Volume 11, Number 1, 2025 53

# Analysis of Methods for Training Robotic Manipulators to Perform Complex Motion Trajectories

Yurii Senchuk*, Fedir Matiko

*Lviv Polytechnic National University, 12 Stepana Bandery St., Lviv, 79013, Ukraine*

Received: April 16, 2025. Revised: June 23, 2025. Accepted: June 27, 2025.



**Abstract**

The article examines current approaches to training robotic manipulators for executing complex tasks in dynamic and changing environments. It provides a comparative analysis of modern training methods, highlighting their advantages and disadvantages. Additionally, the paper outlines the typical areas in which these methods are applied. Particular attention is given to approaches that involve human instructors, self-learning, and reinforcement learning. Special emphasis is placed on training efficiency, robot adaptability to new conditions, human-robot interaction, and the transfer of skills from virtual training environments to the real world. Based on the analysis, the authors recommend imitation learning — specifically, the learning from demonstration approach — as it enables the rapid and safe transfer of skills from humans to robots without the need for task formalization. The article also highlights the challenges of adapting trained models to real-world conditions and ensuring effective human-robot collaboration. It identifies key challenges faced by modern robot training systems. Based on these challenges, the article offers recommendations for selecting optimal training strategies according to the specific task type and available resources.

**Keywords:** robotics; robotic manipulators; teaching methods; adaptability.

## 1. Definition of the problem to be solved

In the context of today's rapid advancements in robotics and industrial automation, there is a growing demand for robotic manipulators to perform tasks that require high accuracy, adaptability, and autonomy. At the same time, traditional programming methods that rely on hard-coded scripts and predefined trajectories often do not permit rapid adaptation of the system to environmental changes. Additionally, these methods require considerable time and highly skilled personnel.

The challenge of teaching robotic manipulators to follow complex trajectories is particularly relevant today. These trajectories often involve multiple stages, interactions with various objects, and operation in dynamic or partially uncertain environments. One promising direction for solving this issue is the use of intelligent learning methods. These methods include approaches based on environmental interaction, action demonstration, and adaptive generalization of prior experience. However, the limitations, capabilities, and suitability of these methods for complex trajectory formation tasks still require further analysis and comparison.

Therefore, there is a need for an approach to training robotic manipulators that ensures rapid learning of new motion paths. This approach should minimize dependence on manual programming, allow adaptation to environmental changes, and reduce implementation time and cost. This need is particularly important for next-generation flexible robotic systems, where self-learning capabilities are a key performance factor.

* Corresponding author. Email address: yurii.m.senchuk@lpnu.ua

**This paper should be cited as:** Y. Senchuk, F. Matiko. (2025) Analysis of methods for training robotic manipulators to perform complex motion trajectories. *Energy Engineering and Control Systems*, Vol. 11, No. 1, pp. 53 – 61. https://doi.org/10.23939/jeecs2025.01.053

### 2. Analysis of the recent publications and research works on the problem

Recent research in the field of robotics demonstrates a growing interest in intelligent methods for training robotic manipulators. The objective is to enhance system flexibility and adaptability while reducing reliance on manual programming.

Papers [1]–[3] provide an overview of imitation learning methods based on demonstrations, along with practical considerations for implementing these systems in industrial settings. Articles [6], [7] introduce the fundamentals of reinforcement learning, while studies [9], [10] focus on deep learning algorithms designed for continuous monitoring. The inverse reinforcement learning approach — which enables the recovery of expert goals from observational data, is presented in [11]. The implementation of collaborative learning and the organization of agent interactions in real-world industrial environments are discussed in [12]–[14].

A number of publications [15]–[22] explore alternative approaches that demonstrate potential for reducing the need for training data and enhancing adaptability. However, these methods remain insufficiently studied in the context of using robotic manipulators for training, particularly for tasks involving the reproduction of complex trajectories in dynamic environments.

Despite significant theoretical progress, the practical application of these methods — especially in reproducing complex trajectories in dynamic environments — remains an open challenge. This underscores the need for further analysis of their effectiveness in relation to the demands of the requirements of modern robotic production.

### 3. Formulation of the goal of the paper

The aim of this article is to review and compare modern methods for training robotic manipulators. It seeks to identify their advantages and limitations and to develop recommendations for their application based on the types of tasks being addressed. In particular, the authors focus on analyzing the use of current approaches in imitation learning, reinforcement learning, collaborative learning, self-learning, and hybrid strategies. These methods are examined in the context of training robotic manipulators to execute complex motion trajectories.

