Robotica (2024), 1–24
doi:10.1017/S0263574724001838

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Robustifying a reinforcement learning agent-based bionic reflex controller through an adaptive sliding mode control

Hirakjyoti Basumatary, Daksh Adhar and Shyamanta M. Hazarika

Biomimetic Robotics and Artificial Intelligence Laboratory (BRAIL), Mechanical Engineering Department, Indian Institute of Technology, Guwahati, India
Corresponding author: Hirakjyoti Basumatary; Email: 23hirak@gmail.com

Received: 27 May 2024; **Revised:** 15 August 2024; **Accepted:** 23 September 2024

Keywords: grasping; control of robotic systems; force control; novel applications of robotics; robotic hands

Abstract

Maintaining object grasp stability represents a pivotal challenge within the domain of robotic manipulation and upper-limb prosthetics. Perturbations originating from external sources frequently disrupt the stability of grasps, resulting in slippage occurrences. Also, if the grasping forces are not optimal while controlling the slip, it may result in the deformation of the objects. This study investigates the robustification of a reinforcement learning (RL) policy for implementing intelligent bionic reflex control, i.e., slip and deformation prevention of the grasped objects. RL-derived policies are vulnerable to failures in environments characterized by dynamic variability. To mitigate this vulnerability, we propose a methodology involving the incorporation of an adaptive sliding mode controller into a pre-trained RL policy. By exploiting the inherent invariance property of the sliding mode algorithm in the presence of uncertainties, our approach strengthens the robustness of the RL policies against diverse and dynamic variations. Numerical simulations substantiate the efficacy of our approach in robustifying RL policies trained within simulated environments.

Acronyms

AFNITSM	Adaptive Fast Nonsingular Integral Terminal Sliding Mode
ASMC	Adaptive Sliding Mode Control
DNN	Deep Neural Network
DR	Domain Randomization/Domain Randomized
FNITSM	Fast Non-singular Integral Terminal Sliding Mode
ISMC	Integral Sliding Mode Control
RL	Reinforcement Learning
SOAFITSMC	Second-Order Adaptive Fast Nonsingular Integral Terminal Sliding Mode Controller
SOFNITSMC	Second-Order Fast Nonsingular Integral Terminal Sliding Mode Controller

1. Introduction

In the domain of robotics and automation, achieving stable object manipulation stands as a pivotal pursuit across diverse applications encompassing industrial automation, service robotics, and prosthetics [1, 2]. However, ensuring a secure grasp encounters significant challenges stemming from factors such as object geometry, material characteristics, and external perturbations, which subject grasped objects to potential slippage occurrences. Effective prevention of slippage necessitates the employment of sophisticated control methodologies. Moreover, beyond the hurdles posed by external disturbances

leading to slippage, suboptimal application of grasp forces resulting in object deformations poses another formidable obstacle impeding the efficiency and safety of robotic manipulation endeavors [3]. Mitigating both slippage and deformation phenomena represents a paramount objective in advancing robotic manipulation techniques. This pursuit has sparked interest in bionic reflex mechanisms, mirroring the control strategies observed in human grasp reflex mechanisms, particularly within the domains of prosthetic and robotic manipulators. While conventional methodologies, including vibration-based approaches, friction model-based techniques, and data-driven methodologies, have demonstrated efficacy under controlled conditions [4], their adaptability to dynamic disturbances remains limited. Notably, data-driven techniques exhibit promise in slip signal detection but often hinge upon labeled training datasets, constraining their adaptability [5]. In light of these challenges, this study delves into application of reinforcement learning (*RL*), as a particularly promising avenue for bionic reflex control.

RL, a subset of machine learning distinguished by its capacity to address intricate challenges in robotic control through trial-and-error learning mechanisms, emerges as a prime candidate for the development of slippage prevention controllers. *RL*'s inherent capability to assimilate real-time sensor feedback holds promise for enabling robotic hands to autonomously adapt their grasping actions, thereby furnishing versatile and robust slippage prevention mechanisms. Nevertheless, *RL* policies frequently encounter challenges when transitioning from simulated training environments to real-world testing environments due to disparities in environmental conditions. Enhancing the generalization capability of *RL* models is tantamount to managing environmental perturbations, where simulated and testing environments correspond to nominal and perturbed states, respectively [6]. Consequently, in this study, we robustify the pre-trained *RL* policy with an adaptive controller to enhance its performance in dynamically changing environmental conditions.

2. Related work

2.1. Slip detection and prevention

Slip detection research can be broadly classified into three main categories: gross slip, involving the complete displacement of the object surface; incipient slip, where partial slippage occurs; and slip prediction, which utilizes tactile features to anticipate slip events [7]. Slip prevention strategies encompass both model-based techniques employing concepts such as friction cones and beam bundle models and model-free approaches including supervised and deep learning methodologies [4, 7]. Detection methodologies encompass friction-based methods utilizing multi-axial force sensing, analysis of tactile sensor signals, and machine learning-based classifiers [4]. An array of sensors, including pressure-resistance, optical, piezoelectric, and thermal sensors, is employed for slip-detection purposes [8]. Advanced signal processing techniques, such as Fourier transforms and wavelet decomposition, play a crucial role in enhancing slip detection capabilities [5]. However, these methodologies often necessitate manual thresholding of tactile sensing signals, thereby constraining the automation of slip detection for objects with unknown properties [4].

Slip prevention strategies are broadly classified into two categories: reactive and proactive methods [9]. Reactive approaches involve responding to detected slippage signals, while proactive methods anticipate and provide warnings of impending slips before they occur. The integration of slip signals into control algorithms for bionic hands poses notable challenges [5]. Advanced control algorithms such as PID, sliding mode control, fuzzy control, and model predictive control are employed by researchers to prevent slippage [8]. Typically, slip prevention entails the use of closed-loop controllers for either force or position control, as shown in Figure 1. Position-based controllers are less favored due to the variability in object stiffness [8]. Force control, complemented by an inner position loop, is commonly adopted to address objects with varying stiffness and to effectively handle unexpected disturbances for slip prevention [10, 11] (shown in Figure 2). Adaptive sliding mode controllers have demonstrated effectiveness in ensuring grasp stability [12], although their integration into bionic hands necessitates the automation and elimination of thresholding signals.

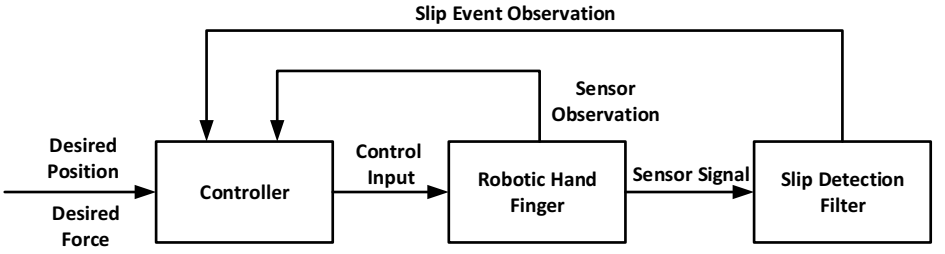

Figure 1. *Slippage avoidance closed-loop control structure presented in literature [8].*

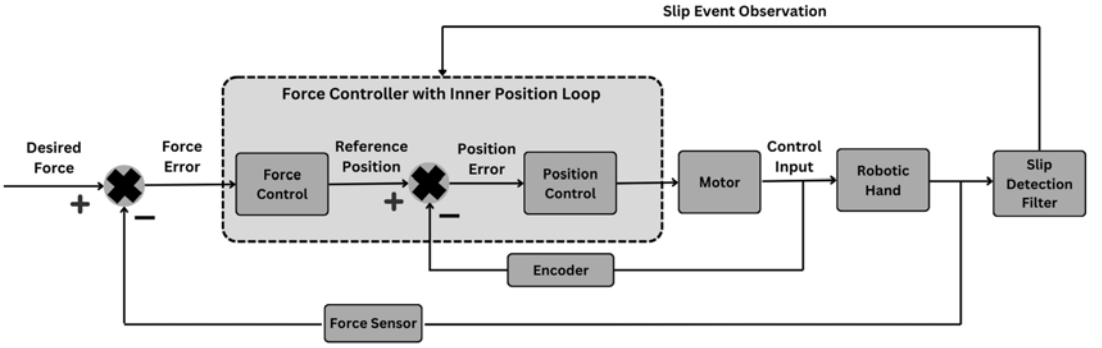

Figure 2. *Slippage avoidance by force control with inner position loop.*

2.2. Deformation detection and control

Preventing deformation in robotic systems necessitates effective stiffness detection or deformation measurement, constituting a challenging and ongoing research endeavor [3]. Current methodologies for stiffness detection and control encompass several approaches. Intrinsic vibration frequency-based signal processing: This method analyzes the vibrational response to ascertain object stiffness via frequency domain decomposition, offering precise measurements albeit typically utilized for offline analysis. Time-domain analysis methods: Monitoring parameters such as equivalent force, deflection, and velocity in real-time facilitates the deduction of object stiffness based on these characteristics. Integration of measuring devices: This approach entails the incorporation of specialized measuring apparatus at the robot gripper's terminus, correlating material stiffness with gripping forces post-contact. However, it may not be suitable for prosthetic hands due to size and weight constraints. Hooke's law: Calculating the stiffness coefficient ($K = F/d$) based on contact force (F) and deformation (d) provides stiffness detection, although instantaneous deformation calculation poses challenges for underactuated prosthetic hands [13]. Some studies explore vision-based techniques for deformation detection [3, 14, 15]. For instance, the Gelsight sensor measures elastomer deformation but may present cost and accessibility limitations [3]. Makihara et al. [15] employ pixel analysis to generate a stiffness map ('pix2stiffness') for grasp pose detection to mitigate damage to deformable objects. However, stiffness map generation entails manual intervention, and no force control based on contact dynamics was considered to minimize deformation. Additionally, other frameworks predict object geometry and dynamics for deformable object manipulation, necessitating training and labeling with human intervention during design [16].

Impedance control serves as a pivotal strategy for averting object deformation within robotic manipulation systems [17–19]. By amalgamating real-time force sensing capabilities with adaptive control algorithms, robots can dynamically modulate compliance and stiffness to align with object characteristics, thereby enhancing precision and reliability in handling delicate objects. In a study by

Hua Deng et al. [20], stiffness was regulated utilizing a polyvinylidene fluoride sensor, employing human-defined voltage thresholds for object categorization. Deformation control techniques leveraging Hooke's law and impedance-based methods necessitate precise knowledge of stiffness and desired model references. Alternative methodologies encompass utilizing kinematics for stiffness detection in underactuated mechanisms [13] and manipulating object weight through reorientation to regulate deformation [21]. Bistable compliant underactuated grippers have demonstrated enhanced grasping capabilities for deformable objects [22], whereas soft grippers offer adaptability although encountering limitations in variability and complexity [23, 24]. Addressing these constraints within prosthetic and robotic hands necessitates the integration of active control systems for adaptable grasping, a focal point of investigation in this paper.

