robotics

MDPI

Article

DDPG-Based Adaptive Sliding Mode Control with Extended State Observer for Multibody Robot Systems

Hamza Khan [1,†], Sheraz Ali Khan [2], Min Cheol Lee [1], Usman Ghafoor [1,3], Fouzia Gillani [4,†] and Umer Hameed Shah [5,*]

[1] School of Mechanical Engineering, Pusan National University, Busan 46241, Republic of Korea; hamzakhan.0496@gmail.com (H.K.); mclee@pusan.ac.kr (M.C.L.); usman@pusan.ac.kr (U.G.)
[2] Department of Mechatronics Engineering, University of Engineering and Technology, Peshawar 25000, Pakistan; sherazalik@uetpeshawar.edu.pk
[3] Department of Mechanical Engineering, Institute of Space Technology, Islamabad 44000, Pakistan
[4] Department of Mechanical Engineering & Technology, Government College University, Faisalabad 37000, Pakistan; fouziagillani@gcuf.edu.pk
[5] Department of Mechanical Engineering and Artificial Intelligence Research Center, College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates
* Correspondence: m.shah@ajman.ac.ae
† These authors contributed equally to this work.

Abstract: This research introduces a robust control design for multibody robot systems, incorporating sliding mode control (SMC) for robustness against uncertainties and disturbances. SMC achieves this through directing system states toward a predefined sliding surface for finite-time stability. However, the challenge arises in selecting controller parameters, specifically the switching gain, as it depends on the upper bounds of perturbations, including nonlinearities, uncertainties, and disturbances, impacting the system. Consequently, gain selection becomes challenging when system dynamics are unknown. To address this issue, an extended state observer (ESO) is integrated with SMC, resulting in SMCESO, which treats system dynamics and disturbances as perturbations and estimates them to compensate for their effects on the system response, ensuring robust performance. To further enhance system performance, deep deterministic policy gradient (DDPG) is employed to fine-tune SMCESO, utilizing both actual and estimated states as input states for the DDPG agent and reward selection. This training process enhances both tracking and estimation performance. Furthermore, the proposed method is compared with the optimal-PID, SMC, and H_∞ in the presence of external disturbances and parameter variation. MATLAB/Simulink simulations confirm that overall, the SMCESO provides robust performance, especially with parameter variations, where other controllers struggle to converge the tracking error to zero.

Keywords: multibody dynamics; sliding mode control; extended state observer; DDPG

Citation: Khan, H.; Khan, S.A.; Lee, M.C.; Ghafoor, U.; Gillani, F.; Shah, U.H. DDPG-Based Adaptive Sliding Mode Control with Extended State Observer for Multibody Robot Systems. *Robotics* **2023**, *12*, 161. https://doi.org/10.3390/robotics12060161

Academic Editor: Raffaele Di Gregorio

Received: 23 October 2023
Revised: 21 November 2023
Accepted: 22 November 2023
Published: 26 November 2023

1. Introduction

The expanding capabilities of multibody robot systems in autonomous operation and their versatility in performing a wide range of tasks have gathered significant attention from both researchers and industries, emphasizing the persistent need for precision and reliability in their operations. As a result, multibody robot systems require robust control algorithms. However, controlling multibody robot dynamics can be a challenging task, especially when the robot dynamics are unknown. In this effort, different robust control algorithms have been proposed in which sliding mode control (SMC) has been of great interest due to outstanding robustness against parametric uncertainties and external disturbances [1,2]. Subsequent developments resulted in different types of SMC, including integral SMC (ISMC) [3], super twisting SMC (STSMC) [4], terminal SMC (TSMC) [5], SMC with a nonlinear disturbance observer known as sliding perturbation observer (SMC-SPO) [6], and SMC with extended state observer (SMCESO) [7].

This research is conducted for the robust control of multibody industrial robot systems with aims of enhancing the trajectory tracking results. Therefore, we consider the nonlinear control SMC with ESO (SMCESO) for the robot control. ESO considers the system dynamics and external disturbances as perturbations to the system. Therefore, with ESO, the system is only affected by the perturbation estimation error because of the compensation provided by the ESO. Another advantage of the ESO is that it requires no system dynamics information and only uses partial state feedback (position) for estimating the states and the perturbation. Subsequently, the robustness of SMCESO now depends on the quality of estimation of the ESO, which is dependent on the selection of control parameters. However, tuning the parameters manually becomes a challenging task. Therefore, optimal parameter selection can be achieved through adapting the parameters for different sliding conditions.

Various methods for adaptive SMC have been explored, including model-free adaptation, intelligent adaptation, and observer-based adaptation. A. J. Humaidi et al. introduced particle swarm optimization-based adaptive STSMC [8]. The adaptation is carried out based on the Lyapunov theory to guarantee global stability. Y. Wang and H. Wang introduced model-free adaptive SMC, initially estimating unknown dynamics through the time delay estimation method [9,10]. Nevertheless, this approach exhibited undesirable chattering in the control input during experiments, which is deemed unacceptable in the present research. On the other hand, R-D. Xi et al. presented adaptive SMC with a disturbance observer for robust robot manipulator control [11]. Observer-based adaptive SMC stands out for its ability to ensure robustness through minimizing the impact of lumped disturbances, a feature similarly emphasized by C. Jing et al. [12]. Conclusively, this study states that implementing a disturbance observer can lead to finite-time stability and specific tracking performance quality. Furthermore, H. Zao et al. introduced fuzzy SMC for robot manipulator trajectory tracking [13]. H. Khan et al. proposed extremum seeking (ES)-based adaptive SMCSPO for industrial robots [14]. A unique cost function is used which consists of estimation error, and error dynamics to guarantee accurate states and perturbation estimation. H. Razmi et al. proposed neural network-based adaptive SMC [15], and Z. Chen et al. presented radial basis function neural network (NN)-based adaptive SMC [16], both demonstrating commendable performance. However, it is worth noting that the systems under consideration in these studies were relatively smaller than the industrial robot in our current research. Furthermore, a model-free reinforcement learning algorithm known as deep deterministic policy gradient (DDPG) has been observed to provide optimal SMC parameters, enhancing performance through learning and adapting to different sliding patterns [17–19].

Considering the diverse literature, initially, the model-free extremum seeking algorithm was a consideration. However, in the current study, the need to tune multiple (four different) parameters simultaneously led to the exploration of learning-based algorithms such as NN and DDPG for adapting controller parameters. Notably, NN is well suited for simpler systems, while DDPG is preferred for complex, high-dimensional systems with unknown dynamics. DDPG is a model-free, online, and off-policy reinforcement learning algorithm. It employs an actor–critic architecture, where the actor focuses on learning the optimal policy, while the critic approximates the Q-function [20]. The Q-function is responsible for estimating the expected cumulative long-term rewards for state–action pairs. The critic achieves this by minimizing the temporal difference error, which represents the disparity between the predicted Q-value and the actual Q-value derived from environmental feedback. This process equips the critic to evaluate action quality in various states, guiding the actor in selecting actions that maximize expected rewards. Ensuring the convergence of the temporal difference error is a pivotal aspect of effective DDPG agent training.

