Shape Servoing of Deformable Objects Using Model Estimation and Barrier Lyapunov Function

Vrithik Raj Guthikonda, *Student Member, IEEE*, and Ashwin P. Dani, *Senior Member, IEEE*

***Abstract*—An adaptive shape servoing control method is presented in this article to manipulate a deformable object into a desired shape in 3-D. A finite-point-based representation of the deformable object is used and the deformation Jacobian matrix is approximated using Fourier series basis functions. The unknown parameters of the deformation Jacobian are learned by using the velocity applied to a control point on the object and corresponding change of positions of the points describing the entire object. An integral concurrent learning (ICL)-based parameter update law is designed along with a constrained controller to satisfy the state constraints on the motion of the control point using Barrier Lyapunov function analysis. ICL-based parameter update law uses data history of velocity and corresponding positions of the points along with their current values. An efficient algorithm to update the history stack using singular value maximization is proposed based on the structure of the regressor matrix. Simulations using a physical simulator and experiments using a robot platform are performed to validate the performance of the proposed controller on two different deformable objects.**

***Index Terms*—Barrier Lyapunov function (BLF)-based constrained control, deformable objects modeling, integral concurrent learning (ICL) parameter estimation, shape servoing.**

I. Introduction

DEFORMABLE objects are commonly found in robotic applications, e.g., in industrial settings, medical robotics, space robotics, and agricultural robotics [1], [2], [3], [4]. They are categorized based on the significant dimension into 1-D deformable objects called deformable linear objects, 2-D deformable objects, such as cloth like objects, and volumetric objects [5]. Manipulating deformable objects to a desired shape measured by an external camera sensor is termed as Shape Servoing [6], [7]. Shape servoing is difficult due to many technical challenges present in deformation modeling, perception of deformable object's shape and design of control methods for real-time manipulation of the shape [5]. Shape servoing is useful for many tasks, such as robotic automation of food processing, manipulation of soft tissues, folding/unfolding cloths, and manipulating and routing cables. There are several approaches developed in the literature for the shape servoing problem, but physical constraint satisfaction while the deformation is in process is not considered. For example, while folding a cloth placed on a table all the points on the cloth should remain above the table surface and transient motions that will take points below the table surface should be avoided. This article presents a shape servoing controller to achieve the objectives of shape matching and physical constraint satisfaction, where the deformation model parameters are updated in real-time as new data becomes available. A Barrier Lyapunov function (BLF) approach [8] is used to design a controller that satisfies the constraints. The deformation model parameters are estimated using an integral concurrent learning (ICL) parameter estimation law developed in [9].

Manuscript received 18 October 2023; revised 6 February 2024; accepted 12 March 2024. This work was supported in part by Space Technology Research Institutes under Grant 80NSSC19K1076 from NASA Space Technology Research Grants Program and in part by National Science Foundation under Grant SMA-2134367. Recommended by Technical Editor D. Shi and Senior Editor Y.-J. Pan. *(Corresponding author: Ashwin P. Dani.)*

The authors are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: vrithik.guthikonda@uconn.edu; ashwin.dani@uconn.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2024.3382590.

Digital Object Identifier 10.1109/TMECH.2024.3382590

A. Object Representation

For shape control of the deformable objects, point-based representation is commonly used, where the object is represented as a collection of points on the object [10]. The shape servoing task then becomes that of matching the current positions of the points on the deformable object to the positions of the point at the desired configuration. Object shape described by the contours measured from the images have also been studied in literature, for example, [6], [11], [12]. The contours are represented using a truncated Fourier series in [6], Bezier curve in [12], B-splines in [13], nonuniform rational B-spline (NURBS) in [11], and using geometric strain-based representation in [14]. These methods require contour estimation in the images, which is often a challenging task. A lattice-based representation is utilized for volumetric objects in [15], which uses geometric constraints to bound the lattice to the object in 3-D. Other representations of the flexible object shape are also developed in terms of extended objects using a collection of ellipses or image moment features, see [16], [17], [18].

B. Deformation Modeling and Control

A deformation model is used to design feedback controllers for servoing the robot such that the desired shape of the deformable object is achieved. The deformation model provides a relationship between the change in the features' positions as a

function of the external velocity input provided to a point (termed as a control point) or in a small region of the deformable object. Estimating a global deformation model of the entire shape is a challenging problem due to high dimensional representation of deformable objects. There are different methods used to represent the object deformation: physics-based methods, geometric methods, and data driven methods.

Physics-based methods use physical/mechanical properties and motion models of the objects to predict the object's shape by solving the equations of motion. Finite element method (FEM) models are widely used mechanical models [19], [20], [21]. An open-loop control with FEM simulations of the soft object is presented in [19]. A force feedback controller with FEM model simulation is used in [21] for shape servoing of a volumetric object. A computationally efficient approach of FEM simulation is developed in [20] by using partitions of the mesh nodes for shape servoing task. In [22], an image-based visual servo (IBVS) controller is developed to cut a deformable object along a desired path based on deformable object's mass-spring-damper shape modeling. Although the physical models provide accurate shape estimation and prediction, which can be used for shape servoing, they often require knowledge of the object's properties and are computationally costly. Since the object's properties are required to be known, the shape servoing solutions are tailored to the specific object.

Geometric methods estimate the shape and deformations using data with certain known geometric properties of the object, such as length or area. In [23], a diminishing rigidity approximation is made to compute the Jacobian matrix, which is used to drive the deformable object to a desired configuration. In [11], the relationship between the control points of Bezier curves and NURBS curve and the velocity is computed by adaptive estimation of the parameters in an adaptive controller loop. In [24], a polynomial representation of the rope is used, where only the knowledge of the length of the rope is required and a polynomial visual servoing control law is designed. In [25], a framework called as-rigid-as-possible (ARAP) shape servoing is presented based on surface modeling of [26], where the deformation Jacobian is estimated using only the current data points.

The data-driven approaches use sensor data and a parametric representation of the object deformation to estimate the deformations in real-time or from offline stored data for initialization of the parameters. In [6], a local linear deformation model is learned by solving a least squares type problem for a given object configuration and deformation model recalibration is achieved by iterative learning at various different configurations. A multilayer neural network is used in [27] to relate the end-effector's movement to the object's deformation with an online learning process to improve model accuracy. In [28], the deformation Jacobian is estimated based on a least-squares minimization, which is further updated based on a user specified confidence threshold. Estimating the deformation Jacobian is a common approach to address this problem, however, the control of such systems is difficult due to the limited number of inputs used to control the high dimensional object [29]. A tangent space mapping algorithm is presented in [30] for the manipulation of deformable objects. In [31], a controller is designed to achieve a desired final shape of the object along a specific deformation path. Deep neural network (DNN)-based learning approaches are presented in [27] and [32], which uses DNN to learn the deformation for visual servo control.

C. Contributions

In this article, a point-based representation of the deformable object is used. It is assumed that the 3-D locations of the points on the object are measured by an RGB-D camera sensor. The deformation Jacobian is approximated using a Fourier series-based regression [6], [33]. Since the parameters of the Jacobian are unknown, an adaptive controller and parameter update law are developed, which estimates the parameters of the Jacobian. A constrained controller is designed based on the BLF analysis, which guarantees the satisfaction of the constraints on the control point's position. The parameter update law uses ICL, proposed in [9], which guarantees the estimation of the ideal parameters along with the convergence of the shape estimation error. A local asymptotic stability of the proposed BLF-ICL shape servoing controller can be concluded from the Lyapunov analysis. The performance of the proposed controller is tested by using simulations and experiments to regulate the shape of a cloth and rope from its current shape to a desired shape. Compared to the literature in [7], [11], and [12] and our prior work in [34], this article develops a BLF-based adaptive controller, where the ICL parameter update law is also modified to account for BLF terms to satisfy state constraints on the control point. The contributions of this article are as follows.

