A DRL-Based Edge Intelligent Servo Control with Semi-Closed-Loop Feedbacks in Industrial IoT

Hao Zheng[1], Haifeng Zhu[2], Hao Wu[1], Changyan Yi[1], Keke Zhu[2], and Xingan Dai[2]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2] Servo Technology Institute of China Aerospace Science and Industry Corporation, Nanjing, China
{zhenghao,cacggzhf,hao.wu,changyan.yi,keke.zhu,xingan.dai}@nuaa.edu.cn

Abstract. In industrial IoT, the design of control algorithms is pivotal to industrial servo systems. Unlike existing work, we study the industrial servo system control through edge computing with semi-closed-loop feedbacks, high-order nonlinear disturbances and lightweight implementation requirements. Particularly, in this paper, we take permanent magnet synchronous motor (PMSM) driven servo system as a typical industrial IoT device, propose a novel deep reinforcement learning based semi-closed-loop control algorithm, i.e., DRL-SCLC, and successfully deploy it on an edge server to provide real-time edge intelligent decision making. The control problem is formulated as a Markov decision process and then solved by the designed DRL-based algorithm, for minimizing the absolute error between reference signal and corresponding system response. To guarantee robustness, we further integrate "three-loop control structure" in traditions to DRL-SCLC for restricting outputs within a desired limit. Experiments on a real-world aerospace servo testbed show that the proposed solution is not only effective but also superior over counterparts.

Keywords: Industrial IoT · edge intelligence · DRL · industrial servo systems

1 Introduction

In industrial internet of things (IoT), industrial servo system has been regarded as a key enabler for advanced automation and manufacturing, being an essential component for the success of Industry 4.0[1]. Permanent magnet synchronous motor (PMSM), renowned for its high efficiency, precise control capabilities and low maintenance requirements, offers an ideal solution to empower the industrial servo system[4]. This promotes the development of PMSM-driven servo systems with the wide applications in industrial IoT, e.g. robotics, electric vehicles and aerospace. Among all issues, the design of control algorithms is pivotal to the functionality of PMSM-driven servo systems, significantly affecting the performance in terms of accuracy, adaptability, stability and robustness[6] for industrial IoT.

In industrial IoT, existing traditional feedback-based control algorithms, such as proportion integration differentiation (PID), active disturbance rejection control (ADRC), and their extended versions, are most commonly adopted for PMSM-driven servo systems due to their simplicity, flexibility and ease of adjustment[5, 9]. Although it has been revealed that such algorithms are effective in general, with the rapid evolution of various application scenarios for industrial IoT and their increasingly high quality-of-service demand, some inherent limitations are emerging.

First and foremost, feedback signals at the control terminal may not always be available or reliable because of harsh working conditions or high implementation costs. For example, in PMSM-driven aerospace applications for industrial IoT, any substantial increase of the weight or the change

of the structure caused by the installment of mechanical sensors is not be permitted[7], meaning that feedback signals of some control terminals, such as aircraft control surfaces, are extremely hard to be obtained directly. In this case, to make the traditional feedback-based control algorithms applicable, one have to estimate the servo system outputs (i.e., feedback signals) based on the easily reachable PMSM feedbacks[3] and the physical configuration of transmission devices. This motivates us to design an intelligent algorithm, circumventing the above difficulty for tackling the servo system control with semi-closed-loop feedbacks (i.e., feedback signals are not available at the terminal end but at the intermediate PMSM).

Furthermore, most of the PMSM-driven servo systems are high-order nonlinear with unknown disturbances[10], challenging traditional linear algorithms in high-precise control due to fixed model parameter settings, since they cannot adapt to time-varying dynamics (e.g., the frequent variations of reference signals and system loads). This motivates us to design a fully nonlinear and potentially model-free control algorithm based on advanced techniques, such as machine learning.

Last but not least, in practical implementations, industrial control systems may require real-time decision making and quick response to fast-changing environments, prompting the deployment of control algorithms on the edge sevrer to realize edge intelligence[12]. Particularly, for PMSM-driven servo systems, digital signal processor (DSP) and field-programmable gate array (FPGA) are the commonly used edge computing devices, responsible for the execution of control algorithms. However, they have relatively limited computational resources compared to those of cloud centers, posing a strong need of highly efficient but lightweight control algorithm design and integration[11].

To address the aforementioned issues, in this paper, we propose a novel deep reinforcement learning based semi-closed-loop control algorithm, i.e., DRL-SCLC, and after model compressing, we further deploy it on an edge server for industrial PMSM-driven servo systems in motor position tracking. The optimal control problem is formulated as a Markov decision process (MDP), aiming to minimize the long-term absolute error between the given reference signal and the corresponding system response, where action is the voltage imposed on PMSM, and the time-varying state includes instantaneous variations of the reference signal and intermediate PMSM feedbacks. A DRL agent is trained and then ran online to provide adaptive outputs, achieving semi-closed loop while nonlinear control. To prevent unsafe actions potentially taken by the DRL agent, we integrate the "three-loop control structure" of traditional control algorithms into the design of the DRL framework, which can restrict the outputs within a desired limit. Experimental study on a real-world PMSM-driven aerospace servo testbed is conducted, verifying the effectiveness of proposed solution.