The primary contribution of this study is the optimal tuning of SMCESO using the DDPG algorithm for a heavy-duty industrial robot manipulator with six degrees of freedom (DOF). Robust performance can be achieved through minimizing estimation errors, ensuring accurate perturbation estimation and compensation. To accomplish this, the

DDPG input states incorporate tracking error, estimation error, current joint angle, and estimated joint angle. A reward has been designed, integrating an overall error tolerance of 0.01 rad for both tracking and estimation errors, yielding positive rewards if error is below the threshold. Conversely, if errors exceed this threshold, negative rewards are assigned. Through this approach, the DDPG agent learns a control pattern based on actual and estimated results, ultimately achieving optimal estimation and robust control performance. The proposed algorithm was implemented and compared with optimal proportional—integral–derivative (PID) control and SMC, and H_∞ control in an extensive MATLAB/Simulink simulation environment. The results demonstrated that SMCESO outperforms all the three controllers, particularly in the presence of variable system parameters, as it effectively reduces the effect of the actual perturbations on system performance.

The reminder manuscript Is organized as follows: Section 2 describes the general multibody dynamics and formulates the SMC. Section 3 presents the ESO and the DDPG algorithm. Section 4 then presents the simulation environment and the results of the proposed algorithm, whereas Section 5 provides the conclusions.

2. Preliminaries

2.1. Multibody Dynamics Description

Consider the second-order multibody dynamics [14] as follows:

$$\ddot{x}_j = f_j(x) + \Delta f_j(x) + \sum_{i=1}^{n} \left[(b_{ji}(x) + \Delta b_{ji}(x)) u_i \right] + d_j(t) \; j = 1, \ldots, n \tag{1}$$

where $x \triangleq [x_1 \ldots \; x_n]^T$ are the state vectors representing the position, and $f_j(x)$ and $\Delta f_j(x)$ are the linear dynamic and dynamics uncertainties, respectively. Similarly, the control gain matrix and their uncertainties are represented by $b_{ji}(x)$ and $\Delta b_{ji}(x)$, respectively. u_i and d_j are the control input and external disturbance, respectively. Combining the system nonlinearities, dynamics uncertainties, and disturbances as the perturbation (ψ) can be written as

$$\psi_j(x,t) = \Delta f_j(x) + \sum_{i=1}^{n} \left[\Delta b_{ji}(x) u_i \right] + d_j(t) \tag{2}$$

whereas it is assumed that the perturbation is bounded by an unknown continuous function, i.e., $\left|\psi_j(x,t)\right| \leq \Gamma > 0$, and, in addition, that it is smooth with the bounded derivative $\left|\dot{\psi}_j(x,t)\right| \leq \overline{\Gamma}$.

2.2. Sliding Mode Control

The main concept of SMC is to design a sliding surface σ in the state space (position x_1, and velocity x_2) [21], which is given as

$$\sigma = \dot{e} + ce \tag{3}$$

where $e = x_d - x$ is the tracking error, and $c > 0$ is a positive constant. Now, in order to drive the system dynamics, the state variable should converge to zero: i.e., $\lim_{t \to \infty} \dot{e}, e = 0$ asymptotically in the presence of perturbation. Therefore, SMC tends to bring system states on the sliding surface by means of control force u. Subsequently, SMC has two phases: The first is the reaching phase, during which the system states are not on the sliding surface and require a switching control u_{sw} to reach the sliding surface. The second phase is the sliding phase, in which the system states have reached the sliding surface and now require continuous control, generally known as equivalent control u_{eq}, to remain on the sliding surface, where the overall control input becomes $u = u_{eq} + u_{sw}$. To compute the control input, the derivative of the sliding surface is defined as follows:

$$\dot{\sigma} = \ddot{e} + c\dot{e} = -K_{smc} \cdot \text{sat}(\sigma) \tag{4}$$

where K_{smc} represents the switching control gain, and 'sat' is the saturation function with a boundary layer thickness ε_c, given as

$$\text{sat}(\sigma) = \begin{cases} \frac{\sigma}{|\sigma|} \; if \; |\sigma| > \varepsilon_c \\ \frac{\sigma}{\varepsilon_c} \; if \; |\sigma| \leq \varepsilon_c \end{cases} \tag{5}$$

Assuming unknown system dynamics, $\ddot{x} = u$ is presumed. Substituting this condition with the dynamics error $\ddot{e} = \ddot{x}_d - \ddot{x}$ in (4) results in the following control input.

$$u = -K_{smc} \cdot \text{sat}(\sigma) + \ddot{x}_d - c\dot{e} \tag{6}$$

Here, $K_{smc} \cdot \text{sat}(\sigma)$ is denotes the switching control (u_{sw}), and the negative sign embodies the error convention. The remaining terms are considered equivalent control (u_{eq}). Subsequently, taking the derivative of the sliding surface, with the system disturbed by perturbation (such as (10) in the subsequent section) yields (7):

$$\dot{\sigma} = \ddot{x} - \ddot{x}_d + c\dot{e} = u + \psi(x,t) - \ddot{x}_d + c\dot{e} \tag{7}$$

Substituting the control law from (6) and solving results in (8):

$$\dot{\sigma} = -K_{smc} \cdot \text{sat}(\sigma) + \psi(x,t) \tag{8}$$

Equation (8) shows that in SMC, the sliding surface is affected by the perturbation. Once the system states have reached the sliding phase, the relationship between the sliding surface and the perturbation is given as the following transfer function [6].

$$\frac{\sigma}{\psi(x,t)} = \frac{1}{p + \frac{K_{smc}}{\varepsilon_c}} \tag{9}$$

where p is the s-domain variable. Increasing the boundary layer will decrease the breaking frequency, making the system less sensitive to the higher frequency perturbations. However, at $\sigma \approx 0$, increasing the boundary layer thickness reduces controller performance, leading to higher tracking error. If the sliding surface is tightly bounded, with a very small boundary layer, chattering occurs.

2.3. Problem Forumlation

Calculating the dynamics of a multibody robot system is a challenging task, further compounded by the presence of inaccurate dynamics, which introduce uncertainties. Therefore, for the later study, considering the complete dynamic model as perturbation and $b = 1$, the resulting dynamics is as follows.

$$\ddot{x} = \psi(x,t) + u \tag{10}$$

Subsequently, to ensure the sliding condition outside the boundary later, the sliding dynamics can be written as

$$\dot{\sigma} = c\dot{e} + \psi(x,t) + u, \; \sigma(0) = \sigma_o \tag{11}$$

For the asymptotic stability of (11) about the equilibrium point, $\dot{V} < 0$ for $\sigma \neq 0$ must be satisfied [22]. The derivative of V is computed as

$$\dot{V} \leq \sigma\dot{\sigma} = \sigma\left[c\dot{e} + \psi(x,t) + u\right] \tag{12}$$

Taking $\eta = c\dot{e} + u$ in (12) will result in

$$\dot{V} \leq \sigma\dot{\sigma} = \sigma[\psi(x,t) + \eta] = \sigma \cdot \psi(x,t) + \sigma \cdot \eta \tag{13}$$

$$\dot{V} \leq |\sigma|\,\Gamma + \sigma \cdot \eta \tag{14}$$

Selecting $\eta = -K_{smc} \cdot \text{sat}(\sigma)$, and with $K_{smc} > 0$, (14) becomes

$$\dot{V} \leq |\sigma|\,\Gamma - |\sigma| \cdot K_{smc} = -|\sigma|(K_{smc} - \Gamma) \tag{15}$$

Consequently, the overall control input becomes

$$u = K_{smc} \cdot \text{sat}(\sigma) + c\dot{e} \tag{16}$$

Equation (15) further emphasizes that for stability, $K_{smc} > \Gamma$ to satisfy the Lyapunov condition. However, obtaining information about Γ can be a complex and tedious task.