1) An adaptive deformation Jacobian model is proposed to compute the deformation matrix in real-time without requiring any prior knowledge of the deformation Jacobian parameters of the object.
2) A new BLF-based adaptive control design with ICL parameter update law is presented, which satisfies the shape regulation and constraint satisfaction objectives. The constraint satisfaction is useful in practice when the deformations are subject to physical limitation, e.g., a cloth placed on a table cannot be folded by moving the control point inside the table surface.
3) The stability and convergence analysis of the BLF with ICL-based update law is provided, which shows constraint satisfaction, convergence of the shape regulation error, and parameter convergence to ideal parameters.
4) The performance of the developed controller is tested using simulation use cases of cloth folding to a desired shape and also using real experiments of a robot arm performing cloth folding and rope displacement tasks. Simulation studies are conducted to test the performance of the BLF-ICL controller due to the variations in the size of history stack, in the number of points used for representing the object, and in the material properties of the cloth. The BLF-ICL controller shows improved performance in terms of constraint satisfaction and shape regulation error compared to three other controllers including a state-of-the-art controller in [35].

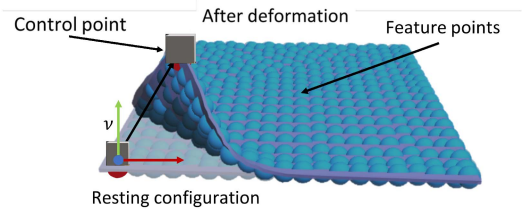

Fig. 1. Deformable object representation for shape servoing.

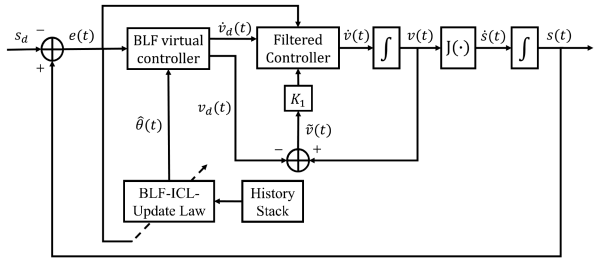

Fig. 2. Block diagram of deformable object shape servoing using adaptive deformation model estimation.

5) The simulation studies and experiments conducted on objects with different material properties and the object types show the applicability of the controller to different types of deformable objects.

The rest of this article is organized as follows. Background of BLF is provided in Section II. The problem formulation and shape representation are discussed in Section III. Details of the BLF-ICL control design and stability analysis are presented in Sections IV and V, followed by singular value maximization for ICL parameter update law in Section VI. The controller validation is provided in Sections VII and VIII using simulation and experiments. Finally, Section IX concludes this article

II. Barrier Lyapunov Functions

In this section, a BLF Lemma is presented from [8], which shows utility of the Lyapunov function and BLF for achieving constraints on the state or part of the state.

A. Preliminaries of BLF Control Design

The objective of BLF controller is to design a controller that achieves a set point regulation task while certain states remain within a specified bound. Let $\eta_c \in \mathbb{R}$, be a state that should satisfy certain constraints defined as $\|\eta_c\| < \eta_b \ \forall t \geq 0$, where $\eta_b \in \mathbb{R}^+$ and $\eta_e \in \mathbb{R}^n$ is a state on which there are no constraints.

Lemma 1: For any positive constant η_b, let $\Phi := \eta_c \in \mathbb{R} : -\eta_b < \eta_c < \eta_b$ and $\Psi := \mathbb{R}^n \times \Phi \subset \mathbb{R}^{n+1}$ be an open set. Consider the system $\dot{\eta} = h(\eta, t)$, where $\eta := [\eta_e, \eta_c]^T \in \Psi$, and $h : \Psi \times \mathbb{R}^+ \subset \mathbb{R}^{n+1}$ is a function piecewise continuous in t and locally Lipschitz in η, uniformly in t. Let there be continuously differentiable positive definite functions $V_1 : \Phi \to \mathbb{R}^+$ with the property that $V_1(\eta_c) \to \infty$ as $\eta_c \to \pm\eta_b$ and $V_2 : \mathbb{R}^n \to \mathbb{R}^+$. Let a combined continuously differentiable positive function $V = V_1(\eta_c) + V_2(\eta_e)$. If the condition holds $\dot{V} = \frac{\partial V}{\partial \eta} \leq 0$, inside set $\eta_c \in \Phi$, then for $\eta_c(0) \in \Phi$, $\eta_c(t) \in \Phi \ \forall t \in [0, \infty)$.

III. Shape Representation and Problem Definition

Let us define a deformable object, such as a cloth using feature points, which are used to describe the shape and position of the cloth in 3-D; and control points, which are used to control the shape of the cloth and carry out the shape servoing task. As shown in Fig. 1, a resting configuration represents the initial state of the soft object before any input is applied. In this article, object deformation is defined as the change in feature points on the soft object as a result of an applied velocity input to the control point. A visual illustration of the deformable object is shown in Fig. 1 and a block diagram of the developed shape servo control system is shown in Fig. 2. The following assumptions are made to capture the deformation behavior of the objects using the model presented in the following section.

Assumption 1: The control points are selected such that the desired configuration is reachable.

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

Assumption 3: The soft object is rigidly grasped at all time with no loose contact during the shape servoing task.

A. System Dynamics

Consider the following system representing the evolution of the points on the deformable object as a function of the velocity applied to a point on the object. The system model can be written as

$$\dot{s} = J(s)v \tag{1}$$

where $s(t) = [s_1^T, \ldots, s_n^T]^T \in \mathbb{R}^{3n}$ denotes the set of n feature points, which describes the deformable object with $s_i(t) = [x_i, y_i, z_i]^T$ denoting the ith feature point's position with respect to a fixed frame, and $v(t) \in \mathbb{R}^m$ represents the input velocity vector. The deformation Jacobian matrix $J(s) \in \mathbb{R}^{3n \times m}$ is expressed by $J(s) = [\theta_1 Y_1(s), \theta_2 Y_2(s), \ldots, \theta_{3n} Y_{3n}(s)]^T$, where $Y_i(s) \in \mathbb{R}^{d \times 1}$ represents a vector of basis functions and $\theta_i \in \mathbb{R}^{m \times d}$ represents its corresponding unknown parameter matrix. Since deriving the Jacobian matrix analytically is challenging, a Fourier series-based regression is proposed to approximate the deformation Jacobian matrix, where each basis function vector $Y_i(s)$ is constructed as follows $Y_i(s) = \frac{1}{\sqrt{d}}\left[\cos\left(\frac{\omega_1^T s}{\tau}\right), \sin\left(\frac{\omega_2^T s}{\tau}\right), \ldots, \cos\left(\frac{\omega_{d-1}^T s}{\tau}\right), \sin\left(\frac{\omega_d^T s}{\tau}\right)\right]^T$ the vector $w_k \in \mathbb{R}^{3n}$ denotes a vector of frequencies with $k = \{1, \ldots, d\}$, d is number of basis functions, and $\tau \in \mathbb{R}$ is a constant selected based on the size of the basis vector. To facilitate the control development, (1) is rewritten as

$$\dot{s} = \bar{Y}^T(s)\bar{V}^T(v)\bar{\theta}. \tag{2}$$

The parameter vector $\bar{\theta} \in \mathbb{R}^{3nmd}$ is constructed by stacking the columns of θ_i as follows $\bar{\theta} = [\theta_{11}^T, \theta_{12}^T, \ldots, \theta_{21}^T \ldots \theta_{3nd}^T]^T$, where the vector θ_{ij} represents the ith matrix's jth column. The regressor matrix $\bar{Y}(s) \in \mathbb{R}^{3nd \times 3n}$ is created such that

$\bar{Y}(s) = \text{blkdiag}[Y_1(s), \ldots, Y_{3n}(s)]$ and $\bar{V}(v) \in \mathbb{R}^{3nmd \times 3nd}$ is constructed using the control input vector $v(t)$ as follows:

$$\bar{V}(v) = \begin{bmatrix} v & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & v \end{bmatrix}. \tag{3}$$

Remark 1: The tradeoff between the model accuracy and computational requirements can be adjusted using the parameter d, see [33].