### 4. Analysis of methods for teaching robot manipulators

The following section provides an overview of current approaches aligned with the stated objective. It focuses on investigating the capabilities of modern learning methods to reproduce complex motion trajectories. Special attention is given to methods that enable robotic manipulators to acquire skills with minimal human intervention.

**Imitation learning** refers to the transfer of skills from a human to a robot through the demonstration of required actions. As noted in [1], during imitation training, a robotic manipulator learns to perform tasks by replicating the movements of an expert — either a human or another agent — who has demonstrated the actions in advance. Demonstrations may be provided in various forms, including direct control (e.g., via a joystick), physical guidance of the manipulator (kinesthetic teaching), or observation through video or sensor data.

This approach enables the acquisition of skills without the need for a formal algorithmic description. Consequently, it significantly reduces the level of user expertise required and simplifies the process of programming the robot. This technique is particularly effective in scenarios where formulating an optimal action strategy is difficult, yet demonstrating the desired behavior is relatively straightforward.

The primary advantage of this method is its ability to enable rapid skill acquisition by a robot through direct demonstration. This significantly reduces the time required for system training compared to alternative approaches. The method is particularly effective for tasks in which constructing an optimal algorithm and formalizing its individual steps is challenging. For example, robots can learn manipulation tasks that require fine motor skills by observing human actions. This approach minimizes the likelihood of errors during the initial stages of system operation, as the learning process is based on demonstrated and validated behaviors [2].

One of the key challenges associated with this approach is its limited generalization capability. A robot may fail to handle situations that differ from those encountered during training. Addressing this limitation often requires additional demonstrations or integration with other learning methods. Furthermore, the effectiveness of the approach largely depends on the quality and precision of the expert's demonstrated actions.

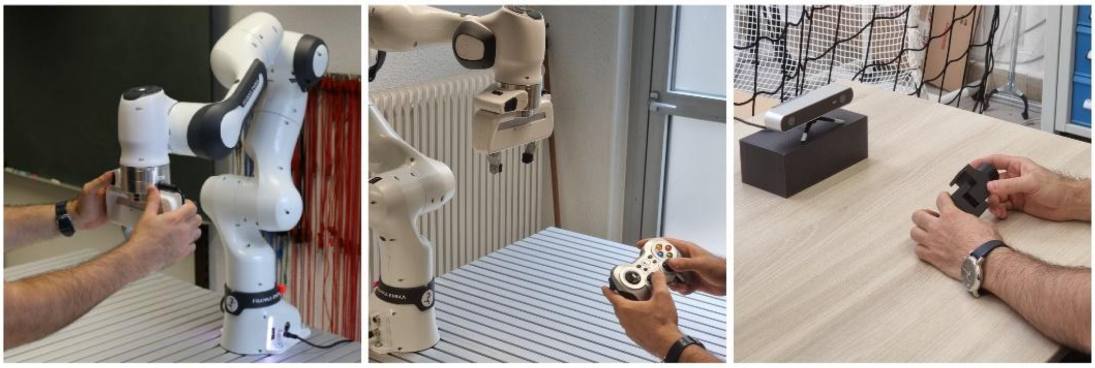
Depending on the mode of interaction between the operator and the robot, the demonstration of the desired behavior can be carried out using different approaches. The most common of these include:

a) **Kinesthetic teaching**, in which the operator physically guides the manipulator, and the system records the movements using its internal sensors.

b) **Teleoperation**, where the robot is operated remotely through specialized interfaces such as joysticks, motion trackers, tactile gloves, or external manipulators.

c) **Passive observation**, where information about the operator's actions is obtained through external sensors — such as video cameras — without direct physical contact with the robot.

The classification of demonstration methods proposed in [3] is relevant for industrial environments. In these settings, factors such as integration flexibility, minimal disruption to existing workflows, and safety are critically important. The second and third approaches — telerobotic control and passive observation — are especially practical for industrial applications, as they enable robot training in scenarios where direct physical contact is either undesirable or technically constrained.



(a) Kinesthetic Teaching (b) Teleoperation (c) Passive Observation

Fig.1. Examples of approaches to demonstrating desired actions during robot training [3].

Among the most commonly used imitation learning algorithms are Behavioral Cloning (BC) and Dataset Aggregation (DAgger). Behavioral Cloning treats the learning process as a classical supervised learning task with a teacher. In contrast, DAgger addresses the issue of error accumulation that is typical of the standard Behavioral Cloning method.