3. Proposed Algorithm

There are two concerns: First, based on (9), as the perturbation affects the sliding dynamics, the correct dynamics are unknown. Therefore, a perturbation observer has been used to estimate and compensate the actual perturbation effects. For this purpose, an extended state observer (ESO) has been implemented, which offers the advantage of not requiring the system dynamics information. Secondly, we optimally tune the control gain for SMC and ESO to stabilize the system in finite time, ensuring that both tracking and estimation error converge to zero. Subsequently, deep deterministic policy gradient (DDPG) has been employed for control gain tuning.

3.1. Extended State Observer

ESO provides real-time estimations of unmeasured system states and perturbations, which is the combination of modelled and unmodelled dynamics and external disturbances, enhancing control system performance and robustness. This means ESO considers the system's linear and nonlinear dynamics as the perturbation and estimates them [23]. Consequently, only the control input u in (1) is known. Furthermore, ESO does not require system dynamics information and uses only partial state feedback (position) for estimation. In addition to the system states (position x_1 and velocity x_2), an extended state x_3 is introduced, such as

$$x_3 = \psi(x,t) = f(x) + \Delta f(x) + \sum_{i=1}^{n} [\Delta b_i(x)u] + d(t) \leq \Gamma \tag{17}$$

Subsequently, the system dynamics in (1) can be simplified as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 + u \\ x_3 &= \Gamma \end{aligned} \tag{18}$$

With the new system information, the mathematical model of nonlinear ESO [7] is then written as

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + l_1 \cdot \rho\left(\tilde{x}_1\right) \\ \dot{\hat{x}}_2 &= \hat{x}_3 + u + l_2 \cdot \rho\left(\tilde{x}_1\right) \\ \dot{\hat{x}}_3 &= l_3 \cdot \rho\left(\tilde{x}_1\right) \end{aligned} \tag{19}$$

where the components with " $\wedge$ ", and " $\sim$ " represent the estimated states and the error between the actual and the estimated value, e.g., $\tilde{x}_1 = x_1 - \hat{x}_1$. ρ is the saturation function, which is selected as

$$\rho\left(\tilde{x}_1\right) = \begin{cases} \tilde{x}_1 / \left|\tilde{x}_1\right| & \text{if} \quad \left|\tilde{x}_1\right| > \varepsilon_o \\ \tilde{x}_1 / \varepsilon_o & \text{if} \quad \left|\tilde{x}_1\right| \leq \varepsilon_o \end{cases} \tag{20}$$

ε_o is the boundary layer of the ESO such that the estimation error should be $\left|\widetilde{x}_1\right| \leq \varepsilon_o$. The estimation errors are calculated as

$$\begin{aligned} \dot{\widetilde{x}}_1 &= \widetilde{x}_2 - l_1 \cdot \rho\left(\widetilde{x}_1\right) \\ \dot{\widetilde{x}}_2 &= \widetilde{x}_3 - l_2 \cdot \rho\left(\widetilde{x}_1\right) \\ \dot{\widetilde{x}}_3 &= \Gamma - l_3 \cdot \rho\left(\widetilde{x}_1\right) \end{aligned} \tag{21}$$

As the estimated error should be bounded by a boundary later, therefore, (21) can be rewritten as follows.

$$\begin{aligned} \dot{\widetilde{x}}_1 &= \widetilde{x}_2 - l_1 \cdot \widetilde{x}_1/\varepsilon_o \\ \dot{\widetilde{x}}_2 &= \widetilde{x}_3 - l_2 \cdot \widetilde{x}_1/\varepsilon_o \\ \dot{\widetilde{x}}_3 &= \Gamma - l_3 \cdot \widetilde{x}_1/\varepsilon_o \end{aligned} \tag{22}$$

Subsequently, the state space of the error dynamics can be written as

$$\dot{\widetilde{x}} = A\widetilde{x} + B\Gamma \tag{23}$$

where

$$A = \begin{bmatrix} -l_1/\varepsilon_o & 1 & 0 \\ -l_2/\varepsilon_o & 0 & 1 \\ -l_3/\varepsilon_o & 0 & 0 \end{bmatrix}, \text{ and } E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{24}$$

The characteristic equation of A can be calculated as follows:

$$|\lambda I - A| = \begin{vmatrix} \lambda + l_1/\varepsilon_o & -1 & 0 \\ l_2/\varepsilon_o & \lambda & -1 \\ l_3/\varepsilon_o & 0 & \lambda \end{vmatrix} = \lambda^3 + (l_1/\varepsilon_o)\lambda^2 + (l_2/\varepsilon_o)\lambda + l_3/\varepsilon_o \tag{25}$$

The error dynamics are stable if the gains l_1, l_2, and l_3 are positive. Therefore, these gains are selected using the pole placement method as follows:

$$(s + \lambda)^3 = s^3 + 3 \cdot s \cdot \lambda^2 + 3 \cdot \lambda \cdot s^2 + \lambda^3 \tag{26}$$

Comparing the coefficients of (25) and (26) results in the following selection of gains:

$$l_1 = 3 \cdot \lambda \cdot \varepsilon_o, \; l_2 = 3 \cdot \lambda^2 \cdot \varepsilon_o, \text{ and } l_3 = \lambda^3 \cdot \varepsilon_o \tag{27}$$

3.2. Extended State Observer-Based Sliding Mode Control (SMCESO)

For enhanced system performance, the final control input u_o for the system with estimated perturbation $\hat{\psi}(x,t)$ from ESO and switching control from SMC can be written as

$$u_o = u - \hat{x}_3 = u - \hat{\psi}(x,t) \tag{28}$$

where u is from (16). Consequently, the system dynamics from (10) can be rewritten as follows.

$$\ddot{x} = u_o - \hat{\psi}(x,t) + \psi(x,t) = u_o + \widetilde{\psi}(x,t) \tag{29}$$

where $\widetilde{\psi}(x,t) = \psi(x,t) - \hat{\psi}(x,t)$ is the perturbation estimation error. Now, it is evident that with ESO, the system is only affected by the perturbation estimation error as compared to the actual perturbation. This follows $\left|\widetilde{\psi}(x,t)\right| \ll |\psi(x,t)|$, ensuring that ESO-based SMC is more stable than the individual SMC. Subsequently, the Lyapunov function in (15) will become

$$\dot{V} \leq -|\sigma|\left(K'_{smc} - \widetilde{\psi}(x,t)\right) \tag{30}$$

The stability of SMCESO with the Lyapunov function $\sigma\dot{\sigma} \leq 0$ can be calculated as

$$\sigma\dot{\sigma} \leq |\sigma|\left(\ddot{e} + c\dot{e}\right) \leq |\sigma|\left(\ddot{x} - \ddot{x}_d + c\dot{e}\right) \leq 0 \tag{31}$$