2.3. Increasing the generalization capability

The generalization capability of *RL* policies can be enhanced through various techniques, including domain randomization/domain randomized (*DR*), adversarial rreinforcement learning (ARL), meta-learning, transfer learning, post-training augmentation, and knowledge distillation [6, 25, 26]. *DR* acts as a bridge between simulation and reality, akin to robust control in control theory, by designing controllers resilient to parameter variations and noise [25]. Even when trained sub-optimally in a simulator, *DR* exhibits effectiveness in real-world scenarios due to its convergence properties [27]. ARL enhances robustness and transferability by training controllers across diverse environment models, leveraging adversarial sub-agents to generate challenging models that minimize cumulative rewards [25]. Robust Adversarial RL frames the problem as a two-player zero-sum game, where a disturbing agent aims to create the worst disturbance, countered by a control agent striving for optimal control input [28, 29]. However, both *DR* and ARL may lead to fixed policies prone to overfitting [30]. Meta-learning, or Meta-RL, focuses on building models capable of adapting and improving performance across new tasks without extensive retraining [31]. It facilitates rapid adaptation of pre-trained policies to dynamic variations, thereby enhancing policy generalization. Nonetheless, learning optimal policies for all possible scenarios may unnecessarily increase complexity, particularly for simpler tasks [26]. Transfer learning encompasses techniques like zero-shot learning, few-shot learning, and domain adaptation, yet may result in learning deterministic policies unsuitable for simulation-to-reality transfer [26]. Meeting strict time constraints is crucial, particularly when implementing controllers on high-frequency physical devices. Deep neural network (*DNN*) policies, especially ensembles, pose challenges in this context. Knowledge distillation, transferring expertise from a large, complex network to a smaller, more efficient one, reduces evaluation time. This technique distills a RL agent's policy, trained in a large network, into a smaller network operating at an expert level [32]. Policy distillation has demonstrated efficiency surpassing domain randomization methods [33, 34]. However, all the above methods tend to learn fixed policies, which may not be suitable for real-time adaptation during the presence of disturbances.

In the post-training augmentation-based strategy aimed at improving generalization capabilities, an augmented robust controller is integrated hierarchically with the *RL* policy to counteract potential disturbances. In ref. [6], the utilization of $\mathcal{L}_1$ adaptive controllers on the application to a pendubot and quadrotor systems is exemplified, showcasing their efficacy in attenuating the impact of matched uncertainties. Furthermore, Jeong Woo Kim et al. [35] introduced a disturbance-based observer to augment an *RL* policy, addressing mismatches between simulated and real-world environments. Similarly, Anubhav Guha and Anuradha Annaswamy [36] employed a model reference adaptive control system to estimate and rectify parametric uncertainties, following a comparable approach. The salient advantage of post-training augmentation, vis-à-vis other policy generalization techniques discussed previously, resides in its capacity to dynamically adapt to real-time disturbances. Unlike fixed policies, this approach accommodates variable disturbances not encountered during simulator training. Consequently, the controller can markedly enhance the performance of the learned policy in real-world settings, even amidst the presence of unforeseen and untrained disturbances.

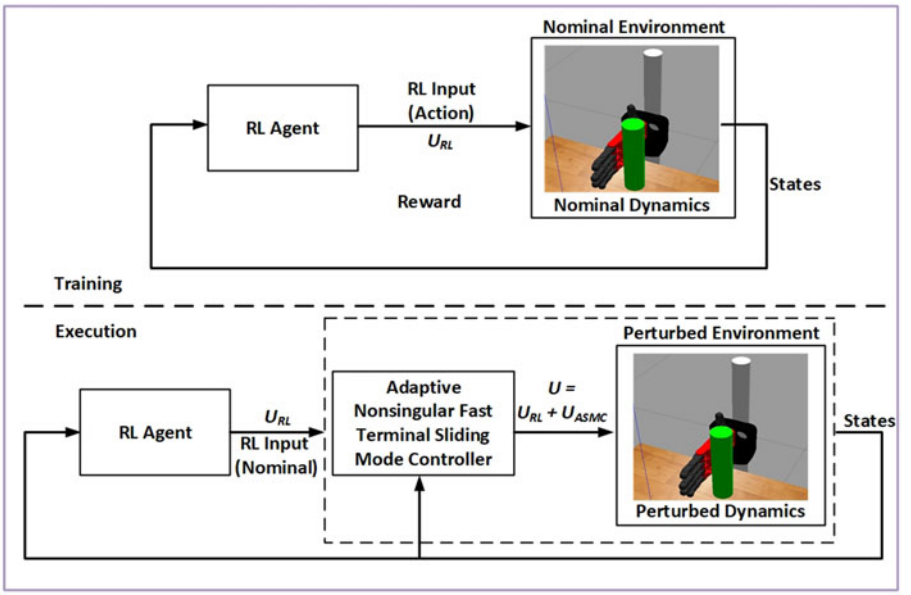

Figure 3. *Proposed approach for reinforcement learning control policy robustness improvement based on adaptive integral sliding mode controller.*

3. Problem formulation

The core problem addressed in this paper is the enhancement of a *RL* based control system to prevent slippage and deformation in robotic grasping and lifting. Conventional *RL* training, even when conducted in a diversified simulated environment with *DR*, often falls short when faced with real-world uncertainties and control variations. Specifically, *RL* agents trained in simulated environments often struggle to generalize to real-world conditions due to discrepancies between the training environment and the actual deployment scenario. These discrepancies include variations in input signals and unmodelled dynamics, which can lead to significant performance degradation. To tackle this issue, we propose a hierarchical approach that combines *RL* with a robust adaptive sliding mode control (*ASMC*), U_{ASMC} strategy. Initially, the *RL* agent is trained in a nominal *DR* environment where object weights, coefficients of friction, and contact stiffness are randomized, enhancing the agent's adaptability to various scenarios. Through this extensive training, a robust nominal policy (U_{RL}) is developed to perform well under these nominal conditions. We then design an adaptive fast nonsingular integral terminal sliding mode (*AFNITSM*) controller [37], denoted as U_{ASMC}, to complement the *RL* policy by using it as a reference force trajectory and providing additional robustness against matched uncertainties and disturbances. The *AFNITSM* control strategy enhances the *RL* policy by achieving ideal dynamics from the outset, bypassing the reaching phase and quickly transitioning to the sliding phase. During policy execution, the *AFNITSM* controller operates alongside the *RL* policy, leveraging the nominal environment's dynamics as an internal model and compensating for discrepancies between this model and actual deployment dynamics. The discontinuous switching function of the *AFNITSM* surface effectively handles matched disturbances, ensuring that the system's dynamics remain within the sliding surface. This robust control approach ensures that the *RL* policy, trained under nominal dynamics, performs effectively even when faced with dynamic variations and disturbances. Figure 3 shows the proposed approach for *RL* control policy robustness improvement. By integrating the *RL* policy with an *AFNITSM* controller, our approach ensures robust performance in preventing slippage and deformation during robotic grasp-and-lift operations, effectively addressing the shortcomings of conventional *RL* training by enhancing the system's adaptability and robustness in real-world scenarios, thereby improving overall reliability and performance.

4. Design Methodology

4.1. Bionic reflex grasping policy

Algorithm 1: Bionic Reflex Control

Input: Grasping environment in PyBullet
Output: Optimal joint torques (τ) to lift the grasped object without slippage and deformation
begin
  foreach *episode* **do**
    Initialize S (observations of states from the grasping environment) ;
    foreach *step of episode* **do**
      Generate actions based on **Soft Actor-Critic** algorithm ;
      Grasp the object ;
      Lift the object ;
      Slip discrimination during object lifting ;
      if *Object Drops* **then**
        episode ends ;
        reset simulation ;
      else
        if *Slip Occurs* **then**
          $\tau \leftarrow \tau + \Delta\tau$;
          if *Deformation Occurs* **then**
            $\tau \leftarrow \tau + \Delta\tau - \lambda$; `// (-λ is a small decrement in force)`
          else
            $\tau \leftarrow \tau + \Delta\tau + \lambda$;
          continue ;
      End lifting motion (Grasp Successful) ;
      if *Grasp Successful* **then**
        episode ends ;
        reset simulation ;
  Until Episode Time is terminal

The entire grasping task is treated as a Model-Free Reinforcement Learning problem, enabling policy learning through direct interaction with the environment and mapping from states to actions. The deformation of the grasping state is represented as a continuous state variable, alongside other observational states such as joint angles, joint velocities, fingertip forces, slip states, wavelet coefficient energy, deformation states, and joint torques, all of which are continuous. Moreover, the action space consists of continuous joint torques. Hence, the Actor-Critic *RL* algorithm is selected for its suitability in handling continuous state and action spaces. The pseudo-code for the *RL*-based bionic reflex controller is outlined in Algorithm 1. This algorithm aims to determine optimal joint torques (τ) necessary for lifting a grasped object without encountering slippage or deformation within a PyBullet [38] grasping environment (as detailed in the subsequent section). The algorithm operates through a sequence of episodes, with each episode comprising steps aimed at executing successful grasping and lifting actions. Initially, the algorithm initializes state observations (s) from the environment. During each step of the episode, actions are generated utilizing the Soft Actor-Critic algorithm [39] to manipulate the robot's joints and grasp the object. Subsequently, the algorithm lifts the object while continually monitoring for signs of slippage or deformation. If the object drops or slips, the episode concludes, and the simulation is reset. In cases where slippage or deformation occurs during lifting, the joint torques (τ) are adjusted accordingly to either increase or decrease the applied force. Initially, the algorithm detects slips, correcting joint torques using $\delta\tau$. Following this, if deformation occurs, the joint torques are further adjusted by a value λ. If the grasp is successful without slippage or deformation, the episode concludes, and the simulation is reset.

This iterative process continues until termination of the episodes, ensuring comprehensive exploration and refinement of the grasping strategy across multiple episodes.

4.2. RL training for the nominal policy

The policy governing the grasping and lifting actions of the robotic hand manipulating an object is trained within the PyBullet Simulator environment. This training utilizes a standard soft-actor critic rreinforcement learning lgorithm from the Stable-Baselines3 framework [40]. The objective is to derive optimal joint torques that enable the generation of requisite fingertip forces, thereby preventing object slippage and deformation during both grasping and lifting maneuvers.

4.2.1. MDP for the RL training

1. **States:** Joint Angles, Joint Velocities, Fingertip Forces, Slip States, Wavelet Coefficients, Joint Torques, Deformation States
2. **Actions:** Joint Torques
3. **Rewards:**

$$\sum_{i=1}^{5}\left(\frac{1}{\ln(x_i+1.1051)}\right)+\sum_{i=1}^{5}(\delta_i.10)-\sum_{i=1}^{5}(\psi_i.10)-C\times\Delta-d^2 \tag{1}$$

The objective is to meticulously guide the hand to grasp the object, ensuring slip-free lifting while mitigating any potential damage caused by deformation. The reward function comprises several terms tailored to achieve this goal. The first term employs an inverse logarithmic relation to compute the reward for each step. Specifically, x_i denotes the distance between the fingertip and the object. As the finger approaches the object, x_i diminishes towards zero, maximizing the reward. To further distinguish between close and actual contact, a discrete term δ is introduced, incrementing a $+10$ reward upon contact of each finger (δ equals 1 if contact is detected, and 0 otherwise). Subsequently, the reward function penalizes slippage during lifting. For each finger, a penalty of 10 units is deducted from the total reward whenever slipping is detected, indicated by the boolean variable, ψ (with ψ equaling 1 upon slip detection and 0 otherwise). However, a limitation of the previous terms lies in their potential to promote excessively tight grasps that may damage the object. To mitigate this, a penalty for deformation is incorporated. This penalty is computed based on the volume gradient (Δ), representing the change in volume, and is scaled accordingly ($C = 50/(initial\ \ volume)$). This value is subtracted from the reward function, penalizing the agent for excessive deformation resulting from the grasp. Finally, the wavelet coefficient energy term serves as a metric for quantifying the degree of slippage [41], providing an additional measure to regulate slip prevention.

