robotics

MDPI

Article

A Novel Actor—Critic Motor Reinforcement Learning for Continuum Soft Robots

Luis Pantoja-Garcia [1], Vicente Parra-Vega [1,*], Rodolfo Garcia-Rodriguez [2] and Carlos Ernesto Vázquez-García [1]

[1] Robotics and Advanced Manufacturing Department, Research Center for Advanced Studies (Cinvestav-Ipn), Ramos Arizpe 25903, Mexico; luis.pantoja@cinvestav.mx (L.P.-G.); ernesto.vazquezg@cinvestav.mx (C.E.V.-G.)

[2] Facultad de Ciencias de la Administración, Universidad Autónoma de Coahuila, Saltillo 25280, Mexico; rogarciar@gmail.com

* Correspondence: vparra@cinvestav.mx; Tel.: +52-844-4389600

Abstract: Reinforcement learning (RL) is explored for motor control of a novel pneumatic-driven soft robot modeled after continuum media with a varying density. This model complies with closed-form Lagrangian dynamics, which fulfills the fundamental structural property of passivity, among others. Then, the question arises of how to synthesize a passivity-based RL model to control the unknown continuum soft robot dynamics to exploit its input–output energy properties advantageously throughout a reward-based neural network controller. Thus, we propose a continuous-time Actor–Critic scheme for tracking tasks of the continuum 3D soft robot subject to Lipschitz disturbances. A reward-based temporal difference leads to learning with a novel discontinuous adaptive mechanism of Critic neural weights. Finally, the reward and integral of the Bellman error approximation reinforce the adaptive mechanism of Actor neural weights. Closed-loop stability is guaranteed in the sense of Lyapunov, which leads to local exponential convergence of tracking errors based on integral sliding modes. Notably, it is assumed that dynamics are unknown, yet the control is continuous and robust. A representative simulation study shows the effectiveness of our proposal for tracking tasks.

Keywords: reinforcement learning; Bellman error; continuum soft robot; constant curvature

check for updates

Citation: Pantoja-Garcia, L.; Parra-Vega, V.; Garcia-Rodriguez, R.; Vázquez-García, C.E. A Novel Actor—Critic Motor Reinforcement Learning for Continuum Soft Robots. *Robotics* **2023**, *12*, 141. https://doi.org/10.3390/robotics12050141

Academic Editor: Po-Yen Chen and Kean Aw

Received: 1 September 2023
Revised: 27 September 2023
Accepted: 8 October 2023
Published: 9 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The essential feature of RL is that it provides a reinforcement signal based on the value function evaluation to compute the present action aiming to learn a task. Such a brief statement deploys a powerful motor learning paradigm [1] from control application through the Actor–Critic scheme, well substantiated by adaptive dynamic programming [2], and optimal control [3] schemes. Recently, model-based (regressor) adaptive and model-free (neural networks) controls have been proposed to deal with uncertainty [4–6].

Conventional RL schemes typically require state and action space exploration, demanding massive trials and data to tune and test the system in broad operational conditions. It makes conventional RL an option for some software applications, but are risky for hardware systems. Then, it has been claimed that further research is needed to introduce explicit stability bounds and clear implementation procedures to implement RL schemes for a physical system, such as a robot. However, a distinct feature of the RL massive literature is lack of stability conditions [7], with few exceptions. Then, translational research is required to yield a novel RL scheme with stability analysis, particularly for highly nonlinear systems such as robots, but stability is definitively a requirement for uncertain and disturbed complex deformable (soft body) robots.

We are interested in the particular class of pneumatic-driven continuum soft robots. For these robots, implementing a conventional RL is prone to failure when attempting to operate it away from its narrow operational margin, thus requiring novel RL designs equipped

with stability analysis. The stability analysis does not represent a nuisance, unnecessary design requirement, or an elegant, irrelevant usage of mathematics; by contrast, stability is tantamount to guaranteeing the operation of the systems under certain conditions. Thus, stability analysis is required for any novel motor control scheme for nonlinear uncertain models, such as continuum soft robots. Given that RL aims at computing the present action that yields the desired state of a system at a given cost, computational architectures have been studied to explore to state-action space, which has led to software architectures rather than RL's stability conditions. Thus, large batches of trials, even millions, are regularly carried out for a particular system until it learns the task and a particular set of discrete admissible controls. However, when such a system is a physical entity, like a robot, there is no room for such trials, but it must operate within stability margins. The trial-and-error mechanistic approach of RL tuning is forbidden for robots since fatal failure may arise. Unfortunately, stability requirements for RL applications for robots have not permeated into the computational RL community; better phrased, why does the immense majority of RL literature lack stability? It is worrisome that this fact is not a worry, but rather the norm, in the practice of RL for robots [7], including for deformable (soft) robots [8]. Although the early approaches of RL dynamical systems are used, the principal developments in recent years have improved the algorithms on a finite set of states and controls.

Nonetheless, RL has an enormous advantageous prevalence over "conventional" control in the sense that RL deals with two metrics (performance index and tracking error) while conventional control deals with only one metric (tracking error). RL evaluates performance to issue a reinforced signal to the action (control), which seems very interesting for novel systems, such as the continuum soft robot subject to varying density and a non-constant center of mass [9]. Then, the question arises of how to design a sound (stability-based) RL scheme considering its deformation, which is the essential feature that distinguishes it against conventional rigid-body robots.

In this paper, we entertain the explicit need for rigorous stability of a model-free RL scheme for continuum soft robots [9], where reward evaluates continuous deformation coordinates. Our proposed scheme is similar to adaptive neurocontrol away from optimal-like control. However, it differs from the former because a second neural network (the Critic or Critic NN) aims to enforce the asymptotic stability of the temporal difference error equation in continuous time. Additionally, a so-called Actor NN aims at inverse dynamics compensation. This scheme is called the Actor–Critic Learning of Motor Control. Unlike traditional Actor–Critic schemes [10,11], novel adaptive mechanisms are introduced for neural weights that allows tightly and complex intertwined nonlinear Actor–Critic neural architectures to emerge, substantiated by the closed-loop stability analysis for tracking tasks of the uncertain continuum 3D soft robot subject to Lipschitz disturbances.

Contribution and Organization

The contribution amounts to a novel RL scheme for a novel continuum soft robot [9], using a particular actuation topology [12]. RL's relevant role evaluation performance is exploited to reinforce the neurocontroller based on the approximation of Bellman's temporal difference. It represents the accumulated reward-based value function, approximated by the Critic NN using a novel adaptive mechanism of weights with nonlinear neural activations, while the Actor NN compensates approximately for inverse dynamics. A chatterless integral sliding mode is introduced to guarantee error tracking [13]. Overall, tracking with performance evaluation is guaranteed, assuming no knowledge of complex dynamics subject to disturbances, yet with a smooth control action.

This manuscript is organized as follows. Section 2 introduces the preliminaries and problem statement. Section 3 presents the RL design, with stability analysis in the Appendix. Simulations are presented in Section 4 for a 3D continuum soft robot motion tracking an aggressive trajectory, with discussions of the overall scheme presented in Section 5. Finally, concluding remarks are given in Section 6, addressing some advantages and concerns of the proposed scheme.

2. Preliminaries and Problem Statement

Interestingly, there exist hundreds of papers indexed in the academic metasearch engine Scopus under the keywords RL and soft robot addressing a variety of RL implementations to non-rigid-body systems; however, only 24 papers deal precisely with soft-robots; only two mention stability [14,15]; yet, none included any formal stability analysis. Though this amazing body of literature is rapidly changing and will hopefully soon be address stability, this situation speaks for itself about the significant worldwide effort to exploit the powerful characteristics of RL to soft robots. However, it also shows that the solid foundations of RL are taken for granted for novel applications; for example, it meets the theoretical assumptions of the original approach, but is without any rigor to study the subtleties on how to include the differences of each novel system with stability. Not only that, but stability analysis also paves the way to substantiate a specific design within the specificities of each new system; that is, stability analysis tailors the design according to what a particular system is. The result is an efficient RL custom design for the system under study, not a general design for a particular system, which typically leads to conservative and inefficient RL designs. Overall, this brief literature assessment shows the tendency of RL implementations to a large class of systems, including deformable-body robots. This approach limits and weakens its effectiveness since, in practice, each new system differs in many aspects from the ideal one considered initially. Impressive empirical results of the literature empower RL as a viable option worth studying; however, its prime will be highlighted, arguably, when synthesized through stability analysis to deliver an asymptotically stable RL approach for a specific system.