IV. Shape Servoing Control Development

A. Controller Objective

The control objective is to regulate the shape of the deformable object given the current measurements of a collection of points $s_i(t)$, $\forall i = \{1, \ldots, n\}$ on the object and the desired collection of points representing the desired shape of the object. In addition, the control point must remain within a user defined set to ensure that the state constraint is not violated. Such constraints may arise from physical constraints or performance constraints, for example, the end-effector holding the control point may collide with a table while a cloth folding task is being performed. To keep the states within certain bounds, a BLF-based controller is designed, which ensures that the control point remains within a predefined safety region.

B. Error Definition

For the control design, the shape regulation error $e(t) \in \mathbb{R}^{3n}$ is defined as

$$e(t) \triangleq s(t) - s_d \tag{4}$$

where $s_d \in \mathbb{R}^{3n}$ are the feature points on the deformable object at the desired configuration. The error for the constrained state $z_e(t) \in \mathbb{R}$ is defined as

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

where $J_z \in \mathbb{R}^{1\times 3n}$ is an operator that selects the elements corresponding to the control point position. A specific example of J_z is given in the experiment section. The parameter estimation error $\tilde{\bar{\theta}} \in \mathbb{R}^{3nmd}$ is defined as

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

where $\hat{\bar{\theta}} \in \mathbb{R}^{3nmd}$ is the estimated parameter vector. Let $v_d(t) \in \mathbb{R}^m$ be a newly defined virtual velocity signal. The velocity error, $\tilde{v}(t) \in \mathbb{R}^m$ can then be defined as

$$\tilde{v}(t) \triangleq v(t) - v_d(t). \tag{7}$$

C. Open-Loop Dynamics

First, the error dynamics is derived. Taking the time derivative of (4) and using (2) yields

$$\dot{e}(t) = \bar{Y}^T(s)\bar{V}^T(v)\tilde{\bar{\theta}} + \hat{J}(s)\tilde{v} + \hat{J}(s)v_d \tag{8}$$

where $\hat{J}(s) \in \mathbb{R}^{3n\times m}$ is an estimate of the deformation Jacobian matrix. The last equation is obtained by adding and subtracting $\hat{J}(s)v$ and using $\hat{J}(s)v = \bar{Y}^T(s)\bar{V}^T(v)\hat{\bar{\theta}}$. Similarly, the error dynamics for BLF error is derived. Taking the time derivative of (5) and substituting (2) yields

$$\dot{z}_e = J_z \left\{ \bar{Y}^T(s)\bar{V}^T(v)\tilde{\bar{\theta}} + \hat{J}(s)\tilde{v} + \hat{J}(s)v_d \right\}. \tag{9}$$

D. Controller Design

To achieve the desired regulation and constraint satisfaction objective, a virtual velocity controller is designed as

$$v_d(t) = -\alpha_v \hat{J}(s)^\dagger \left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2} \right) \tag{10}$$

where $\hat{J}(s)^\dagger = (\hat{J}^T\hat{J})^{-1}\hat{J}^T$ denotes the Moore–Penrose pseudoinverse of $\hat{J}(s)$, $\alpha_v \in \mathbb{R}^{m\times m}$ is a constant positive-definite diagonal gain matrix, $K_b \in \mathbb{R}$ is a control gain for BLF terms, and $k_e \in \mathbb{R}$ is the bound on z_e. Due to the presence of BLF term in (10), the velocities could reach very high values when the states are very close to the boundary. To avoid such undesired effects, inspired by backstepping control design [36], [37], the velocity controller is designed as $\dot{v}(t)$ according to the subsequent Lyapunov analysis

$$\dot{v}(t) = \dot{v}_d - K_1\tilde{v} - \hat{J}(s)^T \left(e + \frac{J_z^T z_e}{k_e^2 - z_e^2} \right) \tag{11}$$

where $K_1 \in \mathbb{R}$ is the velocity error gain, which using (7) yields the following error dynamics:

$$\dot{\tilde{v}}(t) = -K_1\tilde{v} - \hat{J}(s)^T \left(e + \frac{J_z^T z_e}{k_e^2 - z_e^2} \right). \tag{12}$$

By substituting (10) into (8), the closed-loop error dynamics can be written as

$$\dot{e}(t) = \bar{Y}^T(s)\bar{V}^T(v)\tilde{\bar{\theta}} + \hat{J}(s)\tilde{v} - \alpha_v P \left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2} \right) \tag{13}$$

where $P = \hat{J}\hat{J}^\dagger \in \mathbb{R}^{3n\times 3n}$. The Jacobian $\hat{J}$ is estimated by designing an update law for the parameters $\hat{\bar{\theta}}$ using ICL. To simplify the notations, let $Q = \bar{Y}^T\bar{V}^T \in \mathbb{R}^{3n\times 3nmd}$. The ICL-based parameter update law is designed as

$$\dot{\hat{\bar{\theta}}} = \Gamma_\theta^{-1} \left\{ Q^T e + \frac{Q^T J_z^T z_e}{k_e^2 - z_e^2} + K_{\text{cl}} \sum_{i=1}^{N} \bar{Q}_i^T (s(t_i) - s(t_i - \triangle t) - \bar{Q}_i \hat{\bar{\theta}}) \right\} \tag{14}$$

where $\Gamma_\theta^{-1} \in \mathbb{R}^{3nmd\times 3nmd}$ and $K_{\text{cl}} \in \mathbb{R}^{3nmd\times 3nmd}$ are constant, positive definite diagonal gain matrices and $\Delta t \in \mathbb{R}$ is a positive constant representing the size of the window of integration, and $\bar{Q}(t)$ is the integral of Q defined as

$$\bar{Q}(t) = \int_{\max\{t-\triangle t, 0\}}^{t} Q(s(\tau), v(\tau))d\tau. \tag{15}$$

Taking the integral on both sides of (2) and substituting for Q, results in $\int_{t-\triangle t}^{t} \dot{s}(\tau)d\tau = \int_{t-\triangle t}^{t} Q(s, v)\bar{\theta}d\tau$. Using the fact that $\bar{\theta}$ is a constant, result in (15) and the fundamental theorem of

calculus yields

$$s(t) - s(t - \triangle t) = \bar{Q}(t)\bar{\theta}. \tag{16}$$

Since $\bar{\theta}$ is a constant, it can be taken out of the integral. Substituting (16) in (14) results in

$$\dot{\hat{\tilde{\theta}}} = \Gamma_\theta^{-1}\left\{Q^T e + \frac{Q^T J_z^T z_e}{k_e^2 - z_e^2} + K_{\text{cl}} R\tilde{\bar{\theta}}\right\} \tag{17}$$

where $R = \sum_{i=1}^{N} \bar{Q}_i^T \bar{Q}_i \in \mathbb{R}^{3nmd \times 3nmd}$. Taking time derivative of (6), the parameter estimation error dynamics can be written as

$$\dot{\tilde{\bar{\theta}}} = -\Gamma_\theta^{-1}\left\{Q^T e + \frac{Q^T J_z^T z_e}{k_e^2 - z_e^2} + K_{\text{cl}} R\tilde{\bar{\theta}}\right\}. \tag{18}$$

To implement the update law in (14), data collected from the system is stored in a history stack, denoted by $\{s(t_j), s(t_j - \Delta t), \bar{Q}_i(t_j)\}_{j=1}^{N}$, which is recorded at the increasing time sequence $\{t_j\}_{j=1}^{N}$. The closed-loop dynamics of the constrained state error can be derived by substituting (10) into (9)

$$\dot{z}_e = J_z\left\{-\alpha_v P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) + Q\tilde{\bar{\theta}} + \hat{J}(s)\tilde{v}\right\}. \tag{19}$$

V. Stability Analysis

To facilitate the analysis of the controller and parameter estimation law, the following assumption about the history stack is made.