Slip detection. Slip detection is facilitated through Haar wavelet decomposition of the force sensor signal, chosen for its capability to identify transient changes resembling slip signals owing to its non-differentiable, discontinuous, and asymmetric characteristics [4, 42, 43]. To get over the drawback of thresholding, we draw inspiration from the work in [42], which utilized the analysis of the property of the trend of pairwise details in a discrete wavelet transform (DWT)-based methodology to determine the occurrence of slip. Specifically, two subsequent DWT components had the same absolute value but a different sign due to the characteristics of the DWT that is used. Because the sign of paired components shifts from negative to positive in the load phase, as opposed to the slip phase, hence it is possible to discriminate between the two stages [42]. After thorough experimentation, we opted for the 5th-level decomposition, considering factors such as signal representation and slip detection requirements. This level strikes a balance between capturing fine-grained signal details and mitigating excessive noise or artifacts. Figure 4 illustrates the Haar wavelet decomposition-based slippage detection methodology

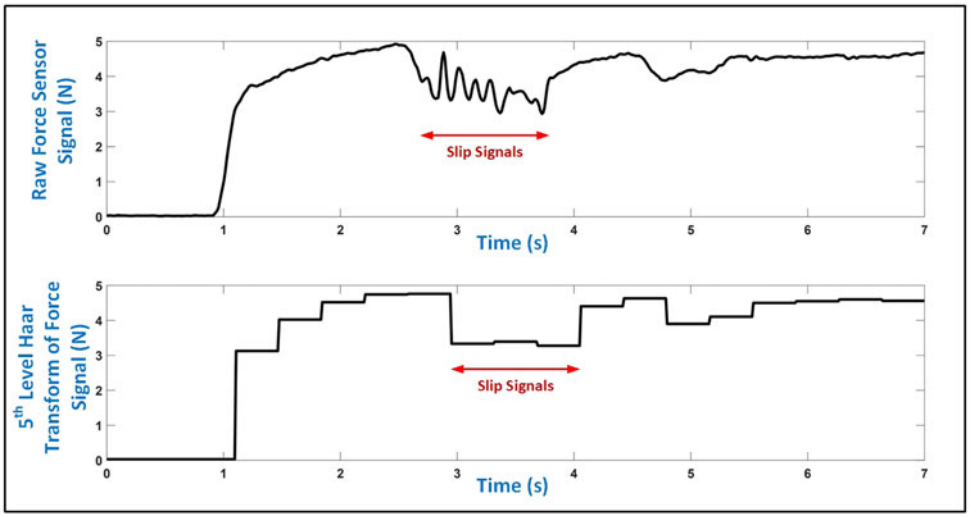

Figure 4. *Force sensor signal while grasping and lifting an object. Fifth-level Haar decomposition of raw force sensor signal is used to detect slip. The positive gradient is a reflection of the load being applied. The opposing variation trend is a representation of slip.*

employed in this study. Our methodology is different from the one mentioned in [42], because following signal decomposition, we analyze the gradient trend of the inverse Haar using the approximation and detailed coefficients of the 5th-level Haar transformed signal. A negative gradient signifies slip occurrence, while a positive gradient indicates its absence. The RL algorithm uses this trend as a slip state observation after detecting it with the slip detector logic. Subsequently, the RL agent's policy is to acquire optimum torques for joint application based on the maximization of the accumulated reward function.

Deformation detection. The process begins with the creation of the deformable object in SolidWorks, resulting in a .stl (Standard Triangle Language/Standard Tessellation Language) file. Subsequently, tetrahedral meshing is executed using fTetWild, yielding a .msh file [44]. The deformable object is then generated in Gmsh as a .vtk file and simulated utilizing the built-in Finite Element Method (FEM) in PyBullet. FEM represents a robust tool for modeling deformable objects, leveraging the discretization of objects into small elements to derive deformation through the solution of partial differential equations. This method enables accurate representation of deformable object dynamics, particularly with fine tessellation, albeit at a computational cost. FEM efficiently approximates the genuine physical behavior of deformable objects [45]. The resulting deformable object, as illustrated in Figure 5, is characterized by parameters detailed in Table I, which are utilized for our physics-based simulation.

4.3. Deformation calculation

In PyBullet, surface mesh data representation entails vertex and triangle lists, as depicted in Figure 5. The function "getMeshData" facilitates the retrieval of mesh information, specifically the vertex indices of triangular meshes constituting the 3D object. Additionally, the reference frame dynamically transitions from the ground to the grasped object (shifted origin) during grasping and lifting operations. This adjustment ensures the accommodation of translations or rotations experienced by the object during manipulation. To estimate volume changes for deformable objects devoid of explicit mathematical formulations, a general point (shifted origin) is defined to construct a tetrahedron, as depicted in Figure 5. Subsequently, the signed volume is computed using the following formula, as given in ref. [46]:

$$Signed\ \ Volume = \mathbf{AO}.(\mathbf{AB} \times \mathbf{AC})/6 \tag{2}$$

Table I. *Simulation parameters for deformable object.*

Parameters	Values
Mass	1 Kg
Friction Coefficient	0.5
NeoHookeanMu (Shear Modulus)	33.78 MPa
NeoHookean Lambda (First Lame Parameter)	810 MPa
NeoHookean Damping	10

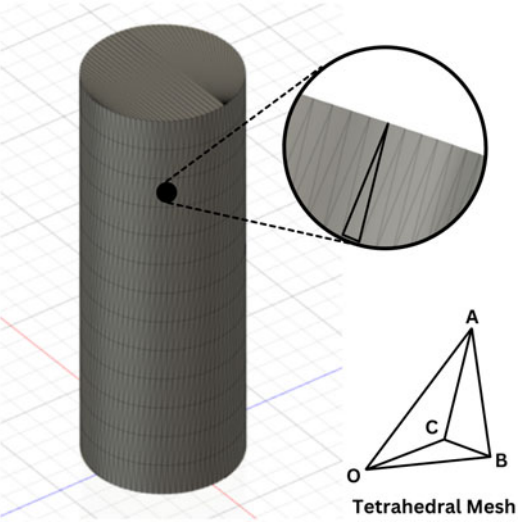

Figure 5. *Triangular mesh description of the deformable cylinder. Point **O** represents reference origin. The tetrahedral mesh, ABCO is shown in the figure as well.*

where points A, B, and C are concurrently selected from the mesh, and O is a reference mesh point taken arbitrarily. The term "signed volume" pertains to the orientation or direction employed for volume computation. Specifically, the surface normal of triangle **ABC** determines the sign and weight of each tetrahedron formed by the vertices, and the aggregation of these solids yields the object's total volume. Subsequently, this volume metric serves as the basis for calculating deformation, denoting the change in volume observed at each time step. Our methodology affords a dynamic and comprehensive evaluation of deformation, facilitating real-time monitoring essential for reward calculation.

4.4. Randomization of the physics parameters

As previously highlighted, DR emerges as a viable strategy for effecting domain adaptation through parameter randomization. In our present experiment, we implement this technique by randomizing the physics parameters within PyBullet. Each episode entails sampling random trajectories from a predetermined range of randomized parameters, thereby diversifying the training data for our RL model. These parameters are subject to randomization around a predefined set of nominal values, encompassing a specified range of limits. This process facilitates robust training by exposing the RL model to a variety of simulated environments characterized by diverse physical properties.

4.4.1. Randomization of the mass

Initially, randomization is applied to the object's mass, thereby evaluating its impact. The nominal mass of the grasped object is set at 1 kg, with an additional variance ranging between 50% and 150% of its

nominal value considered. The manipulation of mass properties is accomplished through the utilization of the mass flag within the changeDynamics option available in PyBullet. This procedure enables the adjustment of mass properties, thereby simulating varying object masses within the experimental setup.

4.4.2. Randomization of the friction properties

In the second phase of randomization, focus is directed toward the surface parameters governing contact properties. Specifically, the frictional properties are subject to modification through the randomization of the coefficient of friction (COF) pertaining to the grasped object. Given that the COF represents a surface property rather than an intrinsic one, its randomization is instrumental in initiating slippage phenomena. Given the material specification of natural rubber, a nominal contact friction coefficient of 0.5 is adopted, aligning with typical values observed in interactions between rubber and plastic materials [47]. Randomization of the friction parameters is implemented with an additive variance ranging between 50% and 150%. The alteration of the friction coefficient is facilitated through the utilization of the "lateralFriction" flag within the changeDynamics function available in PyBullet. This process ensures the simulation of diverse frictional interactions, thereby contributing to the exploration of slippage behaviors within the experimental setup.

4.4.3. Randomization of stiffness properties

In the final stage of randomization, the deformation of the object is addressed to expose the agent to diverse stiffness conditions. This is achieved by utilizing the "contactStiffness" flag within the changeDynamics function in PyBullet. The contact stiffness value is randomized within a predefined range, specifically ranging from 20 N/cm to 500 N/cm [47]. This broad range of stiffness values encompasses various stiffness conditions, enabling the exploration of different deformation behaviors experienced during grasping and lifting tasks.

4.5. Adaptive sliding mode control based augmentation for RL policy robustification

4.5.1. AFNITSM controller for the robotic gripper

The dynamics of the anthropomorphic hand is given as:

$$B(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + \tau_{ext} \tag{3}$$

where $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^n$ stand for the vectors of position, velocity, and acceleration of the joints, $B(\theta) = B_0(\theta) + \Delta B(\theta)$ represents the inertia matrix, $C(\theta, \dot{\theta}) = C_0(\theta, \dot{\theta}) + \Delta C(\theta, \dot{\theta})$ represents the centripetal Coriolis matrix, $g(\theta) = g_0(\theta) + \Delta g(\theta)$ represents the gravitational matrix, τ represents the joint torques, τ_{ext} represents the external disturbance on the joint torque input. $B_0(\theta)$, $C_0(\theta, \dot{\theta})$, and $g_0(\theta)$ are the nominal terms of the dynamic model, and $\Delta B_0(\theta)$, $\Delta C_0(\theta, \dot{\theta})$, and $\Delta g_0(\theta)$ are the uncertainties of the dynamic model.

The dynamical equation in Eq. (3) can be written as:

$$B_0(\theta)\ddot{\theta} + C_0(\theta, \dot{\theta})\dot{\theta} + g_0(\theta) = \tau + \tau_d \tag{4}$$

where $\tau_d = \tau_{ext} - \Delta B(\theta)\ddot{\theta} - \Delta C(\theta, \dot{\theta})\dot{\theta} - \Delta g(\theta)$ represents the lumped disturbances of the robotic gripper.

We define the desired position vector by θ_d, and the tracking error is then defined as $e_1 = \theta - \theta_d$.

To design a robust chattering-free controller for a robotic manipulator with dynamics model (3), which is capable of tracking the desired trajectory accurately, some assumptions are presented as follows:

Assumption 1. *τ_d is bounded, which satisfies the following function:*

$$\|\tau_d\| < a_0 + a_1 \|q\| + a_2 \|\dot{q}\|^2 \tag{5}$$

where a_0, a_1, and a_2 are unknown positive constants. $\|.\|$ represents the 2-norm.

Assumption 2. $\|\dot{\tau}_d\|$ *is bounded,*

The derivatives of the tracking error are:

$$\begin{aligned} \dot{e} &= \dot{\theta} - \dot{\theta}_d \\ \ddot{e} &= \ddot{\theta} - \ddot{\theta}_d \\ \dddot{e} &= \dddot{\theta} - \dddot{\theta}_d \end{aligned} \tag{6}$$

Using Eq. 4, we have:

$$\ddot{e} = B_0{}^{-1}(\theta)[\tau - f(\theta, \dot{\theta}) + \tau_d] - \ddot{\theta} \tag{7}$$

where $f(\theta, \dot{\theta}) = C(\theta, \dot{\theta})\dot{\theta} + G(\theta)$. *Differentiation of the above equation gives:*

$$\begin{aligned} \dddot{e} &= \dot{B}_0^{-1}(\theta)[\tau - f(\theta, \dot{\theta}) + \tau_d] + B_0{}^{-1}(\theta)[\dot{\tau} - f(\theta, \dot{\theta}) + \dot{\tau}_d(\theta, \dot{\theta}, \ddot{\theta})] - \dddot{\theta}_d \\ &= \dot{B}_0^{-1}(\theta)[\tau - f(\theta, \dot{\theta})] + B_0{}^{-1}(\theta)[\dot{\tau} - f(\theta, \dot{\theta})] - \dddot{\theta}_d + F(\theta, \dot{\theta}, \ddot{\theta}) \end{aligned} \tag{8}$$

where $F(\theta, \dot{\theta}, \ddot{\theta}) = \dot{B}_0^{-1}(\theta)\tau_d + B_0^{-1}\dot{\tau}_d(\theta, \dot{\theta}, \ddot{\theta})$

Using Assumptions 1–2, we obtain $\left\|F(\theta, \dot{\theta}, \ddot{\theta}\right\| < b_0 + b_1 \|\theta\| + b_2 \left\|\dot{\theta}\right\|^2$, where b_0, b_1 and b_2 are unknown positive constants.