The main contributions of this paper are as follows:

- We formulate a long-term position tracking control problem for PMSM-driven servo system in industrial IoT with semi-closed-loop feedbacks, system disturbances and load dynamics.
- We propose a novel algorithm, called DRL-SCLC, which can provide adaptive outputs, achieving semi-closed loop while nonlinear control. In such DRL-SCLC design, we integrate the "three-loop control structure" of traditional control algorithms for eliminating unsafe exploration and exploitation.
- We successfully deploy the proposed DRL-SCLC on DSP TMS320C6713B for the edge intelligent control, and evaluate its performance on a real-world PMSM-driven aerospace servo testbed.

[3] Different from terminal feedbacks, PMSM feedbacks can be naturally collected by hall sensors inside PMSM.

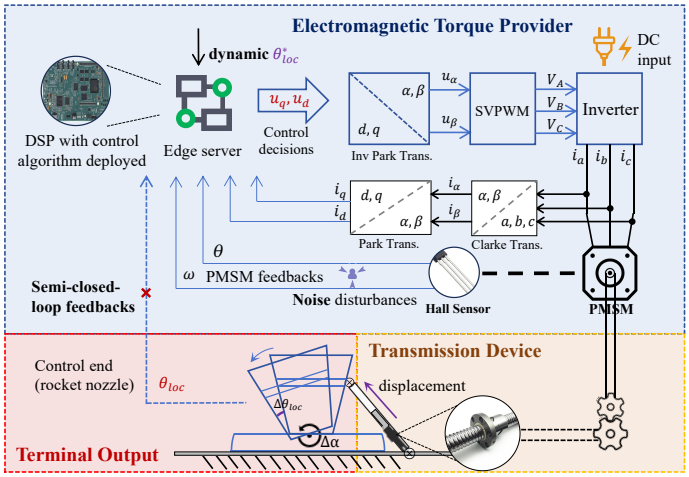

Fig. 1. PMSM-driven servo system with semi-closed-loop feedbacks.

2 System Model and Problem Formulation

2.1 System Overview

Consider a PMSM-driven system in industrial IoT, as shown in Fig. 1, consisting of an edge server (acting as the controller), a surface-mounted PMSM (regarded as the electromagnetic torque input), a lead-screw (regarded as the transmission device), and the control terminal. The edge server receives time-varying reference signals from the upper controller, together with PMSM feedbacks (taking into account the semi-closed-loop settings), to decide the voltage imposed on PMSM for appropriate torque. PMSM then generates the torque to empower the control terminal to track the reference signals. The torque is transmitted to the terminal through the mechanical structure of the transmission device. The terminal, as the control end, however may deviate an uncertain angle due to complex transmission structure and load dynamics. Note that although PMSM feedbacks are utilized for terminal estimation, these feedbacks collected by hall sensors inevitably suffer from noise disturbances introduced by the electromagnetic vibration. Moreover, the estimation based on the mathematical model of the transmission device is commonly inaccurate. These request the controller to perform self-adaptive intelligent decision making. Unfortunately, with limited computation resources, complex control algorithms can hardly be implemented on the edge server, and thus lightweight deployment is necessary.

Overall speaking, we aim to design and practically deploy an intelligent algorithm for optimizing the long-term performance of the PMSM-driven servo system (i.e., the absolute position tracking error between reference and terminal outputs) for industrial IoT by carefully determining the voltage imposed on PMSM instantaneously, with semi-closed-loop feedbacks, high-order nonlinear disturbances and time-varying dynamics.

2.2 Electromagnetic Torque Input

Following the literature, high-order nonlinear model of the PMSM can be specified by synchronous dq coordinate, in which voltages u_d, u_q and currents i_d, i_q are fundamental parameters to the for-

mulation. Based on Clarke's and Park's transformations, the stator current can be expressed as

$$\begin{cases} \dfrac{d\omega}{dt} = \dfrac{1}{J}T_e - \dfrac{1}{J}T_f - \dfrac{1}{J}T_L, \\ \dfrac{di_d}{dt} = -\dfrac{R_s}{L_d}i_d + \dfrac{1}{L_d}u_d + \dfrac{L_q}{L_d}i_q\omega, \\ \dfrac{di_q}{dt} = -\dfrac{R_s}{L_q}i_q + \dfrac{1}{L_q}u_q - \dfrac{L_d}{L_q}i_d\omega - \dfrac{\varphi_f}{L_q}\omega, \end{cases} \tag{1}$$

where T_L is the load torque of the PMSM-driven system, T_f represents the friction torque and can be approximately represented as $T_f = B_m\omega$ with B_m standing for the damping coefcient, T_e represents the electromagnetic torque and can be obtained by

$$\begin{cases} T_e = \dfrac{3}{2}n_p(\varphi_d i_q - \varphi_q i_d), \\ \varphi_d = L_d i_d + \varphi_f, \\ \varphi_q = L_q i_q. \end{cases} \tag{2}$$

For surface-mounted PMSM, the magnetic paths on both dq axes pass through the permanent magnet. The inductances are equal as $L_d = L_q$ and T_e can be simplified as $T_e = \frac{3}{2}n_p\varphi_f i_q$. Basides, we can set "$i_d = 0$" by intuition because flux-producing component i_d does not generate torque.

