A Meta-Reinforcement Learning-Based Adaptive Robot Control for Human-Robot Collaboration in Personalized Production*

Hin Chi Kwok, Chengxi Li, Yatming Pang, Pai Zheng

***Abstract*— Nowadays, with the advancement of production technologies, the manufacturing paradigm has gradually shifted from mass production to a small-batch and high-variety personalized production manner, urged by high flexible automation capabilities. In this paradigm, the existing inspection and assembly processes after manufacturing still rely to a large extent on either human operators with low efficiency or machines with low flexibility. To solve this issue, human-robot collaboration (HRC) has been a prevailing topic of recent concerns. Current robot control strategies in human-machine collaboration are mainly through pre-defined programming and do not yet meet the need for flexible and adaptable tasks in individualised production. To address this challenge, this paper proposes a deep reinforcement learning (DRL) approach based on meta-learning to drive robots in HRC. It enables collaborative robots (cobots) to acquire basic skills and perform tasks based on personalised production requirements, improving learning efficiency and thus quickly adapting to new tasks for human operators. The robot control task was carried out in a simulated environment taken from a real production scenario to assess its efficacy. Experimental results show that our proposed method enables the robot to learn and perform HRC tasks quickly and outperforms the baseline DRL method in terms of success rate.**

I. Introduction

With the rapid development of modern manufacturing technologies and the globalisation of production, the old paradigm of mass production is being challenged by the increasing demand for individualised products and shorter product iteration cycles [1]. Personalised production is a customer-centred production paradigm that translates individual requirements into customised products or services at an appropriate cost, often in small batches [2, 3]. Currently, these products are typically produced manually, at high cost and low-efficiency [2]. Faced with this situation, the industry is concerned with the way manufacturing systems are designed and deployed to improve the production paradigm for personalised production performance. Collaborative robots are a new trend in industrial and service robotics and are part of the Industry 4.0 strategy. Along with the widespread deployment of robots, one possible solution is to use the flexibility of human operators and the reliability and efficiency of robots to perform a variety of manufacturing tasks in human-robot collaboration (HRC) manner.

One of the main issues in the practical deployment of robotic assist systems in HRC is to improve the robustness and design efficiency of cobot control strategies. For robot control in HRC, traditional pre-programmed robots with human involvement, such as demonstration robots that repeat the same array of actions in all operations, require human-guided sequences to execute the required movements, or specialised manipulation language-driven robots that execute actions via programs. These types of robots can only execute instructions to perform pre-learned actions that have been designed by engineers using prior knowledge of a specific production task. And they are unable to transfer control strategies adaptively to a range of new tasks. It is neither time-saving nor flexible in the face of diverse production processes [3], further limiting production power. Although research conducted on HRC has shown that collaborative approaches are most effective for small and medium-sized customised production [4], the issues of reducing programming time and improving the feasibility of collaborative solutions have not been addressed, and most HRCs require advanced adaptive systems to enable system reconfigurability for customised production [5]. To counter this problem, robot systems in HRC should have strong and agile learning capabilities to achieve the goal of personalized manufacturing, adapt precisely and efficiently to different production tasks, and assist humans to complete production tasks in a time-saving and fluent manner. Therefore, this work proposes a Robotic-Assistance System with self-learning capabilities.

Self-learning mechanisms are currently unexplored in depth in HRC, albeit it holds great promise [6]. In a personalised production model, self-learning robotic assisted systems (SLRAS) could self-adapt DRL algorithms to assist human operators by assisting with task goal setting, rather than additional human programming. Although existing DRL algorithms can significantly improve human productivity, reducing the time spent on the learning process is still an issue. Applying meta-learning with DRL is an option to shorten the design cycle of HRC's robot control strategies, providing the feasibility of handling HRC's compatibility in terms of personalised production trends. In theory, meta-reinforcement learning (meta-RL) is an algorithm that trains

*Research funded by the Laboratory for Artificial Intelligence (Project Code: RP2-1) in Design, Hong Kong Special Administrative Region and Undergraduate Research and Innovation Scheme (URIS), The Research Committee of The Hong Kong Polytechnic University (Project Code: 1-TA35)

H. C. Kwok is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong S.A.R. (e-mail:hin-chi.kwok@connect.polyu.hk).

C. X. Li is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong S.A.R., China; Lab for Artificial Intelligence in Design, Hong Kong S.A.R., China (e-mail: chengxi.li@connect.polyu.hk).