4.5.2. Second-order adaptive integral terminal sliding mode controller design:

The integral sliding mode control (*ISMC*) surface can be given as [48]:

$$s = \dot{e} + c_1 e + c_2 \int e d\tau - \dot{e}(0) - c_1 e(0) \tag{9}$$

where $c_1 = diag(c_{11}, c_{12}, \ldots, c_{1n})$ and $c_2 = diag(c_{21}, c_{22}, \ldots, c_{2n})$ are positive-definite matrices. The first and second-time derivatives give:

$$\dot{s} = \ddot{e} + c_1 \dot{e} + c_2 e, \tag{10}$$

$$\ddot{s} = \dddot{e} + c_1 \ddot{e} + c_2 \dot{e}, \tag{11}$$

To expedite finite-time convergence and mitigate singularity issues, we employ a fast non-singular integral terminal sliding mode (*FNITSM*). This approach ensures rapid convergence of the system state s to equilibrium within a finite-time span, both in regions far from and close to equilibrium, while circumventing singularity problems. The design of the second-order fast nonsingular integral terminal sliding mode controller (*SOFNITSMC*) surface is formulated as follows [37, 49, 50]:

$$\sigma = \dot{s} + \int_0^t [\beta_1 \lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) + \beta_2 \lambda_2(\gamma_2, \rho_2, \dot{s}, \varepsilon_2)] d\tau \tag{12}$$

where,

$$\lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) = \begin{cases} \text{sgn} s^{\gamma_1}, & |s| \le \varepsilon_1 \\ {\varepsilon_1}^{\gamma_1 - \rho_1} \text{sgn} s^{\gamma_2}, & |s| > \varepsilon_1 \end{cases} \tag{13}$$

$$\lambda_2(\gamma_2, \rho_2, \dot{s}, \varepsilon_2) = \begin{cases} \text{sgn} \dot{s}^{\gamma_2}, & |\dot{s}| \le \varepsilon_2 \\ {\varepsilon_2}^{\gamma_2 - \rho_2} \text{sgn} \dot{s}^{\gamma_2}, & |\dot{s}| > \varepsilon_2 \end{cases} \tag{14}$$

$\beta_1 = diag(\beta_{11}, \beta_{12}, \ldots, \beta_{1n}$ and $\beta_2 = diag(\beta_{21}, \beta_{22}, \ldots, \beta_{2n}$ are positive-definite matrices, and γ_i, ε_i, and ρ_i are constants that satisfy $0 < \gamma_2 < 1, \gamma_1 = \gamma_2/(2 - \gamma_2), \rho_i \ge 1$ and $\varepsilon_i > 0 \quad (i = 1, 2)$. The parameters γ_1 and γ_2 are critical in shaping the nonlinearity, ensuring finite-time convergence, and improving the smoothness and robustness of the sliding mode control [37].

The time derivative of (12) is:

$$\dot{\sigma} = \ddot{s} + \beta_1 \lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) + \beta_2 \lambda_2(\gamma_2, \rho_2, \dot{s}, \varepsilon_2) \tag{15}$$

The control law of a sliding mode control (SMC) consists of an equivalent control law τ_{eq} and a switching control law τ_{sw}. The SOITSSMC is chosen as:

$$\tau = \tau_{eq} + \tau_{sw} = \int_0^t (\dot{\tau}_{eq} + \dot{\tau}_{sw}) d\tau \tag{16}$$

The τ is designed to guarantee the σ converges to zero. We can get $\dot{\tau}_{eq}$ by making $F(x, \dot{x}, \ddot{x}) = 0$, and then we design $\dot{\tau}_{sw}$ i.e., the discontinuous control action, to deal with the disturbances.

When $F(x, \dot{x}, \ddot{x}) = 0$, and let $\sigma = 0$, this enables us to obtain $\dot{\tau}_{eq}$ and $\dot{\tau}_{sw}$ as follows:

$$\begin{aligned}\dot{\tau}_{eq} = &-B_0(x)\dot{B}_0^{-1}[\tau - f(x, \dot{x})] + B_0(x)(\ddot{x}_d - c_1\ddot{e} - c_2\dot{e}) - B_0[\beta_1\lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) \\ &+ \beta_2\lambda_2(\gamma_2, \rho_2, s, \varepsilon_2)] + \dot{f}(x, \dot{x}),\end{aligned} \tag{17}$$

$$\dot{\tau}_{sw} = -B_0(x)[k\sigma + (b_0 + b_1 \|x\| + b_2 \|\dot{x}\|^2)] \tag{18}$$

where $k = diag(k_1, k_2, \ldots, k_n)$

Adaptive law design:. In practice, we cannot obtain the values of b_0, b_1, and b_2 in Eq. 18. Therefore, we will use the adaptive parameter tuning scheme to estimate them:

$$\dot{\tau}_{asw} = -B_0(x)[k\sigma + (\hat{b}_0 + \hat{b}_1 \|x\| + \hat{b}_2 \|\dot{x}\|^2)] \tag{19}$$

where $\hat{b}_0$, $\hat{b}_1$ and $\hat{b}_2$ are the respective estimates. The adaptive laws for $\hat{b}_i (i = 1, 2, 3)$ are as follows:

$$\begin{aligned}\dot{\hat{b}}_0 &= \|\sigma\| \\ \dot{\hat{b}}_1 &= \|\sigma\| \, \|x\| \\ \dot{\hat{b}}_2 &= \|\sigma\| \, \|\dot{x}\|^2\end{aligned} \tag{20}$$

We define the adaptation error as $\tilde{b}_i = b_i - \hat{b}_i (i = 0, 1, 2)$. The second-order adaptive fast nonsingular integral terminal sliding mode controller (*SOAFITSMC*) is then designed as follows:

$$\tau = \tau_{eq} + \tau_{asw} = \int_0^t (\dot{\tau}_{eq} + \dot{\tau}_{asw}) d\tau \tag{21}$$

where,

$$\begin{aligned}\dot{\tau}_{eq} = &-B_0(x)\dot{B}_0^{-1}[\boldsymbol{\tau} - f(x, \dot{x})] + B_0(x)(\ddot{x}_d - c_1\ddot{e} - c_2\dot{e}) - B_0[\beta_1\lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) \\ &+ \beta_2\lambda_2(\gamma_2, \rho_2, s, \varepsilon_2)] + \dot{f}(x, \dot{x}),\end{aligned} \tag{22}$$

$$\dot{\tau}_{asw} = -B_0(x)[k\sigma + (\hat{b}_0 + \hat{b}_1 \|x\| + \hat{b}_2 \|\dot{x}\|^2)] \tag{23}$$

4.5.3. Stability analysis

In this section, we present the stability analysis of the *SOAFNITSMC*. The Lyapunov function candidate is considered as follows:

$$V = \frac{1}{2}\left(\sigma^T\sigma + \mu_0\tilde{b}_0^2 + \mu_1\tilde{b}_1^2 + \mu_2\tilde{b}_2^2\right) \tag{24}$$

The time derivative of the above equation gives:

$$\begin{aligned}\dot{V} &= \sigma^T\dot{\sigma} + \mu_0\tilde{b}_0\dot{\tilde{b}}_0 + \mu_1\tilde{b}_1\dot{\tilde{b}}_1 + \mu_2\tilde{b}_2\dot{\tilde{b}}_2 \\ &= \sigma^T[\ddot{s} + \beta_1\lambda_1(\gamma_1, \rho_1, s, \varepsilon_1) + \beta_2\lambda_2(\gamma_2, \rho_2, \dot{s}, \varepsilon_2)] + \mu_0\tilde{b}_0\dot{\hat{b}}_0 + \mu_1\tilde{b}_1\dot{\hat{b}}_1 + \mu_2\tilde{b}_2\dot{\hat{b}}_2\end{aligned} \tag{25}$$

Substituting Eqs. 8 and 10 into Eq. 25

$$\dot{V} = \sigma^T[\dot{B}_0^{-1}(x)[\tau - f(x,\dot{x})] + B_0{}^{-1}(x)[\dot{\tau} - f(x,\dot{x})] - \dddot{x}_d + F(x,\dot{x},\ddot{x}) + c_1\ddot{e} + c_2\dot{e}\beta_1\lambda_1(\gamma_1,\rho_1,s,\varepsilon_1) + \beta_2\lambda_2(\gamma_2,\rho_2,\dot{s},\varepsilon_2)] + \mu_0\tilde{b}_0\dot{\hat{b}}_0 + \mu_1\tilde{b}_1\dot{\hat{b}}_1 + \mu_2\tilde{b}_2\dot{\hat{b}}_2 \tag{26}$$

Using Eqs. (22) and (23), we have:

$$\begin{aligned}
\dot{V} &= \sigma^T[-k\sigma - (\hat{b}_0 + \hat{b}_1\|x\| + \hat{b}_2\|\dot{x}\|)sign(\sigma) + F(x,\dot{x},\ddot{x})] + \mu_0\tilde{b}_0\dot{\hat{b}}_0 + \mu_1\tilde{b}_1\dot{\hat{b}}_1 + \mu_2\tilde{b}_2\dot{\hat{b}}_2 \\
&\le \|F(x,\dot{x},\ddot{x})\|\|\sigma\| - \|\hat{b}_0 + \hat{b}_1\|x\| + \hat{b}_2\|\dot{x}\|^2\| + \|b_0 + b_1\|x\| + b_2\|\dot{x}\|^2\|\|\sigma\| \\
&- \|b_0 + b_1\|x\| + b_2\|\dot{x}\|^2\|\|\sigma\| + \mu_0\tilde{b}_0\dot{\hat{b}}_0 + \mu_1\tilde{b}_1\dot{\hat{b}}_1 + \mu_2\tilde{b}_2\dot{\hat{b}}_2 \\
&\le -[\|(b_0 + b_1\|x\| + b_2\|\dot{x}\|^2)\| - \|F(x,\dot{x},\ddot{x}\|\|\sigma\|] - \|\tilde{b}_0\|(\|\sigma\| - \mu_0\sigma) - \|\tilde{b}_1\|(\|\sigma\|\|x\| \\
&- \mu_1\|\sigma\|\|x\|) - \|\tilde{b}_2\|(\|\sigma\|\|\dot{x}\|^2 - \mu_2\|\sigma\|\|\dot{x}\|^2) \\
&\le -\varrho\sqrt{2}\frac{\|\sigma\|}{\sqrt{2}} - \xi_0\sqrt{2\mu_0}\frac{\|\tilde{b}_0\|}{\sqrt{2\mu_0}} - \xi_1\sqrt{2\mu_1}\frac{\|\tilde{b}_1\|}{\sqrt{2\mu_1}} - \xi_2\sqrt{2\mu_2}\frac{\|\tilde{b}_2\|}{\sqrt{2\mu_2}}
\end{aligned}$$

where

$$\begin{aligned}
\varrho &= \left\|(b_0 + b_1\|x\| + b_2\|\dot{x}\|^2\right\| - \|F(x,\dot{x},\ddot{x}\|\,, \\
\xi_0 &= \|\sigma\| - \mu_0\|\sigma\| = (1-\mu_0)\|\sigma\|\,, \\
\xi_1 &= (\|\sigma\|\|x\| - \mu_1\|\sigma\|\|x\|) = (1-\mu_1)\|\sigma\|\|x\|\,, \\
\xi_2 &= (\|\sigma\|\|\dot{x}\|^2 - \mu_2\|\sigma\|\|\dot{x}\|^2) = (1-\mu_2)\|\sigma\|\|\dot{x}\|^2
\end{aligned} \tag{27}$$