Assumption 4: The system in (1) is sufficiently exciting over a finite duration of time, which implies that $\exists \delta > 0$, $\exists T > \Delta t : \forall t \geq T$, $\lambda_{\min}\{R\} > \delta$, where $\lambda_{\min}$ denotes the minimum eigenvalue of the matrix.

Remark 2: Assumption 4 is weaker than the typical persistence of excitation condition required in traditional adaptive control, see [9].

Remark 3: History stack is referred to as full rank when Assumption 4 is satisfied and it is not full rank when the Assumption is not satisfied Assumption 4.

Assumption 5: The matrix $\hat{J}$ is full column rank that is $\text{rank}(\hat{J}) = m$ and for an appropriate $\epsilon > 0$ if $e \in \mathcal{B}_{3n}(0, \epsilon)$, the feature error $e \notin \text{Null}(\hat{J}^T)$.

Remark 4: Assumption 5 requires the construction of $\hat{J}$ using suitable basis functions and parameter values $\hat{\bar{\theta}}$ so that the full column rank condition is satisfied. The second condition requires that the feature motion is realizable and no local minima exists in the region defined by $\mathcal{B}_{3n}(0, \epsilon)$. This condition can be achieved by shrinking ϵ appropriately as outlined in [38].

Two theorems are presented for the stability analysis of the controller and parameter estimation scheme for the cases when Assumption 4 is not satisfied and is satisfied.

Theorem 1: If Assumption 5 is satisfied and Assumption 4 is not satisfied, then for all $t \in [0, T)$, the shape regulation, parameter estimation, and the constrained state errors generated by the closed-loop dynamics in (13), (12), (18), and (19) remain bounded and the state constraint $\|z_e(t)\| < k_e$ are satisfied $\forall t > 0$ given that the initial conditions satisfy, $\|z_e(0)\| < k_e$.

Proof: Let $\Phi = \{z_e | \|z_e\| < k_e\} \subset \mathbb{R}$. To analyze the stability of the constrained state, the following symmetric BLF candidate is selected as $V_1 : \Phi \to \mathbb{R}^+$:

$$V_1 = \frac{1}{2}\log\frac{k_e^2}{k_e^2 - z_e^2}. \tag{20}$$

Since V_1 is positive definite and continuous in Φ, it is a valid Lyapunov function candidate. The time derivative of V_1 in (20) after utilizing (19) is given by

$$\dot{V}_1 = \frac{z_e^T J_z}{k_e^2 - z_e^2}\left\{-\alpha_v P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) + Q\tilde{\bar{\theta}} + \hat{J}(s)\tilde{v}\right\}. \tag{21}$$

Let $\zeta(t) = [z_e(t), e(t), \tilde{\bar{\theta}}(t), \tilde{v}(t)]^T \in \mathcal{D}$, where $\mathcal{D} \subset \Phi \times \mathbb{R}^{3n} \times \mathbb{R}^{3nmd} \times \mathbb{R}^m$. Now, consider a combined continuously differentiable Lyapunov function candidate, $V(\zeta) : \mathcal{D} \to \mathbb{R}^+$

$$V(\zeta) = V_1 + \frac{1}{2}e^T e + \frac{1}{2}\tilde{\bar{\theta}}^T \Gamma_\theta \tilde{\bar{\theta}} + \frac{1}{2}\tilde{v}^T \tilde{v}. \tag{22}$$

Taking the time derivative of (22) and substituting (18), (13), (19), and (21) yields

$$\begin{aligned}\dot{V} = &\frac{z_e^T J_z}{k_e^2 - z_e^2}\left\{-\alpha_v P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) + Q\tilde{\bar{\theta}} + \hat{J}\tilde{v}\right\} \\ &+ e^T\left\{Q\tilde{\bar{\theta}} + \hat{J}\tilde{v} - \alpha_v P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right)\right\} \\ &- \tilde{\bar{\theta}}^T\left\{Q^T e + \frac{Q^T J_z^T z_e}{k_e^2 - z_e^2} + K_{\text{cl}} R\tilde{\bar{\theta}}\right\} + \tilde{v}^T\{\dot{v} - \dot{v}_d\}.\end{aligned} \tag{23}$$

After simplifications the expression of $\dot{V}(\zeta)$ is reduced to

$$\begin{aligned}\dot{V} = &-\alpha_v\left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T\right) P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) + \tilde{v}^T \dot{v} \\ &+ \left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T\right)\hat{J}\tilde{v} - \tilde{\bar{\theta}}^T K_{\text{cl}} R\tilde{\bar{\theta}} - \tilde{v}^T \dot{v}_d.\end{aligned} \tag{24}$$

Substituting the controller from (11) and simplifying terms results in

$$\begin{aligned}\dot{V} = &-\alpha_v\left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T\right) P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) \\ &- \tilde{\bar{\theta}}^T K_{\text{cl}} R\tilde{\bar{\theta}} - K_1 \tilde{v}^T \tilde{v}.\end{aligned} \tag{25}$$

Since Assumption 4 is not satisfied, (25) can be upper bounded as follows by noting that $R \succeq 0$

$$\dot{V} \leq -\alpha_v\left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T\right) P\left(e + K_b \frac{J_z^T z_e}{k_e^2 - z_e^2}\right) - K_1 \tilde{v}^T \tilde{v}. \tag{26}$$

From (26), $\dot{V}(\zeta)$ is negative semidefinite on $\Phi \times \mathbb{R}^{3n} \times \mathbb{R}^{3nmd} \times \mathbb{R}^m$. Invoking Lemma 1, it is concluded that $z_e(t)$ stays bounded within $\pm k_e$ as long as $z_e(0)$ starts within $\pm k_e$. Furthermore, invoking [39, Th. 8.4], it can be concluded that $\zeta(t)$ is bounded and by using the fact that $\dot{V}(\zeta(t)) \leq 0$, we have $V(\zeta(t)) \leq V(\zeta(0))$. Thus, $\|\zeta(t)\| \in \mathcal{L}_\infty$. □

Next theorem analyzes stability of the closed-loop system when Assumption 4 on the history stack is satisfied.

Theorem 2: If Assumptions 5 and 4 are satisfied, then for $\zeta(T) \in \mathcal{D}$, the closed-loop error systems in (13), (12), (18), and

(19) are asymptotically stable in the sense that as $t \to \infty$

$$\|e(t)\| \to 0, \|\tilde{\theta}(t)\| \to 0, \|\tilde{v}(t)\| \to 0, \|z_e(t)\| \to 0. \quad (27)$$

Proof: If Assumptions 4 and 5 are satisfied, then $\lambda_{\min}\{R\} > \delta$. Since, $e \notin \text{Null}(\hat{J}^T)$, the derivative of Lyapunov function in (22) from (25) can be written as

$$\dot{V} = -\alpha_v \left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T \right) P \left(e + \frac{J_z^T z_e}{k_e^2 - z_e^2} \right) - \tilde{\bar{\theta}}^T K_{\text{cl}} R \tilde{\bar{\theta}} - K_1 \tilde{v}^T \tilde{v}. \quad (28)$$