2.1. On Continuum Soft Robots

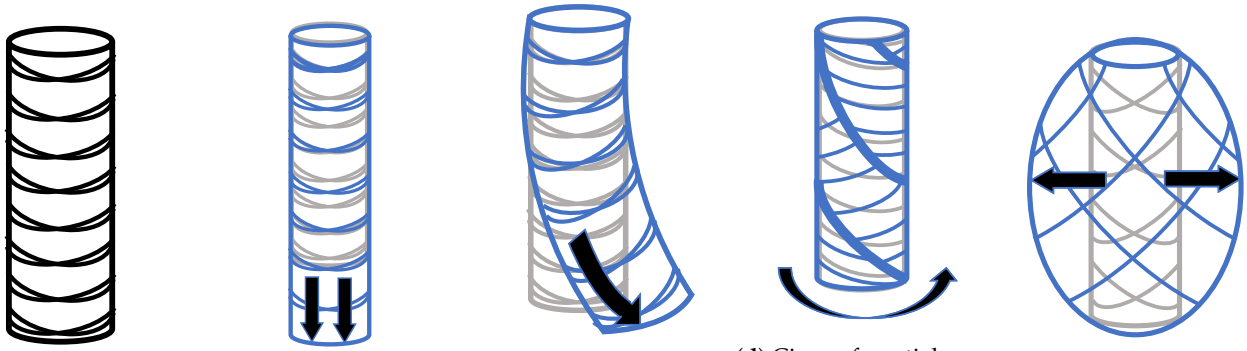
Soft robots can be classified by their actuation into four types [16]:

- Fluidic Elastomer soft robots (FESRs): This type of soft robots has pneumatic/hydraulic chambers embedded into their bodies, which induce body deformation when pressurized. It can generate movements such as bending, elongation, torsion, and a combination of these movements. For example, the STIFF-FLOP comprises a series of identical elastomeric soft actuators with internal pneumatic chambers to unlock three-dimensional movement and a central chamber for stiffness variation via granular interference phenomena [17].
- Cable-driven soft robots (CDSRs): The robot has external or internal cables that generate deformation by tension variation. However, the type of movement and workspace depend on the number and position of cables, which means that there are more control inputs and rigid elements where cables pivot. Furthermore, the exerted force of this type of actuator depends directly on cables tension and not on stiffness. An example is depicted in [18], where a four-cable-driven soft arm is presented.
- Shape-memory polymer soft robots (SMPSRs): This type encompasses robots composed of polymers with a thermally induced effect, which allows them to go from an initial state to a deformed state. However, SMPSRs do not produce high strain and are usually applied when small deformations are required.
- Dielectric/electroactive polymer soft robots (D/EPSRs): This type of robots are based on deformation phenomena in response to electricity. However, due to their high voltage amplification, their doped elastomer is the most disadvantageous and risky option.

By comparing the actuation mechanisms of these types of soft robots, it is recognized that FESRs have the best relationship between applied force and deformation, given that applied energy (either pneumatic or hydraulic) continuously deforms the elastomer, translating into viscoelastic forces of continuum media, i.e., a change in the distance between each pair of particles in the material. On the other hand, there are three types of morphologies for soft robots that show continuous deformation [19]:

- Cylindrical morphology: The robot's body is shaped like a cylinder of elastomeric material, with pressure inputs (chambers) radially distributed along an internal radius. When a chamber is pressurized, the body presents a controlled curvature along the extensible center of the robot. Usually, this morphology is built using inextensible braided threads to mitigate radial and circumferential deformations so that the robot's configuration can be approximated with a minimum set of linearly independent variables principally used as control inputs actuated by pneumatic chambers.
- Ribbed morphology: The robot is composed of three elastomer-based layers. The top and bottom layers have internal ribbed-like structures with multiple rectangular channels connected to fluid transmission lines, whereas the middle layer is a flexible but inextensible restriction. In an active state, where fluid pressurizes a group of chambers, bending is produced. An example is presented in [20] with a soft arm of six ribbed-like segments designed as a manipulation system.
- Pleated morphology: Consists of discrete sections (plates) of elastomeric materials evenly distributed and separated by gaps. At the bottom part, a high-stiffness silicon layer is used to work as an inextensible restriction. Additionally, the top part has hollow cavities (in each plate) connected to a central chamber. When it gets pressurized, each plate experiences balloon-like deformations translated into bending of the high-stiffness silicon layer along the direction of the layer with lower stiffness. An example is presented in [21], where a soft manipulator has six segments with cylindrical cavities, and a pleated-shaped soft gripper is used for grasping purposes.

Among these morphologies, the cylindrical morphology is the one that allows to approximate robot's deformation through a finite number of variables due to the radial distribution of their pressure entries while simultaneously allowing relatively easy characterization of their geometric variables. Additionally, four types of movement can be distinguished [22]: axial (elongation and retraction of length; see Figure 1b), bending (see Figure 1c), circumferential (translated as torsion; see Figure 1d) and radial (expansion and contraction of cross-sectional area; see Figure 1e). Notice that these movements are achieved by imposing different deformation restrictions.



(**a**) Undeformed state (**b**) Axial deformation (**c**) Bending (**d**) Circumferential deformation (**e**) Radial deformation

Figure 1. Characteristic deformations of a cylindrical-shaped soft robot.

Thus, we consider in this paper the soft robot defined as a cylindrical-shaped soft body composed of elastomeric material moving from continuous controlled body deformation. Moreover, we refer to a continuum soft robot as a soft robot with continuous infinitesimal deformation of the distance between their particles, and it must not be confused with what was referred to as a continuum robot 20 years ago [23].

2.1.1. Deformation Coordinates

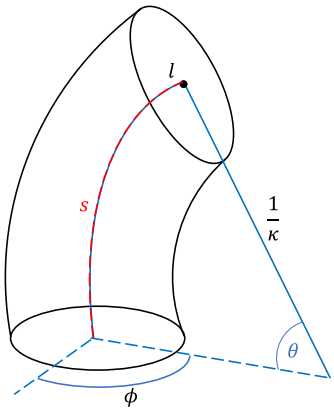
The proposed soft robot has circumferential and radial restrictions so that it can only have axial and bending deformations, i.e., increasing (or decreasing) its length l and performing flexion in the direction of an azimuth angle ϕ (notice that this is achieved

when two or more internal chambers are activated). Given these constraints, we consider constant cross-sectional geometry, which gives rise to defining a curved central axis that passes through the body's geometric center, known as the backbone (see Figure 2a) [24]. By definition, soft robots have variable curvature along the backbone, i.e., for each cross section of the body, there exists a different curvature. To generate low-cost computation modeling, a constant-curvature approach is used, assuming a single s-curve parameterized by an arc (see Figure 2a), which resembles the body as a segment, hence giving a single curvature κ along the segment, which may vary along time. Therefore, a vector of deformation coordinates q_e is defined as

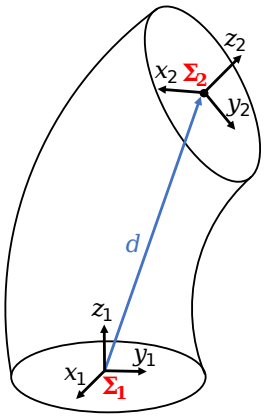
$$q_e = (l \quad \phi \quad \kappa)^T. \tag{1}$$

The constant curvature parameterizes the backbone of a soft robot by a radius of curvature $r_k = \frac{1}{\kappa}$ and a curvature angle $\theta = \kappa l$. On the other hand, the constant-curvature approach has the advantage of enabling an additional space named Actuation space ($\mathcal{AS}$), defined by the l vector which contains all length variables $l_1, l_2, \ldots, l_n$ corresponding to the n-actuation elements of the robot. Hence, two direct kinematic mappings arise:

- From actuation space l to configuration space q_e ($\mathcal{AS} \rightarrow \mathcal{CS}$), related to the actuation mechanism, which in this case is the length of chambers. It is also known as specific mapping.
- From configuration space q_e to operational space x ($\mathcal{CS} \rightarrow \mathcal{OS}$), better known as direct kinematics [25].



(**a**) Deformation coordinates q_e for a constant-curvature segment s (in red the *backbone*).



(**b**) Direct kinematics for a constant-curvature soft robot.

Figure 2. Deformation coordinates with respect to referential frames Σ_1, Σ_2.

2.1.2. Kinematics

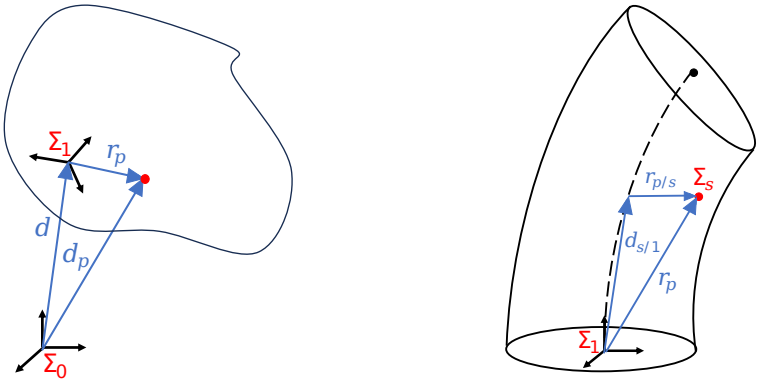
Two frames can describe zeroth-order direct kinematics of a constant-curvature soft robot: the inertial one Σ_1 at the base of the *backbone* and the distal reference Σ_2 at end-effector's frame, which can be seen in Figure 2b. Consider deformation coordinates q_e and the following homogeneous transformations:

$$\begin{aligned} T(q_e) &= \begin{bmatrix} R_{z,\phi} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_{y,\kappa l} & e \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_{z,-\phi} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} S_\phi^2 + C_\phi^2 C_{\kappa l} & -C_\phi S_\phi V_{\kappa l} & S_{\kappa l} C_\phi & C_\phi \frac{V_{\kappa l}}{\kappa} \\ -C_\phi S_\phi V_{\kappa l} & C_\phi^2 + S_\phi^2 C_{\kappa l} & S_{\kappa l} S_\phi & S_\phi \frac{V_{\kappa l}}{\kappa} \\ -S_{\kappa l} C_\phi & -S_{\kappa l} S_\phi & C_{\kappa l} & \frac{S_{\kappa l}}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2}$$

where e is the distal position point obtained by $e = (r_k(1 - cos\theta) \quad 0 \quad r_k sen\theta)^T$ and $C_x = cos(x)$, $S_x = sin(x)$, $V_x = 1 - cos(x)$. Let Cartesian position $d_p^{(0)}$ of a particle p be within the soft body with respect to an inertial reference frame Σ_0 as an analog to movement transformation of a rigid body, i.e.,

$$d_p^{(0)} = d + R_0^1(\theta) r_p^{(1)}, \tag{3}$$

where d is the body's position, and r_p is the relative position of point p with respect to local reference frame Σ_1; see Figure 3 (for the sake of simplicity, Equation (3) is going to be used without superscripts, i.e., $d_p = d + Rr_p$).