Y. M. Pang is with Lab for Artificial Intelligence in Design, Hong Kong S.A.R., China (e-mail: yatmingpang@aidlab.hk).

P. Zheng is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong S.A.R., China; Lab for Artificial Intelligence in Design, Hong Kong S.A.R., China (corresponding author to provide phone: +852 27665633; e-mail: pai.zheng@polyu.edu.hk).

an agent on a set of different tasks and extracts high-level knowledge that is shared across all training tasks. During the testing phase, the agent simply observes a number of trials in the new task to conform to a new task that is similar to the training task. Considering the research gap in the practical application of HRC in the personalised production paradigm, this capability is given to the robot in our work due to the fact that meta-learning can drive the adaptation of the robot in different HRC applications for personalised production.

In our work, a meta-learning approach, called model diagnostic meta-learning (MAML), is used in combination with the proximal policy optimization (PPO) deep reinforcement learning (DRL) algorithm [7, 8]. This meta-DRL approach is adopted to develop control strategies for robots in HRC and to verify whether meta-DRL can be quickly adapted to different manufacturing tasks in HRC. We organise a task set in the MuJoCo simulator [9] that consists of several control tasks in a manufacturing scenario, and the proposed meta-DRL approach is used to validate whether our approach is able to learn general control strategies from multi-task training and drive SLRAS to master new auxiliary tasks in HRC in a fast and advance manner. Our work contributes in two main perspectives: firstly, to prove whether the DRL algorithm is used to improve the generality of the robot in HRC, enabling the robot to learn different tasks autonomously; secondly, meta-RL assigns powerful learning capabilities to the robot to increase the learning efficiency and improve the speed of adaptation to HRC-assisted tasks.

The rest of the paper is organized as follows: In Section II, we provide an overview of the work related to flexible manufacturing and DRL. In Section III, we introduce the notation and concepts of DRL and meta-learning. In Section IV, we describe our proposed approach in detail. In Section V, the experimental results are presented and explained. Section VI demonstrates a case study of the SLRAS implementation, and we conclude the paper and present our future work in Section VII.

II. Related Work

As personalized production commence to be popular in the global market, the flexibility of production models with high product variability and short production cycles becomes an important characteristic of manufacturing systems in the era of Industry 4.0 [10]. Conventional automation has now reached a bottleneck of achieving high flexibility in manufacturing processes.

With breakthroughs in artificial intelligence (AI), the combination of AI technology and manufacturing has gradually moved from the concept to the application stage. In particular, the application of DRL and robotics in manufacturing has won new possibilities for advancing self-learning manufacturing. As self-learning production systems are an emerging branch of smart manufacturing, their prototypes and architectures have been intensively discussed and refined [6, 11]. Besides the basic-level work, cyber-physical system-based production task scheduling and control strategy management for flexible manufacturing through self-learning mechanisms are frequently mentioned [12, 13]. Additionally, data-driven quality control and optimization of process parameters for self-learning production systems are also under investigation [14].

In the context of personalised products, robotic systems address compatibility issues mainly through program switching, manufacturing sequence reorganisation, human-machine collaboration [15, 16], to name a few. A framework for a cyber-physical production system based on digital twins is indicated to overcome challenges in product acquisition, cost and performance, [2, 3] [17] proposes a hybrid manufacturing system called hy-FaaS (factory as a service), which is able to collaborate with humans for personalised production. However, there is little evidence of flexibility when the target product changes. With the success of DRLs in real-world production applications, their ability to learn has been recognised by the industry as accelerating the development of robotic self-learning. Several types of research have focused on safety issues and the mapping of mobile robots [18, 19], but the discussion on controlling industrial robot arms has yet been unfolding. Firstly, the DRL method requires a large amount of annotated training data to optimise performance. As it gains knowledge through trial and error, the robot can only start working with the operator on a task after a significant amount of effort has been made [20]. While it shows great potential within manufacturing, it is impractical to collect data during the actual manufacturing process and therefore becomes one of the limitations in landing. Similarly, designing accurate simulator environments is quite difficult and often not cost-effective [21]. Apart from the data problem, the optimal strategy generated by the DRL algorithm depends on the exploration and exploitation of the environment. However, in reality, extensive exploration in the manufacturing process is limited by safety constraints and production line requirements [22].