According to [4], BC is a fundamental approach in which a training dataset is constructed from expert demonstrations in the form of situation–action pairs. These pairs consist of input data from sensors and the corresponding actions performed by the operator. Based on this data, a model is trained to replicate the expert's behavior. The algorithm is relatively simple to implement and can produce quick results, particularly when the demonstrations are of high quality and sufficiently cover the range of possible behaviors.

However, a significant limitation of this approach is the phenomenon known as **covariate shift**. Even a slight deviation of the robot from the demonstrated trajectory may cause the system to enter a state that is not represented in the training data. In such cases, the robot is unable to determine the correct course of action.

To address this limitation, the DAgger algorithm was introduced in 2011 [5]. It employs an iterative learning process in which the robot performs actions autonomously, while an expert provides optimal action suggestions for each encountered state. As a result, a new dataset is generated that includes situations the robot experiences during its execution. This data is incrementally added to the initial training set, which significantly expands the coverage of possible states and reduces the impact of error accumulation during execution.

This approach substantially improves the robot's behavioral stability and its ability to operate in previously unencountered situations. However, DAgger requires the continuous involvement of an expert at each iteration, which can be burdensome in real-world applications — particularly when real-time precision is necessary.

**Reinforcement Learning (RL)** is based on the principle that a system learns through interactions with its environment, receiving rewards or penalties depending on the outcomes of its actions. The primary objective of this approach is to maximize cumulative reward, thereby encouraging the development of effective behavior [6]. As a result, the robot can gradually learn optimal actions to achieve its goals, even without prior knowledge of the environment.

Figure 2 illustrates the interaction scheme between the agent and the environment in the context of reinforcement learning for a robotic manipulator. Based on observations from the current state $S_t$, the agent generates an action $A_t$ and sends it to the environment. In response, the environment returns a new state $S_{t+1}$ and a reward $R_t$, which are then used to update the agent's action policy.

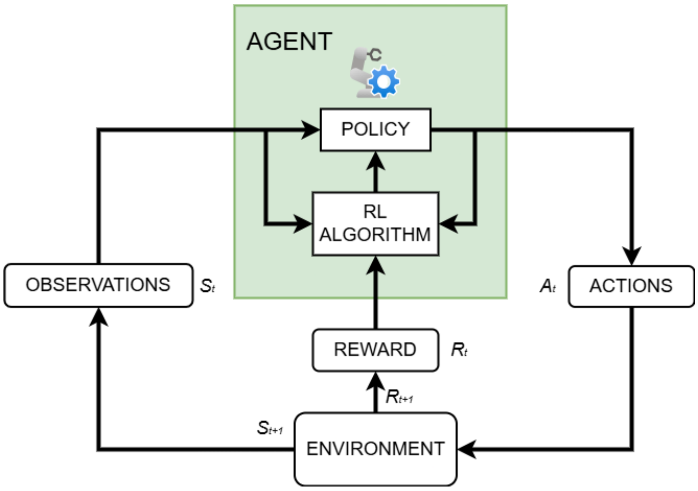



Fig.2. Scheme of interaction between the agent and the environment in the method of reinforcement learning of a robot manipulator.

The key advantages of RL include its capacity for autonomous learning, adaptability to new conditions, and the ability to operate without requiring demonstrations or pre-labeled data. However, the learning process typically demands a large number of episodes, substantial computational resources, and significant time. An additional challenge is ensuring the stability and convergence of RL algorithms, which often requires careful tuning of parameters [7].

Additionally, the risk of hardware damage during real-world experimentation must be considered. For this reason, RL is usually first implemented in simulation environments. However, this practice introduces the well-known problem of sim-to-real transfer — that is, the difficulty of transferring learned behaviors from simulation to physical systems [7].

Article [8] discusses the integration of reinforcement learning with deep neural networks, which led to the emergence of the field known as **Deep Reinforcement Learning**. This approach allows agents to operate in complex state spaces and manage continuous action domains.

The practical reinforcement learning algorithms include Proximal Policy Optimization (PPO) [9] and Soft Actor-Critic (SAC) [10]. PPO promotes stable learning by constraining abrupt changes in the robot's behavior during model updates, thereby increasing the reliability of the training process. In contrast, SAC is based on the principle of maximum entropy, encouraging the robot not only to perform actions with high precision but also to actively explore alternative behaviors. This enhances the system's ability to adapt to complex and unpredictable environmental conditions.

