Robust control of a silicone soft robot using neural networks

Gang Zheng, Yuan Zhou, Mingda Ju

Robust control of a silicone soft robot using neural networks

Gang Zheng[a], Yuan Zhou[b] and Mingda Ju[c]

[a]INRIA Lille-Nord Europe, 40 Avenue Halley 59650, Villeneuve d'Ascq, France.
[b]Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
[c]ETIS, CNRS UMR 8051/ENSEA, Université de Paris Seine, Université de Cergy-Pontoise, 95302 France

ARTICLE INFO

Keywords:
Soft robot, Machine Learning, Robust control

ABSTRACT

This paper deals with the robust controller design problem to regulate the position of a soft robot with elastic behavior, driven by 4 cable actuators. In this work, we first used an artificial neural network to approximate the relation between these actuators and the controlled position of the soft robot, based on which two types of robust controllers (type of integral and sliding mode) are proposed. The effectiveness and the robustness of the proposed controllers have been analyzed both for the constant and the time-varying disturbances. The performances (precision, convergence speed and robustness) of the proposed method have been validated via different experimental tests.

1. Introduction

Most of robots nowadays have the rigid body, such as mobile robot, humanoid and manipulator, for different reasons. Mathematically, rigid body is supposed to have no deformation during its work, so it is relatively easy for the modeling and control design. However, such a rigid robot also has some disadvantages, such as low adaptability and unsafety. Note that, for the robots that serve human, the most important issue is safety. In such a case, we prefer the soft contact between the robot and the human. That is why the research on soft and flexible robots have become one of the hottest topics nowadays. Soft robot can also save energy because it is normally lighter than rigid robot, and it is easier to be integrated into human beings' body, serving as a prosthesis [1], [2], or as sensors [3]. Soft robotics is an emergent research topic. In the literature, it covers two categories: the first one is about the rigid body with flexible actuators, and the second one refers to the deformable body due to the flexibility of materials [4]. The first category can be seen as a natural extension of traditional rigid robots by adding a certain dynamics to describe the behavior of flexible actuators, while the second one is totally new and more attractive since it provides flexibility to robots, for example, to adjust their shapes according to their tasks and environments. Due to the soft property, this type of robot can easily achieve compliant and safe tasks.

For soft robots with deformable body, it is still a difficult problem when considering the control of soft robots. The control theory developed for rigid robots is poorly applicable to this case [1]. It is mainly due to the lack of efficient method to deduce its exact model (either kinematic, or dynamic). One of the most used techniques is to deduce the model via the curvature information of soft robots. In [5], the kinematic model was deduced for a hyper-redundant robot by using the information of backbone curves. In [6], the kinematic model is deduced by using geometric information, and then a computed torque controller is applied to control an eel-like soft robot. A robust feedback control was proposed in [7] to control the trajectory of soft robots, under as well the assumption of piece-wise constant curvature. Other techniques have also been used to get the model. For example, in [8], the authors used Euler-Bernoulli beam theory to model an inflatable robot, and design a force control for such a soft robot. [9] used the Cosserat theory to deduce a static model of a special continuum soft robot, and a 3D steady-state model of a tendon-driven continuum soft octopus-like manipulator has been developed in [10]. In [11], the authors proposed to use the Finite Element Method (FEM) to model the deformation of soft robots, and to check its controllability. A feedback control for soft robots was also realized by using FEM and visual tracking technique [12]. In [13], the dynamics of soft robots has been analyzed, and the reduced-order model technique has been used to design the controller to achieve the fast control of soft robots. Comparing with other approaches to model soft robots' deformation, several advantages have been identified for FEM method. First, it can be easily implemented with the constitutive law of the material which is experimentally measurable, and secondly the accuracy and the computing cost have also been improved. However, for the soft robots with complex shapes or special materials, it is difficult to get the constitutive law of the material.

ORCID(s):

In order to avoid the mentioned disadvantages, this paper proposes to use artificial neural network (ANN) to get an approximated model, for the purpose of describing the relation between actuators (inputs) and the controlled states (outputs) for a soft robot made by silicone. In contrary to FEM, we can train this neural network to get an approximated model without any information about the characteristics of soft robot itself. This strategy allows us to realize the control of soft robots with complex shapes or unknown characteristics. Indeed, it is not new to use neural network to approximate the model of soft robots. In [14], the researchers used neural network to solve the inverse statics of a cable-driven soft arm with non-constant curvature. [15] used FEM to train a neural network to get the model of modular soft robots. Model predictive control for a pneumatic soft robot has been studied in [16] which was based on the model learned via neural networks. We would like to emphasize that most of those mentioned results did not provide any theoretical convergence proof for the designed controllers. Also, no concrete theoretical analysis on the influence of constant (or piece-wise constant) and time-varying disturbances has been carried out. Therefore, compared to those existing results of ANN-based controller design for soft robots, **the main contribution** of this paper is as follows:

1. Based on ANN, two types of controllers were proposed with complete mathematical proofs;
2. Simple and checkable conditions are provided to judge whether the approximated model is precise or not;
3. Robustness analysis has been made by considering constant or time-varying bounded disturbances;

This paper is organized as follows. In Section 2, we briefly discuss the disadvantages of FEM and explain the main motivation of using neural network. Section 3 recalls the classical method to design controller when knowing the exact input-output model of robot, and some properties are investigated as well in this section. In Section 4, we firstly present how to use neural network to obtain the approximated input-output model, based on which robust controllers will be investigated. In the same section, we will mathematically prove that these controllers can work even without the knowledge of the exact model, provided that the neural network can achieve a certain approximation accuracy. The robustness of the proposed controllers have been analyzed as well in Section 4, and this paper is ended up with several experiments on a real soft robot made by silicone in order to highlight the efficiency of the proposed method.