2.3 Transmission Process

For the transmission mechanism, we employ the ball screw feed-drive based one which is mostly utilized in industrial servo systems. The electromagnetic torque generated by PMSM drives the ball screw, converting the rotational motion of PMSM to the linear motion. In such a system, structural vibration significantly impairs the achievable positioning accuracy, due to the inherent structural flexibility of drive components, causing system disturbances. An approximate relationship between the displacement of the ball screw L (in millimeters) and the rotation angle of the PMSM θ (in meters) can be expressed as

$$L \approx 1000 \cdot \theta/(Ni \cdot R_s), \tag{3}$$

where Ni and R_s are the gear ratio and ball screw radius.

2.4 Terminal Output

For the mechanical structure of the control terminal, one end of the ball screw is fixed to the ground, while the other is fixed to the terminal side. The rotation of the terminal is driven by the displacement of the ball screw. The fixed pivot of the terminal, together with two ends of the ball screw, form a triangular linkage mechanism. Therefore, the lead screw's displacement L can be estimated by a cosine formula in terms of the deflection angle of the terminal α as

$$L \approx \sqrt{a^2 + b^2 - 2ab\cos(\alpha + \alpha_0)} - L_0, \tag{4}$$

where a, b represent fixed lengths of two sides of the triangular linkage mechanism, α_0 represents the initial angle of the terminal, and L_0 represents the initial displacement of lead screw. The rotation angle of terminal θ_{loc} is equivalent to the fixed pivot α because they are connected by same bracket.

2.5 Problem Formulation

As the controller, the edge server sends the command to the PMSM, making it generate electromagnetic torque T_e. By the motion transfer, the rotation angle θ is converted into the torque of the transmission device M, finally driving the terminal to rotate an angle θ_{loc}. To evaluate the control efficiency, we take the long-term accumulating position tracking error between the given reference signal θ^*_{loc} and the terminal output θ_{loc} within time range $t \in [0, T], T \to \infty$ as the performance measurement, expressed as $\mathcal{A}_1 = \int_0^T |\theta^*_{loc}(t) - \theta_{loc}(t)|dt$.

With the aim of minimizing $\mathcal{A}_1$, we formulate a position tracking optimization problem by optimizing $u_q(t)$ imposed on PMSM, which can be mathematically formulated as

$$\mathcal{P}_1 : \min_{u_q(t)} \lim_{T\to\infty} \mathcal{A}_1, \tag{5}$$

$$s.t.\ (1), (2), \tag{5a}$$

$$\{\theta^*_{loc}(t) | t \in [0, T]\}, \tag{5b}$$

$$\theta^*_{loc}(t) = \alpha(t), \tag{5c}$$

$$L_{loc}(t) \approx \sqrt{a^2 + b^2 - 2ab\cos(\alpha(t) + \alpha_0)} - L_0, \tag{5d}$$

$$L(t) \approx 1000 \cdot \theta(t)/(Ni \cdot R_s), \tag{5e}$$

$$|\omega(t)| \leq \omega_{\max}, \tag{5f}$$

$$\sqrt{i_q^2(t) + i_d^2(t)} \leq i_{\max}, \tag{5g}$$

$$|u_q(t)| \leq u_{q_{\max}}, \tag{5h}$$

where constraint (5a) is the PMSM model formulation; constraint (5b) indicates that the system tracks the given reference signals; constraints (5c), (5d) and (5e) represent the terminal output estimation in semi-closed-loop feedbacks; constraints (5f), (5g) and (5h) are for guaranteeing safety issues outlined in the traditional "three-loop control structure"[3]. Obviously, solving problem $\mathcal{P}_1$ is very challenging because *i*) the actual terminal output, due to semi-closed-loop feedbacks and system disturbances, are hard to be accurately estimated, making it inappropriate to be solved by classic control algorithms (e.g., PID, LQR, ADRC); *ii*) the considered system, due to system running dynamics, the model parameters cannot be accurately determined, indicating that $\mathcal{P}_1$ cannot be solved within a polynomial time complexity; and *iii*) this is in fact an online problem with uncertainties on not only complex system structure but also load dynamics.

3 Deep Reinforcement Learning based Semi-Closed-Loop Control (DRL-SCLC)

3.1 Problem Transformation

Observed that, problem $\mathcal{P}_1$ is actually a sequential decision making one (where the running state is the imposed voltage), the optimization of problem $\mathcal{P}_1$ can be seen as an MDP in general. However, the coventional MDP solutions (e.g., DRL)[8] may not be directly applicable here. The key reason is that conventional solutions aim to minimize the objective function only, while may occasionally violate constraints causing safety concerns, and possibly neglect inherent correlation between control decisions leading to unstable outcomes. Thus, with a particular consideration of guaranteeing reliablity (i.e., safety and stability), we first conduct a problem transformation.

Reference Signal Transformation We introduce a two-order tracking differentiator (TD) for transforming reference signal $\theta^*_{loc}(t)$ as

$$\begin{cases} \dot{\theta_{v2}}(t) = r^2(\theta^*_{loc}(t) - \theta_{v1}(t)) - rh \cdot \theta_{v2}(t), \\ \dot{\theta_{v1}}(t) = \theta_{v2}(t). \end{cases} \tag{6}$$

Feedback Signal Transformation An extended state observer (ESO) is employed for feedback signals $\theta(t)$ to estimate disturbances in the system. The function is designed based on the difference e between the collected feedback at the current and previous moment. If e is within the threshold, it returns a higher gain, otherwise a lower gain. According to this, we define such a function as

$$fal(e(t), \alpha, \delta) = \begin{cases} e(t)/\delta^{1-\alpha}, & |e(t)| \leq \alpha, \\ |e(t)|^{\alpha} sign(e(t)), & |e(t)| > \alpha, \end{cases} \tag{7}$$

where $sign(e)$ is symbolic function.

