} = 1\text{I}_{6\times6}$, $\gamma_\theta = 0.1$, $\text{K}_{cl} = 25$, $\gamma_a = 0.001$, $\gamma_{a2} = 0.1$, $\gamma_c = 1$, $\text{c}_1 = 3$, and $\text{c}_2 = 50$. The bounds for BT are chosen as $\text{a} = -0.1$, and $\text{A} = 0.1$. A cloth folded along the diagonal is used as the desired configuration. In Fig. 6a, the average 2-norm of the shape error over all experiments and shape error norm for each individual experiment, are shown. The average shape error norm converges close to zero. The control point position remains within the prescribed bounds, as seen from its average position shown in Fig. 6b. The average value function and average control velocities are shown in Figs. 7 and 8, respectively. Due to measurement noise and configuration estimation errors, the average control point position and control velocities converge around zero with small oscillations, as shown in Figs. 6b and 8, respectively.

IX. Conclusion

In this paper, a constrained AAC reinforcement learning controller for VSS is developed. A Fourier-based regression method, along with an ICL-based parameter update law, is used to estimate the deformation Jacobian in real time. A BT method is used to encode constraints on a partial state, thereby guaranteeing that the constraints on the control point position are satisfied. To reduce the model complexity associated with the large state space of the deformable object and improve computational efficiency, the constrained AAC controller is developed using a reduced-dimensional state obtained via PCA. Lyapunov analysis shows that the closed-

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

loop error signals are UUB and that the state constraints are satisfied. The PCA-BT-AAC controller for VSS is evaluated for a cloth folding experiment through simulations with various desired configurations and multiple experiments. The simulation results verify that the PCA-BT-AAC controller achieves comparable error performance in the reduced state while requiring less computation than a controller implemented on a full state representation. In addition, by using a reduced-dimensional representation, the PCA-BT-AAC controller effectively scales to shape servoing tasks involving cloths of various sizes and different desired configurations. Monte Carlo experimental runs demonstrate the effectiveness of the PCA-BT-AAC controller in handling sensor noise and shape estimation errors when implemented on an ABB IRB120 robot equipped with an Intel RealSense camera.

APPENDIX

A. Proof of Lemma 1

Proof. Note that since $\Phi\left(t, e(t_0) = b\left(\tilde{s}(t_0)\right), v_e\right)$ is a complete Caratheodory solution to $\dot{e} = J(s,e)v_e(e,t)$, it is differentiable at almost all $t \in \mathbb{R}_{\geq 0}$. Since b^{-1} is a smooth map, $b^{-1}\left(\Phi\left(t, b\left(\tilde{s}(t_0)\right), v_e\right)\right)$ is also differentiable at almost all $t \in \mathbb{R}_{\geq 0}$. That is,

$$\begin{aligned}&\frac{\mathrm{d}\left(b^{-1}\circ\Phi_i\right)}{\mathrm{d}t}\left(t, b\left(\tilde{s}^0\right), v_e\right)\\&= \left.\frac{\mathrm{d}b^{-1}_{(a_i,A_i)}(y)}{\mathrm{d}y}\right|_{y=\Phi_i(t,b(\tilde{s}^0),v_e)} \frac{\mathrm{d}\Phi_i}{\mathrm{d}t}\left(t, b\left(\tilde{s}^0\right), v_e\right), \quad (57)\end{aligned}$$

for almost all $t \in \mathbb{R}_{\geq 0}$ and all $i = 1, \cdots, n$, where Φ_i denotes the ith component of Φ. As a result,

$$\begin{aligned}&\frac{\mathrm{d}\left(b^{-1}\circ\Phi_i\right)}{\mathrm{d}t}\left(t, b\left(\tilde{s}^0\right), v_e\right)\\&= \frac{\left(J\left(\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right)\right)\right)_i \zeta\left(\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right), t\right)}{B_i\left(\Phi_i\left(t, b\left(\tilde{s}^0\right), v_e\right)\right)}, \quad (58)\end{aligned}$$

for almost all $t \in \mathbb{R}_{\geq 0}$ and all $i = 1, \ldots, n$. By the construction of J, and v,

$$\begin{aligned}\frac{\mathrm{d}\left(b^{-1}\circ\Phi\right)}{\mathrm{d}t}&\left(t, b\left(\tilde{s}^0\right), v_e\right)\\&= J_s\left(s, b^{-1}\circ\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right)\right)\\&\quad v\left(s, b^{-1}\circ\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right), t\right), \quad (59)\end{aligned}$$

for almost all $t \in \mathbb{R}_{\geq 0}$. Clearly $b^{-1}\circ\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right)$ is a Caratheodory solution of (7) on $\mathbb{R}_{\geq 0}$, starting from the initial condition $b^{-1}\left(b\left(\tilde{s}^0\right)\right) = \tilde{s}^0$ under the feedback policy $v(\tilde{s}, t)$. Using the local Lipschitz continuity assumption of J and b inside the barrier region, the solution $\Lambda\left(\cdot, \tilde{s}^0, \xi\right)$ to $\dot{\tilde{s}} = J_s(s)v(s, \tilde{s}, t)$ is complete and $\Lambda\left(t, \tilde{s}^0, v\right) = b^{-1}\left(\Phi\left(t, b\left(\tilde{s}^0\right), v_e\right)\right)$ for all $t \in \mathbb{R}_{\geq 0}$. □