2. Problem statement

This paper investigates the control of the interested point of a soft robot made by silicone. The word *soft* means the material used to build the robot might be deformable. According to the theory of continuum mechanics, the deformability of material depends on different characteristics, such as Poison ratio, Young's modulus, and also on the shape and density of the material.

Generally, given a general configuration of soft robots, with the equipped actuators, it is difficult or impossible to obtain its exact kinematic or dynamic model. Therefore, researchers attempt to approximate those exact models, and then to design the closed-loop controller based on the approximated models. One recent approximation approach is to use FEM. Its basic idea is to spatially discretize the shape of robots by using finite number of fine elements (named as the mesh) to deduce its kinematic or dynamic model. Following the second law of Newton, we can use the following nonlinear model to describe its behavior

$$M(q)\ddot{q} + D(q,\dot{q})\dot{q} + K(q)q = H^T(q)u \tag{1}$$

where $q \in \mathbb{R}^{n_q}$ is the position of the nodes of the mesh, and $u \in \mathbb{R}^m$ represents the magnitude of the actuators. $M(q)$ is the mass matrix which is always invertible, D is the damping matrix, and $K(q)$ represents the stiffness matrix. $H(q)$ represents the force directions (including actuators from the robot itself), and is a rectangular operator, usually sparse, as it has only non-zero values at the points where the actuators are applied.

Using FEM, the control of the interested point of soft robots (named as $x \in \mathbb{R}^3$) can be seen as a projection $q \to x$, and a relation between x and u can be then obtained from (1). Such a method has been applied in [12] to control the silicone soft robot described in Figure 1.

We would like to highlight that FEM method suffers from at least the following problems for the controller design:

- FEM will never give us a precise model.
- FEM method is computationally expensive.
- The designed controller, which is based on FEM model, might not be valid for the real system.

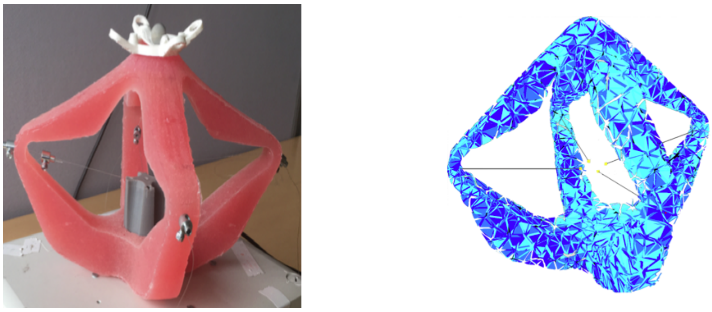

Figure 1: The soft robot made by silicone and the approximated mode via FEM

In order to avoid the above mentioned inconveniences of FEM when approximating the dynamic model of such a soft robot, this paper aims to train (off-line) an artificial neural network to approximate the robot's input-output model, based on which robust controllers will be synthesized to achieve the control (online) of the selected point of interest.

3. Closed-loop control with known input-output model

This section first recalls the classical method of how to design a closed-loop controller for the soft robot described in Figure 1 by assuming that its input-output model is known. At this aim, suppose that the known input-output model is deduced by the following algebraic equation

$$x = f(u) \tag{2}$$

where $x \in \mathbb{R}^3$ represents the position of the interested point to be controlled, $u \in \mathbb{R}^4$ represents the control inputs (the length of the cable linked between the robot and the motor) and f is a map from u to x, i.e., $f : \mathbb{R}^4 \to \mathbb{R}^3$.

Since we are working on the concrete soft robot made by silicone and actuated by cables, the following assumptions are imposed.

Assumption 1. *For the robot described in Figure 1, it is assumed that*

1. *the robot's workspace is bounded, i.e.,* $x \in \mathcal{X}$ *where* $\mathcal{X}$ *represents the workspace boundedness;*
2. *the control input is physically bounded, i.e.,* $u \in \mathcal{U}$ *where* $\mathcal{U}$ *represents the boundedness of the input* u*;*
3. *the control* u *and* x *are changing smoothly, i.e.,* $x, u \in C^\infty$*;*
4. *the point of interest* x *is controllable in its workspace* $\mathcal{X}$*.*

It is worth noting that the above assumptions are not at all restrictive since lots of concrete soft robots satisfy directly these conditions. In fact, these properties are always fulfilled for the soft robot investigated in this paper with $\mathcal{X} \subset \mathbb{R}^3$ and $\mathcal{U} \subset \mathbb{R}^4$. These conditions, stated as assumptions, are only for the purpose of treating general soft robots with different dimensions of $\mathcal{X}$ and $\mathcal{U}$.