Based on (7), we design the ESO to transform $\theta(t)$ with the estimated system disturbances caused by electromagnetic vibration as

$$\begin{cases} e(t) = \theta(t) - \theta_{z1}(t), \\ \dot{\theta_{z1}}(t) = \theta_{z2}(t) + \beta_1 e(t), \\ \dot{\theta_{z2}}(t) = \theta_{z2}(t) + \beta_2 fal(e(t), \alpha_1, \delta_1) + bu(t), \\ \dot{\theta_{z3}}(t) = \beta_3 fal(e(t), \alpha_2, \delta_2). \end{cases} \tag{8}$$

Differential Signal Generation The differential signal can be generated as

$$\begin{cases} e_1(t) = \theta_{v1}(t) - \theta_{z1}(t), \\ e_2(t) = \theta_{v2}(t) - \theta_{z2}(t), \\ \omega_0(t) = k_1 fal(e_1(t), \alpha_3, \delta_3) + k_2 fal(e_2, \alpha_4, \delta_4), \\ \omega^*(t) = sat(\omega_0(t) - z_3(t)/b_0, \omega_{\max}), \end{cases} \tag{9}$$

where $sat()$ is the saturation function.

Based on (9), the reference current on q axis can be generated as

$$i^*_q(t) = sat(K_P e_\omega(t) + K_I \int_0^t e_\omega(t)dt, i_{q_{max}}), \tag{10}$$

where $e_\omega(t) = \omega^*(t) - \omega(t)$. Eventually, referring to $i^*_q(t)$ and $i_q(t)$, the controller will generate the voltage signal u_q.

3.2 Problem Reformulation

In order to track the reference signals more accurately, with the utilization of differential signals, $\mathcal{A}_1$ can be rewritten as

$$\mathcal{A}_2 = \int_0^T [|\theta^*_{loc}(t) - \theta_{loc}(t)| + |\omega^*(t) - \omega(t)| + |i^*_q(t) - i_q(t)|]dt. \tag{11}$$

Replacing the objective of minimizing $\mathcal{A}_1$ by minimizing $\mathcal{A}_2$, along with the transformation conducted in the previous subsection, we can reformulate $\mathcal{P}_1$ into a new form as

$$\mathcal{P}_2 : \min_{u_q(t)} \lim_{T\to\infty} \mathcal{A}_2, \tag{12}$$

$$s.t.\ (1), (2), \tag{12a}$$

$$\{\theta^*_{loc}(t), \theta_{v1}(t), \theta_{v2}(t) | t \in [0, T]\}, \tag{12b}$$

$$\theta^*_{loc}(t) = \alpha(t), \tag{12c}$$

$$L_{loc}(t) \approx \sqrt{a^2 + b^2 - 2ab\cos(\alpha(t) + \alpha_0)} - L_0, \tag{12d}$$

$$L(t) \approx 1000 \cdot \theta(t)/(Ni \cdot R_s), \tag{12e}$$

$$\{\theta(t), \theta_{z1}(t), \theta_{z2}(t), \theta_{z3}(t) | t \in [0, T]\}, \tag{12f}$$

$$(9), (10), \tag{12g}$$

$$|\omega(t)| \le \omega_{\max}, \tag{12h}$$

$$\sqrt{i_q^2(t) + i_d^2(t)} \le i_{\max}, \tag{12i}$$

$$|u_q(t)| \le u_{q_{\max}}, \tag{12j}$$

Note that transformed signals are integrated into the reformulated problem $\mathcal{P}_2$, which takes the tracking signals of position, rotation speed and current on q axis into the objective function. Moreover, constraint (12b) utilizes (6) to rebuild the reference signal collection, which is smoother and more rational; constraint (12f) utilizes (8) to rebuild the position feedback signal collection, which estimates system disturbances and makes feed-forward compensation; constraint (12g) represents the "three-loop control structure" to ensure control reliablity.

3.3 Deep Reinforcement Learning based Solution

After the reformulation, we are now ready to solve problem $\mathcal{P}_2$, whose control process can be regarded as an MDP by a tuple $\{s(t), a(t), r(t), s^{'}(t)\}$, where $s(t)$ is the state space at time slot t, $a(t)$ is the action space, $r(t)$ is the corresponding reward, and $s^{'}(t)$ is the state transition after the agent executing the action at time slot t. Afterwards, the idea of DRL can be adopted. Particularly, we apply the actor-critic network to train the neural network for eventually obtaining the solution.