To address these limitations, a meta-learning mechanism is a solution that allows access to a priori knowledge for learning-to-learn capabilities. In theory, a model with prior knowledge should have the advantage of strong generalisation and rapid adaptation to new tasks. By providing less data, the robot will start to learn itself, and this mechanism can effectively enable the robot to behave as expected without the need for large amounts of data or significant amounts of time. However, applications of meta-learning mechanisms in manufacturing are mainly for computer vision and parameter tuning tasks such as defect detection with a small number of photographs [23] and machining energy tuning [24]. Meta-RL is a generic term for this approach combining meta-learning with DRL algorithms. Mainstream efforts are investigating meta-RL algorithms in the following areas: [25]

- One class learns a latent representation of task distributions [26, 27].
- The other class trains a model (policy function, loss function, etc.) during meta-training, and further improves performance in testing stage [7, 28].

While meta-RL is already an important topic in the field of artificial intelligence, its application in manufacturing is still in its infancy and has not yet been fully explored.

III. Preliminaries

This section introduces the basic principles of notation, DRL and meta-learning, which is essentially a trial-and-error optimisation process that aims to maximise cumulative rewards within a set range. The interaction process can be represented as a Markov Decision Process (MDP) with a tuple consisting of the following elements $(S, A, P_{sa}, R, \gamma)$:

- S is a set of states, where $s \in S$. The state set is used to illustrate the states of the environment.
- A is a set of actions, where $a \in A$. It is used to represent the available actions with discrete or continuous in a specific state.
- P_{sa} represents the transition probability distribution of executing action a under state s, i.e.

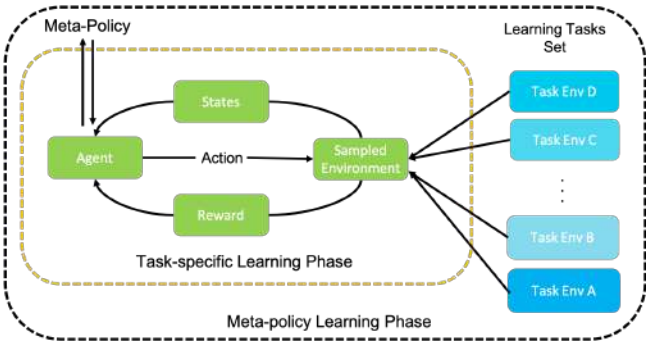
$$P^{a}_{ss'} = P(s_{t+1} = s' \,|\, S_t = s, A_t = a) \tag{1}$$

- $R(s,a)$ is the return reward. It is determined by the state which the agent in and the corresponding action executed by the agent.
- $\gamma \in [0,1]$ is a discount factor. It defines the decay of future accumulated rewards.

In episodic tasks, an episode of the task is treated as an MDP. In each episode, an agent selects an available action $a \in A$ to have interaction with the environment and acquires a reward $R(s,a)$ after every decision interval. The agent attempts to obtain maximum reward expectation from the current state to the planning horizon H. The expectation of the state is a state-value function, estimated by the neural network. The neural network is constantly updated as it is trained to estimate the expected value of the cumulative reward under the current policy π.

$$V^{\pi}(s) = E_{\pi}\left[\sum_{t=i}^{H} \gamma^{t} r_t | s_0 = s\right] \tag{2}$$

DRL is an algorithm for optimising policy π. It reinforces the new policy to produce actions that exceed the desired value of the old policy. During the training of the DRL algorithm, the parameters of the deep neural network are continuously updated. As a result, selected actions will more likely become actions with greater cumulative rewards. Over the past decade, DRL has shown mastery in a variety of situations, such as playing table tennis [29] and fine manipulation skills [30]. However, achieving such high performance with DRL algorithms often requires long-term training. We need to focus not only on the robot's ability to adapt to different tasks, but also on time efficiency in order to reduce production cycles. Therefore, we propose meta-RL to address these challenges and improve on the inefficient sampling and lengthy training deficiencies of current traditional DRL algorithms through a meta-learning mechanism. Essentially, the powerful learning capability and fast adaptation to unseen data (tasks) is achieved by a meta-learning mechanism. This learning mechanism consists of two rounds of learning: a task-oriented adaptation phase and a meta-policy learning phase (meta-policy learning) [31], respectively. The basic mechanism of meta-RL is shown in the Figure 1.



