Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Visual and Force-Driven-Based Assembly Learning Using Collaborative Robots

Dissertation
with the aim of achieving a doctoral degree at the
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Universität Hamburg

submitted by
Yunlei Shi
Matrikelnr.: 7243875
01.2023

Day of oral defense:
31.03.2023

The following evaluators recommend the admission of the dissertation:

Supervisor:
Prof. Dr. Jianwei Zhang,
Department of Informatics,
Universität Hamburg, Germany

1st Reviewer:
Prof. Dr. Jianwei Zhang,
Department of Informatics,
Universität Hamburg, Germany

2nd Reviewer:
Prof. Dr. Stefan Wermter,
Department of Informatics,
Universität Hamburg, Germany

Chair:
Prof. Dr. Janick Edinger,
Department of Informatics,
Universität Hamburg, Germany

Deputy Chair:
Prof. Dr. Stefan Wermter,
Department of Informatics,
Universität Hamburg, Germany

Document identifier:
urn:nbn:de:gbv:18-ediss-108351

Abstract

Collaborative robots are expected to work alongside humans and directly replace human workers in some cases, thus effectively responding to rapid changes in assembly lines. Contact-rich manipulation tasks are commonly found in modern manufacturing settings. However, manually designing a robot controller is considered hard for traditional control methods as the controller requires an effective combination of modalities and vastly different characteristics. In this thesis, several visual and force-based creative skills and learning frameworks are proposed to solve the current issues of force-controlled robotic assembly tasks.

In this thesis, we first consider incorporating operational space visual and haptic information into a Reinforcement Learning (RL) framework to solve the target uncertainty problems in unstructured environments. Moreover, we propose a novel idea of introducing a proactive action to solve a partially observable Markov decision process (POMDP) problem. With these two ideas, our framework can either adapt to reasonable variations in unstructured environments or improve the sample efficiency of policy learning. We evaluated our framework on a task that involved inserting a Random Access Memory (RAM) using a torque-controlled robot and tested the success rates of different baselines used in the traditional methods. We proved that our framework is robust and can tolerate environmental variations.

Moreover, to solve the contact-rich task transparency and pose uncertainty issues during robot teaching, another framework is proposed, which focuses on combining visual servoing-based Learning from Demonstration (LfD) and force-based Learning by Exploration (LbE) to enable the fast and intuitive programming of contact-rich tasks with minimal user efforts. Two learning approaches were developed and integrated into a framework, one relying on human-to-robot motion mapping (visual servoing approach) and the other relying on force-based reinforcement learning. The developed framework implements the noncontact demonstration teaching method based on the visual servoing approach and optimizes the demonstrated robot target positions according to the detected contact state. The developed framework is compared with the two most commonly used baseline techniques, i.e., teach pendant and hand-guiding programming. Furthermore, the efficiency and reliability of the framework are validated via comparison experiments involving the teaching and execution of contact-rich tasks. The proposed framework shows the best performance in terms of teaching time, execution success rate, risk of damage, and ease of use.

Lastly, in order to solve sample efficiency and safety concern issues when training robots in the real world, a sim-to-real transfer learning framework is proposed to address the aforementioned concerns. In this part, we introduce a sim-to-real learning framework for vision-based assembly tasks and perform training in a simulated environment by employing inputs from a single camera. We present a domain adaptation method based on cycle-consistent generative adversarial networks (CycleGAN) and a force control transfer approach to bridge the reality gap. We demonstrate that the proposed framework trained in a simulated environment can be successfully transferred to a real peg-in-hole setup.

Zusammenfassung

Kollaborative Roboter sollen direkt mit menschlichen Mitarbeitern zusammenarbeiten und in manchen Fällen sogar ersetzen können, und somit effektiv auf schnelle Änderungen in Montagestraßen reagieren. Kontaktreiche Manipulationsaufgaben sind in modernen Fertigungsumgebungen weit verbreitet. Der manuelle Entwurf einer Roboterregelung gilt jedoch als schwierig für traditionelle Regelungsmethoden, da die Regelung eine effektive Kombination von Modalitäten und sehr unterschiedlichen Eigenschaften erfordert. In dieser Arbeit werden mehrere visuelle und kraftbasierte kreative Fähigkeiten und Lernsysteme vorgeschlagen, um die aktuellen Probleme kraftgeregelter Roboter-Montageaufgaben zu lösen.

In dieser Arbeit betrachten wir zunächst die Einbeziehung von visuellen und haptischen Informationen in ein Reinforcement Learning (RL)-Framework, um die Probleme der Zielunsicherheit in unstrukturierten Umgebungen zu lösen. Darüber hinaus schlagen wir eine neue Idee zur Einführung einer proaktiven Aktion zur Lösung eines partially observable Markov decision process (POMDP)-Problems vor. Mit diesen beiden Ideen kann sich unser Framework entweder an vernünftige Variationen in unstrukturierten Umgebungen anpassen oder die Stichprobeneffizienz des Policy-Lernens verbessern. Wir haben unser Framework an einer Aufgabe evaluiert, bei der ein Speicherriegel mit Hilfe eines drehmomentgeregelten Roboters eingefügt werden sollte, und die Erfolgsquoten der verschiedenen, in den traditionellen Methoden verwendeten Grundlinien getestet. Wir konnten zeigen, dass unser System robust ist und Umgebungsschwankungen tolerieren kann.

Um die Probleme der Transparenz von kontaktreichen Aufgaben und der Posenunsicherheit während des Roboterlernens zu lösen, wird ein weiteres Framework vorgeschlagen, das sich auf die Kombination von visuellen-Servoing-basiertem Learning from Demonstration (LfD) und kraftbasiertem Learning by Exploration (LbE) konzentriert, um die schnelle und intuitive Programmierung von kontaktreichen Aufgaben mit minimalem Aufwand für den Benutzer zu ermöglichen. Es wurden zwei Lernansätze entwickelt und in ein Framework integriert, von denen einer auf der Abbildung von menschlichen Bewegungen auf Roboterbewegungen (visueller Servoansatz) und der andere auf kraftbasiertem RL beruht. Das entwickelte Framework implementiert die berührungslose Demonstrationsmethode, die auf dem visuellen Servoing-Ansatz basiert, und optimiert die demonstrierten Zielpositionen des Roboters entsprechend dem erkannten Kontaktzustand. Das entwickelte Framework wird mit den zwei gängigsten Methoden der Roboterprogrammierung verglichen, dem Programmieren über ein Handbediengerät und dem Programmieren mittels Handführung. Darüber hinaus werden die Effizienz und Zuverlässigkeit des Frameworks durch Vergleichsexperimente mit dem Teachen und Ausführen von kontaktreichen Aufgaben validiert. Das vorgeschlagene Framework zeigt die beste Leistung in Bezug auf die Lehrzeit, die Erfolgsquote bei der Ausführung, das Risiko von Schäden und die Benutzerfreundlichkeit.