(**a**) Particle position on the solid body.

(**b**) Position inside a soft body with respect to a referential frame associated with the body.

Figure 3. Position of a particle in a continuum soft body.

Particle's velocity $\dot{d}_p$ is easily obtained by the time derivative of its position, Equation (3), resulting in

$$\dot{d}_p = \dot{d} + \dot{R} r_p + R\dot{r}_p. \tag{4}$$

It is important to remark that for rigid bodies $R\dot{r}_p = 0$, the particles position with respect to the inertial reference frame Σ_1 stays constant during transformation. However, this does not occur for soft robots because the distance between particles is variable due to elastic deformation of the body such that $R\dot{r}_p \neq 0$. Additionally, $\dot{R}$ can be expressed in terms of angular velocity ω, using equivalence $\dot{R} = [\omega^{(0)}\times]R = R[\omega^{(1)}\times]$, so that a particle's velocity in a soft body is declared as:

$$\dot{d}_p = \dot{d} + \omega^{(0)} \times r_p^{(0)} + R\dot{r}_p \tag{5}$$

and

$$v_p = v + \omega \times r_p + \dot{r}_p, \tag{6}$$

in inertial and local coordinates, respectively. Now, vector r_p can be calculated as a function of deformation coordinates q_e expressed in toroidal coordinates system $c = (r \quad \psi \quad \mu)^T$, where r and ψ are the radius and angle which allow it to be positioned in any point of a cross-sectional area within the soft robot; and $\mu \in [0 \quad 1]$ is the variable which parameterizes an arc length segment (note that $\mu = 1$ is the distal point). Therefore, r_p is composed by

$$r_p(q_e, c) = d_{s/1}(q_e, \mu) + R_1^s(q_e, \mu) \begin{pmatrix} rC_\psi \\ rS_\psi \\ 0 \end{pmatrix}. \tag{7}$$

where $d_{s/1}$ gives the spatial position of a point s in the *backbone*, and $r_{p/s}$ positions any point along the s-cross section.

From (7), the relative velocity $\dot{r}_p$ of a particle can be obtained by taking its partial derivative with respect to q_e:

$$\dot{r}_p(q_e, \dot{q}_e, c) = \frac{\partial r_p(q_e, c)}{\partial q_e}\dot{q}_e = J_{v_p}\dot{q}_e. \tag{8}$$

where J_{v_p} is the deformation Jacobian given by

$$J_{v_p} = \begin{bmatrix} \mu \cdot sin(\kappa\mu l)cos(\phi)(1-\kappa r \cdot cos(\phi-\psi)) & \frac{(cos(\kappa\mu l)-1)(sin(\phi)-\kappa r \cdot sin(2\phi-\psi))}{\kappa} & -cos(\phi)a_1 \\ \mu \cdot sin(\kappa\mu l)sin(\phi)(1-\kappa r \cdot cos(\phi-\psi)) & -\frac{(cos(\kappa\mu l)-1)(cos(\phi)-\kappa r \cdot sin(2\phi-\psi))}{\kappa} & -sin(\phi)a_1 \\ \mu \cdot cos(\kappa\mu l)(1-\kappa r \cdot cos(\phi-\psi)) & r \cdot sin(\phi-\psi)sin(\kappa\mu l) & a_2 \end{bmatrix}, \tag{9}$$

with $a_1 = \frac{\kappa^2\mu l r S_{\kappa\mu l}C_\phi C_\psi - \kappa\mu l S_{\kappa\mu l} - C_{\kappa\mu l} + \kappa^2\mu l r S_{\kappa\mu l}S_\phi S_\psi + 1}{\kappa^2}$, and $a_2 = -\frac{S_{\kappa\mu l} - \kappa\mu l C_{\kappa\mu l} + \kappa^2\mu l r C_{\phi-\psi}C_{\kappa\mu l}}{\kappa^2}$.

2.1.3. Dynamics

Consider the integral Lagrangian model based on the D'Alembert–Lagrange equation to describes deformation of a cylindrical-shaped pneumatic soft robot of constant-curvature in an inertial base [9]:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - \tau_v = \tau, \tag{10}$$

with $q = \begin{pmatrix} l & \phi & \kappa \end{pmatrix}^T \in \mathbb{R}^3$ being the three-dimension vector of generalized coordinates, $H(q) \in \mathbb{R}^{3\times3}$ the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{3\times3}$ Coriolis matrix, $g(q) \in \mathbb{R}^3$ the gravity vector, and $\tau_v \in \mathbb{R}^3$ the generalized viscoelastic force vector which is assumed to be separated and oversimplified in a pure linear viscous friction term and an elastic restorative one of the form $\tau_v = -D_v\dot{q} + \tau_e$ with positive semi-definite viscous matrix gain $D_v = (d_{q_1}, d_{q_2}, d_{q_3})^T$ and generalized elastic forces τ_e. The Lagrangian dynamic model (10) has the following properties [9]:

- Symmetry and definite positiveness of inertia matrix: $H(q) = H^T(q), \quad H(q) > 0, \quad \forall q.$
- Skew symmetry of Coriolis matrix: $C(\cdot) + C(\cdot)^T = \dot{H}(q)$.
- Passivity: $\int_{t_0}^{t_f} \tau \cdot \dot{q}\, dt = E(t_f) - E(t_0) \geq -E(t_0)$, for any $E(t_0)$.

Elastic forces τ_e are obtained via elastic function $U_e(q) = \frac{1}{2}\frac{AE}{l_0}(l-l_0)^2 + \frac{1}{2}\frac{IE}{l_0}\left(\frac{\kappa l}{2} - \beta_0\right)^2$ proposed in [26] and Castigliano's theorem [27], as

$$\tau_e = \frac{\partial U_e}{\partial q} = \begin{pmatrix} \frac{AE}{l_0}(l-l_0) + \frac{IE}{l_0}\frac{\kappa}{2}\left(\frac{\kappa l}{2} - \beta_0\right) \\ 0 \\ \frac{IE}{l_0}\frac{l}{2}\left(\frac{\kappa l}{2} - \beta_0\right) \end{pmatrix}.$$

2.1.4. Affine Actuation

By considering a pneumatic soft robot with c embedded cylindrical-shaped pneumatic chambers, air injection produces a controlled force field $p = \begin{pmatrix} p_1 & p_2 & \dots & p_c \end{pmatrix}^T \in \mathbb{R}^c$ which causes coupled deformation among all chambers inside the body. Thus, a mapping from c pressure vectors to the n generalized force coordinates can be defined as

$$\tau = B(q)p, \tag{11}$$

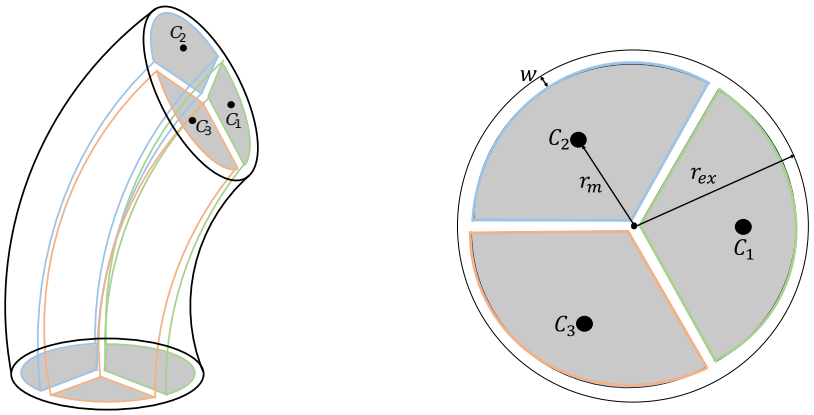
where $B(q) \in \mathbb{R}^{n\times c}$ is the input matrix as a lineal operator given by [9]:

$$B(q) = \frac{\partial}{\partial p}\left\{\frac{\partial U_p}{\partial q}\right\} = \begin{bmatrix} \dots & \frac{\partial V_i(q)}{\partial q} & \dots \end{bmatrix} \in \mathbb{R}^{n\times c} \tag{12}$$

In this work, a three-DoF soft robot is considered, with three identical prismatic-shaped chambers with transverse area A_c evenly positioned such that all centroids of the chambers' area are placed along the circumference described by a radius r_m, see Figure 4b. Thus, the input matrix $B(\cdot) \in \mathbb{R}^{3\times3}$ is full rank and is rewritten as [12]

$$B(q) = A_c \begin{bmatrix} 1-\kappa r_m \cos(-\phi) & 1-\kappa r_m \cos(\frac{2\pi}{3}-\phi) & 1-\kappa r_m \cos(-\frac{2\pi}{3}-\phi) \\ -\kappa l r_m \sin(-\phi) & -\kappa l r_m \sin(\frac{2\pi}{3}-\phi) & -\kappa l r_m \sin(-\frac{2\pi}{3}-\phi) \\ -l r_m \cos(-\phi) & -l r_m \cos(\frac{2\pi}{3}-\phi) & -l r_m \cos(-\frac{2\pi}{3}-\phi) \end{bmatrix}, \tag{13}$$

where $A_c(r_{ex}, w) = \frac{\pi}{3}(r_{ex}-w)^2, r_m(r_{ex}, w) = \frac{2}{\pi}(r_{ex}-w)\sin(\frac{\pi}{3})$.