1. The diagram of meta-RL learning process.

IV. Methodology

In this work, the meta-learning algorithm, MAML, is integrated into HRC's working paradigm. MAML enables collaborative robots to quickly learn and perform ancillary manufacturing tasks for human workers in the HRC system. Specifically, the meta-learning algorithm generates learnable strategies for collaborative work, and the learnable strategies enable the robot to quickly learn the ancillary manufacturing tasks in the HRC process.

As for the learnable meta-policy, it is used as an optimally adaptable initial policy that can be effectively adapted to the new task. The meta-policy generated by MAML is not an off-the-shelf policy for any task, but a policy that can be effectively adapted to the best control policy for the new task. In this approach, three phases are included, as shown in the Figure 1. (1) a task-specific learning phase, where a number of robotic tasks in the HRC are sampled and only a few iterations of the meta-policy are required to learn the corresponding number of policies; (2) a meta-policy learning phase, where the meta-policy is updated with the sampled task data for each specific task from the previous phase. (3) a generalization phase for the new task, in which several iterations are performed to obtain the best control policy after initializing the meta-policy parameters. In the next subsections, we describe how each of these three distinct learning phases enables agile learning of the robotic system.

A. Task-specific Learning Phase

In the task-specific learning phase, the learning process is the same as in traditional DRL, where the PPO algorithm is used to learn potential auxiliary HRC tasks. When the PPO algorithm is used to learn a specific manufacturing task, the agent can interact with the environment, sample new data (as shown in section III), and store them in a pool of experience for further evolution. After learning all potential manufacturing tasks, all collected data are fed to the next learning stage. The PPO DRL algorithm is selected due to its stable performance and recognition by DRL researchers

and institutions. In this stage of the learning process, the parameters of the training strategy are initialised according to a meta-strategy. In this phase, the task-specific objective function is represented as the Formula 3.

$$\theta_i = \underset{\theta'}{\operatorname{argmax}} \mathscr{L}\left(\theta', \mathscr{T}_i\right)\Big|_{\theta_{\text{init}}=\theta} \tag{3}$$

where, the θ represents the parameters of PPO algorithm. It is initialized by meta-policy and $\mathscr{T}_i$ is the vector represent task i. As for the mathematical details of PPO, it can be accessed on Schulman's article[8].

B. Meta-policy Learning Phase

During training, the meta-policy aggregates the sampled data from the previous phase to optimise generalisation. The meta-policy learning phase aims to optimise and obtain the best initialisation weights for the neural network in the DRL model. With the data collected in the previous phase, the PPO algorithm can calculate the gradients to optimise the policy for each task. With all the gradients, we can derive the total average gradient of the meta-policy. As the average gradient is updated, the meta-policy approaches the optimal policy for each task and can further be adapted to similar tasks without much tuning.

$$\theta = \underset{\theta'}{\operatorname{argmax}} \sum_i \mathscr{L}\left(\theta', \mathscr{T}_i\right)\Big|_{\theta_{init}=\theta} \tag{4}$$

C. Meta-policy Generalizing Phase

Through the first two phases of learning, a meta-policy trained in preparation acquires and possesses the learnable ability to adapt to new manufacturing tasks. In the personalised manufacturing system, as new orders are obtained, the corresponding new robot-adapted tasks in the HRC system are sampled. After the working goals of the HRC system have been determined, the robot control policy is initialised with a meta-policy and updated with a gradient descent method during the new task learning phase. The trajectory generated during the task-specific learning phase is accompanied and repeated until the strategy remains stable. The optimal control policy for the new task can be flexibly fine-tuned in a few steps without costing time and money. The meta-policy is initialised arbitrarily at the beginning. A detailed MAML-based RL algorithm is provided in the Algorithm 1 [7].

V. Experiment

Before proceeding with the experiment, it is recommended that the following two questions be asked:

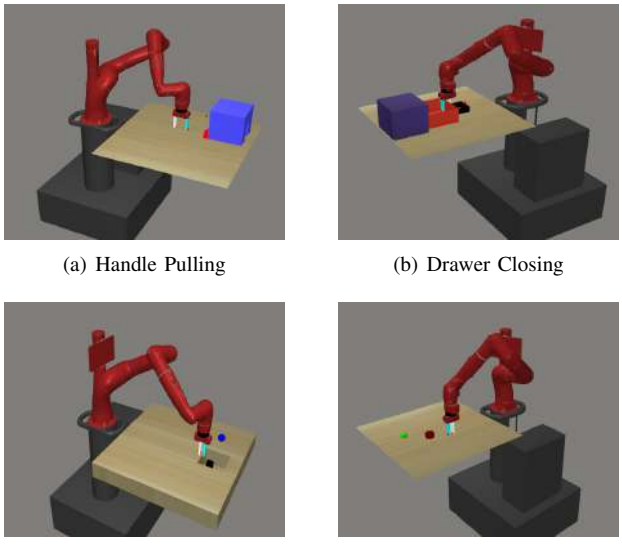
- Could the DRL-based solution improve the learning capability of the robotic system in SLRAS and enable the robots to perform robustly to unseen HRC robot tasks?
- Compared to train-from-scratch DRL, does the proposed meta-RL solution perform better in terms of training time and reward?

Algorithm 1 MAML - Reinforcement Learning

Require: :
α,β learning rate
Initialize θ arbitrarily for meta-policy
while not done **do**
  Sample batch of tasks $\mathscr{T}_i$ from all tasks
  for all $\mathscr{T}_i$ **do**
    Sample trajectories $\mathscr{D} = \{(\mathbf{s}_1, \mathbf{a}_1, \ldots \mathbf{s}_H)\}$ using θ in $\mathscr{T}_i$
    Evaluate $\nabla_\theta \mathscr{L}_{\mathscr{T}_i}(\theta)$ by $\mathscr{D}$ and $\mathscr{L}_{\mathscr{T}_i}$ in Equation 3
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathscr{L}_{\mathscr{T}_i}(\theta)$
    Sample trajectories $\mathscr{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, \ldots \mathbf{x}_H)\}$ using θ_i' in $\mathscr{T}_i$
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathscr{T}_i} \mathscr{L}_{\mathscr{T}_i}(\theta_i')$ using each $\mathscr{D}_i'$ and $\mathscr{L}_{\mathscr{T}_i}$ in Equation 4

A. Environments

In the experiments, to validate the research questions discussed, 12 robot control tasks were randomly selected, which were extracted from similar HRC tasks in meta-world [32]. In the task set, 10 of the tasks were selected as pre-training tasks and the remaining two tasks were set as evaluation tasks. In the ablation experiments, the ab initio trained PPO was chosen as the baseline. Regarding the experimental results, we compared our method (i.e. MAML-based PPO) with the baseline PPO in terms of success rate and reward.



(a) Handle Pulling (b) Drawer Closing

(c) Slider Sweeping (d) Slider Pushing

2. The four scenarios in evaluating experiments.

As we stated in our experiments, we took 10 training tasks at the beginning to train for the meta-policy. This process corresponds to the first two phases of our method. As for the last stage, the obtained meta-policy was evaluated on 2

I. Experiment Settings

Parameter	Value
PPO epochs	3
PPO clip ratio	0.1
Inner learning rate	0.01
Outer learning rate	0.01
Training adapt_steps	1
Validation adapt_steps	5

pre-trained tasks and 2 newly sampled tasks. In the task-specific learning phase, 10 tasks were first sampled and trained, and these tasks can be classified into two types as follows. The first type is the manipulation of different objects (handles, sticks, mugs, etc.) where the manipulator learns to drag, pull, sweep, grasp, etc. The goal is mainly to move the target object to a specified position. The second type attempts to complete a specific practical use task, such as opening a box, closing a drawer, pressing a button, etc. In our evaluation experiments, the tasks were sampled from the pre-training set, i.e. pushing the slider and pulling the handle. The unseen sampled tasks were closing the drawer and sweeping the slider into the bin. The simulation environment is demonstrated in Figure 2.

In this simulated benchmark [32], the state of the environment is returned by the simulator as a multidimensional vector and the action space is the degrees of freedom of the robot system. The reward function is first determined by the L2 distance between the robot's end-effector and the position of the target object. In addition to the distance, the reward is also determined by the state of the target object. Among the various components of the reward, the final reward is determined jointly by those components with different scales.