B. Derivation of the PCA Bound

Consider the term in the LHS of (54)

$$\begin{aligned}\mathbb{E}\left[\|U_p(p_c - p_{dc})\|^2\right] =& \mathbb{E}\left[\|U_p p_c\|^2\right] + \mathbb{E}\left[\|U_p p_{dc}\|^2\right]\\&- 2\mathbb{E}[p_c^T U_p p_{dc}] \quad (60)\end{aligned}$$

where $U_p = I - U^{(n)}U^{(n)T}$. The following expressions can be derived

$$\mathbb{E}\left[\|U_p p_c\|^2\right] = \sum_{i=n+1}^{k} \sigma_i \quad (61)$$

where $p_c \sim \mathcal{D}$, i.e., p_c is from the distribution of the data collected for PCA. The bound is known as the residual bound of the PCA [41]. Since p_{dc} is a deterministic vector

$$\mathbb{E}\left[\|U_p p_{dc}\|^2\right] = \|U_p p_{dc}\|^2 = \epsilon_{pd} \quad (62)$$

The bound on the cross term of (60) can be derived as

$$\begin{aligned}\mathbb{E}[p_c^T U_p p_{dc}] &= \mathbb{E}[tr\left(U_p p_{dc} p_c^T\right)] = tr\left(U_p \mathbb{E}[\left(p_{dc} p_c^T\right)]\right)\\&= tr\left(U_p p_{dc} \mathbb{E}[\left(p_c^T\right)]\right) \quad (63)\end{aligned}$$

where the fact that p_{dc} is deterministic is used. Since p_c is sampled from the mean-centered distribution of PCA as shown in Section IV, $\mathbb{E}\left[p_c\right] = 0$ leading to

$$\mathbb{E}[p_c^T U_p p_{dc}] = 0 \quad (64)$$

REFERENCES

[1] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[2] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-D image contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2018.

[3] F. Xu, Y. Zhang, J. Sun, and H. Wang, "Adaptive visual servoing shape control of a soft robot manipulator using bézier curve features," *IEEE/ASME Transactions on Mechatronics*, vol. 28, pp. 945–955, 2022.

[4] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "As-rigid-as-possible shape servoing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3898–3905, 2022.

[5] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[6] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.

[7] K. G. Vamvoudakis, P. J. Antsaklis, W. E. Dixon, J. P. Hespanha, F. L. Lewis, H. Modares, and B. Kiumarsi, "Autonomy and machine intelligence in complex systems: A tutorial," in *2015 American Control Conference (ACC)*, 2015, pp. 5062–5079.

[8] I. Salehi, G. Rotithor, D. Trombetta, and A. P. Dani, "Safe tracking control of an uncertain euler-lagrange system with full-state constraints using barrier functions," in *IEEE Conference on Decision and Control*, 2020, pp. 3310–3315.

[9] S. M. N. Mahmud, S. A. Nivison, Z. I. Bell, and R. Kamalapurkar, "Safe model-based reinforcement learning for systems with parametric uncertainties," *Frontiers in Robotics and AI*, vol. 8, 2021.

[10] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "FEM-based deformation control for dexterous manipulation of 3D soft objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4007–4013.

[11] A. Koessler, N. R. Filella, B.-C. Bouzgarrou, L. Lequièvre, and J.-A. C. Ramon, "An efficient approach to closed-loop shape control of deformable objects using finite element models," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 1637–1643.

[12] J. Sanchez, K. Mohy El Dine, J. A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Blind manipulation of deformable objects based on force sensing and finite element modeling," *Frontiers in Robotics and AI*, vol. 7, p. 73, 2020.

This article has been accepted for publication in IEEE Transactions on Control Systems Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCST.2026.3721045

[13] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2016–2026, 2020.

[14] L. Smolentsev, A. Krupa, and F. Chaumette, "Shape visual servoing of a tether cable from parabolic features," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 734–740.

[15] H. Wang, B. Yang, J. Wang, X. Liang, W. Chen, and Y.-H. Liu, "Adaptive visual servoing of contour features," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 811–822, 2018.

[16] R. Lagneau, A. Krupa, and M. Marchal, "Active deformation through visual servoing of soft objects," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 8978–8984.