(**a**) The structure of the robot with each pneumatic chamber represented by a distinct color.

(**b**) Cross section showing chambers distribution, the radius of chambers centroid r_m, external radium r_{ex}, and wall width w.

Figure 4. Soft robot proposed geometry.

2.2. Open-Loop Error Equation

Adding and subtracting the functional

$$Y_r = H(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + D_v\dot{q}_r + g(q) \tag{14}$$

to (10), we have the open-loop error equation

$$H(q)\dot{S}_r + C(q,\dot{q})S_r + D_v S_r = \tau + \tau_e - Y_r, \tag{15}$$

where the extended velocity error coordinate is

$$S_r = \dot{q} - \dot{q}_r \tag{16}$$

for $\dot{q}_r$, the continuous nominal reference to be defined. System (15) has mainly been used for control design for many Lagrangian systems, even in neuro-control applications. In the latter case, [28] proposes an adaptive neurocontroller with an underlying integral sliding mode to enforce robust error tracking, which does not require training nor any knowledge of the robot with a smooth control actions.

2.2.1. Nominal Reference Design to Induce Integral Sliding Modes

Let the nominal reference be defined as

$$\dot{q}_r = \dot{q}_d - \alpha\Delta q + S_d - K_i \int \mathrm{sgn}(S_q), \tag{17}$$

where $\Delta q = q - q_d$ is the position error; q_d and $\dot{q}_d$ are the desired position and velocity, respectively; and α and K_i are positive feedback gains, $S_q = S - S_d$, with $S = \Delta\dot{q} + \alpha\Delta q$,

$S_d = S(t_0)e^{-\kappa t}$. Substituting (17) into (16), an extended velocity error coordinates is obtained as

$$S_r = S_q + K_i \int \operatorname{sgn}(S_q). \tag{18}$$

2.2.2. Control Design

Based on the seminal work of [13] for rigid robots and extended for soft robot (10) in [9], now, in this paper, given the non-intuitive free-form deformation of the continuum soft robot, we wonder how to consider the body deformation in the control design to guarantee $S \to 0$ with performance evaluation. More precisely, we are interested in designing τ for unknown (10), considering how deformation coordinates perform, additionally to guarantee tracking error convergence.

2.3. Problem Statement

We are interested in how to control soft robots using reinforcement learning tools, explicitly using the Actor–Critic scheme. Necessarily, it implies introducing an additional metric for task-performance evaluation along with error convergence. Thus, the following problem arises:

"Design a learning mechanism that guarantees simultaneous error convergence and task performance of a closed-loop pneumatic-driven soft robot through a control law that evaluates online learning units for a model-free scheme."

3. Actor–Critic Learning of Motor Control

Now, we proceed to explain in detail the proposed Actor–Critic architecture as well as the main result called Reinforced Neurocontroller; see Figure 5.

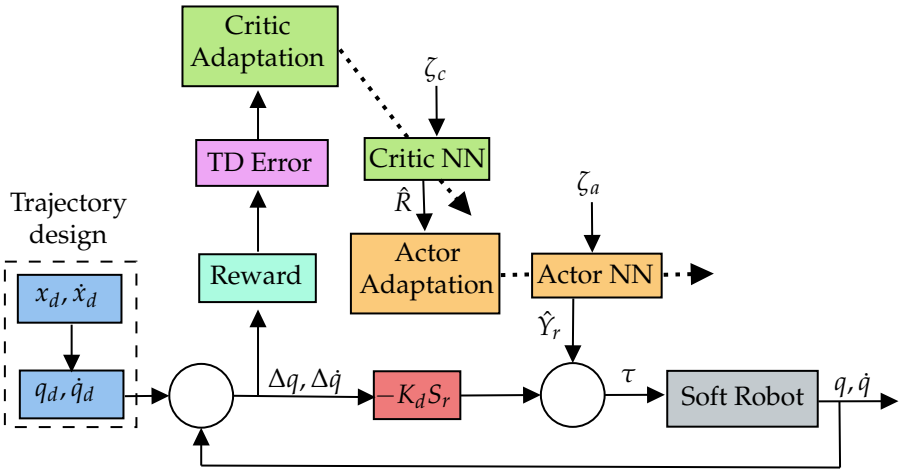


Figure 5. The proposed Actor–Critic scheme where each colored block corresponds to a specific role in the scheme. Notice that it keeps the conventional neurocontroller architecture approximating $\hat{Y}_r$.

3.1. Reward-Based Value Function and Temporal Difference Error

Consider the following continuous value function R that depends only on the task-dependent scalar reward $r \in \mathbb{R}$ [29];

$$R = \int_{t_0}^{\infty} e^{-\frac{m-t}{\psi}} r(m) dm, \tag{19}$$

where ψ is the time constant for discounting future rewards with $t \leq m \leq \infty$. Differentiating (19), one obtains

$$\dot{R} = \frac{1}{\psi} R - r(t), \tag{20}$$

which can be rewritten in terms of an error δ, as the so-called temporal difference error for continuous time [29],

$$\delta = \dot{R} - \frac{1}{\psi}R + r. \tag{21}$$

This consistency equation is used to approximate value function (19) based only on reward information.

Remark 1. *Notice that in the proposed Actor–Critic scheme, the reward and the temporal difference error implement the learning mechanism in the Critic NN that approximates the value function $\hat{R}$, which in turn represents the reinforcement signal used to improve the adaptation mechanism of the Actor NN that approximates Y_r by $\hat{Y}_r$.*

3.2. Critic NN

Given that value function from Equation (19) is smooth, it can be approximated by a neural network with finite constant weights $W_c \in \mathbb{R}^c$ and input basis $Z_c(\cdot) \in \mathbb{R}^c$ such that

$$R = W_c^T Z_c(\cdot) + \epsilon_c, \tag{22}$$

where a small ϵ_c is the neural approximation error. Then, there exists a neural network with adaptive weights $\hat{W}_c \in \mathbb{R}^c$ that approximates (22) as follows

$$\hat{R} = \hat{W}_c^T Z_c(\cdot) \tag{23}$$

where $Z_c(\cdot) = \sigma(V_c^T \zeta_c) \in \mathbb{R}^c$ is the sigmoid bipolar activation function, with $V_c \in \mathbb{R}^{3\times c}$ representing fixed weights and $\zeta_c \in \mathbb{R}^c$ being the input vector. It means that the learning has been made. Now, using (23) instead of (22) by the equivalence principle, the temporal difference error (21) can be written as

$$\begin{aligned}\hat{\delta} =& \dot{\hat{R}} - \frac{1}{\psi}\hat{R} + r \\ =& \dot{\hat{W}}_c^T \sigma(\cdot) + \hat{W}_c^T \dot{\sigma}_c(\cdot) - \frac{1}{\psi}\hat{W}_c^T \sigma_c(\cdot) + r.\end{aligned} \tag{24}$$

Notice that $\hat{\delta} \neq 0$ per se (it is in the neural error domain); thus, the problem becomes in designing the adaptation $\dot{\hat{W}}_c$ such that $\hat{\delta} \to 0$, which translates into an appropriate approximation of (19) by (23). Now, we have the following result.

Proposition 1. *Consider the following adaptation law*

$$\dot{\hat{W}}_c = -K_w sgn(\hat{W}_c) - K sgn(\hat{\gamma}) \frac{\sigma_c(\cdot)}{\sigma_c^T(\cdot)\sigma_c(\cdot)}, \tag{25}$$

and

$$\hat{\gamma} = \hat{R} - \frac{1}{\psi}\int_{t_0}^{t_f} R + \int_{t_0}^{t_f} r, \tag{26}$$

which comes from the temporal difference error $\hat{\delta}$. Then, selecting K_w and K large enough, the convergence of the temporal difference error is achieved such that $\hat{\delta} \to 0$.