**Inverse Reinforcement Learning** (IRL) is a method that enables the training of a robot manipulator by observing the actions of an expert, without requiring a predefined reward function [11]. In contrast to classical Reinforcement Learning (RL), where the system receives explicit feedback indicating which actions are desirable, IRL seeks to deduce the expert's underlying objective solely from observed behavior.

This approach is particularly useful in situations where it is difficult or even impossible to explicitly formalize the desired behavior using a clear reward function. In such cases, manually defining all necessary criteria becomes a major challenge. For example, a human can easily demonstrate how to handle a fragile object with care. However, mathematically specifying all the relevant parameters for this task is extremely difficult. IRL allows the system to learn from such demonstrations. This reduces the need for manual reward function design and enables flexible reproduction of behavior, even under changing conditions.

The primary advantage of this method is its ability to enable learning without explicitly defining the objective. This is particularly valuable for tasks that depend on intuitive human experience. At the same time, Inverse Reinforcement Learning (IRL) has several significant limitations. First, multiple reward functions can explain the same observed behavior, leading to an ambiguity problem. Second, the implementation of IRL is computationally complex and often requires substantial computational resources. In addition, the quality of the results strongly depends on the accuracy and completeness of the expert demonstrations.

**Collaborative Learning (CL)** involves interaction among multiple robotic manipulators or between a robot and a human, with the aim of acquiring shared skills for performing tasks. The core concept of this method is that the experience gained by one agent during training can be transferred to others through knowledge exchange, observation of partners' actions, or direct interaction [12].

Such knowledge transfer can be organized in two ways: centrally, through a shared memory or control module, or in a decentralized manner, via message passing, signaling, or by interpreting the actions of other agents [12]. Additionally, CL often includes human–robot interaction, which enables more intuitive and comprehensible cooperation from the perspective of the human participant.

This approach is especially valuable for tasks that require coordinated actions. Examples include collective assembly, the manipulation of large or heavy objects, and scenarios that involve close interaction with a human operator.

In such cases, CL enhances efficiency by supporting role distribution, joint action planning, and the use of complementary capabilities among agents. Furthermore, human involvement contributes to a more intuitive and natural learning environment, which in turn enhances the overall performance of the system [13].

The advantages of the approach include accelerated learning through shared experience, a reduction in the number of required training episodes, and increased system robustness to environmental changes.

However, the implementation of CL presents several challenges. The most critical among them are the need for effective communication between agents, alignment on shared goals, prevention of action conflicts, and resolution of possible discrepancies in environmental perception or task interpretation.

One example of a modern approach to CL is the DEEPCOBOT project (Collective Efficient Deep Learning and Networked Control for Multiple Collaborative Robot Systems) [14]. This platform focuses on developing decentralized deep learning techniques for groups of collaborative robots that interact with one another and with human operators in real time.

A distinctive feature of the approach is local training performed by each agent, followed by knowledge exchange across the network without relying on a central server. This architecture supports scalability, fault tolerance, and accelerated collective learning. DEEPCOBOT robots are capable of adapting to changes in the production environment, coordinating their actions, and safely cooperating with human operators.

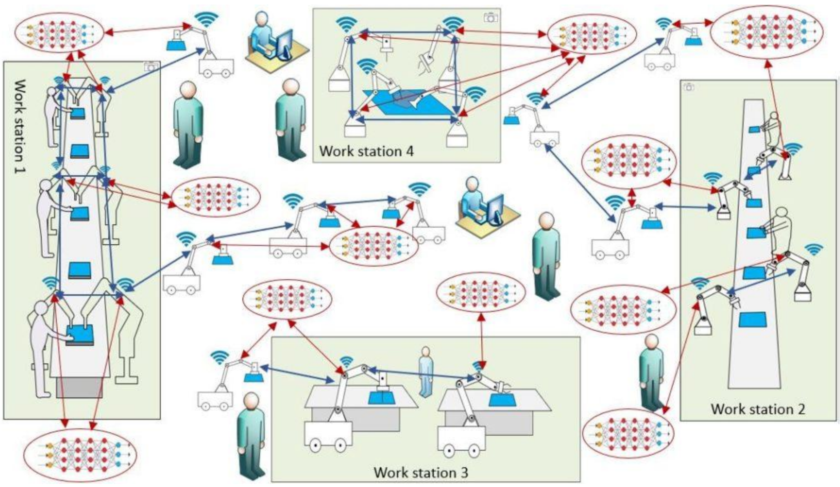



Fig.3. Collaborative interaction of robots in a production environment with decentralized deep learning [14].