With the known and exact input-output model described by (2), since x is smooth, the map f should be smooth, therefore all its components are differentiable and their derivatives are bounded. Consequently, the dynamics of the input-output model can be written as

$$\dot{x} = \frac{\partial f}{\partial u}\dot{u} \tag{3}$$

Then, by noting

$$J = \frac{\partial f}{\partial u} \in \mathbb{R}^{3\times 4} \tag{4}$$

there always exists a constant $\alpha > 0$ such that

$$\|J\|_2 \leq \alpha, \forall u \in \mathcal{U} \tag{5}$$

which can be briefly interpreted as: the velocity of the robot is physically bounded for all admissible control inputs.

In addition, since the point of interest x is assumed to be controllable in $\mathcal{X}$, this means that, for any given $x_d \in \mathcal{X}$, there always exists $u \in \mathcal{U}$ such that the point of interest x can be driven to x_d. Therefore, J defined in (4) needs to be right invertible, i.e., $\exists J_R^{-1} \in \mathbb{R}^{3\times4}$ such that $J J_R^{-1} = \mathrm{I}_{3\times3}$.

This property enables us to design the following classical controller

$$\dot{u} = -\lambda J_R^{-1} e \tag{6}$$

with $\lambda > 0$, where e represents the error between the actual position x and the desired position $x_d \in \mathcal{X}$, i.e., $e = x - x_d$. The proof of the convergence by applying this controller can be found in [17].

Moreover, since J defined in (4) is right invertible, therefore there exists a constant $\beta > 0$ such that

$$\|J J^T\|_2 \geq \beta, \forall u \in \mathcal{U} \tag{7}$$

It is worth noting that the constants α and β defined in (4) and (7) only depend on the studied robot and can be regarded as the characteristic parameters of such a robot. Generally, those parameters can be obtained by performing a prior test.

Concerning the soft robot studied in this paper, the controller (6) cannot be applied as the exact map (2) is unknown in practice (thus it is impossible to obtain J). However, in the following section, we will show that a similar robust controller can be synthesized by using neural networks to approximate the exact map (2).

4. Closed-loop control with approximated model

4.1. Approximation by using neural networks

Neural network is a mechanism which has been developed from cognitive and information theories. It seeks to imitate the learning process of human neurons. When some information comes to the neural system, certain neurons will be activated. This activation is based on the analysis of some former neurons.

For a neuron, we can use the following equation to describe its process

$$y = g\left(\sum_i \omega_i x_i + b\right) \tag{8}$$

where y denotes the output of a neuron, and x_i denotes its former inputs. ω_i represents the significance of the ith former neuron. In other words, it shows how strong the connectivity is between these neurons. In addition, b represents the bias which is used to correct the value when the output y acts as the input of another neuron. And g represents an activation function, which might be linear, sigmoid, hyperbolic tangent and so on [18].

With assembling such type of neurons to create a network with multiple layers, it has been shown in [19, 20, 21, 22] that if the number of neurons and layers are sufficiently large, then any function f can be approximated, to any degree of accuracy, by the output of the neural network.

In our work, we are interested in driving the position of the top point of the robot by controlling the 4 cables linked to motors so that the following neural network is established to approximate the map f in (2).

Concerning the choice of the function g in (8) and due to the fact that f in (2) is smooth and the classical controller (6) contains the derivative of f, a differential function is better to be chosen for g. Inspired by this thought, the activation function is chosen as a sigmoid one, i.e.,

$$y = \mathrm{sgm}(\sum_i \omega_i x_i + b)$$

where sgm represents the following function

$$\mathrm{sgm}(z) = \frac{1}{1 + e^{-z}}$$

By randomly driving the studied robot to explore its whole workspace, we can then acquire the corresponding positions of the interested point x. With these data, the modeling approximation problem can be then transformed to a regression one, which could be solved by applying the existing optimization methods with an approximation performance index.

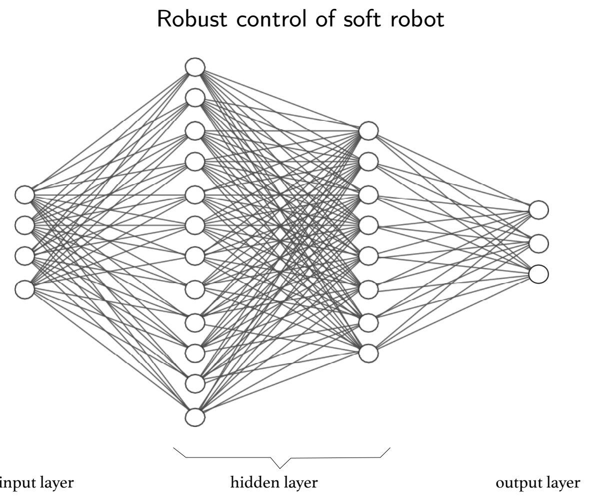

Figure 2: Neural network with 4 inputs (u) and 3 outputs (x)

4.2. Properties of the approximated input-output model

After the training phase, let us assume that the following input-output model has been obtained

$$x = \hat{f}(u) + \delta \tag{9}$$

where $\hat{f} \in \mathbb{R}^{3\times4}$, and δ represents the error between the exact function f and its estimation $\hat{f}$.