Schließlich sollen die Probleme der Probeneffizienz und der Sicherheit beim Training von Robotern in der realen Welt gelöst werden. Dazu wird ein Framework für die Übertragung von Simulationen auf die reale Welt vorgeschlagen. In diesem Teil stellen wir ein Simulation-zu-Realität-Lernsystem für bildverarbeitungsbasierte Montageaufgaben vor und führen das Training in einer simulierten Umgebung durch, indem wir Eingaben von einer einzigen Kamera verwenden. Wir stellen eine Domänenanpassungsmethode vor, die auf zykluskonsistenten generativen adversen Netzen (CycleGAN) und einem Kraftregelungsübertragungsansatz basiert, um die Realitätslücke zu schließen. Wir zeigen, dass das vorgeschlagene Framework, das in einer simulierten Umgebung trainiert wurde, erfolgreich auf eine reale Umgebung übertragen werden kann.

Contents

Chapter 1

Introduction

This chapter gives an introduction to the topic of visual and force combined robotic assembly with collaborative robots. Section 1.1 and Section 1.2 describe the State of the Art (SOTA) and the challenges addressed in this thesis. Section 1.3 presents the achieved contributions in this thesis. Section 1.4 gives the connection between different chapters and the previous publications.

1.1 Motivation

Position-controlled robots are able to handle known objects on well-structured assembly production lines with high efficiency and achieve highly accurate position control. However, they require considerable setup time and tedious reprogramming to fulfill new tasks, and cannot adapt to any unexpected variations [169]. Collaborative robots offer the promise of closing the gap between onerous reprogramming and unexpected variations by combining the capabilities of position-controlled robots with dexterity and flexibility. For example (Figure 1.1), the hand-guiding method enables unskilled users to interact with collaborative robots and facilitates quick programming [122].

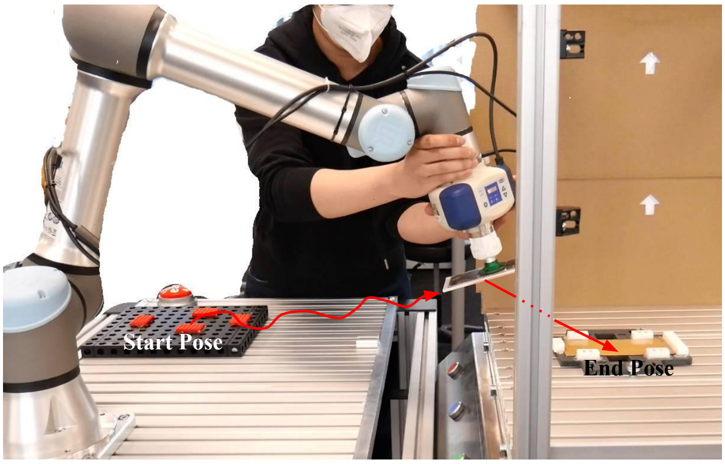


Figure 1.1: Collaborative robots work with human workers in heavily constrained spaces in factories.

Collaborative robots equipped with force control functions can perform certain hybrid position/force operations for contact-rich tasks [5], [56], [38], [77]; however, their effectiveness and variation adaptive capacity in assembly processes are still unsatisfactory [90], [128]. Moreover, a long time is still required to remove and reinstall the robot arms and various attachments during assembly line reconfiguration.

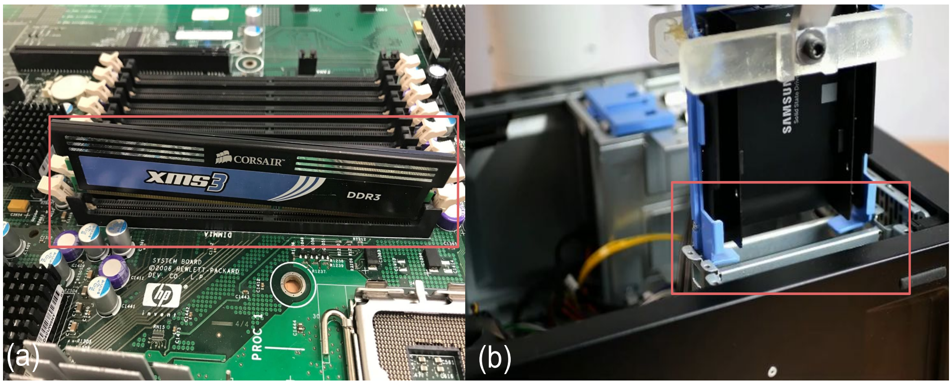


Figure 1.2: High-precision contact-rich insertion tasks. (a) RAM assembly. (b) Solid State Drives (SSD) assembly.

For high-precision contact-rich assembly tasks as shown in Figure 1.2, a robot needs to combine high positioning accuracy with high flexibility. Designing a robot for these tasks is very challenging although such tasks can be easily performed by humans. Several collaborative robots have been designed to perform cooperative tasks in industrial environments such as Light-Weight Robots (LWR) [3], Universal Robots [67], Diana7 [1], and Franka Panda [38]. Most of these robots have seven revolute joints with torque sensors, and similar control algorithms [4], [107], [5]. Currently, torque-controlled robots are safe enough when collisions occur with environments or humans [3], [45]. However, their effectiveness in real-life and production scenarios is still unsatisfactory.

LfD has recently been recommended as an effective technique for accelerating the learning (programming) processes, spanning from high-level assembly planning to low-level control [71], [29]. Guiding robots by means of visual feedback [78], [166], [53] during assembly tasks is an effective way to overcome position uncertainties. For robotic assembly tasks, performing high precision measurements is important. However, visual errors can be introduced by lenses and the imaging sensors, as well as the calibration of intrinsic and extrinsic parameters [83]. Some researchers believe that humans should focus more on execution tasks than vision sensors [83]. Based on this notion, other approaches have been developed, such as intelligent assembly algorithms, in an effort to lower the necessity of vision sensors for given tasks.

LbE has been suggested as an effective method for reducing the programming time, and recent studies have introduced artificial intelligence methods into robotics [70], [169], [2], [12]. Moreover, RL offers a set of tools for designing sophisticated robotic behaviors that are difficult to engineer. RL and its derivative methods

[1] https://www.agile-robots.com/

[5] www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa

have previously been successfully used to address various robotic manipulation problems [77], [89], [56], [90], [76], [35]. Exploration behavior entails interactions between robots and their operational environment. Therefore, a robotic force or impedance controller is required.

For contact-rich manipulations, it is nontrivial to establish a robotic system that can learn a task with a safety guarantee and avoid wear and tear problem. Thus, sim-to-real methods are proposed [113] to address the aforementioned concerns. Style transfer methods based on Generative Adversarial Network (GAN) [41] have been proposed recently in the computer vision field, enabling the use of vision-based manipulation tasks for deploying visual sim-to-real methods; however, owing to poorly simulated dynamics, the sim-to-real reality gap could be an issue when transferred the simulated policies to physical setups [119].

For the reasons outlined above, this thesis(work) focuses on robotic assembly tasks based on visual and force information using collaborative robots.

1.2 Problem Statement

Pose uncertainties are quite normal in human-based production lines as the operation objects are not fixed. Workers could perform high-precision robotic assembly tasks with their strong intelligence, excellent visual ability, and dexterous hands. Whereas these tasks are challenging to robots, especially in unstructured production environments. In addition, the friction and obstruction in contact-rich tasks introduce large positional errors due to the low stiffness design concepts of torque-controlled robots [5]. The limited control stiffness combined with the friction and obstruction in contact-rich tasks gives the position control error at a millimeter level. Torque-controlled robots are expected to achieve the desired dynamic interaction between environmental forces and robot movements to avoid breaking environments or targets, thus the desired position and contact force cannot be satisfied in the same DoF simultaneously. Moreover, the location of the targets is uncertain sometimes due to the insufficient accuracy of industrial assembly lines. Using the visual method to correct the positions of the targets is an intuitive solution, while we still have position control problems when the robot contacts targets or environments, even though we have implemented some explore actions (e.g., the spiral explore method [109]).