Proof. See Appendix A.1. □

3.3. Actor NN

Let (14), according to the NN approximation property, be the continuous nonlinear function; it can be represented as $Y_r = W_a^T Z_a(\cdot) + \epsilon_a$, with $W_a \in \mathbb{R}^{a\times 3}$ finite constant

weights, $Z_a(\cdot) \in \mathbb{R}^a$ being the input basis, and ϵ_a being the reconstruction error. Then, the following neural network and adaptation law are proposed

$$\hat{Y}_r = \hat{W}_a^T Z_a(\cdot), \tag{27}$$

$$\dot{\hat{W}}_a = -\Gamma_a \sigma_a^T(\cdot) S_r^T - \Gamma_a \hat{W}_a (\hat{\gamma} r)^2, \tag{28}$$

where $\hat{W}_a \in \mathbb{R}^{a\times 3}$ is the matrix of adaptive weights, $Z_a(\cdot) = \sigma_a(V_a^T \zeta_a) \in \mathbb{R}^a$ is the bipolar sigmoid activation function, with $V_a \in \mathbb{R}^{4\times a}$ being the matrix of fixed weights and $\zeta_a \in \mathbb{R}^4$ the input vector, and $\Gamma_a \in \mathbb{R}^{a\times a}$ is positive definite gain.

3.4. Passivity-Based Reinforced Neurocontroller

Let the control signal be defined as

$$\tau = -K_d S_r + \hat{Y}_r, \tag{29}$$

with $K_d > 0 \ \in \mathbb{R}^{3\times 3}$. Then, the following main results are in order.

Theorem 1. *Consider the soft robot dynamics* (10) *in a closed-loop with the control signal* (29) *and adaptation laws* (25) *and* (28)*. Thus, from Proposition 1, for high enough K_i and K_d gains, exponential convergence of the tracking errors is guaranteed via integral sliding modes with smooth control signals and without knowledge of the soft robot dynamics.*

Proof. See Appendix A.2. □

4. Numerical Simulations

4.1. The Simulator and Parameters

The soft robot aims to track a tornado-like trajectory at a distal point described by a desired pose X_d; see Figure 6. Notice that X_d was mapped into generalized coordinates q_d via first-order inverse kinematics. Simulations were carried out in Matlab-Simulink 2021b with solver ode23tb running at an adaptive step sampling for a tolerance of 1×10^{-3}. Table 1 shows the dynamic parameters and desired trajectory of the soft robot, with the Young's modulus value being based on experiments [30] using Ecoflex 00–30™. Initial bending β_0 was calculated through initial length and curvature as $\beta_0 = \dfrac{l_0 \times \kappa_0}{2}$.

Table 1. Soft robot parameters and the tornado-like desired trajectory.

Variable	Description	Value
r_{ex}	External radius	0.1 m
w	Wall width	0.001 m
l_0	Initial length	0.9 m
β_0	Initial bending	0.6285 rad
E	Young's modulus	0.15 MPa
$X_d = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix}$	Desired pose	$\begin{bmatrix} 0.7\sin(t/50) + 0.01t\sin(t) + 0.2 \\ 0.7\cos(t/50) + 0.01t\cos(t) - 0.5 \\ 0.1 + 0.01t \end{bmatrix}$

4.2. Neural Network Architectures

The Critic NN has only one hidden layer, with input vector $\zeta_c = (-1 \ \Delta q)$, where weights connect the input layer to the hidden layer with constants V_c; initial adaptive weights $\hat{W}_c(t_0)$ are tuned between 0 and 1. Similarly, the Actor neural network has one hidden layer with input vector $\zeta_a = (1 \ \int S_r)$, where the weights connecting the input layer and the hidden layer V_a are fixed.

4.3. Reward Design

Given that the reward function, r, encodes critical aspects to motivate the fulfillment of the task, and the soft robots are typically equipped with low resolution (due to embedded sensor technology being in progress), it is worth considering weight position over velocity errors. Then, let the reward function be

$$r = \frac{1}{2}\left(\Delta q^T P \Delta q + \Delta \dot{q}^T Q \Delta \dot{q}\right) \tag{30}$$

where $P = diag(9, 9, 9)$, $Q = diag(1, 1, 1)$.

4.4. Feedback Control and Adaptation Gains

The final value of feedback and adaptation gains follow the simulator's theory specifications and numerical performance. The values for feedback gains were $K_d = diag(1000\ 200\ 200)$, $\alpha = diag(20\ 20\ 20)$, $K_i = diag(0.05\ 0.05\ 0.05)$, and $\kappa = 30$; and for the adaptation gains, they were $K = 50$, $K_w = 7$, and $\Gamma_a = I_{10\times 10} \times 4000$. It aims to promote larger reward recollection from smaller position errors, so evaluation-based reinforcement is influenced to take action even by smaller position errors.

4.5. Results

From Figure 6, we notice that despite the fact that initial conditions are selected away from the desired trajectory, the end-effector tracks the desired trajectory with a short transient. This can be seen in Figure 7, where tracking error converges exponentially in about $t = 0.3$ s. Figure 8a shows the bounded extended velocity error S_r that shapes the invariant of stability, which gives rise to sliding mode at $S_q = 0$; see Figure 8b around $t \geq 0.3$ s. The control signals are quite smooth; see Figure 9a. Notice that τ_1 corresponds to the length coordinate l where its magnitude is much greater than τ_2 and τ_3 due to the higher energy that is required to deform the material for elongation tasks. Figure 9c,d shows the integral and temporal difference errors converging quickly, implying an accurate approximation of the value function by the Critic NN. Finally, reward behavior is shown in Figure 10, since it depends on the evaluation of tracking errors (30); since it has a larger value at initial conditions, then at the beginning, it reaches its higher value. Once the end effector reaches the desired trajectory, the reward ceases signaling that motor learning is achieved.

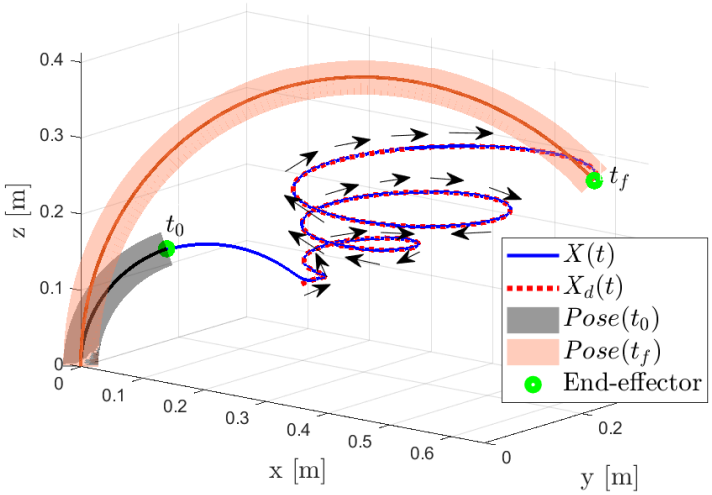
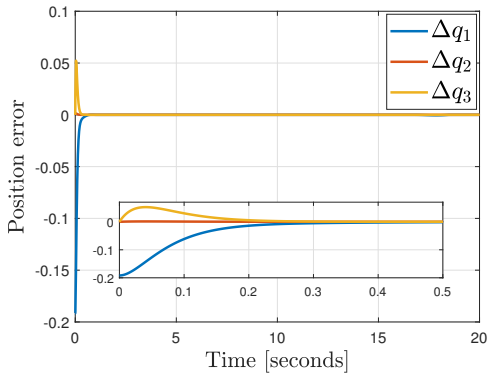
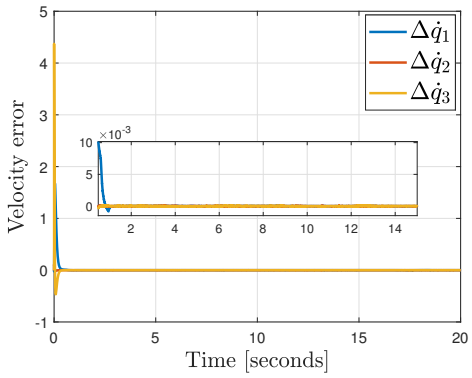


Figure 6. Tracking of 3D soft robot desired (red dotted line) trajectories $X_d(t)$, from initial pose (green dot) shown in grey with a black backbone to a final pose shown in orange, with red backbone.

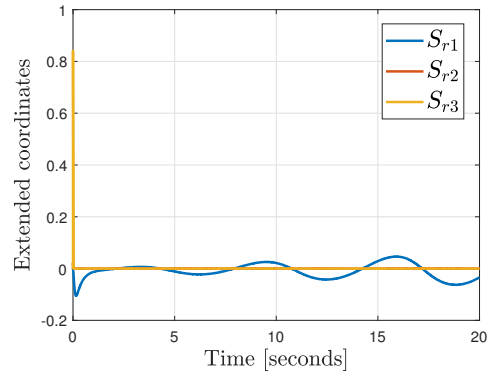


(**a**) Convergence of position tracking errors

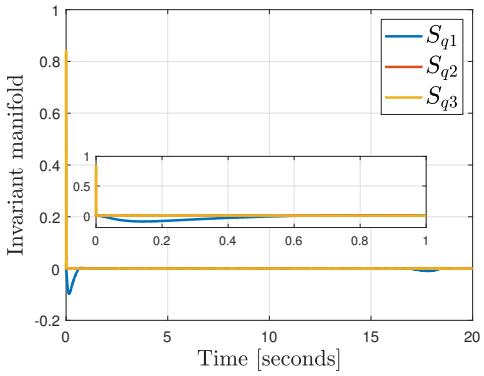


(**b**) Convergence of velocity tracking errors.