1. *State:* The state space is defined as follows.

$$s(t) = \{e_\theta(t), \theta_{v1}(t), \theta_{v2}(t), e_1(t), e_2(t), \omega(t), \omega^*(t), i_q(t), i_q^*(t)\}, \tag{13}$$

   where $e_\theta(t) = \theta^*_{loc}(t) - \hat{\theta_{loc}}(t)$ is the position error between the given reference and estimated feedback signal of terminal, $e_1(t), e_2(t)$ are the differential signals in (8).
2. *Reward:* Based on the state built above, the reward takes the PMSM running status into consideration, i.e., error on position, rotation speed and current on q axis. Coefficients are imposed on these indexes to balance their importance. The reward function is defined as

$$r(t) = -[\gamma_1(\theta^*_{loc}(t) - \theta_{loc}(t))^2 + \gamma_2(\omega^*(t) - \omega(t))^2 + \gamma_3(i_q^*(t) - i_q(t))^2]. \tag{14}$$

3. *Policy:* The optimal policy π^* is expected to maximize the cumulative discounted reward through the interaction with the PMSM working environment, and can be expressed mathematically as

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi[\sum\nolimits_{t=0}^{\infty} \gamma^t r(t) | \pi]. \tag{15}$$

Algorithm 1: DRL-SCLC Training Algorithm

Input: PMSM-driven servo system working simulator
Output: DRL-based model for the controller

for *all PMSM-driven servo systems* **do**
  Initialize actor network parameters ϑ and critic network parameters ξ for agent on edge;
  for *each episode* $= 1, 2, \ldots, N_{epi}$ **do**
    Initialize PMSM-driven servo system;
    Generate reference orders of the robotic arm $\{\theta^*_{loc}(t)\}\ \forall t \in \mathcal{T}$;
    for *each time step* $t = 0, \Delta t, 2\Delta t, \ldots, \mathcal{T}$ **do**
      Sensors collect system information to generate state $s(t)$;
      Actor network outputs $u_q(t)$ based on policy $\pi : s(t) \rightarrow a(t)$;
      for *each time step* $t^{'} = t, t+1, \ldots, t+\Delta t-1$ **do**
        PMSM rotates under received voltage input $u_q(t)$ based on e.q.1, the displacement of PMSM $\theta(t^{'})$ drives the the movement of the lead screw, finally lead to the movement of the robotic arm $\theta_{loc}(t^{'})$.
      Sensors collect feedback for the controller to observe the next state $s^{'}(t)$ and the reward $r(t)$;
      Store $\{s(t), a(t), r^(t), s^{'}(t)\}$ in the replay buffer;
    if *the replay buffer is full* **then**
      Calculate ϑ and ξ loss based on e.q. (17), (18);
      Update actor ϑ and critic ξ network: $\vartheta^{'} = \tau\vartheta + (1-\tau)\vartheta^{'}$, $\xi^{'} = \tau\xi + (1-\tau)\xi^{'}$
return *Controller with optimal policy* π^*

We utilizes proximal policy optimization (PPO) algorithm to solve the MDP above. The employed PPO utilizes the ratio of the new policy to the old one to limit the update range. Moreover, importance sampling for gradient calculation is utilized to allow data reuse for multiple gradient updates, and is defined as

$$L^{PPO}(\vartheta) = \hat{\mathbb{E}}(t)[\frac{\pi_\vartheta(a(t)|s(t))}{\pi_{\vartheta old}(a(t)|s(t))}\hat{A}(t) - \lambda D_{KL}[\frac{\pi_\vartheta(a(t)|s(t))}{\pi_{\vartheta old}(a(t)|s(t))}]]. \tag{16}$$

PPO algorithm is designed based on the actor critic structure. The goal of the actor is to maximize L^{PPO}, which can be divided into two types: L^{KLPEN} in PPO_1 and L^{CLIP} in PPO_2. PPO_2 uses a clip function to limit its ratio within $(1-\epsilon, 1+\epsilon)$, and can be written as

$$L^{CLIP}(\vartheta) = \hat{\mathbb{E}}_t[\min(\frac{\pi_\vartheta(a_t, s_t)}{\pi_{\vartheta old}(a_t, s_t)}\hat{A}_t, clip(\frac{\pi_\vartheta(a_t, s_t)}{\pi_{\vartheta old}(a_t, s_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t)]. \tag{17}$$

The goal of the critic network is to minimize the mean square error between the estimated value and the actual one. The critic loss can be expressed as

$$L(\phi) = -\sum_{t=1}^{T_b}(\sum_{t^{'}>t}\gamma^{t^{'}-t}r(t^{'}) - v_\phi(s(t)))^2, \tag{18}$$

where T_b is the maximum capacity of replay buffer, and ϕ is neural network parameter of the critic.

Considering the high computation resource consumption of DRL-based solution, real-time control at every time slot may lead to a high control latency. Besides, if control frequency f_c equals to sampling frequency of the sensor f_s, there may exist virbation in the output because DRL-based solution output is unstable. Thus, at a given time slot $t \in \{0, \Delta t, 2\Delta t, ...\mathcal{T}\}$, controller observes states $s(t)$ from sensors and outputs action $a(t)$ on the voltage on q axis for time slot $t+1$. Such an action will be hold by controller until time slot $t+\Delta t$, i.e., next action output is produced, implying that $f_c = f_s/\Delta t$. Detailed pseudocode of the proposed solution is shown in Algorithm 1.