Teach pendants are widely used for precision positioning (position and orientation of the End-Effector (EE) in many assembly tasks [122]. However, these devices limit the intuitiveness of teaching processes and are time-consuming. Hand-guiding is a typical physical contact kinesthetic teaching solution, where programming is embodied using demonstration concepts, enabling users to quickly and intuitively program robots. However, it has drawbacks in terms of accuracy, locational separations, and operations involving dangerous objects [169]. Programming based on demonstration approaches has been proposed to solve variations in geometry and configurations for assembly, placement, handling, and picking tasks [94], which can reduce the programming time and user training requirements [169]. Mobile manipulators can considerably reduce robots' and devices' installation time [94]. The use of mobile manipulators introduces a positioning error at the ±5 mm level [148], and errors as small as ±1 mm can induce large huge contact forces and consistent failures in typical assembly tasks [128]. In conclusion, neither the hand-guiding nor the teach pendant programming methods can compensate for the positioning errors that accompany mobile units [148], and can result in the generation of a huge contact force that can damage objects.

Owing to unknown contact mechanics, designing a feedback control mechanism for contact-rich tasks is challenging. RL has shown some progress in robotic contact-rich tasks in unstructured environments; however, sample efficiency and safety concerns are two main problems when performing policy training. Many RL algorithms require millions of steps to train policies for performing complex tasks [82], [78]. In other words, human supervision is always needed in resetting experiments, hardware status monitoring, and safety assurance, which is quite time-consuming and tedious [55]. The sim-to-real approach shows the potential to solve the aforementioned problems; however, one significant difficulty associated with this approach is bridging the reality gap to address the mismatch in distinct distributions of rendered images and real-world counterparts. Another challenge is ascribed to force modeling in simulation as the force interactions will inevitably occur between the target object and environments when performing contact-rich tasks. Moreover, it is expensive to apply the system calibration due to the limitation of the simulation domain expert's ability [144] and accurate requirements [48].

1.3 Contributions

In summary, this thesis focuses on robotic contact-rich assembly tasks using visual and force information with several different collaborative robots. The main contributions of this thesis are summarized as follows:

- **Visual Residual Reinforcement Learning:** A visual RL method that combines a visual-based fixed policy with a contact-based parametric policy is proposed, this method greatly enhances the robustness and efficiency of the RL algorithm. Moreover, a proactive action concept is proposed in the aforementioned residual RL policy to solve a POMDP problem, which could ensure the task success rate and the ability to tolerate environmental variations.

- **Visual Servoing based LfD:** An approach is presented that learns the trajectories of robots from demonstrations based on visual servoing for fast, easy, and accurate robot setup in heavily constrained spaces.

- **RRRL Policy:** A RRRL policy based on force-torque information is trained to overcome pose uncertainty in contact-rich tending operation.

- **CycleGAN and Force Control based Sim-to-Real Transfer of Robotic Assembly:** A vision-based sim-to-real learning framework is proposed to perform assembly tasks, a force controller and a peg-in-hole task that effectively leverages visual information and force control using a simple reward function for a complete insertion, including hole searching, alignment, and insertion.

- **A Pushing-based Hybrid Position/force Assembly Skill** is proposed for contact-rich assembly tasks, and the skill was demonstrated with Diana 7 robot to prove the method's validity. Moreover, the theory of analyzation of maximize the utilization of environmental constraints is proposed.

1.4 Publications and Outline

During the study, ten publications were accepted by different conferences and journals. A list of the publications that are incorporated (or partially incorporated) into this thesis is given below:

- **Yunlei Shi**, Zhaopeng Chen, Hongxu Liu, Sebastian Riedel, Chunhui Gao, Qian Feng, Jun Deng, and Jianwei Zhang. Proactive Action Visual Residual Reinforcement Learning for Contact-Rich Tasks Using a Torque-Controlled Robot. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 765-771. IEEE, 2021 [134].

- **Yunlei Shi**, Zhaopeng Chen, Yansong Wu, Dimitri Henkel, Sebastian Riedel, Hongxu Liu, Qian Feng, and Jianwei Zhang. Combining Learning from Demonstration with Learning by Exploration to Facilitate Contact-Rich Tasks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1062-1069. IEEE, 2021 [135].

- **Yunlei Shi**, Chengjie Yuan, Athanasios Tsitos, Lin Cong, Hamid Hadjar, Zhaopeng Chen, Jianwei Zhang. A Sim-to-Real Learning-based Framework for Contact-Rich Assembly by Utilizing CycleGAN and Force Control. IEEE Transactions on Cognitive and Developmental Systems, Jan. 2023 [136].

- **Yunlei Shi**, Zhaopeng Chen, Lin Cong, Yansong Wu, Martin Craiu-Müller, Chengjie Yuan, Chunyang Chang, Lei Zhang, Jianwei Zhang. Maximizing the Use of Environmental Constraints: A Pushing-Based Hybrid Position/Force Assembly Skill for Contact-Rich Tasks. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2021 [133].

- Chengjie Yuan*, **Yunlei Shi***, Qian Feng, Chunyang Chang, Zhaopeng Chen, Alois Christian Knoll, Jianwei Zhang. Sim-to-Real Transfer of Robotic Assembly with Visual Inputs Using CycleGAN and Force Control. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2022 [162].

- Chunyang Chang*, Kevin Haninger*, **Yunlei Shi**, Chengjie Yuan, Zhaopeng Chen, Jianwei Zhang. Impedance Adaptation by Reinforcement Learning with Contact Dynamic Movement Primitives. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* 2022 [18].

Here is the list of publications that was generated during the period of PhD which are related to the PhD topic but not included in this thesis:

- Lin Cong, **Yunlei Shi** and Jianwei Zhang, Self-supervised Attention Learning for Robot Control, In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2021 [27].

- Lin Cong, Hongzhuo Liang, Philipp Ruppel, **Yunlei Shi**, Michael Görner, Norman Hendrich and Jianwei Zhang, Reinforcement Learning with Vision-Proprioception Model for Robot Planar Pushing, Frontiers in Neurorobotics, Mar.2022 [26] .

- Vincent Mayer, Qian Feng, Jun Deng, **Yunlei Shi**, Zhaopeng Chen, Alois Knoll, FFHNet: Generating Multi-Fingered Robotic Grasps for Unknown Objects in Real-time, In *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021 [95].

- Lei Zhang, Kaixin Bai, Zhaopeng Chen, **Yunlei Shi**, Jianwei Zhang. Towards Precise Model-free Robotic Grasping with Sim-to-Real Transfer Learning.In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2022 [165].