The function $\hat{f}$ is differentiable as it is a combination of a series of sigmoid functions. Consequently, we can define the associated Jacobian matrix as follows

$$\hat{J}(u) = \frac{\partial \hat{f}(u)}{\partial u} = \begin{bmatrix} \frac{\partial \hat{f}_1}{\partial u_1} & \frac{\partial \hat{f}_1}{\partial u_2} & \frac{\partial \hat{f}_1}{\partial u_3} & \frac{\partial \hat{f}_1}{\partial u_4} \\ \frac{\partial \hat{f}_2}{\partial u_1} & \frac{\partial \hat{f}_2}{\partial u_2} & \frac{\partial \hat{f}_2}{\partial u_3} & \frac{\partial \hat{f}_2}{\partial u_4} \\ \frac{\partial \hat{f}_3}{\partial u_1} & \frac{\partial \hat{f}_3}{\partial u_2} & \frac{\partial \hat{f}_3}{\partial u_3} & \frac{\partial \hat{f}_3}{\partial u_4} \end{bmatrix} \tag{10}$$

Assumption 2. *For the studied soft robot, with enough training data, it is assumed that there exists a constant $\gamma > 0$, named as approximation accuracy index, such that*

$$\|J(u) - \hat{J}(u)\|_2 \leq \gamma, \forall u \in \mathcal{U} \tag{11}$$

The above assumption implies that we have used enough data to train the neural network, and the error between the exact value of J and its approximation $\hat{J}$ is bounded in the norm sense. The smaller γ is, the more accurate approximated model we have obtained.

4.3. Robust controller design

For the soft robot described in Figure 1, let us choose its top as the point of interest x, and we want to control x to reach a desired position, noted as x_d. As the model described by (3) contains the derivative of the input u, therefore the following dynamic extension [23] can be introduced for the controller design:

$$\begin{aligned} \dot{x} &= \frac{\partial f}{\partial u} v \\ \dot{u} &= v \end{aligned} \tag{12}$$

It is clear that if we know the exact map f, then an integral controller

$$u = \int_{t_0}^{t} v(\tau) d\tau, \text{ with } v(\tau) = -\lambda \left[\frac{f}{\partial u}\right]^\top e(\tau)$$

can be designed, where $e = x - x_d$. For the case where f is unknown, the following will use its approximation $\hat{f}$ (obtained via the neural network) to design a similar controller as (6), which is of the following form:

$$u = -\int_{t_0}^{t} \lambda \left[\frac{\partial \hat{f}}{\partial u}\right]^{\top} e(\tau) d\tau \tag{13}$$

where $\lambda > 0$ and $\hat{f}$ represents the model learned by the neural network.

The following result ensures the convergence of x to the desired constant position x_d even if the exact input-output model is not known.

Theorem 1. *For the studied soft robot described by (12), if the neural network approximates f with an accuracy index γ satisfying the following inequality*

$$\gamma < \frac{\beta}{\alpha} \tag{14}$$

where α and β are the characteristic parameters of soft robot defined in (4) and (7), then the proposed controller (13) can exponentially drive any $x \in \mathcal{X}$ to the desired constant position $x_d \in \mathcal{X}$, i.e.,

$$\lim_{t\to\infty} \|x(t) - x_d\|_2 = 0.$$

Proof 1. *Note that we want to prove $e(t) \to 0$ when $t \to \infty$, and this is equivalent to prove $V(e) \to 0$ when $t \to \infty$, where $V(e)$ is a Lyapunov function defined as follows*

$$V(e) = \frac{1}{2}e^T e$$

At this aim, let us consider the derivative of V with respect to t, and it yields

$$\begin{aligned} \dot{V} &= e^T \dot{e} = e^T \left[\dot{x} - \dot{x}_d\right] \\ &= e^T \tfrac{\partial f(u)}{\partial u} \dot{u} \end{aligned} \tag{15}$$

where we used the fact that $\dot{x}_d = 0$, since the desired position x_d is constant.

Substituting the proposed robust controller (13) into the above equation, we obtain

$$\dot{V} = -\lambda e^T \frac{\partial f(u)}{\partial u} \left[\frac{\partial \hat{f}(u)}{\partial u}\right]^T e$$

Note now

$$\Delta J = \hat{J} - J = \frac{\partial \hat{f}(u)}{\partial u} - \frac{\partial f(u)}{\partial u}$$

then we have

$$\begin{aligned} \dot{V} &= -\lambda e^T \frac{\partial f(u)}{\partial u} \left[\frac{\partial \hat{f}(u)}{\partial u} + \frac{\partial f(u)}{\partial u} - \frac{\partial f(u)}{\partial u}\right]^T e \\ &= -\lambda e^T J J^T e - \lambda e^T J \Delta J^T e \end{aligned}$$

According to (4) and (7), we can then get

$$\begin{aligned} \dot{V} &\leq -\lambda e^T \beta e + \lambda e^T \alpha \|\Delta J\|_2 e \\ &= -\lambda \left[\beta - \alpha \|\Delta J\|_2\right] e^T e \end{aligned}$$

Consequently, if the neural network approximates f with an accuracy index γ satisfying the inequality (14), then there exists an $\eta > 0$ such that

$$\beta - \alpha\gamma = \eta > 0$$

and we can conclude that

$$\dot{V} \leq -2\lambda\eta V$$

with $\eta > 0$.

The above inequality implies that V will exponentially converge to 0 when $t \to \infty$. Due to the fact that $V(e) = \frac{1}{2}e^T e$, we can finally prove that e will exponentially converge to 0 when $t \to \infty$, i.e., any $x \in \mathcal{X}$ will be exponentially driven by the controller (13) to the desired constant position $x_d \in \mathcal{X}$.