Modern robotics tasks increasingly require systems that are not only accurate but also flexible, adaptable, and capable of rapid learning in complex and dynamic environments. No single training method can fully satisfy all of these requirements. As a result, there is growing interest in **combined** and **hybrid** approaches that integrate the strengths of multiple learning paradigms. These strategies for training robotic manipulators aim to enhance overall system efficiency, adaptability, and robustness. Such hybrid methods are currently being investigated in research and practical applications, particularly in environments characterized by high uncertainty and variability.

For example, in [15], the authors propose a combination of imitation learning and reinforcement learning. In this approach, the robot first acquires initial skills through expert demonstrations. Subsequent improvement is achieved through autonomous interaction with the environment using RL algorithms.

Another common strategy involves integrating classical control methods, such as PID controllers, with neural network models. This hybrid approach enables effective adaptation of robot behavior to changing environmental conditions [16].

The primary advantage of hybrid approaches lies in their ability to eliminate or mitigate the limitations of individual learning methods. These approaches can accelerate the training process, decrease computational requirements, and simultaneously provide the flexibility and adaptability needed for robotic manipulators to perform diverse tasks under varying operating conditions.

However, the use of combined methods presents specific challenges. One of the primary difficulties is selecting appropriate methods for integration and optimizing their interaction. Additionally, hybrid systems often result in increased architectural complexity, which demands a high level of expertise in system design and implementation.

Another promising and emerging area is **few-shot learning** and **unsupervised learning**, which are particularly relevant in environments where collecting large amounts of labeled data are complex, costly, or impractical. These methods allow robotic manipulators to adapt to new situations using only minimal prior information or without any explicit supervision.

In the case of few-shot learning, a robot can acquire a new skill based on only a few demonstrations [17]. This capability is enabled by meta-learning approaches [18], which allow systems to "learn how to learn" — that is, to leverage prior experience from previously solved tasks in order to rapidly adapt to new ones. This method is particularly effective in scenarios where quick adaptation is required, for example, when learning novel actions or interacting with unfamiliar objects that the system has not previously encountered.

Unsupervised learning, as described in [19] and [20], enables a system to learn the structure of its environment without relying on pre-labeled examples. Instead of using manually prepared data, the robot analyzes raw observations to identify patterns, group similar states, or perform clustering. These capabilities form a foundation for subsequent goal-directed learning. As a result, unsupervised learning is considered a promising approach for developing autonomous robotic systems. Such systems are capable of exploring their environment independently and gradually improving their behavior over time.

Despite their significant potential, these methods still face several challenges. Few-shot learning requires complex models that can rapidly adapt with minimal data. Designing and training such models remains a non-trivial task. On the other hand, unsupervised learning offers limited mechanisms for evaluating and controlling the quality of learning, especially in the absence of clearly defined goals or external feedback. These limitations hinder its direct application in tasks that require high levels of reliability and precision.

One of the most important aspects in the context of training robotic manipulators is the use of **simulation environments** for preliminary training. Simulations help reduce the consumption of physical resources and prevent potential equipment damage during the early stages of learning.

Thanks to advanced platforms such as Gazebo, MuJoCo, and PyBullet, researchers can conduct experiments in a virtual space. This significantly accelerates the development and testing of new algorithms and training approaches. Moreover, high-fidelity simulations that accurately model the physical properties of the real world facilitate a smooth transition from virtual to physical deployment without compromising the effectiveness of the trained model.

However, a persistent challenge is the so-called "reality gap" — the discrepancy between the simulated and real environments. This gap can lead to reduced accuracy or instability in the robot’s performance when transferring trained behaviors to real-world conditions [21].

To address this issue, researchers apply **transfer learning** techniques, as described in [22]. These methods aim to preserve learning efficiency when moving from a virtual to a physical space. The main advantages of transfer learning include reduced training time and resource consumption, fewer required demonstrations or training episodes, and improved system adaptability to new conditions.

Nonetheless, transfer learning is not without limitations. If there is a substantial mismatch between the simulated and real tasks, the system may experience negative transfer, which can degrade performance or even lead to failure [22].

A comparative analysis shows that no single method is universally applicable. Each approach has its strengths and limitations, depending on the specific context.