We then obtain:

$$\begin{aligned}
\dot{V} &\le -\min\left\{\sqrt{2}\varrho, \sqrt{2\mu_0^{-1}}\xi_0, \sqrt{2\mu_1^{-1}}\xi_1, \sqrt{2\mu_2^{-1}}\xi_2\right\} \times \left(\frac{\|\sigma\|}{\sqrt{2}} + \frac{\sqrt{\mu_0}}{\sqrt{2}}\|\tilde{b}_0\| + \frac{\sqrt{\mu_1}}{\sqrt{2}}\|\tilde{b}_1\| + \frac{\sqrt{\mu_2}}{\sqrt{2}}\|\tilde{b}_2\|\right) \\
&\le -\alpha V^{1/2}
\end{aligned} \tag{28}$$

where

$$\alpha = \min\left\{\sqrt{2}\varrho, \sqrt{2\mu_0{}^{-1}}\xi_0, \sqrt{2\mu_1{}^{-1}}\xi_1, \sqrt{2\mu_2{}^{-1}}\xi_2\right\}$$

and $\alpha > 0$. The above inequality holds if $\mu_0 < 1$, $\mu_1 < 1$, and $\mu_2 < 1$. The stability of the Lyapunov function in the context of SMC often relies on ensuring that certain parameters or terms within the Lyapunov candidate function remain bounded or decrease over time. For the Lyapunov function V to be positive definite, all terms must be positive or zero. The given function V is quadratic in nature and thus positive definite as long as σ, $\tilde{b}_0$, $\tilde{b}_1$, and $\tilde{b}_2$ are finite and μ_0, μ_1, μ_2 are positive. The parameters μ_0, μ_1, μ_2 scale the estimation errors or uncertainties. If $\mu_i \ge 1$, the corresponding term $\mu_i\tilde{b}_i^2$ could grow large, potentially causing V to increase or not decrease sufficiently, leading to instability. By keeping $\mu_i < 1$, the influence of the estimation errors on the Lyapunov function is bounded, ensuring that the function remains decreasing or non-increasing.

If $t_0 = 0$, then σ in Eq. 12 can converge to zero in a finite time $t_1 = V^{1-1/2}(0)/\alpha(1-1/2) = 2V^{1/2}(0)/\alpha$. Thus, the sliding variable s will converge to zero in a finite time. The tracking error e can asymptotically converge to zero. This completes the proof of stability.

The control architecture of the post-training augmented robust *RL* policy is depicted in Figure 6. This architecture closely resembles an indirect force control setup with an inner position loop [10, 11], wherein the reference force is derived from the trained *RL* policy. Following the application of the adaptive control input to the anthropomorphic hand, force error signals are generated by comparing the experimental force signals F_e with the desired force signals F_d. The desired force F_d is linked to the generated pre-trained *RL* policy (in terms of torques) through the Jacobian transpose. Subsequently, the

Table II. *Physical parameters of the anthropomorphic hand used for simulation.*

Symbol	Definition	Values
m_1	Nominal mass of proximal phalange	0.01 kg
m_2	Nominal mass of medial phalange	0.005 kg
m_3	Nominal mass of distal phalange	0.004 kg
l_1	Length of proximal phalange	5.5 cm
l_2	Length of medial phalange	3.5 cm
l_3	Length of distal phalange	2.5 cm
J_1	Moment of inertia of proximal phalange	0.1 kg m^2
J_2	Moment of inertia of medial phalange	0.85 kg m^2
J_3	Moment of inertia of distal phalange	0.75 kg m^2
g	Acceleration due to gravity	9.81 m/s^2

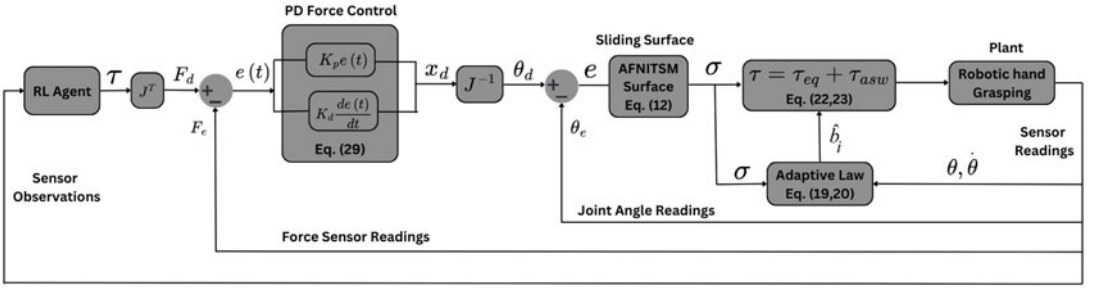

Figure 6. *Post-training augmented robust reinforcement learning control diagram for perturbed environment. This is the detailed diagram of the adaptive fast non-singular integral terminal sliding mode shown in Figure 3.*

error value $e(t)$ serves as the input for a proportional-derivative force control algorithm, comprising a proportional and a derivative regulator. The output x_d of this algorithm is given by:

$$x_d = K_p e(t) + K_d \frac{de(t)}{dt} \tag{29}$$

where K_p and K_d are the proportional and derivative gains, respectively. The values of K_p and K_d can be identified by manual tuning through experimental tests to adjust the reaction time and limit the overshooting of the measured F_e with respect to F_d.

The output x_d is added to the measured value of the finger position θ_e to give the position error signals, e_p. This error signal serves as the input for the sliding surface as given in Eq. 6.

The nominal parameters for the simulation and the control parameters taken are given in Table II (details of the dynamical equation given in Appendix) and Table III.

5. Results and discussions

5.1. Reward plots of the nominal and domain randomized agents

The reward plots, depicting the average reward attained by *RL* agents trained under different scenarios (one in the nominal environment and the other in the *DR* environment), are presented in Figure 7. The black-colored graph illustrates the reward plot of the *RL* agent trained in the nominal environment, while the orange-colored graph represents the reward plot of the *RL* agent subjected to randomization of mass, friction coefficient, and object stiffness. Although the convergence times of the reward plots are comparable, the *DR* agent's rewards reach convergence slightly later than those of agents trained in the nominal environment. This delay can be attributed to disturbances induced by the randomized

Table III. *Parameters of the proposed controller and their values.*

Parameters	Values
c_1	10
c_2	85
β_1	3
β_2	5
ρ_1	1.3
ρ_2	1.1
γ_1	0.1
γ_2	0.1
k_1	150
k_2	150
ε_1	0.02
ε_2	0.02
ϵ	0.1
$b_i(0)$	0

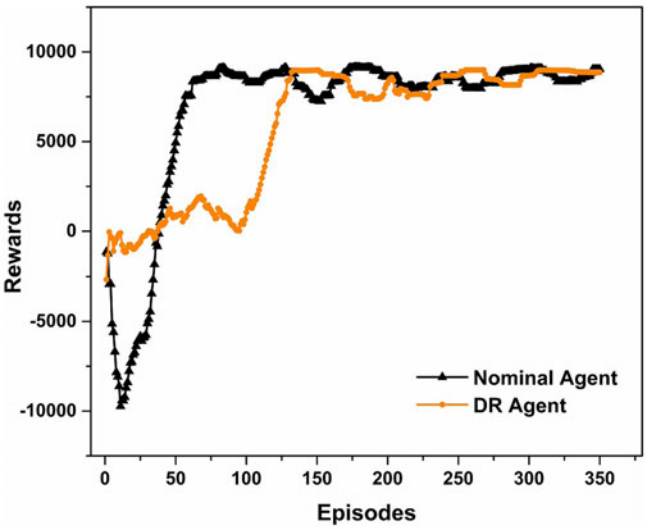

Figure 7. *Average reward plots of nominal and domain randomization/domain randomized agent.*

parameters namely, weight, friction, and stiffness–resulting in increased occurrences of slippage and deformation, thereby causing a decline in cumulative reward during the initial stages. Nonetheless, the domain-randomized *RL* agent exhibits superior performance in tasks involving the grasping of unknown objects, as discussed in the subsequent section. The success of the trained agent's grasp simulation is evident in Figure 8, where the agent adeptly grasps and lifts the object without experiencing slippage while minimizing deformation, as illustrated in Figure 8c.

5.2. Success rates for nominal and domain randomized agents

Figure 9 shows performance tests of both the agents trained on the nominal environment as well as on the *DR* environment on the unknown object grasping task. The unknown objects are randomized to

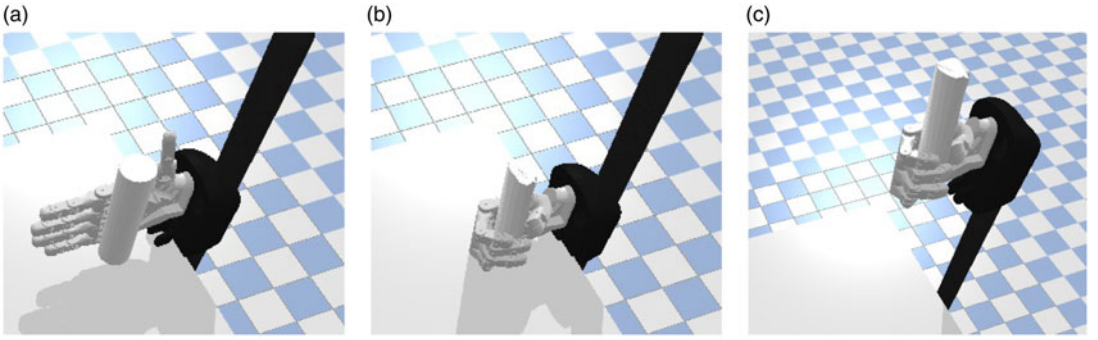

Figure 8. *Learned grasp simulation: (a) Initial grasp pose, (b) Grasping the object, and (c) Object lift without slippage.*

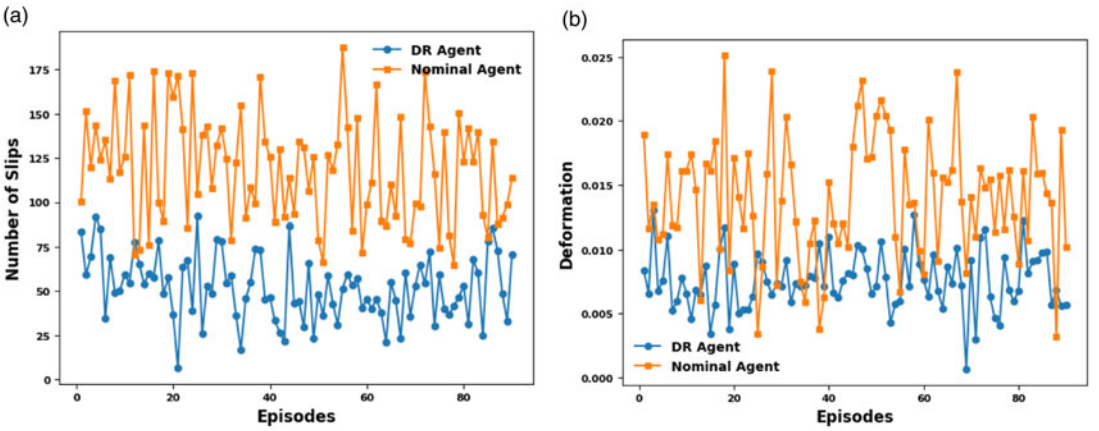

Figure 9. *Performance tests of success rates on unknown objects. Parameters randomized were the object weights, stiffness, and friction coefficients while grasping: (a) Slips prevented by nominal and domain randomization/domain randomized (DR) agent at unseen objects task. (b) Amount of deformation (in mm) prevented by nominal and DR agent on unknown objects task.*

have different weights, stiffness, and friction properties. Figure 9a shows the success rates of preventing slippage while grasping unknown objects for nominal and *DR* agents. The x-axis shows the number of episodes for which the learned agent was tested. The y-axis represents the frequency of slips occurring. From the success rates plot, it is evident that the agent trained in a *DR* environment has been able to prevent more slips than the nominal agent. Figure 9b shows the deformation occurring on the deformable object when grasped by the nominal and the *DR*-trained agent on unknown objects tasks with different object properties. The x-axis shows the number of episodes for which the learned agent was tested. The y-axis represents the amount of deformation undergone while the object is being grasped. It is seen that while grasping objects with unknown properties, the *DR* agent is able to grasp the object with lesser deformation than the nominal agent, together with preventing slip and droppage.