The following part concerns the trajectory tracking problem. Using the same variable $x_d(t)$ to present the desired trajectory and define their error as $e(t) = x(t) - x_d(t)$, then the following assumption of $x_d(t)$ is needed to design a robust controller.

Assumption 3. *The reference trajectory stays in $\mathcal{X}$ (i.e., $x_d(t) \in \mathcal{X}, \forall t > 0$), and is differentiable (i.e., $x_d \in C^1$).*

The above assumption implies that the first derivative of $x_d(t)$ should be bounded, i.e., $\exists \rho > 0$, such that

$$||\dot{x}_d||_\infty < \rho \tag{16}$$

We design the following controller of the sliding mode type [24]:

$$u = -\int_{t_0}^{t} \left[\frac{\partial \hat{f}}{\partial u}\right]^\top \left[\lambda_1 e(\tau) + \lambda_2 \left[sign(e_1), sign(e_2), sign(e_3)\right]^T\right] d\tau \tag{17}$$

where $\lambda_1 \geq 0$ and $\lambda_2 > 0$, and the following result can be stated.

Theorem 2. *For the studied soft robot described by (12), if the neural network approximates f with an accuracy index γ satisfying (14), then there exists a positive λ_2 satisfying*

$$\lambda_2 > \frac{\rho}{(\beta - \alpha\gamma)} \tag{18}$$

with α, β and ρ being defined in (4), (7) and (16), such that the proposed robust controller (17) can drive x to track the desired trajectory x_d in a finite time, i.e.,

$$\|x(t) - x_d(t)\|_2 = 0, \exists t > T$$

where T is the setting time.

Proof 2. *Note $e = x - x_d$ and choose the same Lyapunov function defined as follows*

$$V(e) = \frac{1}{2}e^T e = \frac{1}{2}\|e\|_2^2$$

Following the similar procedure as that in the proof of Theorem 1, we can obtain

$$\dot{V} = -\lambda_1 e^T J J^T e - \lambda_1 e^T J \Delta J^T e - \lambda_2 e^T J J^T w(e) - e^T \dot{x}_d$$

where $w(e) = \left[sign(e_1), sign(e_2), sign(e_3)\right]^T$ for simplicity. Note that

$$e^T w(e) = \|e\|_1 \geq \|e\|_2$$

then we have

$$\begin{aligned}\dot{V} &\leq -\lambda_1 \left[\beta - \alpha\gamma\right] \|e\|_2^2 - \lambda_2\beta\|e\|_2 + \lambda_2\alpha\gamma\|e\|_2 + \rho\|e\|_2 \\ &= -\lambda_1 \left[\beta - \alpha\gamma\right] \|e\|_2^2 - \left[\lambda_2\beta - \lambda_2\alpha\gamma - \rho\right] \|e\|_2\end{aligned}$$

Consequently, if the inequalities (14) and (18) are both satisfied, then we get

$$\dot{V} \leq -2\lambda_1\eta_1 V - \sqrt{2}\eta_2\sqrt{V}$$

with $\eta_1 = \beta - \alpha\gamma > 0$ and $\eta_2 = \lambda_2\beta - \lambda_2\alpha\gamma - \rho > 0$. This implies that there exists a finite time T such that $V(t) = 0, \forall t > T$, and this is equivalent to prove that $e(t)$ will converge to 0 in a finite time.

4.4. Robustness analysis

Let us consider now some disturbances ϵ (which might be either the external perturbation, the modeling error, or the approximation error introduced by ANN) are added into the system as:

$$x = f(u) + \epsilon$$

then system (12) can be rewritten as:

$$\begin{aligned} \dot{x} &= \frac{\partial f}{\partial u} v + \dot{\epsilon} \\ \dot{u} &= v \end{aligned} \tag{19}$$

It is evident that, if the disturbance ϵ is constant or piece-wise constant, then $\dot{\epsilon} = 0$ (or almost every where equal to 0 for the piece-wise constant case). In this case, the disturbed system (19) is exactly equivalent to system (12). Therefore, the designed controllers (13) and (17) can still drive the robot to the desired position or trajectory.

For the case where the disturbance is time-varying, the boundedness assumption should be imposed for $\dot{\epsilon}$ in order to analyze the robustness of the proposed controllers.

Assumption 4. *It is assumed that $\epsilon \in C^1$, and $\dot{\epsilon}(t)$ is bounded, i.e., $\exists \rho_\epsilon > 0$, such that $||\dot{\epsilon}(t)||_\infty < \rho_\epsilon$ for all $t > 0$.*

Based on the above assumption, let us firstly consider the controller (13) for the case where x_d is constant. Using the same Lyapunov function $V = \frac{1}{2} e^T e$, then we have

$$\begin{aligned} \dot{V} &= e^T \dot{e} = e^T \left[\dot{x} - \dot{x}_d \right] \\ &= e^T \frac{\partial f(u)}{\partial u} \dot{u} + e^T \dot{\epsilon} \end{aligned} \tag{20}$$

By following the same procedure as that in the proof of Theorem 1, we obtain

$$\dot{V} \le -2\lambda\eta V + e^T \dot{\epsilon}$$

where $\eta > 0$. Therefore, if Assumption 4 is satisfied, then by applying Young's inequality we have

$$\dot{V} \le -2\lambda\eta V + \kappa \|e\|_2^2 + \frac{1}{\kappa} \|\dot{\epsilon}\|_2^2 \le -2\lambda\eta V + \frac{\kappa}{2} V + \frac{1}{\kappa} \|\dot{\epsilon}\|_\infty^2 \le -(2\lambda\eta - \frac{\kappa}{4}) V + \frac{1}{\kappa} \rho_\epsilon^2$$

where $\kappa > 0$ is an arbitrary positive constant. The above inequality implies that x will be controlled via (13) to enter and stay in a neighborhood of x_d whose size is mainly determined by the boundedness of $\dot{\epsilon}$.