The structure and chapter connections of this thesis are shown in Figure 1.3. The main contents of this thesis are organized as follows: Chapter 1 gives a brief introduction of the thesis from motivation, problem statement, and contribution. Chapters 2, 3 and 4 belonging to the preliminaries and state of the art part, contact-rich robotic assembly (Chapter 2), robot force control (Chapter 3) and RL (Chapter 4) are introduced first as they are preliminary knowledge for the work in this thesis. Chapter 5 introduces the reader to the system design of the Diana7 robot, the force control performance is also evaluated, and a pushing-based hybrid position/force assembly skill is developed for tending tasks. The proposed framework visual residual RL is given in Chapter 6 and a PC assembly evaluation including RAM and SSD insertion based on the aforementioned framework is given in Chapter 9. Chapter 7 present visual servoing based LfD and the RRRL policy, then two methods combined into a robot tending skill; then the new skill is compared with two most commonly used baseline techniques using an UR5e robot for phone part tending task in Chapter 10. In order to solve the sample efficiency and safety concerns when performing policy training, Chapter 8 gives a framework design of CycleGAN and force control based sim-to-real transfer RL for robotic assembly. The sim-to-real Peg-in-Hole (PiH) experiment is demonstrated in Chapter 11. In the Summary part, the conclusion, limitations, as well as the outlook for the future are discussed.

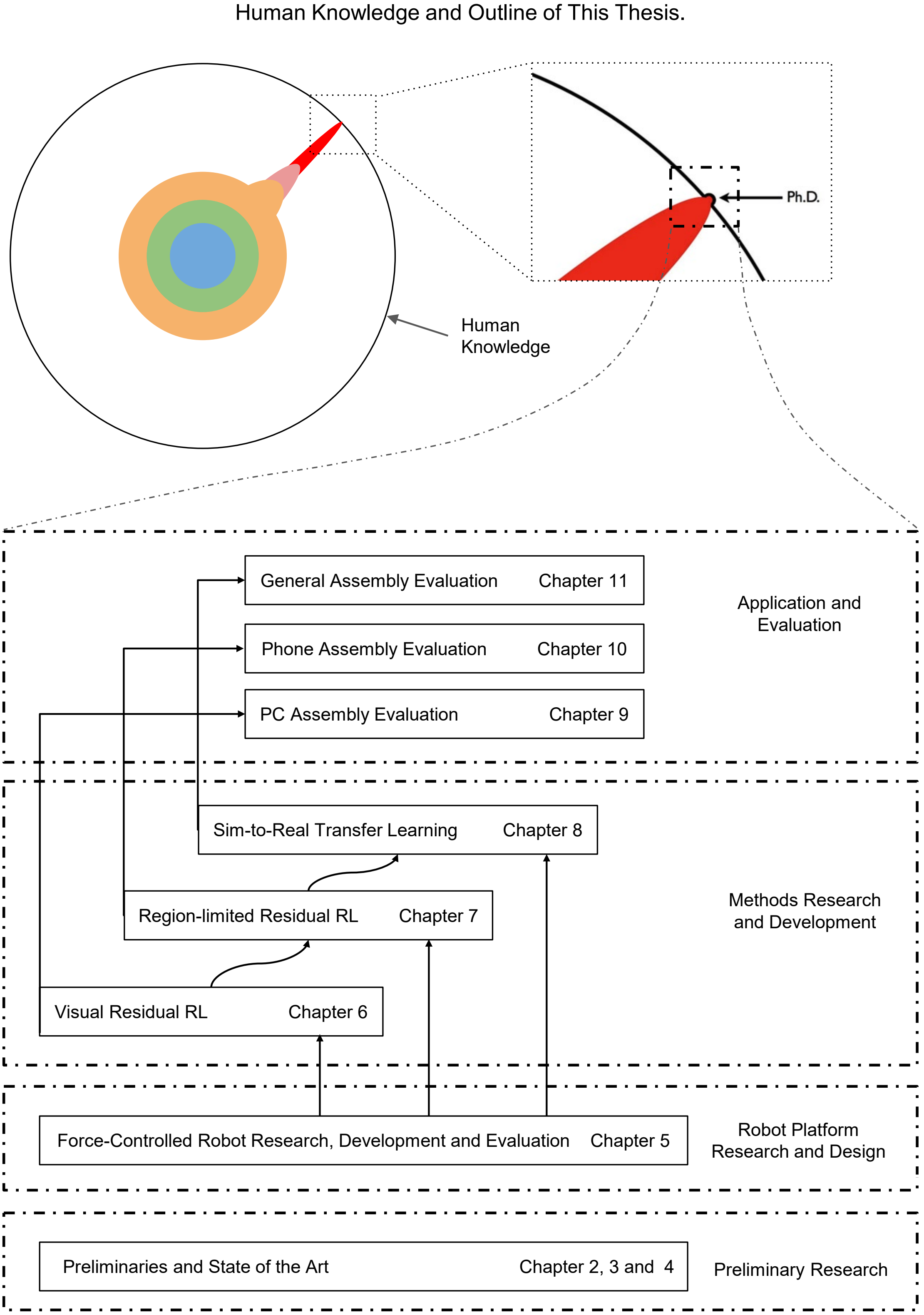

Human Knowledge and Outline of This Thesis.
Ph.D.
Human Knowledge
General Assembly Evaluation Chapter 11
Phone Assembly Evaluation Chapter 10
PC Assembly Evaluation Chapter 9
Application and Evaluation
Sim-to-Real Transfer Learning Chapter 8
Region-limited Residual RL Chapter 7
Visual Residual RL Chapter 6
Methods Research and Development
Force-Controlled Robot Research, Development and Evaluation Chapter 5
Robot Platform Research and Design
Preliminaries and State of the Art Chapter 2, 3 and 4
Preliminary Research

Figure 1.3: Outline of the thesis and relation of the main aspects.

Part I

Preliminaries and State of the Art

Chapter 2

Contact-Rich Robotic Assembly

2.1 Assembly and Contact Modeling

2.1.1 Assembly Sequence

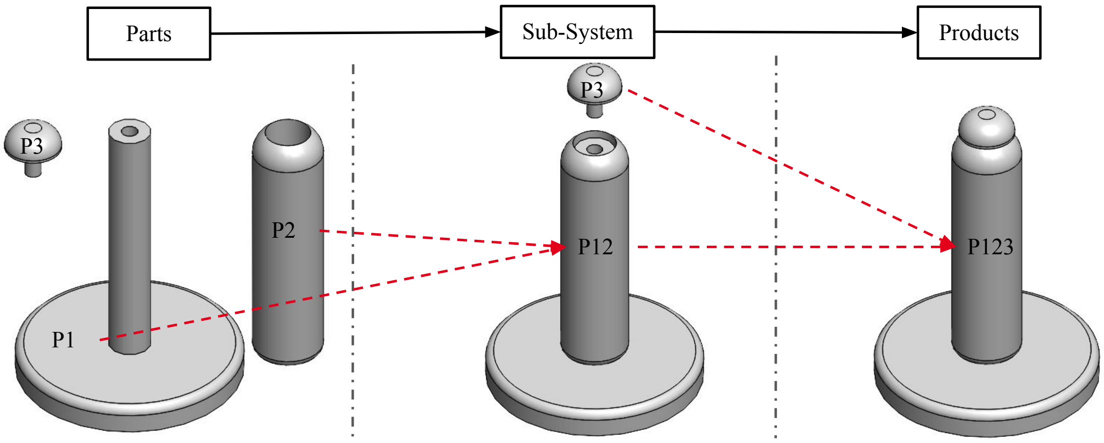


Figure 2.1: Graphical representation of mechanical parts assembly sequence.