Figure 7. Position and velocity tracking errors exhibit smooth convergence, with a short transient of about 300 ms. Notice that such a performance for such aggressive trajectories, since Coriolis (centrifugal and centripetal) forces increases as time goes by.

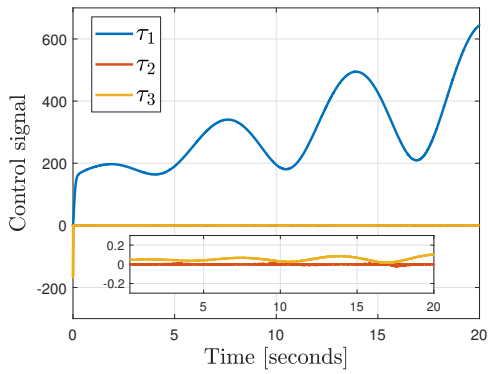


(**a**) Extended velocity error, shows that the bound of S_r increases due to Coriolis forces increases.

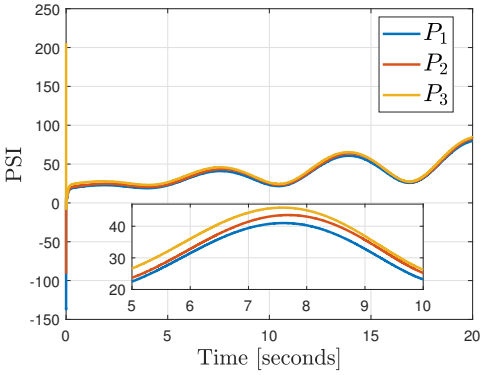


(**b**) Invariant error manifolds are enforced after a short transient, due to the large initial error.

Figure 8. Performance of extended velocity error S_r and invariant manifolds $S_q = 0$ suffers from the increment of Coriolis forces; nonetheless, sliding modes are enforced.

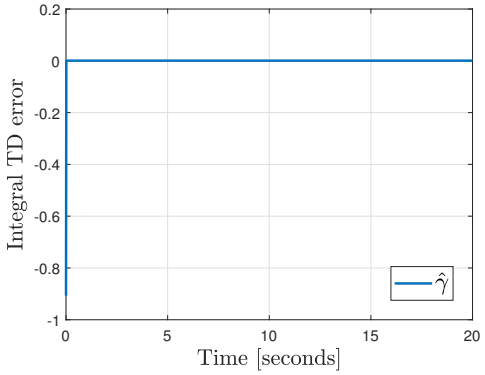


(**a**) Control signals τ are smooth, though increasing for length coordinate in accordance to the task.

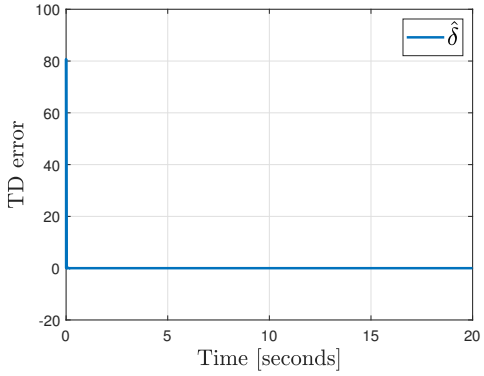


(**b**) Pressure of internal control chambers related as $P = B^{-1}\tau$.

Figure 9. *Cont.*



(**c**) Integral of temporal difference error quickly converges.



(**d**) Convergence of the neural approximation of temporal difference error, implies the learning mechanism is in place after a short initial period.

Figure 9. (**a**) Control signals and (**b**) pressure behavior, in accordance to the convergence of (**c**) integral temporal difference error and (**d**) temporal difference error.

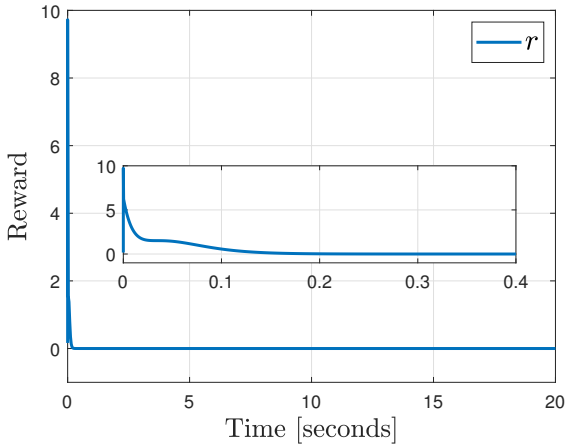
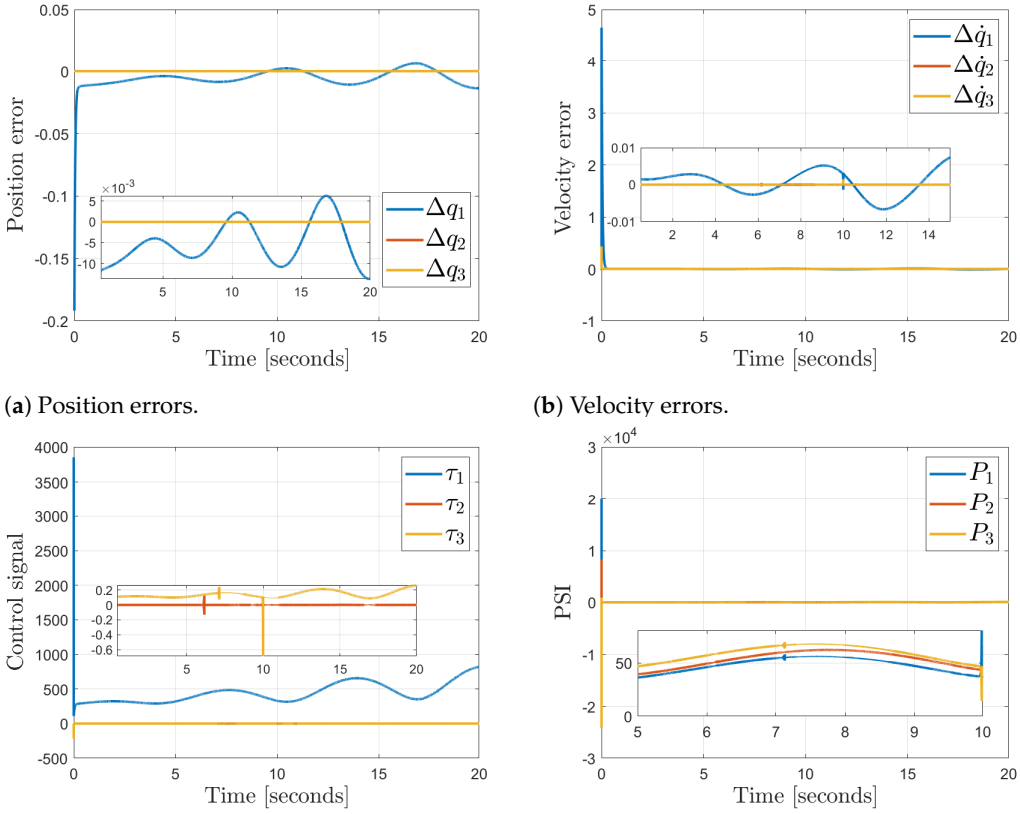


Figure 10. Surprisingly, reward converges after a very short transient.

Comparative Results vs. Classical PID Controller

In order to compare how our proposal performs against others, simulations were also carried out for the very well-known controller, such as classical model-free PID regulator, under the same desired trajectories and initial conditions. The results are shown in Figure 11. Generalized position and velocity errors are shown in Figures 11a and 11b, respectively. Notice that trajectories remain bounded after a short initial response. In contrast, our scheme converges to the origin; see Figure 7. Figure 11c,d shows the control signals and the demanded pressures. Notice that for initial time $t < 0.1$, the system has a high demand of control effort, translating into a high-pressure demand, which can be detrimental in practice. In contrast, our scheme ameliorates this effect and achieves convergence of tracking errors with smooth control; see Figures 7 and 9a,b.



(**a**) Position errors.

(**b**) Velocity errors.

(**c**) Control signals.

(**d**) Demanded pressures.

Figure 11. [Classical PID controller]. Results obtained using PID: (**a**) position errors remains bounded, similarly, (**b**) the velocity errors also remain bounded. (**c**) control signals exhibit high demand and produces pressures (**d**).

5. Discussions

5.1. On the Actor–Critic Architecture with Adaptive Neural Weights

The Actor and Critic neural networks interplay to guarantee tracking, taking into consideration the soft robot performance. The Critic NN approximates the value function by enforcing convergence of the integral and the temporal difference error. In contrast, the Actor NN approximates the nonlinear dynamics using online reinforcement from the Critic NN throughout its weights and the reward. Reward design is fundamental to yield information about the robot task performance. Given the subtleties of the soft robot, reward design is based on the weighted sum of the position and velocity tracking errors. However, the reward can be designed otherwise, since its interpretation depends on what promotes learning. Adaptation of Actor–Critic's weights is proposed based on Lyapunov stability, in contrast to other works that use variants of the gradient descent method.