B. Experiment Settings

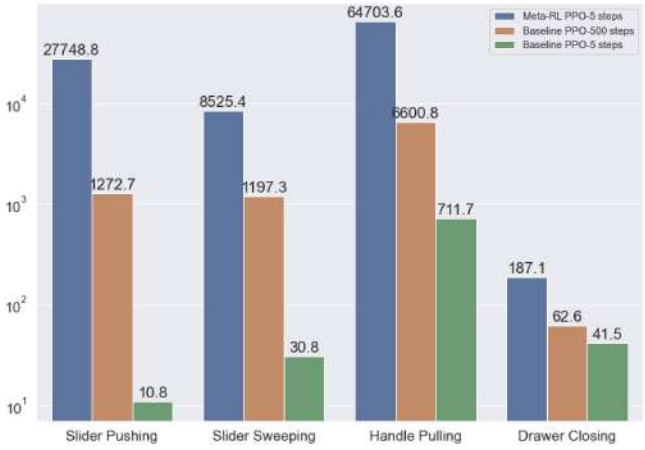
In our experiments, the DRL algorithm is optimised by a multi-layer neural network consisting of two fully connected layers with 100 hidden units in each layer. The activation function is non-linear and ADAM is used as the optimiser [33]. The detailed algorithm training settings are listed in Table I.

The experiments were executed on a CPU Intel Core i5-5200U 2-Core Processor 2.20GHz and each iteration took an average of 280s.

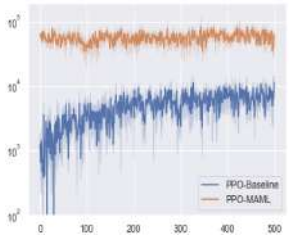
C. Results and Analysis

In this subsection, the performance obtained by the proposed method and the baseline PPO on the four evaluation tasks is presented and compared. The policy of the baseline PPO is initialised randomly and our approach is similar to the baseline PPO in terms of initialisation in the experiments. However, the MAML-based PPO differs from the baseline in that a meta-policy is used as the initialisation for learning these evaluation tasks, and the training of the meta-policy was mentioned in the previous section.

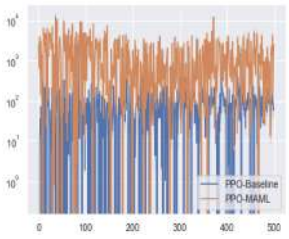
In the evaluation, the performance of the two algorithms is analysed separately in terms of success rate and reward. To better see the curves in Figure 4, the values on the vertical axis are logarithmic and averaged through a moving window of 10 elements in length. Although the two metrics are correlated, the value of the reward can also provide some advantage of the policy in difficult tasks when the task is not completed within the specified horizon.



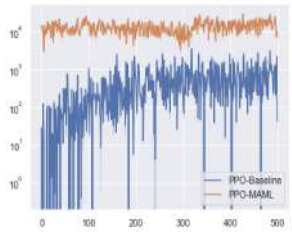
3. The reward gained with MAML-based PPO and baseline PPO in evaluating scenarios with different learning periods.



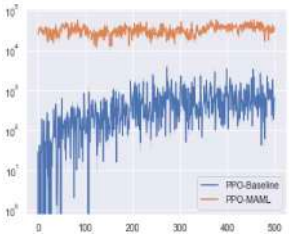
(a) Handle Pulling



(b) Drawer Closing



(c) Slider Sweeping



(d) Slider Pushing

4. The reward curves of evaluating experiments. Higher reward value means better performance.

Based on the success rates and rewards, it can be concluded that our proposed MAML-based PPO can achieve better performance than the baseline PPO in 5 and 500 steps of adaptation when evaluating task scenarios, especially in terms of success rate and learning speed. These results also respond to the expected research questions illustrated at the beginning of this section. As Table II shows, our method also achieves a respectable success rate, but the baseline PPO only achieves a limited improvement in the same period. Compared to the 500-step PPO baseline, the success rate of the MAML-based 5-step PPO improves by about 75%

II. Success Rate of Evaluating Tasks

Policy	Slider Pushing	Slider Sweeping	Handle Pulling	Drawer Closing
Baseline-PPO 5 steps	0%	0%	0%	0%
Baseline-PPO 500 steps	30%	66%	0%	0%
MAML-PPO 5 steps	100%	100%	100%	86.7%

on average. In conclusion, the meta-learning mechanism MAML allows the PPO algorithm to be flexibly adapted to new control tasks. In particular, with the help of the meta-learning mechanism, the robot was able to acquire self-learning capabilities and to improve the application of HRC in personalised production systems, while aligning with the goals of SLRAS at the same stage.