With the system dynamics and combined control input from (6) and (28), according to (7), this will result in the following condition:

$$\sigma\dot{\sigma} \leq |\sigma|\left(-K'_{smc}\cdot\text{sat}(\sigma) + \ddot{x}_d - c\dot{e} - \hat{\psi}(x,t) + \psi(x,t) - \ddot{x}_d + c\dot{e}\right) \leq 0 \tag{32}$$

Simplifying (32) yields (33):

$$\sigma\dot{\sigma} \leq |\sigma|\left(-K'_{smc}\cdot\text{sat}(\sigma) + \widetilde{\psi}(x,t)\right) \leq 0 \tag{33}$$

Subsequently, to keep the system stable, the control gain should follow the following condition.

$$K'_{smc} > \left|\widetilde{\psi}(x,t)\right| \tag{34}$$

Now, the new control gain K'_{smc} is small in comparison to conventional gain K_{smc}, with $K'_{smc} < K_{smc}$. The reduced gain will result in smoother switching control, eliminating any chattering for improved performance. Furthermore, the control parameters K'_{smc}, c, ε_c, and λ are then optimally tuned using DDPG to reduce manual tuning efforts.

3.3. DDPG-Based SMCESO

Deep deterministic policy gradient (DDPG) is a reinforcement learning algorithm designed for solving continuous action space problems. It combines elements of deep neural networks and the deterministic policy gradient theorem to achieve remarkable performance in control tasks. DDPG employs an actor–critic architecture, with the actor network modeling the policy and the critic network estimating the state–action value function. A key innovation in DDPG is the use of target networks to stabilize training, with periodic updates to slowly track the learned networks. This approach, coupled with experience replay, enables stable and efficient learning, making DDPG a prominent choice for complex, high-dimensional control problems.

Similar to other reinforcement learning algorithms, the DDPG algorithm operates within the framework of a Markov decision process (MDP) [24], denoted by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ represent the environment's state space and the agent's action space, respectively. $\mathcal{P}$ signifies the probability of state transitions. During agent training, the reward function $\mathcal{R}$ serves as the training target. In core, while training the agent, the system's state $s \,\epsilon\, \mathcal{S}$ is observed, and the associated reward $r \,\epsilon \mathcal{R}$ is acquired. Subsequently, the optimal policy $\pi^a(a|s)$ is determined through maximizing the state–action value function.

$$Q(s,a) = E[R_c|S_t = s,\ A_t = a] \tag{35}$$

where R_c represents the cumulative reward, and $R_c = \sum_{k=0}^{\infty}\gamma^k r_{k+1}$ with $0 \leq \gamma \leq 1$ is the discount factor that reflects the importance of the reward value at future moments. To enhance the controller performance, the DDPG has to study the regulation strategy μ (actor network) and calculate the probability of each action. Consequently, the controller parameters are updated in real time to maximize the total reward [25,26].

$$\begin{cases} \max\limits_{\mu}\left[\sum_{k=0}^{\infty}\gamma^k r_k(x_1(k), \hat{x}_1(k))\right] \\ st:\ \theta(k) = \mu(K'_{smc},\ c, \varepsilon_c, \lambda) \\ \quad\ \theta_{\min} \leq \theta(k) \leq \theta_{\max} \end{cases} \tag{36}$$

$\theta(k)$ is the set of action parameters, with minimum limit $\theta_{\min}$ and maximum limit $\theta_{\max}$. The structure of DDPG is presented in Figure 1. The selection of a suitable state space is crucial for ensuring the convergence of reinforcement learning. In the context

of the present challenges, the chosen state space should inherently pertain to the robot's position and its estimated dynamics. As a result, for the sake of computational efficiency and enhanced learning, the state space is straightforwardly defined as $S = [x(k), \hat{x}(k)]$, and the state vector is defined as $s_k = \left[x_1,\ \hat{x}_1, e, \widetilde{x}_1\right]$.

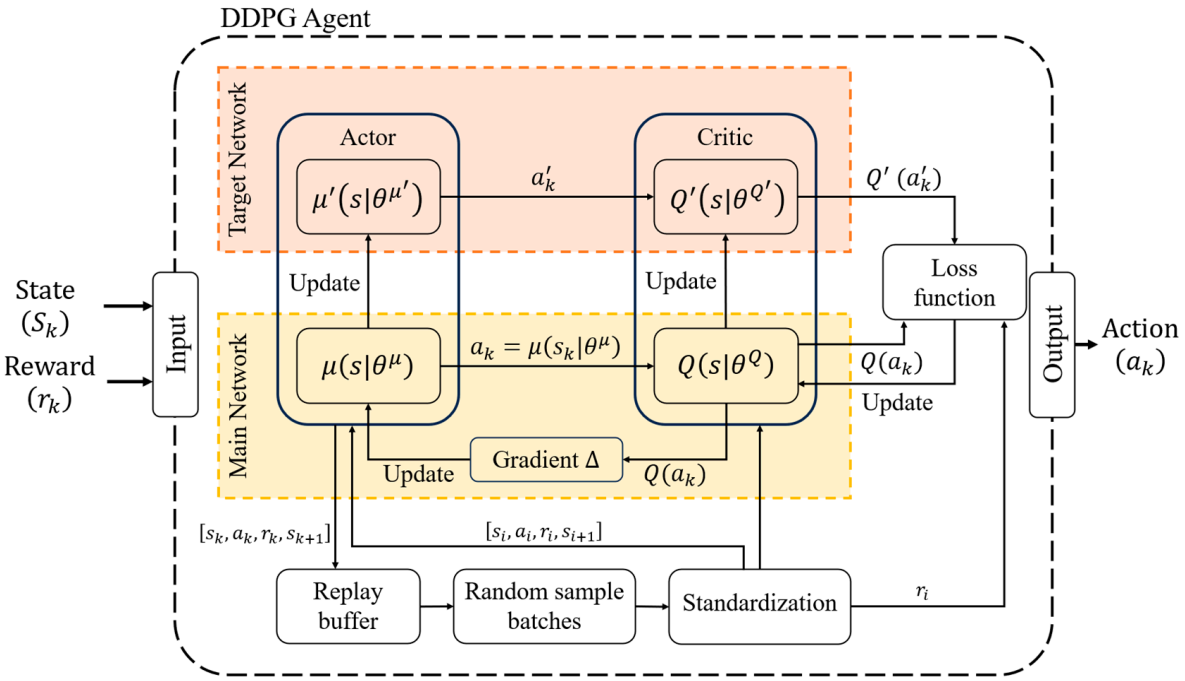

Figure 1. Structure of DDPG.

The actor–critic value network for the robot system is established, which is a double-layer structure including the target network and main network. The replay buffer stores data in the form of $[s_k, a_k, r_k, s_{k+1}]$, which is used for network training. Both the main networks and target networks share the same structure but differ in their parameters. The actor network is denoted by $a_k = \mu(s_k|\theta^{\mu})$, with θ^{μ} as the network parameter. The critic network is denoted as $Q(s_k,\ a_k|\theta^{Q})$, with the network parameter as θ^{Q}. When training, small batches of sample information $[s_i, a_i, r_i, s_{i+1}]$ are randomly selected from the replay buffer for learning. In brief, the training process involves the four networks to ensure that actions generated by the actor network can be used as input for the critic network to maximize the state–action value function in (35). The training process is provided in Algorithm 1.