Since $\dot{V} \leq 0$, it can be concluded that $V(t) \leq V(0)$, which implies that $e, \tilde{\bar{\theta}}, \tilde{v} \in \mathcal{L}_\infty$. From Lemma 1, it is concluded that $z_e \in \mathcal{L}_\infty$. Next, to check the uniform continuity of $\dot{V}(t)$, the derivative of $\dot{V}$ yields

$$\ddot{V} = -\alpha_v \left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T \right) \dot{P} \left(e + \frac{J_z^T z_e}{k_e^2 - z_e^2} \right) - \tilde{\bar{\theta}}^T K_{\text{cl}} \dot{R} \tilde{\bar{\theta}} - 2\alpha_v \left(\frac{z_e^T J_z}{k_e^2 - z_e^2} + e^T \right) P \left(\dot{e} + \frac{J_z^T \dot{z}_e}{k_e^2 - z_e^2} + \frac{2\, J_z^T z_e^2 \dot{z}_e}{(k_e^2 - z_e^2)^2} \right) - 2\tilde{\bar{\theta}}^T K_{\text{cl}} R \dot{\tilde{\bar{\theta}}} - 2K_1 \tilde{v}^T (\dot{v} - \dot{v}_d) \quad (29)$$

where $\dot{P} = \frac{d}{dt}(\hat{J}\hat{J}^\dagger)$ and $\dot{R} = \frac{d}{dt}(\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i)$. Since $e, \tilde{\bar{\theta}}, \tilde{v}$, $z_e \in \mathcal{L}_\infty$, from (13), (18), (19), (11), and (10), it can be concluded that $\dot{e}, \dot{\tilde{\bar{\theta}}}, \dot{z}_e, \dot{\tilde{v}}$, and v_d are bounded. Since $\tilde{v} \in \mathcal{L}_\infty$ implies v is bounded. Furthermore, from taking derivative of $J(s)$ and using $Y_i(s)$, and $\dot{\tilde{\bar{\theta}}} \in \mathcal{L}_\infty$, it can be proved that $\hat{J}$ and $\dot{\hat{J}}$ are bounded. Using (16), it can be shown that $\bar{Q} \in \mathcal{L}_\infty$ and using (15), it can be shown that $\dot{R} \in \mathcal{L}_\infty$. Then, from (29), it can be observed that $\ddot{V}(t) \in \mathcal{L}_\infty$. This shows that $\dot{V}(t)$ is uniformly continuous. Given that $V(t) \geq 0$, invoking Barbalat's lemma, the right-hand side of (28) converges to 0, which implies that $\|\tilde{\bar{\theta}}\| \to 0$, $\|\tilde{v}\| \to 0$, and $\|e\| \to 0$ and $\|J_z^T z_e\| \to 0$ as $t \to \infty$ given that $J_z^T z_e + e = 0$ only when z_e and e are identically 0. Furthermore, since $z_e \notin \text{Null}(J_z^T)$, $\|z_e\| \to 0$ as $t \to \infty$. □

Remark 5: In general, $P \succ 0$ is very hard to achieve when the dimension of $s(t)$ is larger than the dimension of $v(t)$. Similar to the stability analysis for the IBVS shown in [38] a local stability of the error dynamics can be concluded.

VI. Singular Value Maximization for ICL

In this section, a method that chooses the data points of history stack based on eigenvalue maximization, is presented (cf. [40]). By maximizing the minimum eigenvalue of $\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i$ the convergence rate of $\tilde{\bar{\theta}}$ can be improved as seen from (28). Since the dimension of $\bar{Q}$ can be very high due to large number of points representing the deformable object, maximizing $\lambda_{\min}(\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i)$ is computationally expensive. The structure of $\bar{Q}(s(t), v(t)) = \int_{\max\{t-\triangle t,0\}}^t \bar{Y}^T(s(\tau)) \bar{V}^T(v(\tau)) d\tau \in \mathbb{R}^{3n \times 3nmd}$ can be used to reduce the computation, which is shown next. Consider $\bar{Q}(s(t), v(t))$

$$\bar{Q} = \int_{\max\{t-\triangle t,0\}}^t \begin{bmatrix} Y_1^T(s) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Y_{3n}^T(s) \end{bmatrix} \begin{bmatrix} v^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & v^T \end{bmatrix} d\tau \quad (30)$$

which when multiplied can be written as follows since all Y_is are identical

$$\bar{Q} = \int_{\max\{t-\triangle t,0\}}^t \begin{bmatrix} q & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q \end{bmatrix} d\tau \quad (31)$$

where $q \in \mathbb{R}^{1 \times md}$. Let $\bar{q} = \int_{\max\{t-\triangle t,0\}}^t q d\tau$. The product $\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i$ can be written as

$$\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i = \begin{bmatrix} \sum_{i=1}^N \bar{q}_i^T \bar{q}_i & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sum_{i=1}^N \bar{q}_i^T \bar{q}_i \end{bmatrix}. \quad (32)$$

Since $\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i$ contains repeating entries of $\sum_{i=1}^N \bar{q}_i^T \bar{q}_i$ on the block diagonals, the minimum eigenvalue of $\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i$ is the same as the minimum eigenvalue of $\sum_{i=1}^N \bar{q}_i^T \bar{q}_i$. Therefore, maximizing $\lambda_{\min}(\sum_{i=1}^N \bar{q}_i^T \bar{q}_i)$ is sufficient to maximize $\lambda_{\min}(\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i)$, which is a much reduced computation due to the size of $\bar{q}_i^T \bar{q}_i$. The computation of $\lambda_{\min}(\sum_{i=1}^N \bar{Q}_i^T \bar{Q}_i)$ scales independent of the size of the feature points and only depends on sizes of the basis vector d and the control input m. Details of the singular value maximization method can be found in [9], [40], and references therein.

Remark 6: For both simulation and experiment, the history stack is initialized using data collected from an offline phase. Predefined constant velocity motions in different directions are applied to the end-effector to simulate the motion of deformable object. The collected data is used to fill up the history stack such that Assumption 4 is satisfied.

VII. Simulation Studies

A. Simulation Setup

A particle-based simulation library from NVIDIA Flex is used to evaluate the proposed BLF-ICL controller on an Alienware Aurora R12 Intel-i7 Desktop with a NVIDIA GeForce GTX 1660 Ti GPU and 16-GB RAM. A cloth consisting of 30 cm $\times$ 30 cm is chosen as a deformable 3-D object as shown in Fig. 3 for the simulation. A rigid cube is attached to one point on the corner of the cloth to simulate the grasped point on the deformable object by a robot end-effector. At each iteration, the velocity input is computed and applied to the rigid cube to manipulate the deformable object to its desired configuration. A small subset of points are sampled symmetrically from the object from its edges and middle portion. The desired configuration s_d is obtained by storing the 3-D positions of each point at the desired shape as shown in Fig. 3, where the height of the control point is 0.3433 cm above the surface. Simulation studies are conducted to test the performance of the developed BLF-ICL

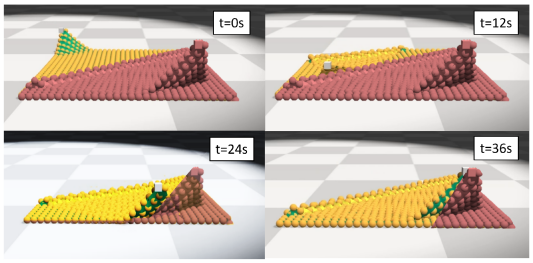

Fig. 3. Simulation results showing (a) initial configuration of the cloth, (b) intermediate configuration, (c) intermediate configuration, and (d) desired configuration of the cloth.

shape servoing controller against three other controllers, and for variations in the size of the history stack, N, number of feature points, and cloths with different material properties.

B. Cloth Folding Simulation

The deformable object Jacobian is approximated using Fourier basis functions with $d = 20$ basis functions. The weight parameters w_k of the basis functions, where $k = \{1, \ldots, d\}$, are initialized from a normal distribution with zero mean, and unit variance and a constant $\tau = 1$. The dimension of the control input $v(t)$ is $m = 3$. A total of $n = 17$ points are symmetrically sampled, 12 points from the edges, and 5 from the middle of the cloth to represent the 3-D configuration in the camera frame. The performance of the BLF-ICL controller is compared with an ICL controller, a BLF controller with gradient update law, and a shape servoing controller developed in [35]. For the BLF-ICL controller, the controller gains are selected to be $\alpha_v = 5\mathrm{I}_3$, $K_1 = 60$, and $K_b = 1$. Whereas the parameter update gains are selected as $\Gamma_\theta^{-1} = 2\mathrm{I}_{3nmd}$ and $K_{\mathrm{cl}} = 10\mathrm{I}_{3nmd}$ for the ICL term. The bound on z_e is selected to be $k_e = 0.3$, which ensures that the control point remains within $\pm k_e$ of the height of the desired point above the table surface. The time window of integration for ICL is $\Delta t = 0.5$ s and the size of the history stack is computed using the inequality $N \geq dm$ and is selected as $N = 70$. The input vector is initialized as $v = [0.01, 0.01, 0.01]^T$ cm/s. The same parameters are chosen for both the ICL controller and the BLF-Gradient controller, except α, Γ_θ^{-1}, and k_b, which are selected as $\alpha_v = 0.5\mathrm{I}_3$, $\Gamma_\theta^{-1} = 0.2\mathrm{I}_{3nmd}$, $K_b = 0.5$. The parameters of the controller in [35] are selected as Jacobian update parameter $\Gamma = 0.95$, and gains $c = 0.7$, $C = 1\mathrm{I}_n$.