VI. Case Study

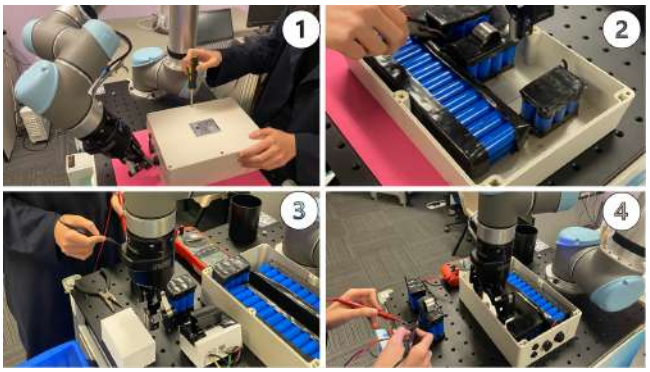
In this section, a case study of HRC disassembly of a retired battery is presented to better demonstrate the feasibility and efficiency of the SLRAS approach in personalised production. Assuming that all target locations are detected and known, a 6-axis cobot is employed to perform the auxiliary robot control task.

Efficient dismantling procedures are urgently needed as the traditional methods of mechanical recycling and chemical decomposition of used batteries are environmentally unfriendly and lead to wasted resources. In this process, although battery dismantling procedures are mostly similar, conventional autonomous robots lack the ability to learn and need to be reprogrammed for even small changes, creating time and cost issues. The battery removal process cannot be fully automated due to the different makes and models of batteries and their state of age. Faced with such small batches and customised dismantling tasks, HRC has become a worthwhile alternative to manual dismantling or robotic dismantling that requires repetitive programming.

As shown in Figure 5, HRC allows humans to concentrate on delicate tasks such as cutting wires and dealing with unexpected events, while SLRAS can quickly adapt and adjust its procedures to new situations, reducing time and costs for each specific case and assisting the human operator with the task. The following 4 separate scenarios can be referred to:

- In Figure 5.1, the SLRAS robot implements meta-learning based on how to press a button to observe the state of the battery while a human unscrews the battery box, which is consistent with the first simulation experiment.
- In Figure 5.2, the human operator cuts the cable to the battery and the robot learns to remove the battery from the box. The robot's behaviour is similar to the second simulation experiment.
- In Figure 5.3, after the human operator diagnoses the recycling value of the parts in the battery, the robot eventually sweeps the disassembled parts into the relevant containers according to the classification. The control strategy of SLRAS is the same as in the third simulation experiment.
- In Figure 5.4, the robot is used to push new items to the operator and the strategy implemented is the same as in the final simulation experiment.

From the above formulation, it appears that SLRAS assumes different responsibilities for different batteries and different task sequences in HRC, and our proposed meta-RL control method provides a viable solution for these cases.



5. Case Study of HRC for aging battery disassembly

VII. Conclusions and Future Work

In this work, the DRL algorithm based on a meta-learning mechanism was expected to improve the learning capability of robots in HRC for personalised production. MAML-PPO was used to obtain meta-strategies to improve learning efficiency, allowing the robot to adapt to the evaluation task in several training iterations. As a result, SLRAS gained agile learning capabilities and outperformed RL train-from-scratch training with faster adaptation. Furthermore, the experimental results revealed that the proposed approach has greater flexibility and learning capability and could be adopted in future HRCs for personalised production.

Despite the above contributions, there are still some limitations inadequately addressing the agile learning capabilities of the robot, such as the simplification of the model and simulation environment. Therefore, in the near future, potential research could be explored to 1) manipulate robots in more complex task scenarios to ensure their performance and 2) transfer control strategies obtained in simulators to physical robots without any performance degradation.

References

[1] Yoram Koren. *The global manufacturing revolution: product-process-business integration and reconfigurable systems*. Vol. 80. John Wiley & Sons, 2010.

[2] Daqiang Guo et al. “A framework for personalized production based on digital twin, blockchain and additive manufacturing in the context of Industry 4.0”. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2020, pp. 1181–1186.

[3] Kyu Tae Park et al. “Digital twin-based cyber physical production system architectural framework for personalized production”. In: *The International Journal of Advanced Manufacturing Technology* 106.5 (2020), pp. 1787–1810.

[4] Dario Antonelli, Sergey Astanin, and Giulia Bruno. “Applicability of human-robot collaboration to small batch production”. In: *Working Conference on Virtual Enterprises*. Springer. 2016, pp. 24–32.