4 Experimental Study

4.1 Experimental Setup

Table 1. Parameter Settings

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
J	$2.1994e^{-5}$	N_i	2.68	n_p	2	γ	0.99
R_s	1570 rad/m	L_d, L_q	$1.5e^{-3}$H	B_m	1.59(N·m)/(rad·s)	N_{epi}	2048
k_1	5	k_2	2.5	K_P	25	τ	$3e^{-4}$
K_I	5	$u_{q_{\max}}$	146.67V	$\omega_{\max}$	732.67(rad/s)	γ_1	175
a	283.72mm	b	503.016mm	$i_{\max}$	50A	γ_2	1
α	0.5	α_0	24.067°	σ	0.1	γ_3	1
L_0	270mm	$\mathcal{T}$	10s	f_s	10000Hz	f_c	200Hz

To implement the proposed control algorithm, i.e., DRL-SCLC, and evaluate its performance, we particularly build a PMSM-driven aerospace servo testbed. A DSP is employed as the edge server for deploying the algorithm, which is specified as TMS320C6713B with 300Mhz frequency, 2MB RAM and 32-bit floating-point number computational accuracy. A FPGA is responsible for executing Park conversions and inverted Park conversions, and it is specified as XC4VLX60 with 60Mhz frequency. Considering the vibration caused by fluctuations and sensor noise, a kalman filtering algorithm is applied on DSP for processing feedback signals before executing the control algorithm. Besides, to suppress the potential vibrations on the output, a lowpass filtering is imposed after executing DRL-SCLC algorithm. The trained model is further lightweight deployed by exporting model parameters of input, hidden and output layers, translating into the efficient C language code, and then burn into DSP.

According to conventional settings of control systems and specific configurations of the considered aerospace application for industrial IoT[2], we train the proposed DRL-SCLC with step signals, low frequency high amplitude sine wave signals and high frequency low amplitude sine wave signals.

For comparison purpose, besides proposed DRL-SCLC algorithm, a PID-based control scheme with improved quantum genetic optimization[9] and an ADRC-based contorl scheme[5] with an improved position-speed loop and parameter self-tuning are also simulated as benchmarks. Since both benchmarks are not originally designed for semi-closed-loop control, $\theta_{loc}(t)$ and $\theta^*_{loc}(t)$ are converted into L based on (3) and (4) to form closed-loop responses, respectively. Moreover, experience-based linear feedforward compensations are also added on position loop to close the gap caused by the inaccurate estimation. The compensation in the complex field is defined as $s = \frac{10s}{s+10}$ on $\theta^*_{loc}(t)$, and $s = \frac{0.5s}{s+4}$ on $\theta_{loc}(t)$. Table 1 summarizes main parameter settings in experiment.

4.2 Experimental Evaluations

Table 2. Control result on $\delta^+_{max}, \delta^-_{max}, \Delta\delta_{max}, \delta_0, \bar{\omega}, \sigma\%$ under given $\theta^*_{loc}(t)$

Benchmark	$\delta^+_{max}(°)$	$\delta^-_{max}(°)$	$\Delta\delta_{max}(°)$	$\delta_0(°)$	$\bar{\omega}$(rad/s)	$\sigma\%$
DRL-SCLC	**3.3085**	**-3.2566**	**0.5777**	**3.424e-04**	77.0339	0.90%
PID-based	3.2786	-3.2168	0.6409	4.708e-04	**106.6639**	2.78%
Improved-ADRC	3.2691	-3.2059	0.6286	8.609e-04	40.6078	**0.31**%
DRL-SCLC/PID-based	+0.91%	+1.24%	-9.85%	-172.73%	-27.77%	-67.63%
DRL-SCLC/Improved-ADRC	+1.21%	+1.58%	-8.10%	-60.23%	+162.67%	+190.32%
PID-based/Improved-ADRC	+0.29%	+0.34%	+1.95%	-45.31%	+89.70%	+786.77%

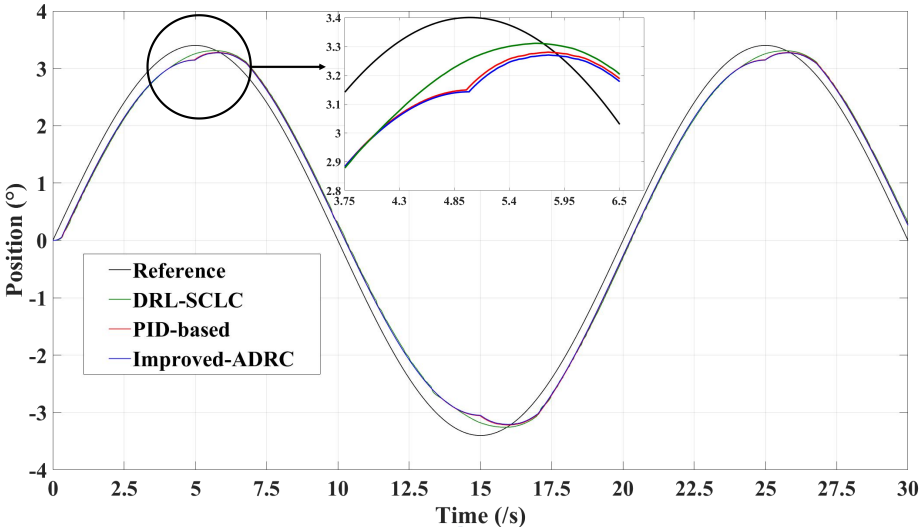

Fig. 2. Load position experiment result.