For example, imitation learning enables the rapid acquisition of basic skills but often demonstrates limited generalization to new or unforeseen situations. In contrast, reinforcement learning offers flexibility and autonomy, but it requires substantial computational resources and time. Inverse reinforcement learning provides a deeper understanding of the expert's intentions, but it is complex to implement and computationally demanding. Combined and collaborative approaches deserve particular attention. They enable the integration of complementary advantages from multiple learning paradigms, supporting the development of more robust and adaptive robotic systems.

Ultimately, the choice of an optimal learning strategy should be guided by several factors. These include the nature of the task, environmental complexity, the availability of training data, and technical constraints. A brief comparative overview of the discussed methods is presented in Table 1.

Table 1. Brief comparative characteristics of teaching methods.

| Method | Advantages | Disadvantages | Typical scenarios of possible application |
|---|---|---|---|
| Imitation Learning | 1) Fast learning without complex setup<br>2) No reward function required<br>3) Suitable for clear demonstrations<br>4) Low risk of hardware damage | 1) Poor generalization to new situations<br>2) Requires many high-quality demonstrations<br>3) Hard to teach complex or error-prone behaviors | 1) Simple motions and action sequences<br>2) Industrial tasks in controlled settings<br>3) Social robotics with human-like behavior |
| Reinforcement Learning | 1) Learns through trial and error<br>2) No demonstrations or labels needed<br>3) Works well in complex, dynamic environments<br>4) Optimizes long-term strategies. | 1) High computational cost and long training<br>2) Requires many training episodes<br>3) Risk of hardware damage in real-world training<br>4) Sim-to-real transfer issues (reality gap) | 1) Autonomous learning in dynamic settings<br>2) Real-time control optimization<br>3) Complex behaviors hard to program manually |
| Inverse Reinforcement Learning | 1) Learns expert's hidden goals<br>2) Useful when reward is hard to define<br>3) Flexible generalization to new situations | 1) High complexity and computational cost<br>2) Reward ambiguity for the same behavior<br>3) Requires many expert demonstrations | 1) Intuitive tasks hard to formalize<br>2) Manual or medical skills (e.g., surgery) |
| Collaborative Learning | 1) Faster learning via collaboration or knowledge sharing<br>2) Enables teamwork and human–robot interaction<br>3) Task distribution reduces individual agent load | 1) Complex coordination between agents<br>2) Requires communication protocols<br>3) Risk of goal or action conflicts | 1) Multi-robot collaboration in cooperative systems<br>2) Human–robot cooperation<br>3) Joint manipulation of large or complex objects |
| Hybrid / Combined Learning | 1) Combines strengths of multiple methods<br>2) Flexible and scalable solutions<br>3) Improved stability for complex tasks | 1) Complex implementation and integration<br>2) Requires deep customization and testing<br>3) Higher hardware and software demands | 1) Error correction and policy refinement<br>2) Low-data or high-risk domains |
| Transfer Learning | 1) Reduces training time and data requirements<br>2) Reuses prior knowledge for new tasks<br>3) Effective for sim-to-real transfer | 1) Risk of negative transfer with dissimilar tasks<br>2) Requires good task alignment<br>3) May need model architecture adaptation | 1) Transferring grasping skills across objects<br>2) Retraining after sim-to-real deployment |

## 5. Conclusion

Based on the conducted analysis, imitation learning is the most suitable choice for training robotic manipulators. In particular, training through demonstration proves effective in tasks where high-quality expert demonstrations are feasible.

This method enables fast and safe acquisition of basic skills, which is particularly important when working with robotic manipulators, as it minimizes the risk of equipment damage during experimentation. It is especially well suited for tasks involving well-defined trajectories or repetitive actions. In such cases, imitation learning significantly reduces the time and resources required at the training stage. This is particularly useful when an operator intuitively knows how to perform a task but is unable to formalize it in the form of a program.

Moreover, demonstration-based learning supports safe training in real-world environments and lowers computational demands compared to methods relying solely on autonomous exploration, such as reinforcement learning. These characteristics make imitation learning an optimal choice for training robotic manipulators in applications that require high levels of reliability and repeatability.