Algorithm 1: Training DDPG Agent

Initialize the networks $\mu(s_k|\theta^{\mu})$, and $Q(s_k,\ a_k|\theta^{Q})$ randomly.
Initialize the target network $\mu'\left(s_k\middle|\theta^{\mu'}\right)$, and $Q'\left(s_k,\ a_k\middle|\theta^{Q'}\right)$ with weights.
Initialize the replay buffer.
While $ep\ \leq ep_{max}$
Randomly initialize the process $\mathcal{N}$ for action exploration.
Receive the states s_k
while $k < k_{max}$
$a_k = \mu(s_k|\theta^{\mu}) + \mathcal{N}$.
Execute the environment to update the reward r_k, and s_{k+1}.
Store $[s_k, a_k, r_k, s_{k+1}]$ in replay buffer R.
Sample a random minibatch of m transitions transitions $[s_i, a_i, r_i, s_{i+1}]$ from R.
Set target $y_i = r_i + \gamma \cdot Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{\mu'}\right)$.
Update the critic by minimizing the loss function $J = \frac{1}{m}\sum\left(y_i - Q(s_i, a_i|\theta^{Q})\right)^2$.
Update the actor using the sampled policy gradient
$\nabla_{\theta^{\mu}} J \approx \frac{1}{m}\sum \nabla_a Q(s, a|\theta^{Q})|_{s=s_i,\ a=\mu_s} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})$.
Update the target network with soft update
$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'},\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$.
If isdone == 1
Reset.
End if
end while k
if $r_{average} \geq r_{stopping}$
Stop training.
End
end

The DDPG-based SMCESO block diagram is presented in Figure 2. For robust performance, the tracking error should be eliminated. Subsequently, the estimation should be accurate, i.e., $\widetilde{x} \to 0$. Consequently, the true perturbation will be estimated and well compensated. Therefore, the reward function for the current study is designed as follows.

$$\begin{aligned} r &= R_1 + R_2 + R_3 \\ R_1 &= \begin{cases} -1 & e \geq e_{tol} \\ 5 & e < e_{tol} \end{cases} \\ R_2 &= \begin{cases} -1 & \widetilde{x}_1 \geq \widetilde{x}_{1,tol} \\ 5 & \widetilde{x}_1 < \widetilde{x}_{1,tol} \end{cases} \\ R_3 &= \begin{cases} -100 & x_1 \geq x_{1,stop} \\ 0 & x_1 < x_{1,stop} \end{cases} \end{aligned} \tag{37}$$

where e_{tol} is the error tolerance for accepting good performance of tracking control. Similarly, R_2 is for the good performance of ESO with estimation error tolerance as $\widetilde{x}_{1,tol}$. R_3 is for the stopping condition (isdone in Algorithm 1), meaning the robot is not stable exceeding the movement limits $x_{1,stop}$.

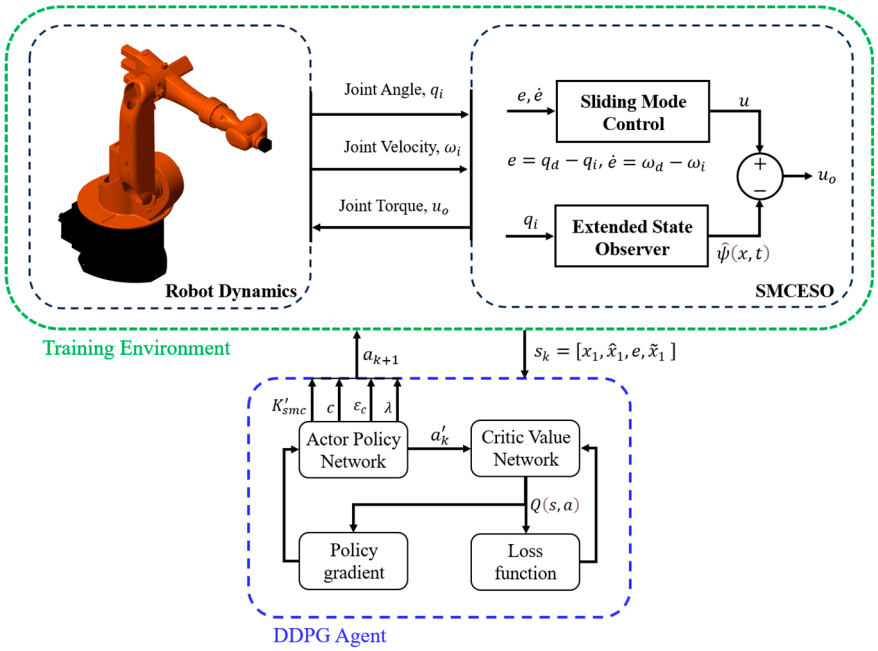

Figure 2. Block diagram of DDPG-based SMCESO.

4. Simulations and Discussion

This section provides details about the simulation system and the environment. It also includes the presentation of results and their subsequent discussion.

4.1. System and Environment Descrption

For the DDPG implementation, a simulation environment is created in MATLAB/Simulink, featuring an object pick-and-place task using the Simscape Multibody model of the KUKA KR 16 S industrial robot arm, as presented in Figure 2. The KR 16 is a six-degrees-of-freedom (DOF) high-speed, heavy-duty industrial robot arm with a substantial payload capacity. Demonstrating robust performance with such robot will validate the efficiency of the proposed method. Consequently, the robot must exhibit robust performance and a minimal tracking error in the presence of nonlinear dynamics. The sampling time for the DDPG algorithm is set to 0.5 s, while the control algorithm operates with a sampling time of 5 ms. The computations are carried out on a computer equipped with an Intel i7 processor and an RTX 3090 ti GPU.

4.2. Simulations

Simulations are conducted in two phases. First is the implementation of the proposed algorithm on a simple linear system to explain the basics or the workings of the ESO. Second is the implementation on the multibody dynamics of the robot arm, with a sine wave as the desired position. For simulation, the DDPG hyperparameters are presented in Table 1

4.2.1. Simple System Implementation

For a simple linear system, consider the following second-order dynamics.

$$\ddot{x} = u_o - b\dot{x} - kx + d(t),\ d(t) = a \cdot \sin(t) \tag{38}$$

$$\psi(x,t) = -10\dot{x} - 50x + d(t) \tag{39}$$

where a is the magnitude of disturbance ($d(t)$), $b = 10$ is the damping coefficient, and $k = 50$ is the stiffness. The performance of DDPG-based SMCESO has been compared with SMC, proportional–integral–derivative (PID) control optimally tuned using the Control

System Tuner toolbox in Simulink, and H_∞ control [27]. The control gains are provided in Table 2, and the trajectory tracking error is shown in Figure 3.

Table 1. DDPG parameters.

Reinforcement	Parameters	
	Parameter	**Value**
Critic	Learn rate	1×10^{-3}
	Gradient Threshold	1
Actor	Learn rate	1×10^{-4}
	Gradient threshold	1
Agent	Sample time	0.5
	Target smooth factor	1×10^{-3}
	Discount factor	1
	Minibatch size	64
	Experience buffer length	1×10^{6}
	Noise variance	0.3
	Noise variance decay rate	1×10^{-5}
Training	Maximum episode	2000
	Maximum steps	20
	Average reward window length	10

Table 2. Control gains.