The same argument can be applied for the case of trajectory tracking. With the same Lyapunov function $V = \frac{1}{2} e^T e$, by applying the controller (17) to the disturbed system (19), we can then obtain

$$\dot{V} \le -2\lambda_1\eta_1 V - \sqrt{2}\eta_2 \sqrt{V} + e^T \dot{\epsilon} ||\dot{\epsilon}||_2^2 \le -2\lambda_1\eta_1 V - \sqrt{2}\eta_2 \sqrt{V} + \rho_\epsilon \sqrt{V} \tag{21}$$

This implies that, if the neural network gives a precise approximation such that the inequalities (14) and (18) are satisfied, then we can find an η_2 such that $\eta_2 > \rho_\epsilon / \sqrt{2}$ which will guarantee the finite-time convergence of $x(t)$ to $x_d(t)$.

Remark 1. *The proposed controller is to control the soft robot made by silicone with 4 actuators. This methodology can be easily generalized to control more general type of soft robots. However, the proposed approach works only if we can obtain a precise approximation model via the used neural network. In other words, during the training stage of neural network to get precisely approximated model, we need to actuate the robot to cover as largely as possible its workspace to collect enough input-output data set. This in fact might be the main disadvantage of such a method, since the model needs to be re-trained once the workspace is changed.*

Remark 2. *The above analysis shows that, the proposed controller (13) can eliminate constant (or piece-wise constant) disturbance (or the approximation error introduced by the neural network) with a usual gain since it is in fact an integral controller, but it can only attenuate time-varying bounded disturbance by using a high gain. If we want to eliminate as well time-varying disturbance (of type C^1), then we should use the controller (17) by combining the sliding mode technique.*

Figure 3: Experimental setup with the soft robot and an OptiTrack system

Remark 3. *Assumption 4 supposes that the disturbance (or uncertainty) is differentiable, which however in some situations is not true. For example the dry friction which leads to the dead-zone of servo-motor is indeed not differentiable, but piece-wise differentiable. In this case, big jumps will appear in $\dot{\epsilon}$ at the non-differentiable points, which will lead to positive large jumps of V in (21). One possible solution is to approximate as well this type of uncertainty, by using for example another neural network as presented in [25], and compensate it in the close-loop control design.*

5. Validation of the proposed approach

5.1. Experimental setup

In the experiment, the robot described in Figure 1 was printed by 3D printer with silicone. It has 4 holders which are linked with 4 independent cables, stretched by 4 independent motors inside the box. The height of the printed silicone robot is 12cm, and the farthest distance between two holders is 22cm.

In our test, we are interested in controlling the top point of such a silicone robot. Therefore, a white sensing ball was stuck on the top of the robot (see Figure 1). In order to capture the position of this sensing ball in real time, an OptiTrack system with 4 ultra-red cameras was installed around this robot. These cameras were fixed above the robot and can localize the position of the sensing ball with a high precision (in millimeter). The whole setup of the experiment is depicted in Figure 3.

5.2. Neural network training

Since the studied silicone robot has infinity numbers of degrees of freedom, it is therefore very hard to establish its model by using traditional analytical methods, such as Euler-Lagrange approach. Hence, as discussed in Section 4, this paper uses the neural network to approximate this model.

During the test, we first sent random but admissible values to the 4 motors (u) for the purpose of generating a corresponding static position x. This was regarded as a group of input-output data. In order to well train the neural network, we tried to cover the whole admissible workspace of the robot, by collecting 6000 groups of data.

Base on those data, an estimation of α and β was obtained as follows:

$$\alpha = 2.413, \text{ and } \beta = 0.166$$

Before training the neural network, we chose to minimize the mean square error between the real position (provided by OptiTrack system) and its approximation (using neural network) for all 6000 groups of data.

Generally, it is known that a more complicated network will often lead to a smaller mean square error. In our test, we adopted a neural network with 2 hidden layers, in which the first layer has 32 neurons and the second one has 40 neurons. Each neuron has the same sigmoid activation function. We chose this structure since the increase of the complexity for such a neural network does not make a big change in the final mean square error. As for the activation function, we used the sigmoid function as the input-output model moves smoothly without the sudden changes.