Load Position Experiment To measure the effectiveness of control algorithms on position tracking under low frequency signals, a load position experiment is conducted. The order from the upper controller θ^*_{loc} is given as a sine wave with fixed frequency and amplitude, and the feedback positions θ_{loc} are collected. $\theta^*_{loc}(t)$ is given as $\theta^*_{loc}(t) = 3.4 \cdot \sin(0.05 \times 2\pi(t)), t \in [0, 20]$. The performance indicators of load position experiment are calculated as

$$\delta^+_{max} = \max_{t\in[0,20]} \theta_{loc}(t), \delta^-_{min} = \min_{t\in[0,20]} \theta_{loc}(t), \Delta\delta_{max} = \max_{t\in[0,20]} |\theta^*_{loc}(t) - \theta_{loc}(t)|, \delta_0 = |\theta_{loc}(t)| \text{ when } \theta^*_{loc}(t) = 0$$

where $\delta^+_{max}, \delta^-_{min}$ are the maximum and minimum swing angle in positive and negative directions, which indicates the position tracking speed and terminal estimation performance of control algorithm; $\Delta\delta_{max}$ represents the maximum of position loopback width, which represents the maximum of the position tracking error; δ_0 is the zero-position deviation, which measures the symmetry of control algorithm in both directions when faced with a symmetrical low-frequency sinusoidal order.

Fig. 2 and Table 2 illustrate the load position experiment result under given $\theta^*_{loc}(t)$. It is shown in figure that for all control algorithms, the position tracking performance is almost the same at the beginning. However, when $\theta^*_{loc}(t)$ reaches 3.4°, DRL-SCLC outperforms benchmarks with higher δ^+_{max} and smaller δ^-_{min} because it can learn real-time terminal position during contorl, so as to minimize position error, while benchmarks suffer from inaccurate estimations. Besides, in Table 2, we can see that DRl-SCLC is more efficient on smaller $\Delta\delta_{max}$ and δ_0 as "three-loop control structure" is integrated, avoiding instability during positoin tracking.

Rotation Speed Experiment To measure the effectiveness of control algorithms on position tracking under step signals, a rotation speed experiment is conducted. θ^*_{loc} is set to 3° from the initial time slot. The performance indicators of rotation speed experiment are calculated as

$$N(t) = \{t | 0.3° < \theta_{loc}(t) < 2.7°\}, \bar{\omega} = \frac{1}{|N(t)|} \sum\nolimits_{t\in N(t)} \omega(t), \sigma\% = (c(t_p) - c(\infty))/c(\infty) \times 100\%,$$

Table 3. Control result on L, ϕ under given $\theta^*_{loc}(t)$

Benchmark		0.5°		0.8°		1.1°	
		ϕ(°)	L(dB)	ϕ(°)	L(dB)	ϕ(°)	L(dB)
2.0 rad/s	DRL-SCLC	**-34.6186**	**-2.6942**	**-22.1154**	**-1.5647**	**-16.4562**	**-1.0819**
	PID-based	-39.1228	-3.5124	-25.2032	-1.8631	-19.5374	-1.2248
	Improved-ADRC	-37.0531	-3.4548	-22.7996	-1.8675	-16.6131	-1.2561
4.0 rad/s	DRL-SCLC	**-35.2117**	**-3.4199**	**-22.3902**	**-1.8475**	**-16.5903**	**-1.2009**
	PID-based	-41.0283	-4.1134	-27.7989	-2.0887	-22.8683	-1.3606
	Improved-ADRC	-36.6057	-3.8738	-22.6770	-2.0121	-16.6416	-1.3075
8.0 rad/s	DRL-SCLC	**-35.9444**	-3.7979	**-22.6479**	**-1.8465**	**-16.6614**	**-1.1269**
	PID-based	-45.6413	-4.4581	-33.6918	-2.3496	-30.1233	-1.5641
	Improved-ADRC	-36.8707	**-3.7132**	-23.2074	-1.9041	-17.0674	-1.2918
16.0 rad/s	DRL-SCLC	**-36.0237**	-2.9298	**-23.5125**	**-0.9440**	**-17.7739**	**-0.1003**
	PID-based	-56.3407	-4.7633	-46.6562	-2.7073	-44.9531	-2.1349
	Improved-ADRC	-40.1180	**-2.8961**	-25.2925	-1.6275	-18.6574	-0.7646

where $N(t)$ is the rise time, which represents the duration when terminal output is between 0.3° to 2.7°; $\bar{\omega}$ is the corresponding average rotation speed in duration; $\sigma\%$ represents the overshoot of the control algorithm; $c(t_p)$ is the instantaneous maximum position, and $c(\infty)$ is the steady position.

Table 2 shows the rotation speed experiment result under given $\theta^*_{loc}(t)$. It is obvious that PID-based algorithm has the fastest acceleration at the beginning, and also has the highest overshoot due to rapid rise time. DRL-SCLC, however, balances the trade-off between rise time and overshoot, in which rotation speed is faster than improved-ADRC, and overshoot is smaller than PID-based algorithm. Furthermore, we can observe that DRL-SCLC is the closest one to $\theta^*_{loc}(t)$ when $\theta_{loc}(t)$ stabilizes. This is because DRL-SCLC utilizes actual terminal feedbacks for training, helping the agent to learn terminal output more accurately.