To validate the efficacy of the results obtained, we repeat the above experiments over 10 trials. We then present the error bar plots for each iteration to provide the visual representation of variability and illustrate the statistical significance of the experimental outcomes (shown in Figure 10). From both performance tests and error bar plots, we can conclude that the learned *DR* agent is better equipped to improve the generalization capability than the nominal agent.

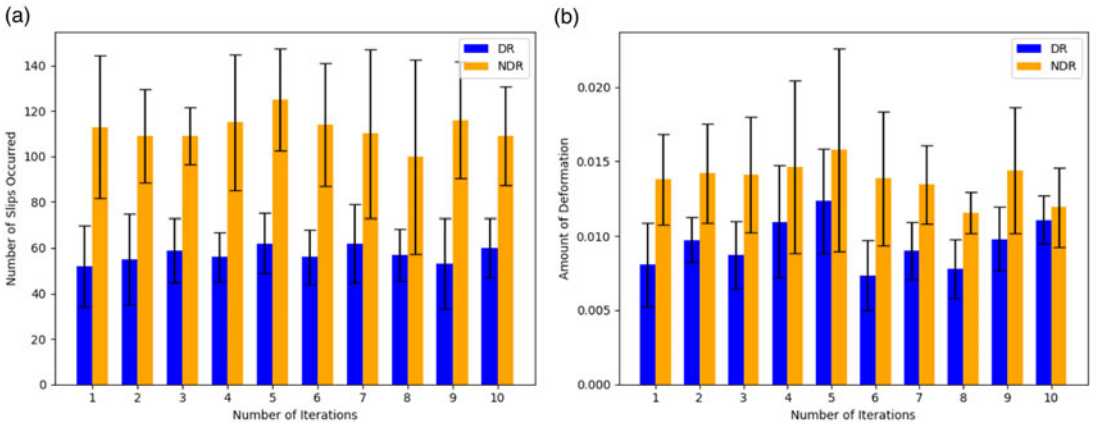

Figure 10. *Error bar plots for (a) Slips prevented by nominal and domain randomization/domain randomized (DR) agent at unknown objects task. (b) Amount of deformation (in mm) prevented by nominal and DR agent on unknown objects task. Here DR represents a domain-randomized agent, and Non-Domain Randomized (NDR) represents non-domain-randomized agent, i.e., nominal agent.*

5.3. Success rates of domain randomized and robustified agent

To evaluate the efficacy of the robustified post-augmented *RL* controller relative to the nominal domain-randomized agent, we performed a success rate assessment. This evaluation involved quantifying the occurrences of slips and the extent of deformation during the process of object grasping and lifting under conditions characterized by randomized parameters (including object weights, friction, and contact stiffness), alongside the introduction of sinusoidal disturbances.

$$\tau_d = \left[2 \sin(t) + 0.5 \sin(200\pi t) \right] \tag{30}$$

The time-varying external input disturbances described above serve as a standardized benchmark commonly employed in robotic manipulator control problems [51, 52]. The success rate plot presented in Figure 11 quantifies the occurrence of slips and the extent of deformation for both the *DR* and robustified agents. Figure 11a illustrates the success rates in preventing slippage during the grasping of unknown objects by both nominal and *DR*-trained agents. The x-axis denotes the number of episodes during which the learned agent was tested, while the y-axis denotes the frequency of slip occurrences. The results indicate that the agent trained in a *DR* environment exhibits a superior capability in preventing slips compared to the nominal agent. Furthermore, Figure 11b illustrates the deformations observed in the deformable object when grasped by both nominal and *DR*-trained agents across tasks involving unknown objects with varying properties. Notably, the success rate plots underscore the enhanced performance of the post-training augmented robust controller, which manifests in fewer instances of slip and deformation occurrences compared to the nominal domain-randomized agent.

To validate the robustness of our experimental results, we conducted multiple iterations of the success rate tests, totaling ten trials. Error bar plots were generated for each iteration to visually represent the variability and assess the statistical significance of the experimental outcomes (as depicted in Figure 12). In comparing the two groups of error bar plots concerning slippage (Figure 12a), namely the DR agent group and the robust agent, we computed the difference in means between the two groups and contrasted it with their combined standard deviations. This approach provides insight into the magnitude of the difference relative to the variability within each group. The resulting percentage difference was calculated as 103.35%, with a Cohen's D test value of 4.08. Cohen's D test assesses the effect size, quantifying the magnitude of the difference between groups. Similarly, for the deformation error bar plots (Figure 12b), the percentage difference relative to the variability was determined to be 197%, with a corresponding Cohen's D test value of 5.35. These statistical analyses, coupled with the performance tests, indicate

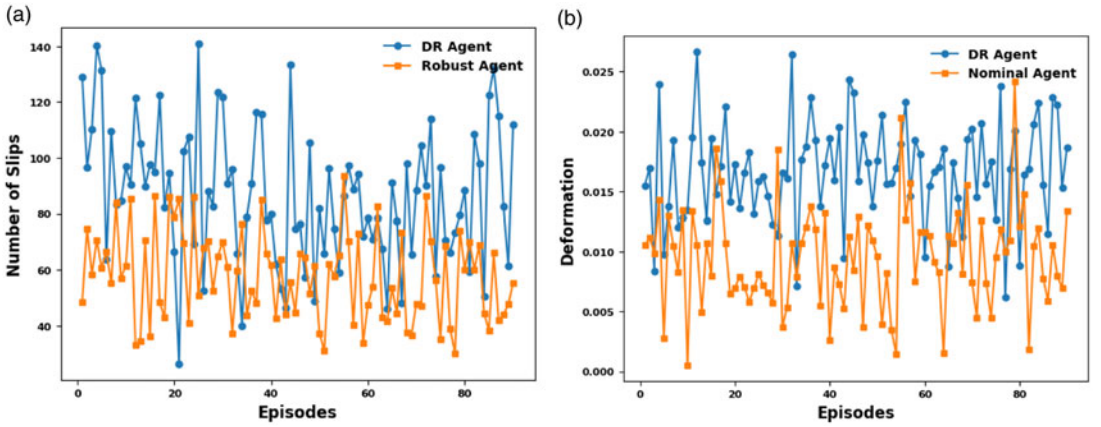

Figure 11. *Performance tests of success rates on unseen objects. Parameters randomized were the object weights, stiffness, and friction coefficients while grasping: (a) Slips prevented by domain randomization/domain randomized (DR) agent and robust agent at unseen objects task. (b) Amount of deformation (in mm) prevented by DR agent and robust agent on unknown objects task.*

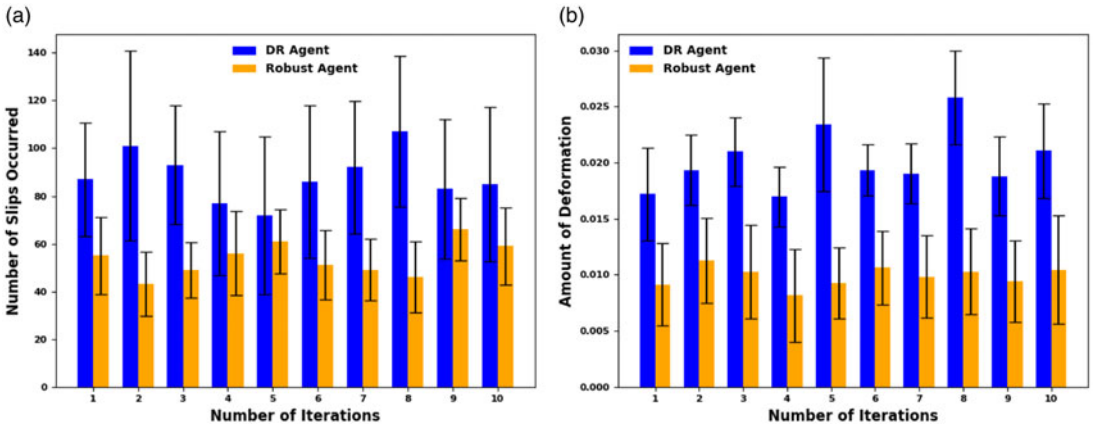

Figure 12. *Error bar plots of statistical tests for (a) Slips prevented by domain randomization/domain randomized (DR) agent and robust agent at unseen objects task. (b) Amount of deformation (in mm) prevented by DR agent and robust agent on unknown objects task.*

that the post-training-based robustified agent demonstrates superior generalization capability compared to the nominal *DR* agent.

6. Comparison with state-of-the-art

Although this article represents a novel effort in implementing bionic reflex control using reinforcement learning, a comparative analysis of our bionic reflex control system against state-of-the-art slippage and deformation prevention methods is reported. Table IV lists the values for various elements of comparison, including well-recognized indicators of the performance of our method and several recent slippage and deformation prevention methods. The comparison criteria include the type of robotic hand, the sensor technology employed, whether simulation study or real-world implementation, the slip response time, and object deformation minimization. Though our main baselines are five-fingered hands, I also list two and three-fingered grippers. Compared to both grippers and five-fingered hands, our

Table IV. *Comparison of our method with the state-of-the-art.*

Method	Hand Type	Fingertip Sensor	Simulation/ Real	Slip Response Time	Object Deformation Minimized
Multi-threshold-based control [8]	Five Fingered Hand	FSR	Real	<160 ms	No
ON-OFF Control [5]	Five Fingered Hand	FSR	Real	≈ 200 ms	No
Switched Adaptive Regrasping [53]	Two Fingered Gripper	FSR Laser Sensor	Real	≈ 500 ms	Yes
Hybrid Position/Force Control [54]	Three-link soft finger	NA	Simulation	≈ 300 ms	No
Feedback Linearization [55]	Two Fingered Gripper	NA	Simulation	≈ 750 ms	No
Variable Impedance Control [56]	Two Fingered Gripper	NA	Simulation	≈ 200 ms	No
Ours	Five Fingered Hand	Force sensor	Simulation	≈ 200 ms	Yes

proposed methodology performs competitively, validating the efficacy of reinforcement learning-based control.