Control Algorithm	Gains
PID	$K_p = 200$, $K_i = 1000$, and $K_d = 20$.
SMC	$K_{smc} = 300$, $c = 35$, and $\epsilon_c = 0.5$.
SMCESO	$K'_{smc} = 50$, $c = 30$, $\epsilon_c = 0.5$, $\lambda = 137.31$, and $\epsilon_o = 1$.
H_∞	Sensitivity Function $W_s = {}^{s+40}/_{4s+0.36}$

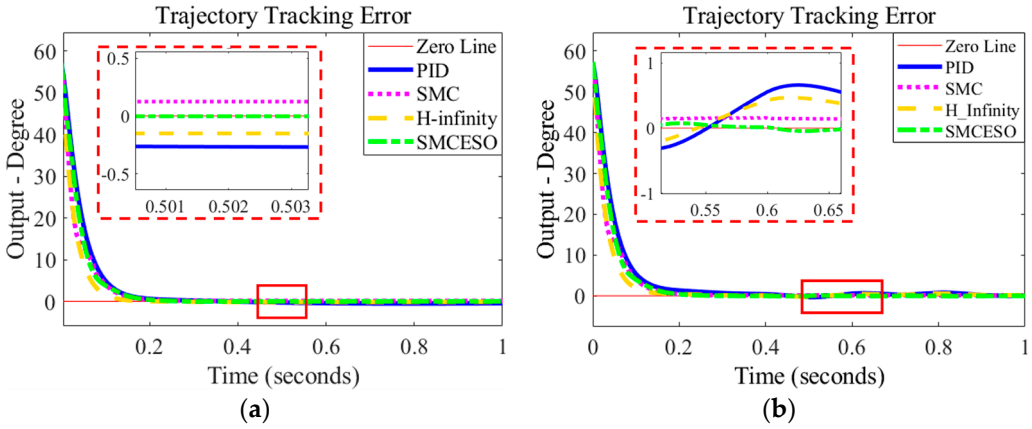

Figure 3. Controller performance evaluation: (**a**) with disturbance; (**b**) with parameter variation.

The error results of the step response in Figure 3a reveal that when a disturbance (a = 10) is present, all three controllers except for SMCESO demonstrate good performance with high control gains but fail to fully converge the error to zero. In contrast, SMCESO effectively estimates and compensates for the perturbation, as depicted in Figure 4 (on the next page), leading to error convergence toward zero. Moreover, as anticipated in Section 3.2, the new control gain K'_{smc} is notably smaller than the conventional gain K_{smc} (in Table 1), which is tuned using the DDPG algorithm. Additionally, the algorithms underwent testing with parameter changes, where the stiffness was chosen as $k = 50 \pm 8$. These variations were introduced using the Simulink random number block with a variance of 20. The tracking errors for variable stiffness are presented in Figure 3b, illustrating that

PID exhibits the maximum deviation, while H_∞ outperforms PID. However, SMC now surpasses H_∞ due to a model mismatch between the actual system and the dynamics used for controller synthesis. Finally, SMCESO outperforms all three controllers through maintaining the error very close to zero. This validates that SMCESO effectively estimates system uncertainties and compensates for their effects on the system response, resulting in robust performance.

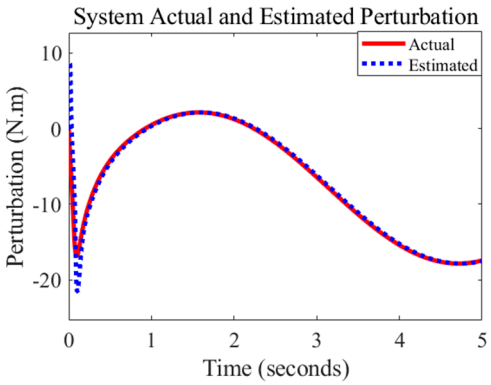

Figure 4. Actual and estimated perturbation comparison.

4.2.2. Adaptive SMCESO with Multibody Robot

With a multibody robot system, the DDPG agent has been trained to fine-tune the controller parameters. For controller evaluation, Joint 2 (q_2) of the robot manipulator has been considered as it holds the maximum weight of the robot against gravity. Therefore, the robot arm is fully extended, and only q_2 is moving. The desired trajectory is defined as $q_{2,d} = \sin(w{\cdot}t)$, with initial frequency $w_o = 1$, which resets after every episode as $w = 1 + rand[-0.5, 0.5]$. Furthermore, the total simulation time is 10seconds, with an ideal reward $r_{max} = 210$. The training stops when the average award reaches $r_c \geq 199$, considering the average reward window length. The DDPG agent took 343 episodes for the training. The episode reward and the cumulative reward are presented in the following Figure 5, and subsequently, the tuned parameters are shown in Figure 6 and the trajectory tracking error and joint torques are in Figure 7.

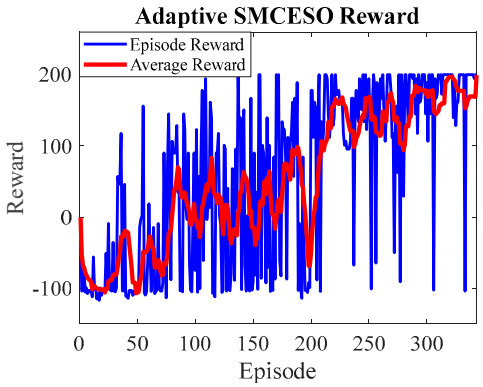

Figure 5. SMCESO training reward.

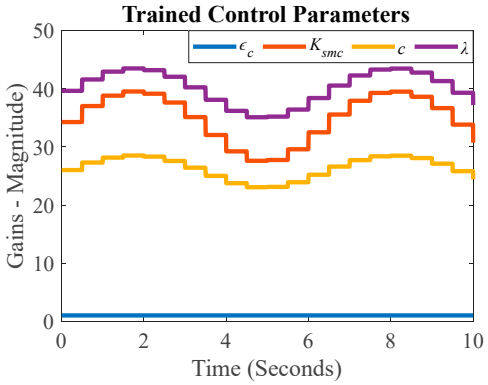

Figure 6. SMCESO fine-tuned gains.

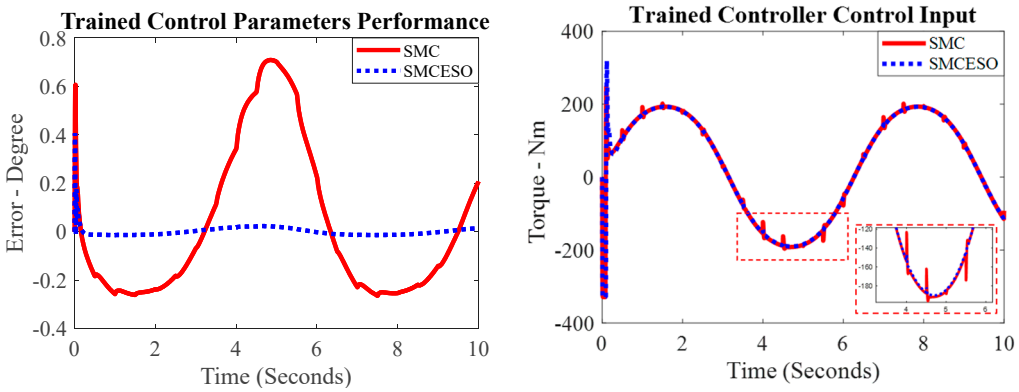

Figure 7. Fine-tuned controller tracking performance.