A figure has been made to clearly explain the mechanical parts assembly process. The three pieces $p : 1,2,3$ as shown in Figure 2.1, are assembled in the sequence indicated by the arrows, here, $p2$ is inserted into $p1$ and generate $p12$, and the rest has the same definition. In this thesis, we focus on the assembly process.

As shown in Figure 2.1, several assembly operations are required, each assembly sequence consists of the following principle stages [24]:

- first of all, pick up the particular component part;
- secondly, place it into the assembly jig;
- thirdly, mating it into the desired component part;
- lastly, return the manipulator for the next pick-up movement.

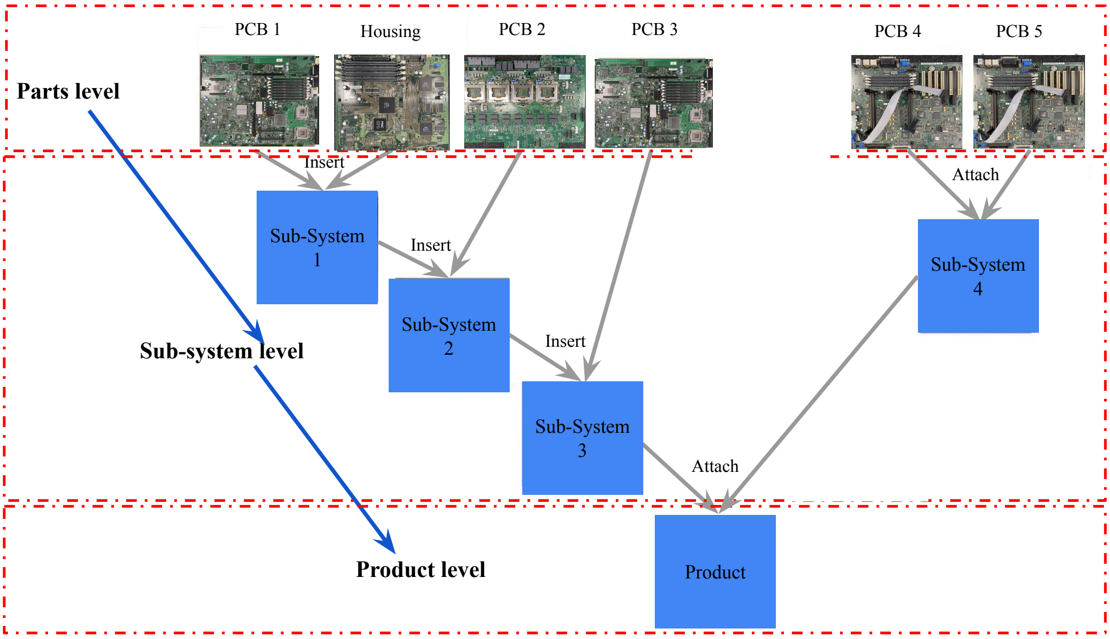


Figure 2.2: Graphical representation of Programmable Logic Controller (PLC) product assembly, this figure is redrawn based on author understanding [163] to explain the PCB assembly sequence.

In the PLC I/O Module assembly application as shown in Figure 2.2. The assembly process is done as follows: Firstly, three Printed Circuit Boards (PCB) (PCB 1, PCB 2 and PCB 3) are inserted into a housing. Then, a light cover is assembled by attaching different components to a plastic plate via one snap-fit. Finally, the cover is attached to the housing via another snap-fit [163].

The PLC I/O Module assembly application was separated as different tasks, such as "Housing with PCB", and this task contains three different actions:

- Pick up PCB1;
- Move PCB1 to insertion pose;
- Insert PCB1 into housing.

In order to execute the tasks, the skills can be built based on the actions [163]. The relationship between skills, motion primitives and tasks will be explained as follows.

- Tasks :

  In terms of parameters and state variables, the task layer contains an abstract description of what the robot is doing. This layer must communicate with end users, as well as systems that run the manufacturing line, by scripting and starting activities.

- Skills :

  which is visible to the end-user when the operator programs new tasks on the robot, is the layer that deserves the most attention.

- Motion primitives:

  a bottom-level motion layer in charge of implementing the robot's real-time control loops. For example, a hybrid force/position controller or an impedance controller can be part of this layer.

According to the estimate from [60], PiH assembly is one of the most typical task in assembly processes (approximately 40% of the total assembly task). Thus understanding the contact model of PiH assembly is important.

2.1.2 Contact Modeling

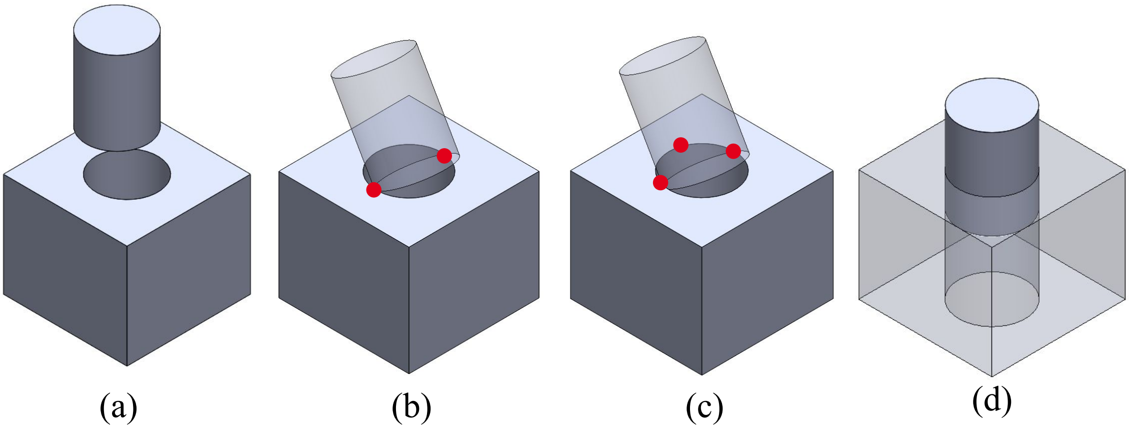


Figure 2.3: Four contact states that are commonly encountered in the PiH procedure.

Figure 2.3 depicts four states involving a peg and a hole when the peg is in touch with the hole and is pushed toward the hole by a force [112]. The bottom of the peg and the top of the hole are shown in planar contact in Figure 2.3 (a). When the center of the bottom of the peg is positioned beyond the surface of the hole, this condition occurs. The peg is slanted and two-point contact occurs if the center of the peg is near to the hole (assuming the presence of a compliant robot), as illustrated in Figure 2.3 (b). Figure 2.3 (c) shows three-point contact, which happens when the center of the bottom of the peg and the hole are near enough but the peg's tilting angle maintains. The peg will naturally fall into the hole if the tiling angle is zero, as shown in Figure 2.3 (d). The two-point contact condition is critical to understand since it is the most common occurrence in the PiH operation [112]. In the two-point contact condition, the peg is angled in the direction of the straight line and the assembly force pushes the peg to the hole.