The results from the simulation are summarized in Fig. 4(a)–(c). In Fig. 4(a), shape regulation error norm of the BLF-ICL shape servoing controller is compared with that of an ICL-based controller without the constraint, BLF controller with gradient parameter update law and the controller in [35]. The BLF-ICL controller uses constrained state error in the control law and the adaptive update law in addition to the history stack in the adaptive parameter update law. This allows the shape regulation error to converge to a small value of 0.04 cm^2 while maintaining the control point above and below 0.312 cm of the height of its desired location from the table surface, which avoids the control point hitting the table or going too far above the table surface. On the other hand, the ICL-based method and the controller in [35] that does not use any constraints in the controller design may cause the control point to go through the table surface by violating the bound in vertical direction from 35 to 40 s and 5 to 10 s, respectively. When the constraints on the vertical motion of the control point are not enforced, the control point may take nonfeasible paths before reaching the desired shape. The constrained state error for both the controllers is presented in Fig. 4(b), which shows that the BLF-ICL and BLF-gradient controllers ensure constraint satisfaction of the control point within the predefined safety region. Between 9–10 s and 31–32 s, the control point approaches the boundary, which causes the higher control velocities near that time as seen from Fig. 4(c) for the BLF controllers, which causes the control point to move away from the boundary after around those time instances for BLF-based controllers. However, the controllers without the BLF term violate the constraints on the control point during transient motion. The controller in [35] also shows high velocities spikes throughout the simulation, which may cause issues in practical settings.

TABLE I
EFFECT OF THE SIZE OF HISTORY STACK

Size of history stack (N)	RMSE (cm)	Time to convergence
60	1.739135	49.28
70	1.4899	30.133
80	1.3874	25.43
90	1.254	19.77
100	1.1831	17.8

C. Performance With Respect to History Stack Size

In this section, simulations are carried out to test the performance of the shape regulation error and convergence time with respect to the size of the history stack N. The same simulation parameters are selected for all simulations except the size of the history stack. The parameters are selected as: $d = 20$, $n = 51$, $K_{\mathrm{cl}} = 10$, $K_1 = 60$, $K_b = 1$, $\alpha_v = 5I_3$, $\Gamma_\theta^{-1} = 2I_{3nmd}$ and the bounds on the control point is $k_e = 0.3$. The results are summarized in Table I. The root mean squared error (RMSE) is computed over 100 s at the sample rate of 60 frames per second. Time to converge is computed as time after which the error norm is always less than 7% of the initial error. From Table I, it is observed that as the size of the stack N increases, the RMSE of the shape regulation and the time to converge decrease. However, increasing N too high can be computationally expensive while providing negligible improvement.

D. Performance of Variation in the Number of Feature Points

In this section, simulations are carried out to test the performance of the BLF-ICL controller when the cloth is represented using different number of discrete set of feature points. The same simulation parameters are selected for all simulations. The parameters are: $d = 20$, $n = 51$, $N = 70$, $K_{\mathrm{cl}} = 20$, $K_1 = 60$, $K_b = 1$, $\alpha_v = 1.5I_3$, $\Gamma_\theta^{-1} = 1I_{3nmd}$, and the bounds on the control point is $k_e = 0.3$. The final shape regulation error using all

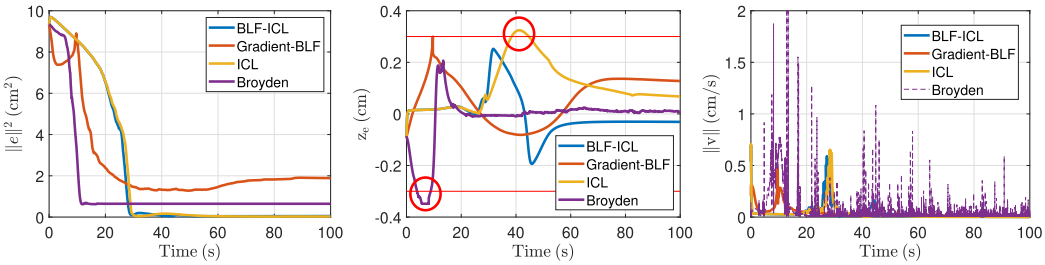

Fig. 4. Results of the simulation, and comparison between BLF-ICL controller, Gradient-BLF, ICL-controller without BLF terms, and a controller using Broyden update rule [35]. (a) Shape error norm $\|e\|^2$. (b) Control point error z_e. (c) Norm of the control velocities.

TABLE II
EFFECT OF NUMBER OF FEATURE POINTS

# of feature points	Time to convergence	Final shape error (cm)
9	14.4167	5.3311
17	30.133	3.1923
25	29.55	2.1614

TABLE III
EFFECT OF DIFFERENT MATERIAL PROPERTIES

Stretch stiffness	Bend stiffness	Shear stiffness	Mass (g)	RMSE (cm)	Time to convergence
0.1	0.1	0.1	1.0	1.299	30.583
0.5	0.5	0.5	1.0	1.233536	24.5
1	1	1	1.0	1.238	30.9
0.5	0.5	0.5	0.1	1.4313	72.26
0.5	0.5	0.5	0.5	1.406196	79.73
0.5	0.5	0.5	5.0	1.3622	64.967
0.5	0.5	0.5	10.0	1.3458	53.467
0.5	0.5	0.5	20.0	1.468	85.81

the points on the cloth and time to converge metrics are reported in Table II to evaluate the controller performance with respect to the expressiveness of the cloth representation based on number of feature points. As the number of feature points to match (size n of the state vector) increases, the time of convergence also increases but the final shape regulation error decreases achieving the desired shape more accurately. With fewer feature points, the task can be carried out faster with larger final shape error.

E. Performance of Variation in Material Properties

In this section, simulations are carried out to test the performance of the controller on the cloth object with different stretch, bend, shear stiffness, and mass properties. In NVIDIA Flex simulator cloth motions are simulated with different material properties. The shape servoing controller's performance is tested with three different sets of material properties by keeping a constant mass parameter, and by keeping stiffness properties same with variation in mass. The same simulation parameters used to test the variation of history stack are selected for all simulations. The size of the history stack is selected as $N = 70$ and corresponds to data collected from a cloth with stretch stiffness 1.0, bend stiffness 0.5, and shear stiffness 0.9. The results are reported in Table III. The history stack for varying material properties is initialized using the data collected on the same object with different material properties. From the metrics reported in Table III, it is observed that history stack collected from an object can work on a similar object with different material properties to achieve a good shape regulation performance.

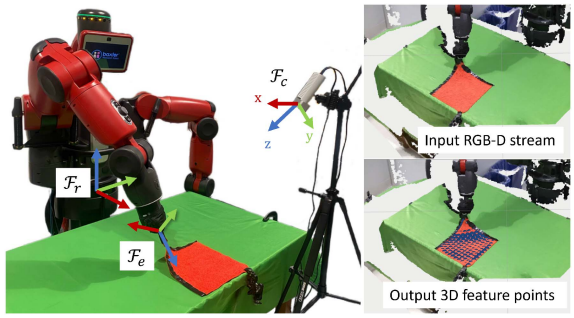

Fig. 5. Experimental setup for cloth shape manipulation control. Reference frames $\mathcal{F}_r$, $\mathcal{F}_e$, and $\mathcal{F}_c$ correspond to the robot body, robot end-effector, and Microsoft Kinect camera. Images on the right show RGB-D data from camera.