The joints were equipped with electromechanical motor dynamics with the motor parameters given in Table 3. Consequently, both control algorithms (SMC and SMCESO) can achieve joint tracking errors with the range ± 1 *degree*. However, it is evident from the control input that SMC has sudden spikes throughout the simulation. Reducing gains can eliminate these spikes but will reduce the control performance, resulting in larger errors. Similarly, to reduce the error of SMC, higher gains (more than double those of SMCESO) are required. This, in turn, increases the spikes and occasionally introduces chattering in the response. In contrast, SMCESO shows very smooth performance and keeps the error within the range of ± 0.1 *degree*. This validates the robustness of SMC integrated with ESO, which overcomes the perturbation effects of the system with a total mass $m > 55$ Kg on joint 2. Overall, the initial jump in the control input is primarily attributed to motor dynamics such as friction, which stabilizes once the robot starts moving. Moreover, for a deeper understanding of achieving robust performance, observing the estimated states in Figure 8.

Table 3. Motor dynamics parameters.

Parameter	Value
Inductance, L	0.573×10^{-3} H
Resistance, R	0.978 Ω
Torque constant, k_t	33.5×10^{-3} N·m/A
Voltage constant, k_e	33.5×10^{-3} V·s/rad
Gear Ratio	100

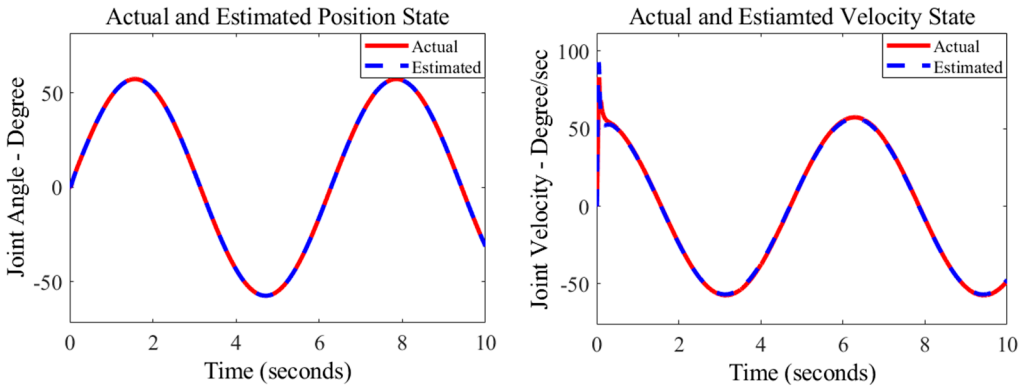

Figure 8. Actual and estimated states of the system.

The position and velocity results show that the state observer is performing very well, with estimations showing nearly zero error. This suggests that the system may have highly effective perturbation estimation and compensation capabilities to enhance tracking performance. Moreover, the Simscape multibody toolbox allows obtaining the dynamics components of the robot system, including the mass matrix $M(q)$, velocity product torque $C(q,\dot{q})\cdot\dot{q}$ with $C(q,\dot{q})$ Coriolis terms, and gravitational torque $G(q)$. This can be achieved through first creating the rigid body tree and then utilizing the Manipulator Algorithm library from Robotics System Toolbox. Subsequently, similar to (10), the expected perturbation is presumed as

$$\psi(x,t) = C(q,\dot{q})\cdot\dot{q} + G(q) \tag{40}$$

The assumed and estimated perturbations are presented in Figure 9, below.

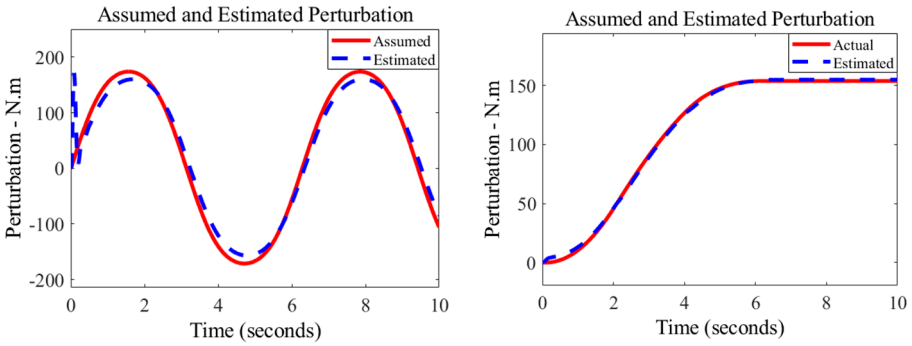

Figure 9. Perturbation results.

The estimated perturbation closely aligns with the assumed perturbation. With the desired trajectory being a sine wave, the velocity is continuously changing, leading to some perturbation estimation error, as expected due to the motor dynamics, which are not factored into the perturbation calculation. However, this error can be compensated by the SMC in Equation (16), further validating the theory in Equation (29) that, with ESO, the system dynamics are primarily influenced by the perturbation estimation error. From a magnitude perspective, it is evident that the perturbation estimation error is considerably smaller than the actual perturbation, making the system achieve robust performance. Furthermore, when the robot comes to a stop, the estimated perturbation converges to match the assumed perturbation, confirming the accurate working of the ESO.

5. Conclusions

In this study, an approach to control and stabilize multibody robotic systems with inherent dynamics and uncertainties is presented. The approach leverages extended

state observer (ESO) and sliding mode control (SMC) (SMCESO), combined with the optimization capabilities of deep deterministic policy gradients (DDPGs). One of the advantages of ESO is that it requires only partial state feedback (position) to estimate the perturbation, which includes the system dynamics and external disturbances. Initially, the proposed algorithm is implemented on a simple second-order system with introduced sinusoidal disturbance. Subsequently, the control parameters were fine-tuned using a DDPG agent, which was trained based on system tracking error, joint angle, estimated joint angle, and estimation error. This training allowed the DDPG-based SMCESO to outperform the optimally tuned PID control (via a control tuner toolbox), conventional SMC (tuned through DDPG), and H_∞ control in terms of robustness, significantly enhancing system stability and performance. Even in the presence of disturbances, the SMCESO consistently converges to zero error due to its perturbation rejection capabilities. It was also demonstrated that with ESO, the system dynamics are primarily affected by the perturbation estimation error, which was validated through simulations showing close alignment between estimated and actual perturbations, leaving only minor estimation errors to be handled by the SMC control input. As a result, the multibody robot system's overall performance is highly robust.