Frequency Experiment To measure the effectiveness of control algorithms on position tracking under high frequency signals, a frequency experiment is conducted. θ^*_{loc} is given as a sine wave signal, whose $\mathcal{A}$ is among 0.5°, 0.8°, 1.1°, and ω is among 2 rad/s, 4 rad/s, 8 rad/s and 16 rad/s. The performance indicators of frequency experiment are calculated as

$$A = \sqrt{a^2 + b^2}, \Phi = \arctan(a/b), L = 20 \cdot \lg A/A^*, \phi = (\Phi - \Phi^*) \cdot (180/\pi).$$

where a^*, b^*, a, b are results of orthogonal decomposition of θ^*_{loc} and θ_{loc} on sine and cosine bases.

Table 3 demonstrates frequency experiment result under given $\theta^*_{loc}(t)$ with different frequencies and amplitudes. We can see that with the increase of ω, all control algorithms experience phase and amplitude attenuation because of rapid changes in frequency and system dynamics. However, DRL-SCLC and improved-ADRC outperform PID-based algorithm due to their system disturbances feedforward compensation mechanism. Moreover, DRL-SCLC suffers the least phase attenuation because it makes self-adaptions to these changes, indicating its reliability. In addition, with the help of more accurate estimation, DRL-SCLC also has the least amplitude attention in most cases.

Table 4. Performance comparison of DRL-SCLC on edge and cloud

Deployment	δ^+_{max}(°)	δ^-_{max}(°)	$\Delta\delta_{max}$(°)	δ_0(°)	$\bar{\omega}$(rad/s)	$\sigma\%$
Edge	**3.3085**	**-3.2566**	**0.5777**	**3.424e-04**	**77.0339**	**0.90%**
Cloud	3.2835	-3.0100	0.6908	0.07505	76.6023	6.15%

DRL-SCLC on Edge or on cloud To show the effectiveness of lightweight deploying the proposed DRL-SCLC algorithm on the edge (i.e., DSP TMS320C6713B), Table 4 compares its performance

running on edge and cloud servers. Both load position and speed experiments are conducted. We can obviously see that deploying the algorithm on the edge can always reach a better performance. This is because deploying the algorithm on the cloud inevitably introduces a much larger communication delay, causing vibration and instability on control decisions, further leading to the tracking error.

5 Conclusion

In this paper, the industrial PMSM-driven servo system control in industrial IoT with semi-closed-loop feedbacks has been studied. Particularly, aiming to minimize the position trakcing error, a position tracking problem is formulated for determining the voltage imposed on PMSM. Moreover, for control safety and reliability, "three-loop control structure" is integrated for reformulating the problem. A novel solution, called DRL-SCLC, is proposed, which is further successfully deployed on an edge server for real-time intelligent decision making in motor position tracking. Experiment on a real-world aerospace servo testbed show that DRL-SCLC is superior compared to counterparts on not only reducing the long-term position tracking error, but also improving the control reliability.

References

1. Ahmed, I., Jeon, G., et al.: From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where. IEEE Trans. Industr. Inform. **18**(8), 5031–5042 (2022). https://doi.org/10.1109/TII.2022.3146552
2. Golnaraghi, F., Kuo, B.C.: Automatic control systems. McGraw-Hill Education (2017)
3. Hadi, N.H., Ibraheem, I.K.: Speed control of an spmsm using a tracking differentiator-pid controller scheme with a genetic algorithm. Int. J. Electr. Comput. Eng. **11**(2), 1728 (2021)
4. Liu, L., Guo, Y., et al.: Design and optimization technologies of permanent magnet machines and drive systems based on digital twin model. Energies **15**(17), 6186 (2022)
5. Lu, W., Li, Q., Lu, K., Lu, Y., Guo, L., Yan, W., Xu, F.: Load adaptive pmsm drive system based on an improved adrc for manipulator joint. IEEE Access **9**, 33369–33384 (2021). https://doi.org/10.1109/ACCESS.2021.3060925
6. Mohd Zaihidee, F., Mekhilef, S., et al.: Robust speed control of pmsm using sliding mode control (smc)—a review. Energies **12**(9), 1669 (2019)
7. Verkroost, L., Vandenabeele, T., et al.: Multi-agent position estimation in modular motor drives using low-resolution sensors. IEEE open j. Ind. Electron. **3**, 105–115 (2022). https://doi.org/10.1109/OJIES.2022.3146302
8. Viswanadhapalli, J.K., et al.: Deep reinforcement learning with reward shaping for tracking control and vibration suppression of flexible link manipulator. Appl. Soft Comput. **152**, 110756 (2024)
9. Wang, H., Xu, S., et al.: Pid controller for pmsm speed control based on improved quantum genetic algorithm optimization. IEEE Access **11**, 61091–61102 (2023). https://doi.org/10.1109/ACCESS.2023.3284971
10. Yang, C., Song, B., et al.: Adaptive identification of nonlinear friction and load torque for pmsm drives via a parallel-observer-based network with model compensation. IEEE Trans. Power. Electron. **38**(5), 5875–5897 (2023). https://doi.org/10.1109/TPEL.2023.3239609
11. Zhang, C., Meng, Y., et al.: A framework for mapping drl algorithms with prioritized replay buffer onto heterogeneous platforms. IEEE Trans. Parallel Distrib. Syst. **34**(6), 1816–1829 (2023). https://doi.org/10.1109/TPDS.2023.3264823
12. Zhou, Z., Chen, X., et al.: Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proc IEEE Inst. Electr. Electron Eng. **107**(8), 1738–1762 (2019). https://doi.org/10.1109/JPROC.2019.2918951