[17] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning visual shape control of novel 3d deformable objects from partial-view point clouds," in *International Conference on Robotics and Automation*, 2022, pp. 8274–8281.

[18] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D deformable object manipulation using deep neural networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, 2019.

[19] V. R. Guthikonda and A. P. Dani, "Shape servoing of deformable objects using model estimation and barrier lyapunov function," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2024.

[20] V. Guthikonda, G. Rotithor, and A. P. Dani, "Shape servoing of deformable objects using adaptive deformation model estimation," in *IFAC World Congress*, 2023, pp. 10 793–10 798.

[21] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 12, pp. 1775–1787, 2019.

[22] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[23] V. H. Giraud, M. Padrin, M. Shetab-Bushehri, C. Bouzgarrou, Y. Mezouar, and E. Ozgur, "Optimal shape servoing with task-focused convergence constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 2197–2202.

[24] A. Magassouba, M. Aranda, L. Lequièvre, J. A. C. Ram, R. I. Rodriguez, Y. Mezouar *et al.*, "Multi actor-critic DDPG for robot action space decomposition: A framework to control large 3d deformation of soft linear objects," *IEEE Robotics and Automation Letters*, 2023.

[25] A. P. Dani and S. Bhasin, "Reinforcement learning for image-based visual servo control," in *IEEE Conference on Decision and Control*, 2023, pp. 4358–4363.

[26] ——, "Adaptive actor-critic based optimal regulation for drift-free nonlinear systems," *IEEE Open Journal of Control Systems*, vol. 4, pp. 117–129, 2025.

[27] A. Astolff, C. L. Beck, D. Bhattacharjee, Y. Kawano, A. Moreschini, H. Sandberg, and J. M. Scherpen, "Forty plus years of model reduction and still learning," in *IEEE Conference on Decision and Control*, 2024, pp. 4480–4493.

[28] J. Zhu, D. Navarro-Alarcon, R. Passama, and A. Cherubini, "Vision-based manipulation of deformable and rigid objects using subspace projections of 2d contours," *Robotics and Autonomous Systems*, vol. 142, p. 103798, 2021.

[29] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.

[30] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[31] V. S. Donge, B. Lian, F. L. Lewis, and A. Davoudi, "Data-efficient reinforcement learning for complex nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 54, no. 3, pp. 1391–1402, 2024.

[32] ——, "Accelerated reinforcement learning via dynamic mode decomposition," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 4, pp. 2022–2034, 2023.

[33] M. Netto, Y. Susuki, V. Krishnan, and Y. Zhang, "On analytical construction of observable functions in extended dynamic mode decomposition for nonlinear estimation and prediction," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 1868–1873, 2021.

[34] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2022.

[35] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[36] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[37] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1976, vol. 3.

[38] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 6443–6450.

[39] G. Yao, R. Saltus, and A. P. Dani, "Shape estimation for elongated deformable object using b-spline chained multiple random matrices model," *International Journal of Intelligent Robotics and Applications*, vol. 4, no. 4, pp. 429–440, 2020.

[40] G. Rotithor and A. P. Dani, "Deep interacting multiple model filtering," in *2022 American Control Conference (ACC)*, 2022, pp. 958–963.

[41] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, 2006, vol. 4, no. 4.

Vrithik R. Guthikonda (Student Member, IEEE) received the B.S. degree in electrical engineering in 2021 from the University of Connecticut, Storrs, CT, USA, where he is currently working toward the Ph.D. degree in Electrical and Computer Engineering. His research interests involve deformable object manipulation using adaptive control methods.

Rounak Bhattacharya (Student Member, IEEE) received the B.Tech degree in electronics and communication engineering in 2018 from Heritage Institute of Technology, Kolkata, India. He is currently working towards the Ph.D. degree in Electrical and Computer Engineering from the University of Connecticut, Storrs, CT, USA. His research interests primarily involve human-robot collaboration and safety and control of time delay systems.

Ashwin P. Dani (M'11–SM'18) received the M.S. degree and the Ph.D. degree from the University of Florida (UF), Gainesville, FL. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Connecticut, Storrs. He was a post-doctoral research associate at the University of Illinois at Urbana-Champaign. His research interests are in the areas of estimation, learning for control, vision-based estimation and control, and human-robot collaboration. He served as an Associate Editor of IEEE Transactions on Mechatronics and as a member of the Conference Editorial Board for IEEE Control System Society. He is currently an Associate Editor of IEEE Transactions on Automatic Control. He is a co-recipient of the 2020 IFAC Applications Paper Award Finalist, 2016 FUSION student paper award - 1st runner up, 2015 ASME Dynamics, Systems and Controls Conference Robotics paper award and the 2012 Technology Innovator Award from UF.