2.1.2.1 Quasi-Static Modeling Notation (Figure 2.4)

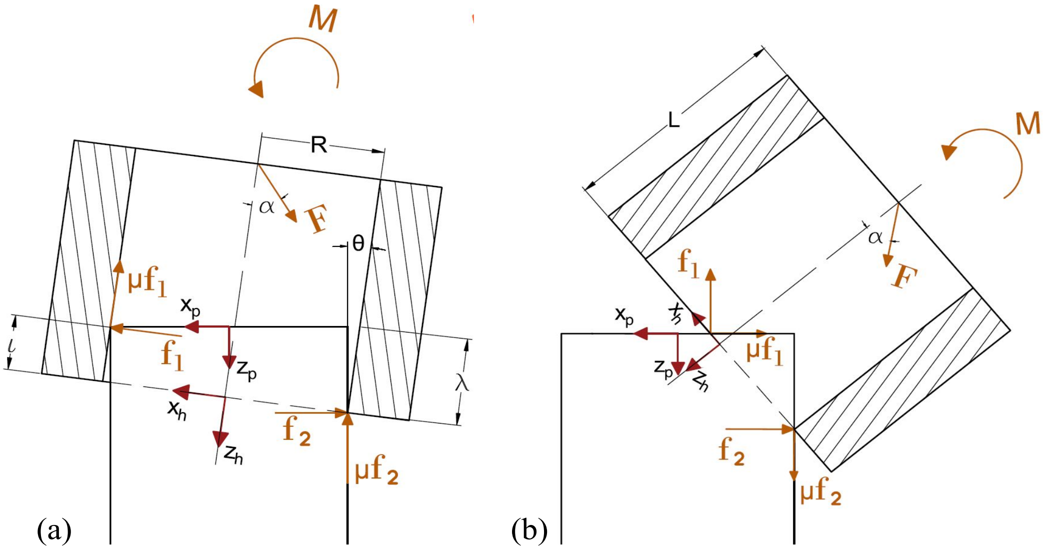


Figure 2.4: Three-point and two-point contact models [123]. Reprinted Image: ©2020 IEEE.

- θ denotes the tilt angle between the peg and the axes of the hole and ;
- L denotes the length of the hole;
- μ denotes the friction coefficient;
- f_1, f_2 are the reaction forces generate at the contact points;
- The applied wrench:

  the force F along the vector $n_\alpha = \begin{bmatrix} -s_\alpha & 0 & c_\alpha \end{bmatrix}^T$ forms an angle α with respect to the moment M about y_h and the axis of the hole z_h.
- The quasi-static equilibrium equations:

$$\begin{aligned} Fn_\alpha &= -F_{r_1} - F_{r_2} \\ MR_h^T y_h &= r_1 \times F_{r_1} + r_2 \times F_{r_2} \end{aligned} \tag{2.1}$$

- For different contact cases, $F_{r_i}, r_i, i \in \{1,2\}$ can be expressed differently.

2.1.2.2 Two-Point Contact Model (Figure 2.4) (a)

In this situation, $F_{r_i}, r_i, i \in \{1,2\}$ are expressed as:

$$\begin{aligned}
r_{\mathbf{1}} &= \begin{bmatrix} R & 0 & (L-\ell) \end{bmatrix}^T \\
r_2 &= \begin{bmatrix} -R & 0 & L \end{bmatrix}^T \\
F_{r_1} &= \begin{bmatrix} f_1 & 0 & -\mu f_1 \end{bmatrix}^T \\
F_{r_{\mathbf{2}}} &= R_{y_h}(\theta) \begin{bmatrix} -f_2 & 0 & -\mu f_2 \end{bmatrix}^T \\
\ell &= (Dc_\theta - d)\, s_\theta
\end{aligned} \tag{2.2}$$

When the tilt angle is smaller than a threshold, a two-point contact case occurs:

$$\theta^\star = \arccos(\rho) \tag{2.3}$$

where $\rho = \frac{r}{R}$ represents the ratio between peg radius and hole radius. To keep an internal two-point contact, f_1 and f_2 must be positive, thus the lower and higher constraints for the force direction angle γ is obtained:

$$\arctan\left(-\frac{1}{\mu}\right) + \theta < \gamma < \arctan\left(\frac{1}{\mu}\right) \tag{2.4}$$

In the end, the assumption of the direction of the friction forces satisfies the requirement of a positive rotation for:

$$\frac{M}{F} > \mathscr{H}_{2\mathrm{pc}}(r, \mu, \rho\theta, \gamma, L) \tag{2.5}$$

$\mathscr{H}_{2\mathrm{pc}}$ is a function that can be expressed in closed form [16] by substituting f_1 and f_2 obtained from Equation (2.2) and solve with respect the ratio $\frac{M}{F}$. As the two and three-point contact cases modeling follows the same steps as [16], the three-point contact model will not be explained here.

2.2 Robotic Assembly

2.2.1 Passive Compliant Approaches

Many methods have been developed to employ the passive compliant approaches which can be separated into following groups [24]:

- Compliant EE or work station:

  Regarding the compliant methods, a variety of compliance concepts have been developed as well as the mating theories associated with this method. The most successful device using this method is the remote-center-compliance (RCC) wrist [24].

- Air stream:

  The air stream-assisted approaches use a suction cup to generate necessary forces by an air stream to mate the parts [15], [24]. Normally, the cycle time cost is much less than Remote-Center-Compliance (RCC) wrist [24].

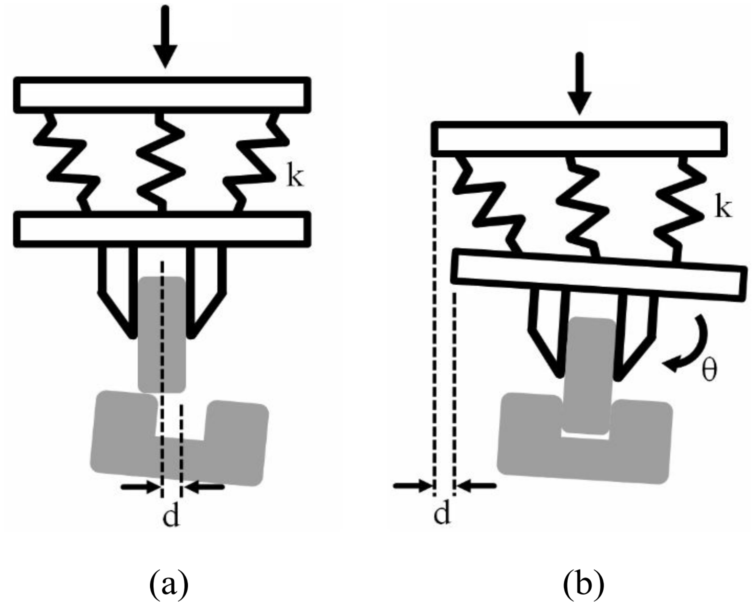
- Magnetic force [39]:

  This method utilizes a magnetic force to align the mating objects, a magnetic field formed between two mating components will generate the magnetic force.

- Vibratory motion [49]:

  The vibratory insertion method implemented a random search, a component is vibrated with respect to the mating component. The vibration can be performed either by the robot or by the specially designed EE. This kind of EE could be similar to the traditional RCC, however, the inserting concepts are not the same.

Some passive compliant approach examples can be found as follows.



(a) (b)

Figure 2.5: Compliant mechanism work concept. [110]. Reprinted Image: ©2014 IEEE.

In the passive compliant assembly approaches shown as Figure 2.5, in Figure 2.5(a), the peg contacts with hole before insertion, compliant mechanism shape haven't been changed. Then after the insertion, compliant mechanism shape is changed due to the adaption of distance d and θ) as shown in Figure 2.5(b).