During the training process, we initialized the network with random weights and bias. Then, we adjusted these parameters little by little in order that the output could act as expected by the training data. Thanks to the smoothness

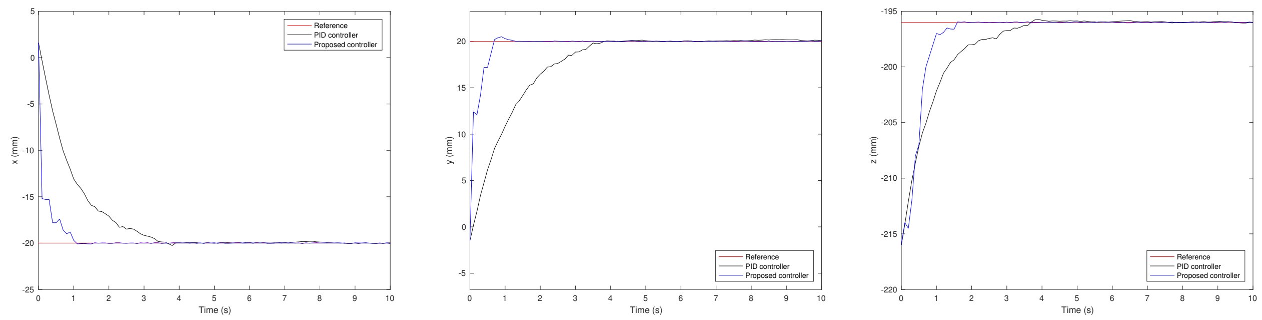

Figure 4: Performance comparison of the controller (13) and the PID.

of the sigmoid function, a small change of ω_i and b only corresponded a rather small change of output. With this characteristic, we were able to modify the output of the network for a certain input without interfering much the other inputs.

By iteratively tuning those parameters which was realized via classical gradient method, we finally obtained an acceptable mean square error which was equal to 0.0027, and it was regarded as small enough for our experiment. Based on the approximation function $\hat{f}$, we found out that

$$\gamma = 0.053 < \frac{\beta}{\alpha} = 0.0688$$

Therefore, the inequality (14) was satisfied, and then we would implement and validate the proposed controller (13) on the real silicone robot. It is noticed that the tuning process can be accelerated by using modern algorithm such as the adaptive population extremal optimization approach to improve the performance of the neural network [26].

5.3. Experimental result

A python script has been written for sending the command to the robot (according to the controller (13)) and for receiving the position data of the robot, which is provided by OptiTrack system.

For the experiments, the initial position of the ball stuck on the top of robot was $(1.6, -1.5, -216)$, and its desired position was set as $(-20, 20, -196)$. The unit of those values were millimeter. In the experiment, we chose $\lambda = 1.8$. It is worth noting that it is important to choose an appropriate value of λ. In fact, if this value is too big, the robot will over-react, which results in the visual oscillation. If it is too small, the convergence speed will be slow, i.e., it will take much time to drive x to its desired position.

Figure 4 shows the position of the ball stuck on the top of robot (blue line) and its desired value (red line) in $x-y-z$ axis, respectively. In the same figure, the performance of the proposed controller (13) is compared with that of the classical PID controller. We would like to emphasize that 12 gains need to be tuned for such a soft robot actuated by 4 cables when applying the PID controller, it is therefore not an easy task to find out the optimal gains. It can be seen that, for the PID controller, (x, y, z) converges to the desired position around $(3.5s, 5s, 3s)$ while the values for the proposed controller are around $(1s, 1.5s, 2s)$ with only very small overshoot for y. From the performance comparisons in Figure 4, it can be concluded that the controller (13) makes the point of interest converging to its desired position faster with a comparable precision.

In order to show the robustness of the proposed controller, another small piece of soft material has been stuck on the body of the soft robot which can be regarded as an external, permanent and constant disturbance. The experimental results have been depicted in Figure 5. It can be clearly seen that a disturbance was added around $5s$ during the experiment, and the proposed controller is robust in the sense that the top ball position still converges to the desired position after a small transition interval.

6. Conclusion

It is quite challenging to have the precise kinematic or dynamic model for a general shape of soft robot made by silicone, and that is the reason why researchers used the approximation method to obtain an approximated model,

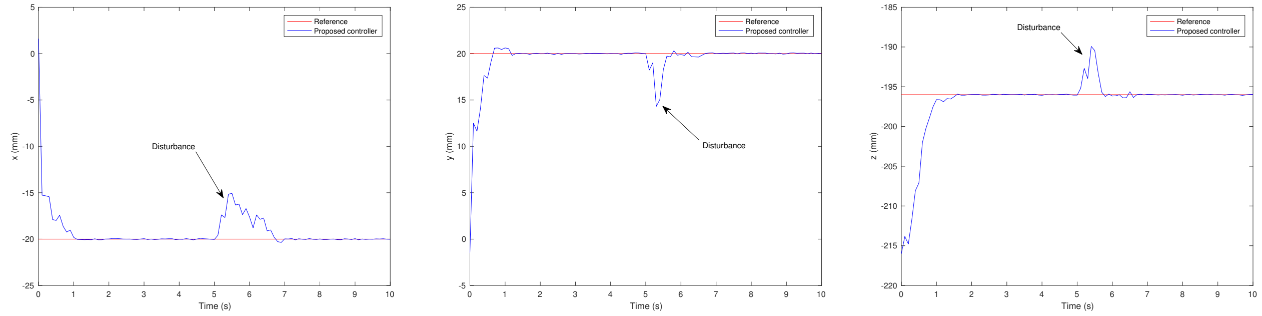

Figure 5: Robustness of the controller (13) when adding an external disturbance around $5s$.

based on which the controller can be then designed. This paper used an artificial neural network to learn the input-output model of soft robots. Based on the obtained approximated model, two robust controllers have been proposed, which can always guarantee the exponential or finite-time convergence of the interested point of soft robot to a desired reference (constant or time-varying). These results were proven by using Lyapunov function, and were validated by implementing them to control a real soft robot, made by silicone. As we have discussed in Remark 1 that the main disadvantage of this method is that the training phase for soft robots needs to collect enough data for the purpose of largely covering the robot's workspace, which is obviously time-consuming, therefore our future work will be to train the neural network by using a virtual soft robot modeled via high-dimensional FEM method. Instead of learning the input-output model and then designing the robust controller, another interesting research direction is to learn directly the robust controller for soft robots via the reinforcement learning technique.