This research has led to the development of a real-time adaptive bionic reflex controller trained in a physics-based simulator that is deployed within a Sim-to-Sim testing environment. While experimental validation in a physical setting is undeniably valuable, we would like to emphasize the following key points supporting the adequacy of simulation-based validation in this context: Simulation Fidelity and Realism: The simulation environment used in the research was designed with high fidelity to replicate real-world conditions accurately. This includes detailed modeling of the underactuated prosthetic hand, incorporating realistic joint dynamics, sensor noise, and frictional interactions between the hand and various objects. Such use of advanced state-of-the-art physics engines ensures that the simulation outcomes are representative of real-world performance [57]. Theoretical Foundation and Control Strategy: The novel grasp reflex control strategy, based on deep reinforcement learning, addresses the challenge of minimizing slippage and deformation. The simulation studies provided a rigorous framework to test and refine the control policy across a wide range of object and contact properties, along with domain randomization induced to capture uncertainties of the real world [58]. Future investigations may be taken up toward Sim-to-Real implementation by integrating the controller into a 3D-printed anthropomorphic hand equipped with embedded control.

7. Conclusion

In this study, we introduce a novel technique aimed at increasing the robustness of pre-trained *RL* policies via post-training augmented adaptive control. The primary objective is to enhance the generalization capabilities of learned policies. Our approach entails a hierarchical integration strategy, where an adaptive sliding mode controller is incorporated into the existing *RL* policy framework. This augmentation is designed to robustify the pre-trained agent's bionic reflex capability. Through extensive evaluation via success rate tests of minimizing slip occurrences and deformation levels during object manipulation under matched disturbances, we validate the efficacy of our proposed robustification methodology. The

utilization of adaptive controllers exhibits potential for enhancing the performance of robotic manipulators and upper-limb prosthetic devices. Subsequent research endeavors will prioritize the refinement of adaptive control algorithms by incorporating predefined convergence criteria and introducing greater variability in physical parameters during grasping and manipulation tasks.

Author contributions. Hirakjyoti Basumatary and Shyamanta M. Hazarika conceived and designed the work. Hirakjyoti Basumatary performed the simulations and generation of results. Daksh Adhar also performed part of the simulations. Shyamanta M. Hazarika guided the progress and reviewed the work.

Financial support. This work was supported in part by MHRD, Government of India, through Indian Institute of Technology, Guwahati, for Doctoral Research. Financial support received from DST, Government of India, under Project Grant TDP/BDTD/21/2019 is gratefully acknowledged.

Competing interests. None.

Ethical considerations. None.

Acknowledgements. None.

References

[1] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *Int. J. Robot. Res.* **37**(7), 688–716 (2018).
[2] H. Basumatary and S. M. Hazarika, "State of the art in bionic hands," *IEEE T. Hum-MACH. Syst.* **50**(2), 116–130 (2020).
[3] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, L. Xiang, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robot. Autom. Mag.* **29**(3), 67–77 (2022).
[4] R. A. Romeo and L. Zollo, "Methods and sensors for slip detection in robotics: A survey," *IEEE Access* **8**, 73027–73050 (2020).
[5] R. A. Romeo, C. Lauretti, C. Gentile, E. Guglielmelli and L. Zollo, "Method for automatic slippage detection with tactile sensors embedded in prosthetic hands," *IEEE T. Med. Robot. Bion.* **3**(2), 485–497 (2021).
[6] Y. Cheng, P. Zhao, F. Wang, D. J. Block and N. Hovakimyan, "Improving the robustness of rreinforcement learning olicies with l1 adaptive control," *IEEE Robot. Auto. Lett.* **7**(3), 6574–6581 (2022).
[7] J. W. James and N. F. Lepora, "Slip detection for grasp stabilization with a multifingered tactile robot hand," *IEEE T. Robot.* **37**(2), 506–519 (2020).
[8] D. Yang and G. Wu, "A multi-threshold-based force regulation policy for prosthetic hand preventing slippage," *IEEE Access* **9**, 9600–9609 (2021).
[9] K. Nazari and W. Mandil, "roactive slip control by learned slip model and trajectory adaptation," (2022). arXiv preprint arXiv: 2209.06019.
[10] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo. *Force Control* (Springer, 2009).
[11] G. Carbone, S. Iannone and M. Ceccarelli, "Regulation and control of LARM Hand III," *Robot. Comp-INT. Manuf.* **26**(2), 202–211 (2010).
[12] E. D. Engeberg and S. G. Meek, "Adaptive sliding mode control for prosthetic hands to simultaneously prevent slip and minimize deformation of grasped objects," *IEEE/ASME T. Mechtron.* **18**(1), 376–385 (2011).
[13] Y. Zhang, X. Xu, R. Xia and H. Deng, "Stiffness-estimation-based grasping force fuzzy control for underactuated prosthetic hands," *IEEE/ASME T. Mechatron.* **28**(1), 140–151 (2022).
[14] A.-M. Cretu, P. Payeur and E. M. Petriu, "Soft object deformation monitoring and learning for model-based robotic hand manipulation," *IEEE T. Syst. Man Cybern. Part B (Cybernetics)* **42**(3), 740–753 (2011).
[15] K. Makihara, Y. Domae, I. G. Ramirez-Alpizar, T. Ueshiba and K. Harada, "Grasp pose detection for deformable daily items by pix2stiffness estimation," *Adv. Robot.* **36**(12), 600–610 (2022).
[16] B. Shen, Z. Jiang, C. Choy, L. J. Guibas, S. Savarese, A. Anandkumar and Y. Zhu, "Acid: Action-conditional implicit visual dynamics for deformable object manipulation," (2022). arXiv preprint arXiv: 2203.06856.
[17] W. Ji, J. Zhang, B. Xu, C. Tang and D. Zhao, "Grasping mode analysis and adaptive impedance control for apple harvesting robotic grippers," *Comput. Electron. Agr.* **186**, 106210 (2021).
[18] X.-G. Duan, Y. Zhang and H. Deng, "A simple control method to avoid overshoot for prosthetic hand control," In 2014 IEEE International Conference on Information and Automation (ICIA), IEEE (2014) pp. 736–739.
[19] L. Jiang, X. Tian, Q. Zhan, Q. Xu and Y. Zhang, "Impedance control of an anthropomorphic hands without finger force sensors," *IEEE T. Autom. Sci. Eng.* **21**(4), 5779–5789 (2023).
[20] H. Deng, G. Zhong, X. Li and W. Nie, "Slippage and deformation preventive control of bionic prosthetic hands," *IEEE/ASME T. Mechatron.* **22**(2), 888–897 (2016).

[21] M. Kaboli, K. Yao and G. Cheng, "Tactile-based manipulation of deformable objects with dynamic center of mass," In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), IEEE (2016) pp. 752–757.

[22] N. Mouaze and L. Birglen, "Bistable compliant underactuated gripper for the gentle grasp of soft objects," *Mech. Mach. Theory* **170**, 104676 (2022).

[23] W. Wang and S.-H. Ahn, "Shape memory alloy-based soft gripper with variable stiffness for compliant and effective grasping," *Soft Robot.* **4**(4), 379–389 (2017).

[24] A. Milojević, S. Linß, Žarko Ćojbašić and H. Handroos, "A novel simple, adaptive, and versatile soft-robotic compliant two-finger gripper with an inherently gentle touch," *J. Mech. Robot.* **13**(1), 011015 (2021).

[25] E. Salvato, G. Fenu, E. Medvet and F. A. Pellegrino, "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning," *IEEE Access* **9**, 153171–153187 (2021).

[26] L.ía Güitta-López, J. Boal and Á. lvaro J López-López, "Learning more with the same effort: How randomization improves the robustness of a robotic deep rreinforcement learning gent," *Appl. Intell.* **53**(12), 14903–14917 (2023).

[27] X. Chen, J. Hu, C. Jin, L. Li and L. Wang, "Understanding domain randomization for sim-to-real transfer," (2021). arXiv preprint arXiv: 2110.03239.

[28] L. Pinto, J. Davidson, R. Sukthankar and A. Gupta, "Robust adversarial reinforcement learning," In International Conference on Machine Learning, PMLR (2017) pp. 2817–2826

[29] J. Morimoto and K. Doya, "Robust reinforcement learning," *Neural Comput.* **17**(2), 335–359 (2005).

[30] L. Rice, E. Wong and Z. Kolter, "Overfitting in adversarially robust deep learning," In International Conference on Machine Learning, PMLR (2020) pp. 8093–8104.

[31] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," (2018). arXiv preprint arXiv: 1803.11347.

[32] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu and R. Hadsell, "Policy distillation," (2015). arXiv preprint arXiv: 1511.06295.

[33] Y. Kadokawa, L. Zhu, Y. Tsurumine and T. Matsubara, "Cyclic policy distillation: Sample-efficient sim-to-real rreinforcement learning ith domain randomization," *Robot. Auton. Syst.* **165**, 104425 (2023).

[34] Z. Niu, J. Yuan, X. Ma, Y. Xu, J. Liu, Y.-W. Chen, R. Tong and L. Lin, "Knowledge distillation-based domain-invariant representation learning for domain generalization," *IEEE T. Multimedia*, (2023).

[35] J. W. Kim, H. Shim and I. Yang, "On improving the robustness of reinforcement learning-based controllers using disturbance observer," In 2019 IEEE 58th Conference on Decision and Control (CDC), IEEE (2019) pp. 847–852.

[36] A. Guha and A. Annaswamy, "Mrac-rl: A framework for on-line policy adaptation under parametric model uncertainty," (2020) arXiv preprint arXiv: 2011.10562.

[37] S. Hao, L. Hu and P. X. Liu, "Second-order adaptive integral terminal sliding mode approach to tracking control of robotic manipulators," *IET Control Theory A.* **15**(17), 2145–2157 (2021).

[38] E. Coumans and Y. Bai "Pybullet, a python module for physics simulation for games, robotics and machine learning," (2016). (https://pybullet.org/wordpress/).

[39] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta and P. Abeel, "Soft actor-critic algorithms and applications," (2018), arXiv preprint arXiv: 1812.05905.

[40] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus and N. Dormann, "Stable-baselines3: Reliable rreinforcement learning mplementations," *J. Mach. Learn. Res.* **22**(268), 1–8 (2021).

[41] H. Deng, Y. Zhang and X.-G. Duan, "Wavelet transformation-based fuzzy reflex control for prosthetic hands to prevent slip," *IEEE T. Ind. Electron.* **64**(5), 3718–3726 (2016).

[42] H. Yang, X. Hu, L. Cao and F. Sun, "A new slip-detection method based on pairwise high frequency components of capacitive sensor signals," In 2015 5th International Conference on Information Science and Technology (ICIST), IEEE (2015) pp. 56–61.

[43] R. A. Romeo, U. B. Rongala, A. Mazzoni, D. Camboni, M. C. Carrozza, E. Guglielmelli, L. Zollo and C. M. Oddo, "Identification of slippage on naturalistic surfaces via wavelet transform of tactile signals," *IEEE Sens. J.* **19**(4), 1260–1268 (2018).

[44] Y. Hu, T. Schneider, B. Wang, D. Zorin and D. Panozzo, "Fast tetrahedral meshing in the wild," *ACM T. Graphics (TOG)* **39**(4), 117–111 (2020).

[45] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano and J. L. Wyatt, "Modeling of deformable objects for robotic manipulation: A tutorial and review," *Front. Robot. AI* **7**, 82 (2020).

[46] C. Zhang and T. Chen, "Efficient Feature Extraction for 2d/3d Objects in Mesh Representation," **In:** *Proceedings 2001 International Conference On Image Processing (Cat No. 01CH37205)*, Vol. 3, (IEEE, 2001) pp. 935–938.

[47] X. Ma, L. Chen, Y. Gao, D. Liu and B. Wang, "Modeling contact stiffness of soft fingertips for grasping applications," *Biomimetics* **8**(5), 398 (2023).

[48] V. Utkin and J. Shi, "Integral sliding mode in systems operating under uncertainty conditions," In Proceedings of 35th IEEE conference on decision and control, Vol. 4, IEEE, (1996) pp. 4591–4596.