The passive compliant mechanism was used to reduce the contact force or torques between the object and the environment. The assembly system can generate natural compliance with external forces by a parallel spring mechanism. Charles Stark Draper Laboratory has developed the most successful device using the RCC wrist method [158], [31].

F is the assembly force, θ is the misalignment of orientation and d is the misalignment of position. k is the spring constant parameter of the RCC device. k_x is the translation spring constant and k_w is the orientation spring constant. In most RCC devices, the spring constant sets are fixed, thus the compliance of the manipulator is fixed.

Jeong *et al*. developed a pneumatic vibratory wrist for robotic assembly operations, the vibratory wrist can perform the random motion of the hole to compensate for the position uncertainty. The results showed that this assembly method can compensate for considerable initial XY plane errors (maximum to 0.6 mm) for various combinations of frequency ratio f (16Hz $< f <$ 20Hz).

The advantages of passive compliant approaches are [24], [60]:

- Low cost, no need for expensive sensors;
- The structure is simple and the response is quick.

However, the disadvantages are [24], [60]:

- Poor adaptability, can only handle small misalignment;
- Low assembly accuracy;
- Lack of ability to measure the external force;
- The contradiction between the high flexibility of the devices and the high stiffness due to the spring constant and the geometrical direction.

2.2.2 Active Compliance Approaches

Different with passive compliant approaches, active compliance approaches introduce more sensing information, EE and actuating mechanism design, as well as the associated control algorithms.

Regarding the sensing method, the main following schemes and the details can be found as follows:

- Force sensors [116]:

  The force sensors measure the contact forces and torques generated by the misalignment. These forces and torques signals are fed back to the relevant controller to control the robot's motion. For example, the admittance controller is one of the typical controllers.

- Touch sensors [108]:

  The touch sensor provides a rich signal to indicate the different contact states. The angle of contact can be estimated by a Convolutional Neural Network (CNN).

- Vision sensors [137]:

  Vision cameras, optical fibers and laser beams are commonly used in visual guided assembly. The vision sensors can provide information on the relative position and orientation of mating parts.

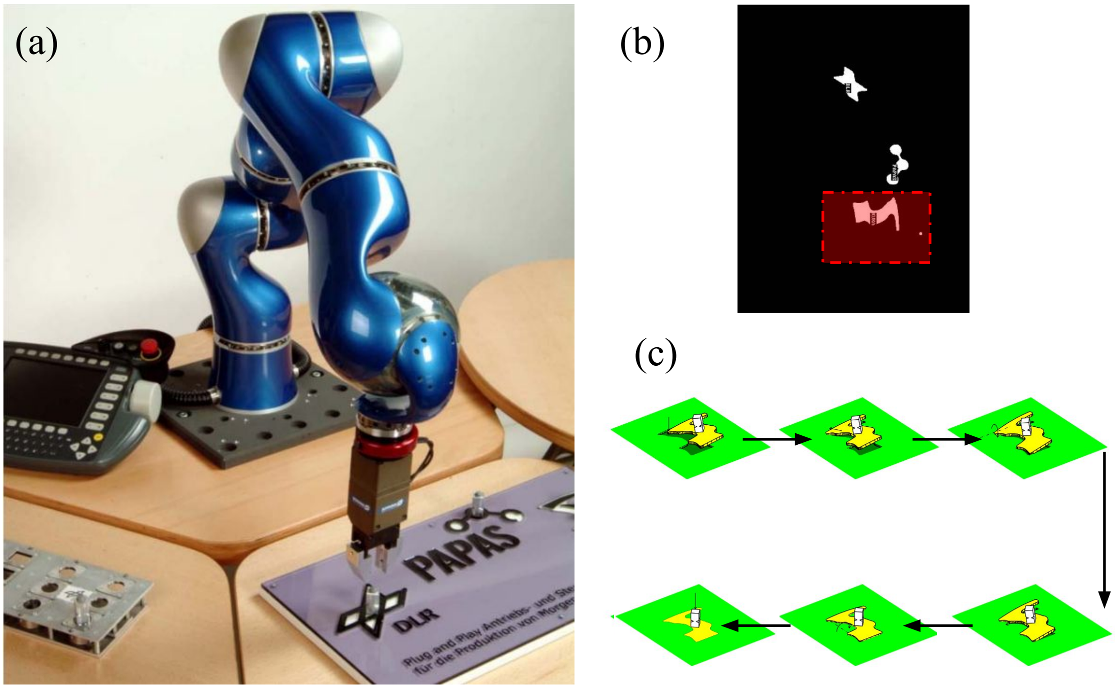


Figure 2.6: The Region of Attraction (ROA) compliance based assembly strategy [146]. (a): Assembly experimental setup for a set of planar objects with complex, nonconvex geometric forms task using the German Aerospace Center (DLR) LWR. (b): Binary image after color classification and post-processing with object labeling. (c):Basic sensing-based assembly strategy. Reprinted Image: ©2006 IEEE.

Some active compliance approaches measure the contact force/torque and feedback them to the controller to generate the compliance trajectory of the EE [100]. Active compliant control can overcome the disadvantages of passive compliant such as lack of ability to measure the external force and poor adaptability. Thus, active compliant approaches have a wider application area. According to the characteristics of implementation, several different categories of active control strategies can be found [100]:

- Admittance control strategy;
- Impedance control strategy;
- Force control strategy;
- Hybrid force/position control strategy.

Some sensing-based approaches take visual and force information as a feedback input for the assembly. Stemmer *et al.* [146] took vision and force information as the input of the ROA compliance-based assembly strategy which guarantees the local convergence of the assembly process by considering the parts' geometry as shown in Figure 2.6. Complicated shapes such as prisms and splines geometries are used in this research.