## References

[1] Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. Robotics and Autonomous Systems, 57(5), 469–483. https://doi.org/10.1016/j.robot.2008.10.024

[2] Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In Springer handbook of Robotics (pp. 1371-1394). Springer. https://doi.org/10.1007/978-3-540-30301-5_60

[3] Barekatain, A., Habibi, H., & Voos, H. (2024). A practical roadmap to learning from demonstration for robotic manipulators in manufacturing. Robotics, 13(3), 100. https://doi.org/10.3390/robotics13070100

[4] Underactuated Robotics. (n.d.). Ch. 21 – Imitation Learning. Retrieved from https://underactuated.mit.edu/imitation.html.

[5] Ross, S., Gordon, G., & Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. *AISTATS*. 2011. – P. 627–635.

[6] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd Edition)*. MIT Press.

[7] Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. International Journal of Robotics Research, 32(11), 1238–1274. https://doi.org/10.1177/0278364913495721

[8] Lillicrap, T. P., Hunt, J. J., Pritzel, A., et al. (2015). Continuous control with deep reinforcement learning.

[9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms.

[10] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.

[11] Ng, A., & Russell, S. (2000). Algorithms for inverse reinforcement learning. Proceedings of the 17th International Conference on Machine Learning (ICML), 663–670.

[12] Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems, 11(3), 387–434. https://doi.org/10.1007/s10458-005-2631-2

[13] Nikolaidis, S., & Shah, J. A. (2013). Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. ACM/IEEE International Conference on Human-Robot Interaction. https://doi.org/10.1109/HRI.2013.6483499

[14] DEEPCOBOT – Collective Efficient Deep Learning and Networked Control for Multiple Collaborative Robot Systems. UiA WISENET Lab. 2024. URL: https://deepcobot.uia.no/about

[15] Zhu, Y., Mottaghi, R., Kolve, E., et al. (2018). Reinforcement and imitation learning for diverse visuomotor skills. Proceedings of Robotics: Science and Systems (RSS), 33–40. https://doi.org/10.15607/RSS.2018.XIV.009

[16] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-End Training of Deep Visuomotor Policies. Journal of Machine Learning Research, 17(39), 1–40.

[17] Yaqing Wang, Quanming Yao, James Kwok, Lionel M. Ni (2019). Generalizing from a Few Examples: A Survey on Few-Shot Learning. Science, 53(3), 1–34. https://doi.org/10.1145/3386252

[18] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. Proceedings of the 34th International Conference on Machine Learning (ICML), 1126–1135.

[19] Naeem, S., Ali, A., & Anam, S. (2023). An unsupervised machine learning algorithms: Comprehensive review. International Journal of Computing and Digital Systems, 12(1), 1–10. https://doi.org/10.12785/ijcds/130172

[20] Hjelm, R. D., Fedorov, A., Lavoie, E., et al. (2019). Learning deep representations by mutual information estimation and maximization. International Conference on Learning Representations (ICLR).

[21] James, S., Davison, A. J., & Johns, E. (2019). "Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. https://doi.org/10.1109/CVPR.2019.01291

[22] Pan, S. J., & Yang, Q. (2010). "A Survey on Transfer Learning." IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

# Аналіз методів навчання роботів-маніпуляторів для виконання складних траєкторій руху

Юрій Сенчук, Федір Матіко

*Національний університет «Львівська політехніка», вул. С. Бандери, 12 , Львів, 79013, Україна*

**Анотація**

У статті розглянуто актуальні підходи до навчання роботів-маніпуляторів, які застосовуються для виконання складних завдань у динамічних та змінних умовах середовища. Проведено порівняльний аналіз сучасних методів, визначено їхні основні переваги, недоліки, а також окреслено типові сфери їхнього практичного застосування, зокрема методи із залученням людини-інструктора, самонавчання та навчання з підкріпленням. Особливу увагу приділено питанню ефективності навчання, адаптивності роботів до нових умов, взаємодії з людиною та перенесення навичок з віртуального навчального середовища у реальне. На основі аналізу рекомендованим визначено імітаційне навчання, зокрема підхід навчання за демонстрацією, що дозволяє швидко та безпечно передавати навички від людини до робота без необхідності формалізації завдань. Крім того, в статті акцентовано увагу на проблемах адаптації навчених моделей до реальних умов і взаємодії роботів із людиною. Визначено ключові виклики, що стоять перед сучасними системами навчання роботів та сформульовано рекомендації щодо вибору оптимальних стратегій навчання залежно від типу завдань і доступних ресурсів.

**Ключові слова:** робототехніка; робот-маніпулятор; система керування; методи навчання; траєкторія руху; адаптивність.