[49] P. Li, J. Ma, Z. Zheng and L. Geng, "Fast nonsingular integral terminal sliding mode control for nonlinear dynamical systems," In 53rd IEEE conference on decision and control, IEEE (2014) pp. 4739–4746.

[50] K. A. Alattas, S. Mobayen, U. D. Sami, H. A. Jihad, Fekih Afef, A. Wudhichai and T. V. Mai, "Design of a non-singular adaptive integral-type finite time tracking control for nonlinear systems with external disturbances," *IEEE Access* **9**, 102091–102103 (2021).

[51] S. Mondal and C. Mahanta, "Adaptive second order terminal sliding mode controller for robotic manipulators," *J. Frankl. Inst.* **351**(4), 2356–2377 (2014).

[52] M. Boukattaya, N. Mezghani and T. Damak, "Adaptive nonsingular fast terminal sliding-mode control for the tracking problem of uncertain dynamical systems," *ISA T.* **77**, 1–19 (2018).
[53] M. Al-Mohammed, R. Adem and A. Behal, "A switched adaptive controller for robotic gripping of novel objects with minimal force," *IEEE T. Contr. Syst. T.* **31**(1), 17–26 (2022).
[54] A. Fakhari, I. Kao and M. Keshmiri, "Modeling and control of planar slippage in object manipulation using robotic soft fingers," *ROBOMECH. J.* **6**(1), 15 (2019).
[55] A. Fakhari, M. Keshmiri, I. Kao and S. H. Jazi, "Slippage control in soft finger grasping and manipulation," *Adv. Robotics* **30**(2), 97–108 (2016).
[56] M. Logothetis, G. C. Karras, K. Alevizos and K. J. Kyriakopoulos, "A variable impedance control strategy for object manipulation considering non–rigid grasp," In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE (2020) pp. 7411–7416.
[57] J. Collins, S. Chand, A. Vanderkop and D. Howard, "A review of physics simulators for robotic applications," *IEEE Access* **9**, 51416–51431 (2021).
[58] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger and J. Peters, "Robot learning from randomized simulations: A review," *Front. Robot. AI* 31, (2022).
[59] C.-H. Chen and D. S. Naidu, "*Fusion of Hard and Soft Control Strategies for the Robotic Hand*," (John Wiley & Sons, 2017).

Appendix

A. Dynamics of three linked finger

Dynamics of a finger are obtained as [59]:

$$B(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau + \tau_{ext} \tag{1}$$

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} {\tau_{ext}}^1 \\ {\tau_{ext}}^2 \\ {\tau_{ext}}^3 \end{bmatrix}$$

$$\begin{aligned} B_{11} = {} & 2m_2L_1l_2\sin(\theta_1)\sin(\theta_1+\theta_2) + 2m_2L_1l_2\cos(\theta_1)\cos(\theta_1+\theta_2) \\ & + 2m_3L_1L_2\sin(\theta_1)\sin(\theta_1+\theta_2) + 2m_3L_1L_2\cos(\theta_1)\cos(\theta_1+\theta_2) \\ & + 2m_3L_1l_3\sin(\theta_1)\sin(\theta_1+\theta_2+\theta_3) + 2m_3L_1l_3\cos(\theta_1)\cos(\theta_1+\theta_2+\theta_3) \\ & + 2m_3L_2l_3\sin(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) + 2m_3L_2l_3\cos(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_1{l_1}^2 + m_2{L_1}^2 + m_2{l_2}^2 + m_3{L_1}^2 + m_3{L_2}^2 + m_3{l_3}^2 + I_{zz1} + I_{zz2} + I_{zz3}, \end{aligned} \tag{2}$$

$$\begin{aligned} B_{12} = {} & m_2L_1l_2\sin(\theta_1)\sin(\theta_1+\theta_2) + m_2L_1l_2\cos(\theta_1)\cos(\theta_1+\theta_2) \\ & + 2m_3L_2l_3\sin(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) + 2m_3L_2l_3\cos(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_3L_1L_2\sin(\theta_1)\sin(\theta_1+\theta_2) + m_3L_1L_2\cos(\theta_1)\cos(\theta_1+\theta_2) \\ & + m_3L_1l_3\sin(\theta_1)\sin(\theta_1+\theta_2+\theta_3) \\ & + m_3L_1l_3\cos(\theta_1)\cos(\theta_1+\theta_2+\theta_3) + m_2{l_2}^2 + m_3{L_2}^2 + m_3{l_3}^2 + I_{zz2} + I_{zz3} \end{aligned} \tag{3}$$

$$\begin{aligned} B_{13} = {} & m_3L_1l_3\sin(\theta_1)\sin(\theta_1+\theta_2+\theta_3) + m_3L_1l_3\cos(\theta_1)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_3L_2l_3\sin(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) + m_3L_2l_3\cos(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_3{l_3}^2 + I_{zz3} \end{aligned} \tag{4}$$

$$B_{21} = B_{12} \tag{5}$$

$$\begin{aligned} B_{22} = {} & 2m_3L_2l_3\sin(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) + 2m_3L_2l_3\cos(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_2{l_2}^2 + m_3{L_2}^2 + m_3{l_3}^2 + I_{zz2} + I_{zz3} \end{aligned} \tag{6}$$

$$\begin{aligned} B_{23} = {} & m_3L_2l_3\sin(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) + m_3L_2l_3\cos(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) \\ & + m_3{l_3}^2 + I_{zz3} \end{aligned} \tag{7}$$

$$B_{31} = B_{13}, B_{32} = B_{23} \tag{8}$$

$$B_{33} = m_3 {l_3}^2 + I_{zz3} \tag{9}$$

$$\begin{aligned} G_1 = {} & g(m_1 l_1 \cos(\theta_1) + m_2 L_1 \cos(\theta_1) + m_3 L_1 \cos(\theta_1) + m_1 l_2 \cos(\theta_1 + \theta_2) \\ & + m_3 L_2 \cos(\theta_1 + \theta_2) + m_3 l_3 \cos(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{10}$$

$$G_2 = g(m_2 l_2 \cos(\theta_1 + \theta_2) + m_3 L_2 \cos(\theta_1 + \theta_2) + m_3 l_3 \cos(\theta_1 + \theta_2 + \theta_3)) \tag{11}$$

$$G_3 = g(m_3 l_3 \cos(\theta_1 + \theta_2 + \theta_3)) \tag{12}$$

$$\begin{aligned} C_1 = {} & (2m_2 L_1 l_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - 2m_2 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + 2m_3 L_1 L_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - 2m_3 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + 2m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3)) \\ & \times \left(\frac{\partial \theta_1}{\partial t}\right)\left(\frac{\partial \theta_2}{\partial t}\right) \\ & + 2m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3) \\ & + 2m_3 L_2 l_3 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_2 l_3 \cos(\theta_1 + \theta_2)\sin(\theta_1 + \theta_2 + \theta_3) \\ & \times \left(\frac{\partial \theta_1}{\partial t}\right)\left(\frac{\partial \theta_3}{\partial t}\right) \\ & + 2m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3) \\ & + 2m_3 L_2 l_3 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_1 l_3 \cos(\theta_1 + \theta_2)\sin(\theta_1 + \theta_2 + \theta_3) \\ & \times \left(\frac{\partial \theta_2}{\partial t}\right)\left(\frac{\partial \theta_3}{\partial t}\right) \\ & + (m_2 L_1 l_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - m_2 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + m_3 L_1 L_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - m_3 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3)) \\ & \times \left(\frac{\partial \theta_2}{\partial t}\right)\left(\frac{\partial \theta_2}{\partial t}\right) \\ & + (m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3) \\ & + m_3 L_2 l_3 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2 + \theta_3) - m_3 L_1 l_3 \cos(\theta_1 + \theta_2)\sin(\theta_1 + \theta_2 + \theta_3)) \\ & \times \left(\frac{\partial \theta_3}{\partial t}\right)\left(\frac{\partial \theta_3}{\partial t}\right) \end{aligned} \tag{13}$$

$$\begin{aligned} C_2 = {} & (m_2 L_1 l_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - m_2 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + m_3 L_1 L_2 \sin(\theta_1)\cos(\theta_1 + \theta_2) - m_3 L_1 l_2 \cos(\theta_1)\sin(\theta_1 + \theta_2) \\ & + m_3 L_1 l_3 \sin(\theta_1)\cos(\theta_1 + \theta_2 + \theta_3) - m_3 L_1 l_3 \cos(\theta_1)\sin(\theta_1 + \theta_2 + \theta_3)) \times \\ & \left(\frac{\partial \theta_1}{\partial t}\right)\left(\frac{\partial \theta_2}{\partial t}\right) \\ & + 2m_3 L_2 l_3 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2 + \theta_3) - 2m_3 L_2 l_3 \cos(\theta_1 + \theta_2)\sin(\theta_1 + \theta_2 + \theta_3) \times \\ & \left(\frac{\partial \theta_1}{\partial t}\right)\left(\frac{\partial \theta_3}{\partial t}\right) \end{aligned}$$

$$
\begin{aligned}
&+ 2m_3L_2l_3 \sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) - 2m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3)\times \\
&\left(\frac{\partial\theta_2}{\partial t}\right)\left(\frac{\partial\theta_3}{\partial t}\right) \\
&+ (-m_2L_1l_2\sin(\theta_1)\cos(\theta_1+\theta_2) + m_2L_1l_2\cos(\theta_1)\sin(\theta_1+\theta_2) \\
&- m_3L_1L_2\sin(\theta_1)\cos(\theta_1+\theta_2) + m_3L_1l_2\cos(\theta_1)\sin(\theta_1+\theta_2) \\
&- m_3L_1l_3\sin(\theta_1)\cos(\theta_1+\theta_2+\theta_3) + m_3L_1l_3\cos(\theta_1)\sin(\theta_1+\theta_2+\theta_3))\times \\
&\left(\frac{\partial\theta_1}{\partial t}\right)\left(\frac{\partial\theta_1}{\partial t}\right) \\
&+ (m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) - m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3))\times \\
&\left(\frac{\partial\theta_3}{\partial t}\right)\left(\frac{\partial\theta_3}{\partial t}\right)
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
C_3 = {} & (2m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) - 2m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3)\times \\
&\left(\frac{\partial\theta_1}{\partial t}\right)\left(\frac{\partial\theta_2}{\partial t}\right) \\
&+ (m_3L_1l_3\sin(\theta_1)\cos(\theta_1+\theta_2+\theta_3) - m_3L_1l_3\cos(\theta_1)\sin(\theta_1+\theta_2+\theta_3) \\
&+ m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) - m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3)\times \\
&\left(\frac{\partial\theta_1}{\partial t}\right)\left(\frac{\partial\theta_3}{\partial t}\right) \\
&+ (m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3) - m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3)\times \\
&\left(\frac{\partial\theta_2}{\partial t}\right)\left(\frac{\partial\theta_3}{\partial t}\right) \\
&+ (m_3L_1l_3\cos(\theta_1)\sin(\theta_1+\theta_2+\theta_3) - m_3L_1l_3\sin(\theta_1)\cos(\theta_1+\theta_2+\theta_3) \\
&+ m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) - m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3))\times \\
&\left(\frac{\partial\theta_1}{\partial t}\right)\left(\frac{\partial\theta_1}{\partial t}\right) \\
&+ (m_3L_2l_3\cos(\theta_1+\theta_2)\sin(\theta_1+\theta_2+\theta_3) - m_3L_2l_3\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2+\theta_3))\times \\
&\left(\frac{\partial\theta_2}{\partial t}\right)\left(\frac{\partial\theta_2}{\partial t}\right)
\end{aligned} \tag{15}
$$

Cite this article: H. Basumatary, D. Adhar and S.M Hazarika, "Robustifying a reinforcement learning agent-based bionic reflex controller through an adaptive sliding mode control", Robotica. https://doi.org/10.1017/S0263574724001838