[5] Anil Kumar Inkulu et al. “Challenges and opportunities in human robot collaboration context of Industry 4.0-a state of the art review”. In: *Industrial Robot: the international journal of robotics research and application* (2021).

[6] Giovanni Di Orio, Gonçalo Cândido, and José Barata. “Self-learning production systems: A new production paradigm”. In: *Sustainable Design and Manufacturing* (2014), pp. 887–898.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[8] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[9] Emanuel Todorov, Tom Erez, and Yuval Tassa. “Mujoco: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[10] Keliang Zhou, Taigang Liu, and Lifeng Zhou. “Industry 4.0: Towards future industrial opportunities and challenges”. In: *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. IEEE. 2015, pp. 2147–2152.

[11] Giovanni Di Orio, Gonçalo Cândido, and José Barata. “The Adapter module: A building block for self-learning production systems”. In: *Robotics and Computer-Integrated Manufacturing* 36 (2015), pp. 25–35.

[12] Benjamin Lindemann et al. “Cloud-based control approach in discrete manufacturing using a self-learning architecture”. In: *IFAC-PapersOnLine* 51.10 (2018), pp. 163–168.

[13] Jay Lee et al. “Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing”. In: *IET Collaborative Intelligent Manufacturing* 2.1 (2020), pp. 34–36.

[14] Peter Schlegel, Kristof Briele, and Robert H Schmitt. “Autonomous data-driven quality control in self-learning production systems”. In: *Congress of the German Academic Association for Production Technology*. Springer. 2018, pp. 679–689.

[15] Jose Luis Outón et al. “Innovative mobile manipulator solution for modern flexible manufacturing processes”. In: *Sensors* 19.24 (2019), p. 5414.

[16] SM Mizanoor Rahman and Yue Wang. “Mutual trust-based subtask allocation for human–robot collaboration in flexible lightweight assembly in manufacturing”. In: *Mechatronics* 54 (2018), pp. 94–109.

[17] Eunseo Lee et al. “A Study on Human-Robot Collaboration based Hybrid Assembly System for Flexible Manufacturing”. In: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1. IEEE. 2019, pp. 4197–4202.

[18] Valentin Robu, David Flynn, and David Lane. “Train robots to self-certify their safe operation”. In: *Nature* 553.281 (2018).

[19] Kin Lok Keung et al. “Cloud-based cyber-physical robotic mobile fulfillment systems: a case study of collision avoidance”. In: *IEEE Access* 8 (2020), pp. 89318–89336.

[20] Afonso Castro, Filipe Silva, and Vitor Santos. “Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics”. In: *Sensors* 21 (June 2021), p. 4113. DOI: 10.3390/s21124113.

[21] Dorothea Schwung et al. “Self learning in flexible manufacturing units: a reinforcement learning approach”. In: *2018 International Conference on Intelligent Systems (IS)*. IEEE. 2018, pp. 31–38.

[22] Mohamed El-Shamouty et al. “Towards Safe Human-Robot Collaboration Using Deep Reinforcement Learning”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4899–4905.

[23] Shen Zhang et al. “Few-Shot Bearing Anomaly Detection Based on Model-Agnostic Meta-Learning”. In: *arXiv preprint arXiv:2007.12851* (2020).

[24] Qinge Xiao et al. “Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations”. In: *IEEE Transactions on Automation Science and Engineering* (2019).

[25] Gerrit Schoettler et al. “Meta-Reinforcement Learning for Robotic Industrial Insertion Tasks”. In: *arXiv preprint arXiv:2004.14404* (2020).

[26] Yan Duan et al. *RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning*. 2016. arXiv: 1611.02779 [cs.AI].

[27] Marcin Andrychowicz et al. “Learning to learn by gradient descent by gradient descent”. In: *arXiv preprint arXiv:1606.04474* (2016).

[28] Erin Grant et al. “Recasting gradient-based meta-learning as hierarchical bayes”. In: *arXiv preprint arXiv:1801.08930* (2018).

[29] Jan Peters, Katharina Mulling, and Yasemin Altun. “Relative entropy policy search”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 24. 1. 2010.

[30] Hai Nguyen and Hung La. “Review of deep reinforcement learning for robot manipulation”. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, pp. 590–595.

[31] Mathew Botvinick et al. “Reinforcement learning, fast and slow”. In: *Trends in cognitive sciences* (2019).

[32] Tianhe Yu et al. “Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2020, pp. 1094–1100.

[33] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).