5.2. On Simulation Study

It is considered to be a Lagrangian soft robot assuming constant cross-section geometry along the backbone, even after being exposed to exogenous and endogenous forces. In practice, this is achieved by manufacturing inextensible threads that braid the soft robot [17] to restrict deformation when pressurized. However, this constraint may not be enforced for negative absolute or relative pressure or when a vacuum emerges. Further research is needed from the material science community to considered these cases.

5.3. Advantages, Disadvantages, and Limitations

The advantages of this method are as follows: the proposed AC scheme is model-free, i.e., no information on the soft robot dynamics is needed to implement the scheme. In addi-

tion, neither pre-training nor initialization is required in the learning process. Surprisingly, Actor–Critic neural network topologies are of low dimension for such difficult and complex nonlinear soft robot dynamics, with only one hidden layer of neurons. We address the difficult and complex soft robot continuum dynamics based on density variations that yield a varying center of mass and a varying inertia tensor. On the other hand, as a limitation, the model has a kinematic singularity at $\kappa = 0$, which is common with other soft robot models, as well as the assumption of constant cross area, which is hard to enforce in practice. Then, for practical implementations of this RL scheme, we surmise that the challenge also includes the soft robot hardware, actuation, and sensory system.

Other modeling domains, such as FEM, have been used as an alternative to study the approximate mechanical properties of soft robots [31,32], with quite some success in designing and manufacturing deformable bodies. Certainly, FEM is needed to analyze structurally optimal designs and comparisons to its continuum domain.

6. Conclusions

Contributions from many research fields have enriched soft robot knowledge, giving rise to novel paradigms on modeling, control, and design. In this note, a novel RL controller for a class of continuum soft robot model is addressed, contributing to the state-of-the-art in learning how to yield 3D trajectories, taking into account online evaluation of deformation performance. The stability-guaranteed, model-free neurocontroller oversees the convergence of the tracking error and reward recollection while exploiting its structural properties, including passivity. The key contribution comes from novel designs of nonlinear adaptive weights of the Critic NN and Actor NN, the former of which uses discontinuous terms and nonlinear activations functions to enforce convergence of TD errors $\hat{\delta}$. In contrast, the latter is influenced by the reinforcement signals given by the reward and temporal difference error. The soft robot under consideration is of the class of continuum body deformation driven by internal pneumatic chambers. This soft robot model has a Lagrangian structure; henceforth, our RL scheme is not limited to soft robots, but it can be implemented in systems characterized by Lagrangian dynamics. Real-time experimental testing has major importance in pursuing real compliance with the theory based on the axioms and the assumptions. Ongoing effort is occuring in the development of such a platform with special care on non-invasive measurement system and low-level pneumatic control instrumentation [9].

Author Contributions: Methodology, L.P.-G., V.P.-V., R.G.-R. and C.E.V.-G.; Formal analysis, L.P.-G., V.P.-V., R.G.-R. and C.E.V.-G.; Writing—original draft, L.P.-G., V.P.-V., R.G.-R. and C.E.V.-G.; Writing—review & editing, L.P.-G., V.P.-V. and C.E.V.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Stability Proof

Appendix A.1. Critic Neural Network

Consider the following Lyapunov candidate function

$$V_\gamma = \frac{1}{2}\hat{\gamma}^2 + \frac{1}{2}\hat{W}_c^T\hat{W}_c. \tag{A1}$$

Taking the time derivative and using (24) and (25), we obtain

$$\begin{aligned}\dot{V}_\gamma =&\hat{\gamma}\dot{\hat{\gamma}} + \hat{W}_c^T\dot{\hat{W}}_c\\ =&\hat{\gamma}\left(\left\{-K_w\text{sgn}(\hat{W}_c) - K\text{sgn}(\hat{\gamma})\frac{\sigma_c(\cdot)}{\sigma_c^T(\cdot)\sigma_c(\cdot)}\right\}^T\sigma_c(\cdot) + \hat{W}_c^T\dot{\sigma}_c(\cdot) - \frac{1}{\psi}\hat{W}_c^T\sigma_c(\cdot) + r\right)\\ &+\hat{W}_c^T\left\{-K_w\text{sgn}(\hat{W}_c) - K\text{sgn}(\hat{\gamma})\frac{\sigma_c(\cdot)}{\sigma_c^T(\cdot)\sigma_c(\cdot)}\right\}\\ =&-\hat{\gamma}K\text{sgn}(\hat{\gamma}) - K_w\hat{W}_c^T\text{sgn}(\hat{W}_c) - K_w\hat{\gamma}\text{sgn}(\hat{W}_c)^T\sigma_c(\cdot) + \hat{\gamma}\hat{W}_c^T\dot{\sigma}_c(\cdot) - \frac{\hat{\gamma}}{\psi}\hat{W}_c^T\sigma_c(\cdot)\\ &+\hat{\gamma}r + K\hat{W}_c^T\text{sgn}(\hat{\gamma})\frac{\sigma_c(\cdot)}{\sigma_c(\cdot)^T\sigma_c(\cdot)}.\end{aligned} \tag{A2}$$

Selecting $\gamma >> 1$ and expressing the bounded terms as ϵ_i, we obtain

$$\begin{aligned}\dot{V}_\gamma =&-\hat{\gamma}K\text{sgn}(\hat{\gamma}) - K_w\hat{W}_c^T\text{sgn}(\hat{W}_c) - K_w\hat{\gamma}\epsilon_1 + \hat{\gamma}\hat{W}_c^T\dot{\sigma}_c(\cdot) - \hat{W}_c^T\epsilon_2 + \hat{\gamma}r + K\hat{W}_c^T\epsilon_3\\ \leq&-K|\hat{\gamma}| - K_w|\hat{W}_c| + |\hat{\gamma}||\hat{W}_c||\dot{\sigma}_c(\cdot)| + \hat{\gamma}r - K_w\hat{\gamma}\epsilon_1 - |\hat{W}_c|\epsilon_2 + K|\hat{W}_c|\epsilon_3\\ =&-(K + K_w\epsilon_1)|\hat{\gamma}| - (K_w - |\hat{\gamma}||\dot{\sigma}_c(\cdot)| + \epsilon_2 - K\epsilon_3)|\hat{W}_c| + \epsilon_{\gamma r}\\ \leq&-\chi_1|\hat{\gamma}| - \chi_2|\hat{W}_c| + \epsilon_{\gamma r}.\end{aligned} \tag{A3}$$

where $\chi_1 = K + K_w\epsilon_1, \chi_2 = K_w - |\hat{\gamma}||\dot{\sigma}_c(\cdot)| + \epsilon_2 - K\epsilon_3$. Thus, we can always select K and K_w such that $\chi_1, \chi_2 > 0$; this implies that there exists constants ϵ_4 and ϵ_5 modulated by $\epsilon_{\gamma r}$ such that $(\hat{\gamma}, \hat{W}_c)$ converges to a compact set of size $sup(\epsilon_4, \epsilon_5)$.

Appendix A.2. Proof of Theorem

Consider the following Lyapunov candidate function of the closed-loop system

$$V_{AC} = \frac{1}{2}S_r^TH(q)S_r + \frac{1}{2}tr(\tilde{W}_a^T\Gamma_a^{-1}\tilde{W}_a) + V_\gamma. \tag{A4}$$

Taking its time derivative, we obtain

$$\dot{V}_{AC} = S_r^TH(q)\dot{S}_r + S_r^T\frac{\dot{H}}{2}S_r + tr(-\tilde{W}_a\Gamma_a^{-1}\dot{\hat{W}}_a) + \dot{V}_\gamma. \tag{A5}$$

Using (15) and passivity property, Equation (A5) becomes

$$\begin{aligned}\dot{V}_{AC} =&S_r^T\left(-D_vS_r + \tau_e - K_dS_r - \tilde{W}_a^T\sigma_a - \epsilon_a\right) - tr\left(\tilde{W}_a^T(-\sigma_aS_r^T - \hat{W}_a(\hat{\gamma}r)^2)\right) + \dot{V}_\gamma\\ \leq&-S_r^T(D_v + K_d)S_r + S_r^T\epsilon_{\tau e} - S_r^T\tilde{W}_a^T\sigma_a + S_r^T\tilde{W}_a^T\sigma_a - S_r^T\epsilon_a\\ &-(\hat{\gamma}r)^2tr(\tilde{W}_a^T(\tilde{W}_a - W_a)) - \chi_1|\hat{\gamma}| - \chi_2|\hat{W}_c| + \epsilon_{\gamma r}\\ <&-\lambda_{min}(K_d)\parallel S_r\parallel^2 + \parallel S_r\parallel(\epsilon_{\tau e} + \epsilon_a) - (\hat{\gamma}r)^2tr(\tilde{W}_a^T(\tilde{W}_a - W_a)) - \chi_1|\hat{\gamma}|\\ &-\chi_2|\hat{W}_c| + \epsilon_{\gamma r}\\ <&-(\lambda_{min}(K_d)\parallel S_r\parallel - (\epsilon_{\tau e} + \epsilon_a))\parallel S_r\parallel - (\hat{\gamma}r)^2tr(\tilde{W}_a^T(\tilde{W}_a - W_a)) - \chi_1|\hat{\gamma}|\\ &-\chi_2|\hat{W}_c| + \epsilon_{\gamma r}\end{aligned} \tag{A6}$$

For high enough values of K_d and of K, K_w according to A.1, there arises an invariant bounded set in terms of $(S_r, \tilde{W}_a)$ that guarantees the boundedness of all closed-loop signals $S_r, \tilde{W}_a, \tilde{W}_c, \hat{\gamma}$. It also implies the boundedness of $\dot{S}_r$ by a constant η.