Acknowledgments

This work is partially supported by project ROBOCOP [ANR PRCE 19 CE19] , Inventor (I-SITE ULNE, le programme d'Investissements d'Avenir, Métropole Européenne de Lille, France), and by project VALID (CPER DATA).

References

[1] S. Kim, C. Laschi, B. Trimmer, Soft robotics: a bioinspired evolution in robotics, Trends in Biotechnology 31 (5) (2013) 287 – 294.
[2] D. Rus, M. T. Tolley, Design, fabrication and control of soft robots, Nature 521 (2015) 467–475.
[3] S. Escaida Navarro, O. Goury, G. Zheng, T. Morales Bieze, C. Duriez, Modeling Novel Soft Mechanosensors based on Air-Flow Measurements, IEEE Robotics and Automation Letters (2019) In press.
[4] D. Trivedi, C. D. Rahn, W. M. Kier, I. D. Walker, Soft robotics: Biological inspiration, state of the art, and future research, Appl. Bionics Biomechanics 5 (3) (2008) 99–117.
[5] G. S. Chirikjian, J. W. Burdick, A modal approach to hyper-redundant manipulator kinematics, IEEE Transactions on Robotics and Automation 10 (3) (1994) 343–354.
[6] F. Boyer, M. Porez, W. Khalil, Macro-continuous computed torque algorithm for a three-dimensional eel-like robot, IEEE Transactions on Robotics 22 (4) (2006) 763–775.
[7] C. Della Santina, R. K. Katzschmann, A. Biechi, D. Rus, Dynamic control of soft robots interacting with the environment, in: 2018 IEEE International Conference on Soft Robotics (RoboSoft), IEEE, 2018, pp. 46–53.
[8] J. B. M. Siddharth Sanan, C. G. Atkeson, Robots with inflatable links, in: The 2009 IEEE/RSJ International Conference on Inte lligent Robots and Systems October 11-15, 2009 St. Louis, USA, IEEE, 2009.
[9] B. A. Jones, R. L. Gray, K. Turlapati, Three dimensional statics for continuum robotics, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 2659–2664. doi:10.1109/IROS.2009.5354199.
[10] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, C. Laschi, A 3d steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm, Bioinspiration & biomimetics 7 (2) (2012) 025006.
[11] G. Zheng, O. Goury, M. Thieffry, A. Kruszewski, C. Duriez, Controllability pre-verification of silicon soft robots based on finite-element method, in: Robotics and Automation (ICRA), 2019 IEEE International Conference on.
[12] Z. Zhang, J. Dequidt, C. Duriez, Vision-Based Sensing of External Forces Acting on Soft Robots Using Finite Element Method, IEEE Robotics and Automation Letters 3 (3) (2018) 1529 – 1536. doi:10.1109/LRA.2018.2800781.
[13] M. Thieffry, A. Kruszewski, O. Goury, T.-M. Guerra, C. Duriez, Dynamic control of soft robots, in: IFAC World Congress, 2017.
[14] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, C. Laschi, Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature, IEEE Transactions on Robotics 31 (4) (2015) 823–834.

[15] G. Runge, M. Wiese, A. Raatz, Fem-based training of artificial neural networks for modular soft robots, in: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2017, pp. 385–392.
[16] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, M. D. Killpack, Learning nonlinear dynamic models of soft robots for model predictive control with neural networks, in: 2018 IEEE International Conference on Soft Robotics (RoboSoft), IEEE, 2018, pp. 39–45.
[17] B. Siciliano, O. Khatib, Springer Handbook of Robotics, Springer-Verlag, Berlin, Heidelberg, 2007.
[18] P. A. Ioannou, J. Sun, Robust Adaptive Control, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.
[19] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359 – 366. doi:https://doi.org/10.1016/0893-6080(89)90020-8.
[20] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems (MCSS) 2 (4) (1989) 303–314. doi:10.1007/BF02551274.
[21] E. J. Hartman, J. D. Keeler, J. M. Kowalski, Layered neural networks with gaussian hidden units as universal approximations, Neural Computation 2 (2) (1990) 210–215. doi:10.1162/neco.1990.2.2.210.
[22] J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural Computation 3 (2) (1991) 246–257. doi:10.1162/neco.1991.3.2.246.
[23] A. Isidori, Nonlinear control systems (3rd edition), London: Springer-Verlag.
[24] V. I. Utkin, Sliding mode control design principles and applications to electric drives, IEEE transactions on industrial electronics 40 (1) (1993) 23–36.
[25] K. Guo, Y. Pan, H. Yu, Composite learning robot control with friction compensation: A neural network-based approach, IEEE Transactions on Industrial Electronics 66 (10) (2019) 7841–7851. doi:10.1109/TIE.2018.2886763.
[26] M.-R. Chen, Y.-Z. Lu, G. Yang, Population-based extremal optimization with adaptive lévy mutation for constrained optimization, in: International Conference on Computational and Information Science, Springer, 2006, pp. 144–155.