VIII. EXPERIMENT

A. Experiment Setup

The experimental platform, which includes a Baxter robot, a deformable object, such as a cloth or a rope, and a Microsoft Azure Kinect camera sensor, is shown in Fig. 5. The developed controller is tested using two types of deformable objects, a cloth and a rope. The cloth and rope are placed horizontally on a table top and one arm of the Baxter robot is used to grasp one end of the object. The other end is securely fastened to the table. Images of the cloth object are obtained using the Kinect sensor at the frame rate of 30 Hz are shown in Fig. 5. Based on the Kinect images, the current shape of the object is obtained from the constrained deformable coherent point drift (CD-CPD) algorithm proposed in [41]. CD-CPD provides a point-based representation to describe the location of the object with respect to the camera reference frame. Out of 100 points produced by the CD-CPD algorithm used as its input, 13 points are selected to represent the cloth as shown in Fig. 6. In the case of the rope, the point cloud input to CD-CPD, outputs ten 3-D keypoints representing the current configuration of the rope out of which all 10 are used as the feature points by the shape servoing controller.

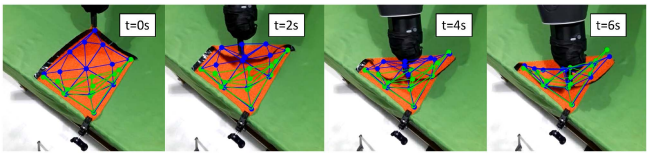

Fig. 6. Experimental results showing initial, intermediate, and final (desired) configurations of the cloth, where green points show the desired configuration and blue points show the current configuration of the cloth.

The designed BLF-ICL controller is implemented such that the control point maintains a constant position above the table. To compute the bounds, a distance from the control point to the plane describing the table top is calculated. Since all the measurements are taken from a coordinate frame attached to the camera $\mathcal{F}_c$, which is pointed down at an angle to the table surface, the height of the control point cannot be determined solely using measurements along the optical axis. Therefore, three points on the deformable object kept flat on the table are used to define a plane and the normal vector $[n_x, n_y, n_z]^T$ is calculated in the camera reference frame. The operator J_z in (5) is computed as $J_z = [[n_x, n_y, n_z]^T, 0_{1\times(3n-3)}]^T$. Marker-based systems can also be used to compute the normal vector.

B. Cloth Shape Servoing Experiment

Details of a cloth folding experiment are presented in this section. The deformable object's Jacobian matrix is approximated using a set of basis functions and unknown parameters. The frequencies of the basis functions w_k, where $k = \{1, \ldots, d\}$, are selected from a normal distribution with zero mean and unit variance. The number of frequencies and the constant τ is selected as $d = 40$ and $\tau = 1$, respectively. The dimension of the control input $v(t)$ is $m = 3$. Two controllers are implemented, namely, the proposed controller BLF-ICL and a controller without the BLF terms.

For the BLF-ICL controller, the control gains of the virtual controller is selected as $\alpha_v = 0.0251\mathrm{I}_3$, the proposed filtered controller gain is selected as $K_1 = 45$, the constrained state gain is chosen as $K_b = 1$, the constraint on z_e is selected as $k_e = 0.04$, the adaptation gain is set to $\Gamma_\theta^{-1} = 0.3\mathrm{I}_{3nmd}$, and the gain for ICL terms is used as $K_{\mathrm{cl}} = 0.001\mathrm{I}_{3nmd}$. The time window of integration for ICL terms is selected to be $\Delta t = 0.75$ s and the size of the history stack is selected as $N = 140$. For the controller without BLF terms, same parameters are used except the ones that are mentioned next, which are selected as $\alpha_v = 0.0575\mathrm{I}_3$, $\Gamma_\theta^{-1} = 0.03\mathrm{I}_{3nmd}$, $K_{\mathrm{cl}} = 10\mathrm{I}_{3nmd}$, and $K_1 = 30$. The input vector is initialized as $v(t_0) = [0.01, 0.01, 0.01]^T$ m/s.

A sequence of images showing the experiment in progress is shown in Fig. 6. The results from the experiment are summarized in Fig. 7. In Fig. 7, the performance of the proposed BLF-ICL-based controller is compared with an ICL-based controller in terms of the shape regulation error. The shape error converged to a small error near 0m with the proposed controller. It is observed that the constrained state error remains within the safety region ensuring that the control point stays away from the boundaries, i.e., without hitting the table surface, whereas for the controller without BLF, the constraints are violated and the control point

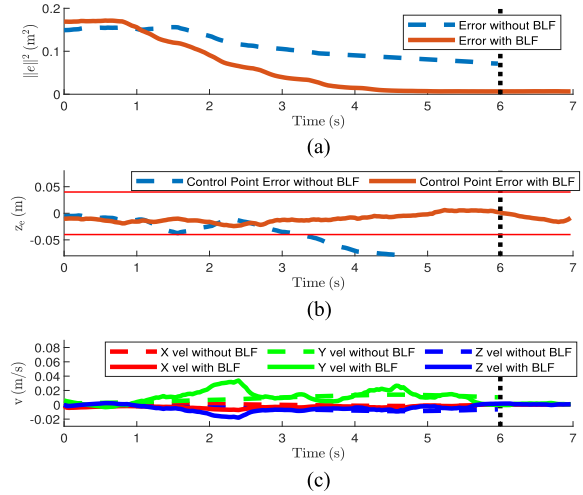

Fig. 7. Results for the cloth folding experiment, and comparison between BLF-ICL controller and ICL-controller without BLF terms. (a) Shape error norm squared. (b) Control point error. (c) Control velocities.

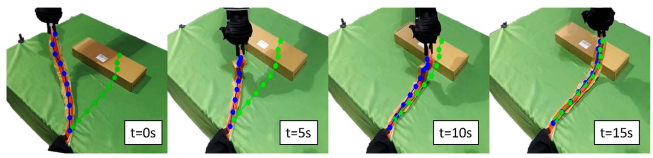

Fig. 8. Experimental results showing initial, intermediate, and desired configurations of a rope displacement task, where green points show the desired configuration and blue points show the current configuration.

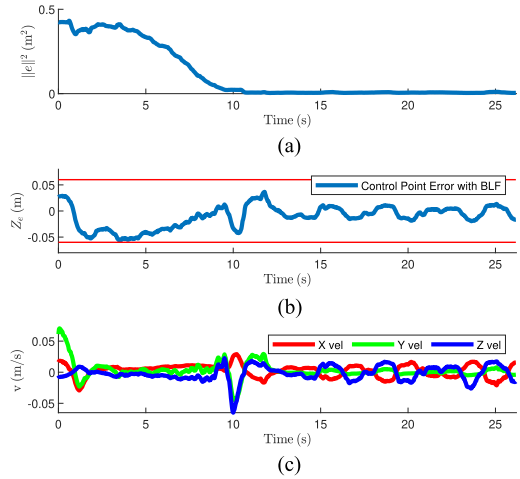

Fig. 9. Results for the rope experiment with BLF-ICL controller. (a) Shape error norm squared. (b) Control point error. (c) Control velocities.

reaches the table surface near $t = 6$ s, shown using a vertical line in Fig. 7. The control velocity shown in Fig. 7, where it is observed that the higher control velocities are generated by the BLF-ICL controller when the constrained state reaches near the constraint boundaries.