Author Contributions: Conceptualization, H.K. and M.C.L.; Data curation, S.A.K.; Formal analysis, F.G.; Funding acquisition, U.H.S.; Investigation, S.A.K. and F.G.; Methodology, H.K.; Project administration, M.C.L.; Resources, M.C.L.; Software, H.K.; Validation, U.G. and U.H.S.; Writing—original draft, H.K.; Writing—review and editing, U.G. and U.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This work was supported by the Deanship of Graduate Studies and Research (DGSR) Program, Ajman University, United Arab Emirates.

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

References

1. Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. Introduction: Intuitive Theory of Sliding Mode Control. In *Sliding Mode Control and Observation*; Control Engineering; Birkhäuser: New York, NY, USA, 2014; pp. 1–42.
2. Afifa, R.; Ali, S.; Pervaiz, M.; Iqbal, J. Adaptive Backstepping Integral Sliding Mode Control of a MIMO Separately Excited Dc Motor. *Robotics* **2023**, *12*, 105. [CrossRef]
3. Khan, H.; Abbasi, S.J.; Lee, M.C. DPSO and Inverse Jacobian-based Real-time Inverse Kinematics with Trajectory Tracking using Integral SMC for Teleoperation. *IEEE Access* **2020**, *8*, 159622–159638. [CrossRef]
4. Hollweg, G.V.; de Oliveira Evald, P.J.; Milbradt, D.M.; Tambara, R.V.; Gründling, H.A. Design of continuous-time model reference adaptive and super-twisting sliding mode controller. *Math. Comput. Simul.* **2022**, *201*, 215–238. [CrossRef]
5. Mobayen, S.; Bayat, F.; ud Din, S.; Vu, M.T. Barrier function-based adaptive nonsingular terminal sliding mode control technique for a class of disturbed nonlinear systems. *ISA Trans.* **2023**, *134*, 481–496. [CrossRef] [PubMed]
6. Khan, H.; Abbasi, S.J.; Lee, M.C. Robust Position Control of Assistive Robot for Paraplegics. *Int. J. Control Autom. Syst.* **2021**, *19*, 3741–3752. [CrossRef]
7. Abbasi, S.J.; Khan, H.; Lee, J.W.; Salman, M.; Lee, M.C. Robust Control Design for Accurate Trajectory Tracking of Multi-Degree-of-Freedom Robot Manipulator in Virtual Simulator. *IEEE Access* **2022**, *10*, 17155–17168. [CrossRef]
8. Humaidi, A.J.; Hasan, A.F. Particle Swarm Optimization-Based Adaptive Super-Twisting Sliding Mode Control Design for 2-Degree-of-Freedom Helicopter. *Meas. Control* **2019**, *52*, 1403–1419. [CrossRef]
9. Wang, Y.; Zhu, K.; Yan, F.; Chen, B. Adaptive Super-Twisting Nonsingular Fast Terminal Sliding Mode Control for Cable-Driven Manipulators using Time-Delay Estimation. *Adv. Eng. Softw.* **2019**, *128*, 113–124. [CrossRef]
10. Wang, H.; Fang, L.; Song, T.; Xu, J.; Shen, H. Model-free Adaptive Sliding Mode Control with Adjustable Funnel Boundary for Robot Manipulators with Uncertainties. *Rev. Sci. Instrum.* **2021**, *92*, 065101. [CrossRef]
11. Xi, R.-D.; Xiao, X.; Ma, T.-N.; Yang, Z.-X. Adaptive Sliding Mode Disturbance Observer-Based Robust Control for Robot Manipulators Towards Assembly Assistance. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6139–6146. [CrossRef]
12. Jing, C.; Xu, H.; Niu, X. Adaptive Sliding Mode Disturbance Rejection Control with Prescribed Performance for Robotic Manipulators. *ISA Trans.* **2019**, *91*, 41–51. [CrossRef]

13. Zhao, H.; Tao, B.; Ma, R.; Chen, B. Manipulator trajectory tracking based on adaptive fuzzy sliding mode control. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7620. [CrossRef]
14. Khan, H.; Lee, M.C. Extremum Seeking-Based Adaptive Sliding Mode Control with Sliding Perturbation Observer for Robot Manipulators. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 5284–5290.
15. Razmi, H.; Afshinfar, S. Neural Network-Based Adaptive Sliding Mode Control Design for Position and Attitude Control of a Quadrotor UAV. *Aerosp. Sci. Technol.* **2019**, *91*, 12–27. [CrossRef]
16. Chen, Z.; Huang, F.; Chen, W.; Zhang, W.; Sun, W.; Chen, J.; Zhu, S.; Gu, J. RBFNN-Based Adaptive Sliding Mode Control Design for Delayed Nonlinear Multilateral Telerobotic System with Cooperative Manipulation. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1236–1247. [CrossRef]
17. Wang, D.; Shen, Y.; Sha, Q.; Li, G.; Kong, X.; Chen, G.; He, B. Adaptive DDPG Design-Based Sliding-Mode Control for Autonomous Underwater Vehicles at Different Speeds. In Proceedings of the IEEE Underwater Technology (UT), Kaohsiung, Taiwan, 16–19 April 2019; pp. 1–5.
18. Mosharafian, S.; Afzali, S.; Bao, Y.; Velni, J.M. A Deep Reinforcement Learning-Based Sliding Mode Control Design for Partially Known Nonlinear Systems. In Proceedings of the European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 2241–2246.
19. Lei, C.; Zhu, Q. U-Model-Based Adaptive Sliding Mode Control using a Deep Deterministic Policy Gradient. *Math. Probl. Eng.* **2022**, *2022*, 8980664. [CrossRef]
20. 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. [CrossRef]
21. Abbasi, S.J.; Lee, S. Enhanced Trajectory Tracking via Disturbance-Observer-Based Modified Sliding Mode Control. *Appl. Sci.* **2023**, *13*, 8027. [CrossRef]
22. Raoufi, M.; Habibi, H.; Yazdani, A.; Wang, H. Robust Prescribed Trajectory Tracking Control of a Robot Manipulator Using Adaptive Finite-Time Sliding Mode and Extreme Learning Machine Method. *Robotics* **2022**, *11*, 111. [CrossRef]
23. Saleki, A.; Fateh, M.M. Model-free control of electrically driven robot manipulators using an extended state observer. *Comput. Electr. Eng.* **2020**, *87*, 106768. [CrossRef]
24. Zheng, Y.; Tao, J.; Sun, Q.; Zeng, X.; Sun, H.; Sun, M.; Chen, Z. DDPG-Based Active Disturbance Rejection 3D Path-Following Control for Powered Parafoil Under Wind Disturbances. *Nonlinear Dyn.* **2023**, *111*, 1–17. [CrossRef]
25. Sun, M.; Zhang, W.; Zhang, Y.; Luan, T.; Yuan, X.; Li, X. An Anti-Rolling Control Method of Rudder Fin System Based on ADRC Decoupling and DDPG Parameter Adjustment. *Ocean. Eng.* **2023**, *278*, 114306. [CrossRef]
26. Yang, J.; Peng, W.; Sun, C. A Learning Control Method of Automated Vehicle Platoon at Straight Path with DDPG-Based PID. *Electronics* **2021**, *10*, 2580. [CrossRef]
27. Dey, N.; Mondal, U.; Mondal, D. Design of a H-Infinity Robust Controller for a DC Servo Motor System. In Proceedings of the 2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI), Kolkata, India, 21–23 October 2016; pp. 27–31.