Now, so far, we have proven that all signals remain bounded. To show that tracking errors converge, we need to show that an integral sliding mode is enforced at $S_q = 0$ in finite time. To this end, consider the following function

$$V_{sq} = \frac{1}{2}S_q^TS_q. \tag{A7}$$

Taking its time derivative of (A7) along the flow (derivative) of $S_r = S_q + K_i \int \text{sgn}(S_q)$, we obtain

$$\begin{aligned} \dot{V}_{s_q} =& S_q^T(\dot{S}_r - K_i \text{sgn}(S_q)) \\ \leq& -K_i|S_q| + |S_q|\eta \\ \leq& -(K_i - \eta)|S_q|. \end{aligned} \tag{A8}$$

We can always choose $K_i > \eta$ to enforce a sliding mode condition at $S_q = 0$, guaranteeing the local exponential convergence of tracking errors, i.e., $\Delta q, \Delta\dot{q} \to 0$ as $t \to \infty$ [13].

References

1. Barto, A.G.; Sutton, R.S.; Anderson, C.W. Looking Back on the Actor—Critic Architecture. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *51*, 40–50. [CrossRef]
2. Wang, F.Y.; Zhang, H.; Liu, D. Adaptive Dynamic Programming: An Introduction. *IEEE Comput. Intell. Mag.* **2009**, *4*, 39–47. [CrossRef]
3. Lewis, F.; Vrabie, D.; Syrmos, V. *Optimal Control*; EngineeringPro Collection; Wiley: Hoboken, NJ, USA, 2012.
4. Guo, K.; Pan, Y. Composite adaptation and learning for robot control: A survey. *Annu. Rev. Control.* **2023**, *55*, 279–290. [CrossRef]
5. Jin, L.; Li, S.; Yu, J.; He, J. Robot manipulator control using neural networks: A survey. *Neurocomputing* **2018**, *285*, 23–34. [CrossRef]
6. He, W.; Chen, Y.; Yin, Z. Adaptive Neural Network Control of an Uncertain Robot with Full-State Constraints. *IEEE Trans. Cybern.* **2016**, *46*, 620–629. [CrossRef] [PubMed]
7. Song, B.; Slotine, J.J.; Pham, Q.C. Stability Guarantees for Continuous RL Control. *arXiv* **2022**, arXiv:cs.RO/2209.07324.
8. Bhagat, S.; Banerjee, H.; Ho Tse, Z.T.; Ren, H. Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges. *Robotics* **2019**, *8*, 4. [CrossRef]
9. Trejo-Ramos, C.A.; Olguín-Díaz, E.; Parra-Vega, V. Lagrangian and Quasi-Lagrangian Models for Noninertial Pneumatic Soft Cylindrical Robots. *J. Dyn. Syst. Meas. Control.* **2022**, *144*, 121004. [CrossRef]
10. Guan, Z.; Yamamoto, T. Design of a Reinforcement Learning PID controller. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–6. [CrossRef]
11. He, W.; Gao, H.; Zhou, C.; Yang, C.; Li, Z. Reinforcement Learning Control of a Flexible Two-Link Manipulator: An Experimental Investigation. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7326–7336. [CrossRef]
12. Vázquez-García, C.E.; Trejo-Ramos, C.A.; Parra-Vega, V.; Olguín-Díaz, E. Quasi-static Optimal Design of a Pneumatic Soft Robot to Maximize Pressure-to-Force Transference. In Proceedings of the 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE), Natal, Brazil, 11–15 October 2021; pp. 126–131.
13. Parra-Vega, V.; Arimoto, S.; Liu, Y.H.; Hirzinger, G.; Akella, P. Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans. Robot. Autom.* **2003**, *19*, 967–976. [CrossRef]
14. Yang, T.; Xiao, Y.; Zhang, Z.; Liang, Y.; Li, G.; Zhang, M.; Li, S.; Wong, T.W.; Wang, Y.; Li, T.; et al. A soft artificial muscle driven robot with reinforcement learning. *Sci. Rep.* **2018**, *8*, 14518. [CrossRef]
15. Ishige, M.; Umedachi, T.; Taniguchi, T.; Kawahara, Y. Exploring Behaviors of Caterpillar-Like Soft Robots with a Central Pattern Generator-Based Controller and Reinforcement Learning. *Soft Robot.* **2019**, *6*, 579–594. [CrossRef] [PubMed]
16. Boyraz, P.; Runge, G.; Raatz, A. An overview of novel actuators for soft robotics. *Actuators* **2018**, *7*, 48. [CrossRef]
17. Cianchetti, M.; Ranzani, T.; Gerboni, G.; De Falco, I.; Laschi, C.; Menciassi, A. STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots And Systems, Tokyo, Japan, 3–7 November 2013; pp. 3576–3581.
18. Xu, F.; Wang, H.; Au, K.W.S.; Chen, W.; Miao, Y. Underwater dynamic modeling for a cable-driven soft robot arm. *IEEE/ASME Trans. Mechatronics* **2018**, *23*, 2726–2738. [CrossRef]
19. Marchese, A.D.; Katzschmann, R.K.; Rus, D. A recipe for soft fluidic elastomer robots. *Soft Robot.* **2015**, *2*, 7–25. [CrossRef]
20. Marchese, A.D.; Komorowski, K.; Onal, C.D.; Rus, D. Design and control of a soft and continuously deformable 2d robotic manipulation system. In Proceedings of the 2014 IEEE International Conference ON Robotics And Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2189–2196.
21. Katzschmann, R.K.; Marchese, A.D.; Rus, D. Autonomous object manipulation using a soft planar grasping manipulator. *Soft Robot.* **2015**, *2*, 155–164. [CrossRef]
22. Connolly, F.; Walsh, C.J.; Bertoldi, K. Automatic design of fiber-reinforced soft actuators for trajectory matching. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 51–56. [CrossRef]
23. Hannan, M.W.; Walker, I.D. Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *J. Robot. Syst.* **2003**, *20*, 45–63. [CrossRef]
24. Sadati, S.H.; Naghibi, S.E.; Shiva, A.; Noh, Y.; Gupta, A.; Walker, I.D.; Althoefer, K.; Nanayakkara, T. A geometry deformation model for braided continuum manipulators. *Front. Robot. AI* **2017**, *4*, 22. [CrossRef]

25. Webster, R.J., III; Jones, B.A. Design and kinematic modeling of constant curvature continuum robots: A review. *Int. J. Robot. Res.* **2010**, *29*, 1661–1683. [CrossRef]
26. Godage, I.S.; Branson, D.T.; Guglielmino, E.; Medrano-Cerda, G.A.; Caldwell, D.G. Shape function-based kinematics and dynamics for variable length continuum robotic arms. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 3–9 May 2011; pp. 452–457.
27. Odom, E.M.; Egelhoff, C.J. Teaching deflection of stepped shafts: Castigliano's theorem, dummy loads, heaviside step functions and numerical integration. In Proceedings of the 2011 Frontiers in Education Conference (FIE), Rapid City, SD, USA, 12–15 October 2011; pp. F3H-1–F3H-6. [CrossRef]
28. Garcia, R.; Parra-Vega, V. Tracking control of robot manipulators using second order neuro sliding mode. *Lat. Am. Appl. Res.* **2009**, *39*, 285–294.
29. Doya, K. Temporal Difference Learning in Continuous Time and Space. In *Advances in Neural Information Processing Systems*; Touretzky, D., Mozer, M., Hasselmo, M., Eds.; MIT Press: Cambridge, MA, USA, 1995; Volume 8, pp. 1073–1079.
30. Kandasamy, S.; Teo, M.; Ravichandran, N.; McDaid, A.; Jayaraman, K.; Aw, K. Body-powered and portable soft hydraulic actuators as prosthetic hands. *Robotics* **2022**, *11*, 71. [CrossRef]
31. Bieze, T.M.; Largilliere, F.; Kruszewski, A.; Zhang, Z.; Merzouki, R.; Duriez, C. Finite Element Method-Based Kinematics and Closed-Loop Control of Soft, Continuum Manipulators. *Soft Robot.* **2018**, *5*, 348–364. [CrossRef] [PubMed]
32. Sun, Y.; Zhang, D.; Liu, Y.; Lueth, T.C. FEM-Based Mechanics Modeling of Bio-Inspired Compliant Mechanisms for Medical Applications. *IEEE Trans. Med. Robot. Bionics* **2020**, *2*, 364–373. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.