C. Rope Shape Servoing Experiment

An experiment is conducted to test the applicability of the shape servoing method to a rope. Shape servoing of a rope like object to displace it from one location to another is selected as a use case as shown in Fig. 8. Similar to the cloth like object, the rope is represented using a discrete set of 10 points. The other parameters for the deformation model are selected as $d = 20, \tau = 1$. The BLF-ICL controller parameters are selected as: $\alpha_v = 0.3$, $K_1 = 60$, $k_e = 0.06$, $K_b = 1.0$, $\Gamma_\theta^{-1} = 1.0\mathrm{I}_{3nmd}$, $K_{\mathrm{cl}} = 50\mathrm{I}_{3nmd}$. For ICL, the size of history stack is selected as $N = 90$, and $\Delta t = 0.67$ s. The shape estimation error norm, control point error, and control velocities are shown in Fig. 9. It can be observed that the shape estimation error converges to 0, and the control point stays within the specified bounds.

IX. Conclusion

In this article, an adaptive controller that satisfies physical constraints for achieving shape regulation of the deformable objects is designed based on the BLF and ICL methodologies. The unknown deformation Jacobian is approximated using a Fourier series-based regression with unknown parameters. An ICL-based parameter update law is designed to estimate the parameters. The controller is able to regulate the shape of the object to a desired shape and satisfy constraints on the system state simultaneously. The ICL-based law uses a history stack of the prior data in the parameter estimation law, which yields convergence of the parameter estimates to the ideal parameters. Lyapunov stability theorems are presented to prove the boundedness of the closed-loop signals and convergence of the shape regulation and parameter estimation errors. The BLF-ICL adaptive controller is implemented using NVIDIA Flex simulator on a cloth folding use case. The comparison studies show that the BLF-ICL controller is able to maintain the physically inspired constraints and shows a good shape regulation performance. Simulation studies for variations in control parameters provide insights into how to select parameters, such as size of history stack and number of points used to represent the object. Experiments are performed using a robot platform, and a Kinect RGB-D camera for a cloth folding and rope displacement tasks. This shows the applicability of the controller to different types of deformable objects. In future, extensions of the BLF-ICL controller to manipulate deformable objects using multiple points and rotational velocities will be explored, which can achieve more complex manipulation tasks.

References

[1] A. Shah, L. Blumberg, and J. Shah, "Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1823–1838, Oct. 2018.

[2] O. Ogunmolu, A. Kulkarni, Y. Tadesse, X. Gu, S. Jiang, and N. Gans, "Soft-NeuroAdapt: A 3-DOF neuro-adaptive patient pose correction system for frameless and maskless cancer radiotherapy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3661–3668.

[3] S. Kamtikar, S. Marri, B. T. Walt, N. K. Uppalapati, G. Krishnan, and G. Chowdhary, "Towards autonomous berry harvesting using visual servoing of soft continuum arm," in *Proc. AI Agriculture Food Syst.*, 2022, pp. 1–7.

[4] S. Mehta and T. Burks, "Vision-based control of robotic manipulator for citrus harvesting," *Comput. Electron. Agriculture*, vol. 102, pp. 146–158, 2014.

[5] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Sci. Robot.*, vol. 6, no. 54, 2021, Art. no. eabd8803.

[6] 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 Trans. Robot.*, vol. 34, no. 1, pp. 272–279, Feb. 2018.

[7] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1462–1480, 2014.

[8] K. P. Tee and S. S. Ge, "Control of nonlinear systems with partial state constraints using a barrier Lyapunov function," *Int. J. Control*, vol. 84, no. 12, pp. 2008–2023, 2011.

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

[10] M. Aranda, J. A. C. Ramon, Y. Mezouar, A. Bartoli, and E. Özgür, "Monocular visual shape tracking and servoing for isometrically deforming objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7542–7549.

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

[12] 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 Trans. Mechatronics*, vol. 28, no. 2, pp. 945–955, Apr. 2023.

[13] P. Li, F. Chaumette, and O. Tahri, "A shape tracking algorithm for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2847–2852.

[14] A. AlBeladi, G. Krishnan, M.-A. Belabbas, and S. Hutchinson, "Vision-based shape reconstruction of soft continuum arms using a geometric strain parametrization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11753–11759.

[15] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "Lattice-based shape tracking and servoing of elastic objects," *IEEE Trans. Robot.*, vol. 40, pp. 364–381, Nov. 2023.

[16] G. Yao, R. Saltus, and A. P. Dani, "Image moment-based extended object tracking for complex motions," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6560–6572, Jun. 2020.

[17] G. Yao, R. Saltus, and A. P. Dani, "Shape estimation for elongated deformable object using b-spline chained multiple random matrices model," *Int. J. Intell. Robot. Appl.*, vol. 4, no. 4, pp. 429–440, 2020.

[18] G. Yao and A. P. Dani, "Image moment-based random hypersurface model for extended object tracking," in *Proc. Int. Conf. Inf. Fusion*, 2017, pp. 1–7.

[19] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "FEM-based deformation control for dexterous manipulation of 3D soft objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4007–4013.

[20] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 1637–1643.

[21] 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," *Front. Robot. AI*, vol. 7, p. 73, 2020.

[22] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 4, pp. 2016–2026, Aug. 2021.

[23] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4525–4532.

[24] L. Smolentsev, A. Krupa, and F. Chaumette, "Shape visual servoing of a tether cable from parabolic features," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 734–740.

[25] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "As-rigid-as-possible shape servoing," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3898–3905, Apr. 2022.

[26] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proc. Symp. Geometry Process.*, 2007, pp. 109–116.

[27] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D deformable object manipulation using deep neural networks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4255–4261, Oct. 2019.

[28] R. Lagneau, A. Krupa, and M. Marchal, "Active deformation through visual servoing of soft objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 8978–8984.
[29] 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 Trans. Robot.*, vol. 39, no. 1, pp. 417–436, Feb. 2023.
[30] T. Tang, C. Liu, W. Chen, and M. Tomizuka, "Robotic manipulation of deformable objects by tangent space mapping and non-rigid registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2689–2696.
[31] V. H. Giraud, M. Padrin, M. Shetab-Bushehri, C. Bouzgarrou, Y. Mezouar, and E. Ozgur, "Optimal shape servoing with task-focused convergence constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2197–2202.
[32] 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 *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 8274–8281.
[33] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[34] V. Guthikonda, G. Rotithor, and A. P. Dani, "Shape servoing of deformable objects using adaptive deformation model estimation," in *Proc. IFAC World Congr.*, 2023, pp. 10793–10798.
[35] D. Navarro-Alarcón, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1457–1468, Dec. 2013.
[36] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*. Hoboken, NJ, USA: Wiley, 1995.
[37] T. Garg and S. B. Roy, "Barrier Lyapunov function based controller design for Euler-Lagrange systems with reduced control effort," *IFAC-PapersOnLine*, vol. 53, no. 1, pp. 459–464, 2020.
[38] F. Chaumette and S. Hutchinson, "Visual servo control. I. basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
[39] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Hoboken, NJ, USA: Prentice-Hall, 2002.
[40] G. Chowdhary, T. Yucelen, M. Muhlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adaptive Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.
[41] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6443–6450.

Vrithik Raj 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 engineering.

His research interests involve deformable object manipulation using adaptive control methods.

Ashwin P. Dani (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the University of Florida (UF), Gainesville, FL, USA, in 2008 and 2011, respectively.

He was with the Nonlinear Controls and Robotics Group, Mechanical and Aerospace Engineering Department, University of Florida. He was a Postdoctoral Research Associate with the University of Illinois, Urbana-Champaign, IL, USA. In 2013, he joined the faculty of Electrical and Computer Engineering (ECE) as Assistant Professor with the University of Connecticut, Storrs, CT, USA, where he is currently an Associate Professor. His current research interests include machine learning for control, human–robot collaboration, and robot perception.

Dr. Dani is a member of the conference editorial board of IEEE Control Systems Society (CSS). He was a member of the conference editorial board of IEEE Robotics and Automation society (RAS) and a Technical Editor for IEEE TRANSACTIONS ON MECHATRONICS for years 2021–2023.