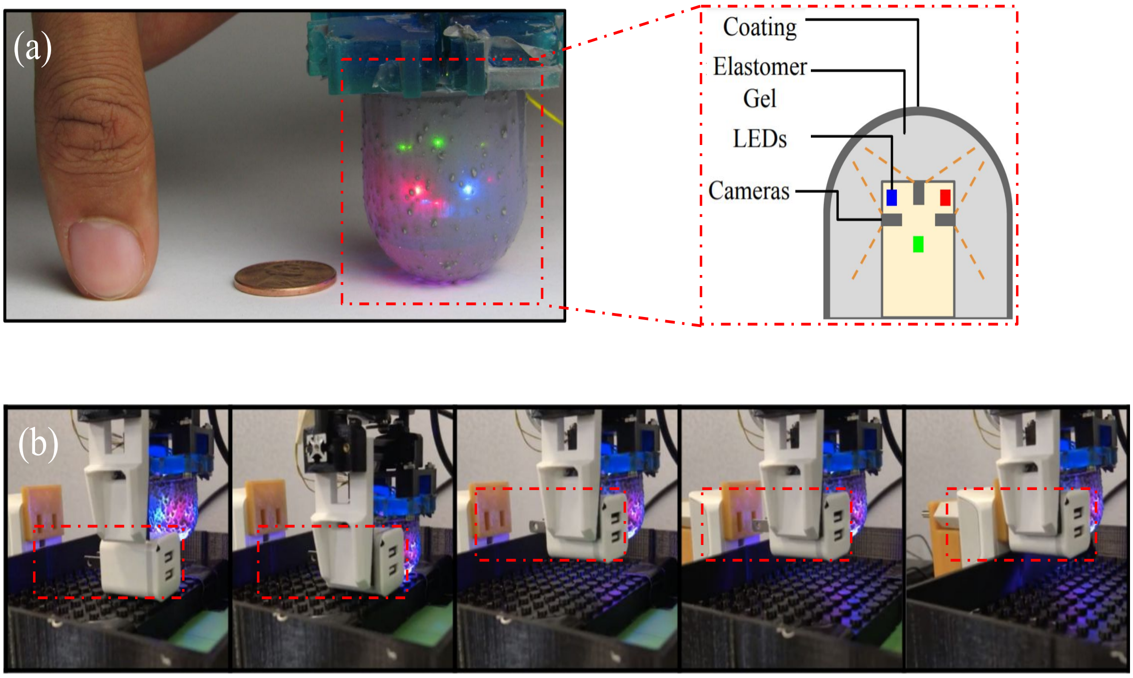


Figure 2.7: (a): OmniTact dimension and key design, human thumb and a US penny for scale compare. (b): OmniTact sensor is used to execute an insertion task of electrical connector into a wall outlet [108]. Reprinted Image: ©2020 IEEE.

Tactile information has been also used to guide assembly. For example, OmniTact [108] is a multi-directional tactile sensor designed for robotic operations, such as assembly and grasping. This concept shows that high-resolution tactile has the ability to sense and to "feel" curved surfaces and that both may be accomplished simultaneously by combining several micro-cameras. The experiment demonstrated how a CNN can detect the contact angle with a finger pressing on a flat surface, moreover, OmniTact can also be used to execute tactile control to insert an electrical connector into an outlet using a CNN. The cost of the cameras is a drawback of the existing design. The endoscopic cameras utilized in the sensor cost 600 US dollars apiece, bringing the total cost of the sensor prototype to 3200 US dollars.

In Figure 2.7 (b), a tactile sensing-based insertion operation is demonstrated from left to right: first of all, the connector is grasped by the gripper jaws, secondly, a random offset is applied to the EE gripper position. The sensor touches the floor and saves the touch information received from the top camera. Then, a fixed pick-up policy is called to grasp and lift the connector, then the gripper and connector move to the outlet, and the policy network is called to decide how to adjust the gripper position for insertion. Finally, the robot applies adjustment movement and inserts the connector successfully.

However, the tactile sensor has issues with a non-linear response, temperature and moistness dependence, fatigue, permanent deformation, and hysteresis [65], which make the tactile sensing-based insertion methods are hardly used in industrial scenarios.

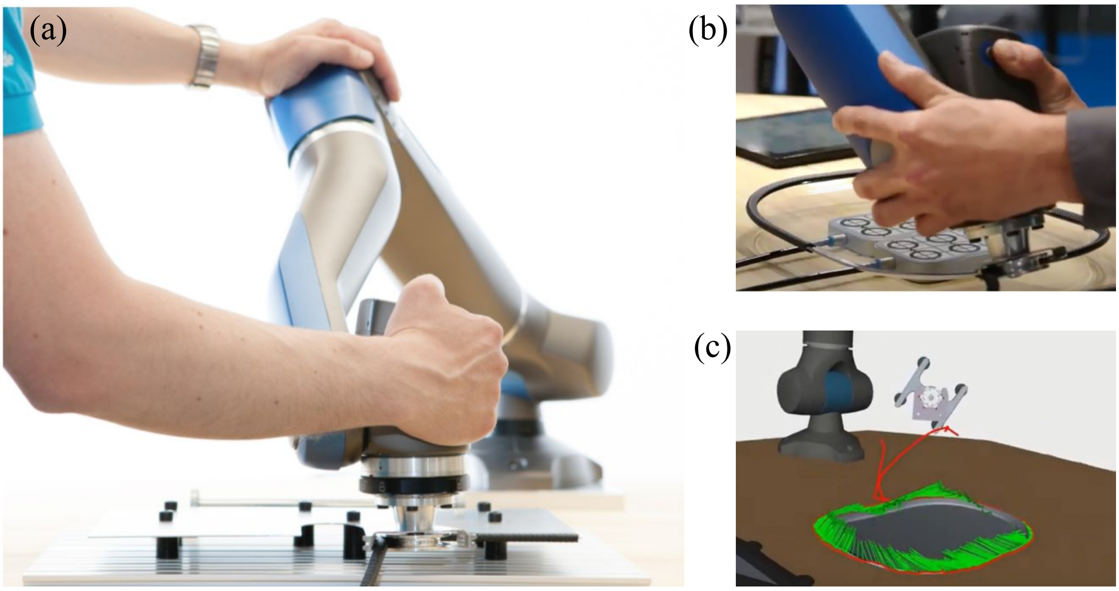


Figure 2.8: (a): Safe Autonomous Robotic Assistant (SARA) robot performs a soft rubber strip assembly task. (b): Hand-guiding method is used to teach SARA robot a position/force hybrid trajectory. (c): The position/force hybrid trajectory is visualized in the Graphical User Interface (GUI) [58]. Reprinted Image: ©2021 IEEE.

The SARA robot is equipped with redundant force sensors which enable a high-resolution force and torque measurement at the robot flange and thus allow measuring the contact forces during the hand-guiding teaching process Figure 2.8.

In general, the advantages of active compliance approaches are [24], [60]:

- The active method can handle comparably large positioning errors;
- The structure is simple and the response is quick;

However, the disadvantages are [24], [60]:

- A comparably long insertion time is required due to long search motion and signal processing;
- Lower reliability and applicability.

2.2.3 Learning-based Approaches

2.2.3.1 Learning by Exploration

The compliant-based and sensing-based approaches are always passive which means that the device is not adaptable and has no self-learning capability [60]. Thus, learning-based approaches are proposed to solve the above issues.

Inoue *et al.* used a robot to perform a tight clearance PiH task successfully by training a neural network with deep RL [56]. For the tight clearance PiH challenge, their technique has a good fitting performance and resilience against positional and angular faults. By taking the force and position sensors input from a robot to predict the system status, the neural network learns to execute the best action [56]. Inoue *et al.* separates the PiH task into two main phases: search and insertion.

In this research, the state that inputs the RL net is defined as:

$$\mathbf{s} = \left[F_x, F_y, F_z, M_x, M_y, \tilde{P}_x, \tilde{P}_y\right] \tag{2.6}$$

F and M are the force and torque information generated by the force-torque sensor; the subscript x, y, z denotes the Cartesian axis. The action space is defined as:

$$\mathbf{a} = \left[F_x^d, F_y^d, F_z^d, R_x^d, R_y^d\right] \tag{2.7}$$

F^d is the force command, R^d is the peg rotation command, a hybrid position/force controller is used to execute the commands. In real experiments, the action spaces are defined differently in the search phase and insertion phase, respectively.
For search phase:

$$\begin{gathered}\left[+F_x^d, 0, -F_z^d, 0, 0\right]\\ \left[-F_x^d, 0, -F_z^d, 0, 0\right]\\ \left[0, +F_y^d, -F_z^d, 0, 0\right]\\ \left[0, -F_y^d, -F_z^d, 0, 0\right]\end{gathered} \tag{2.8}$$

with $F_x^d = F_y^d = F_z^d = 20$ N.
For insertion phase:

$$\begin{gathered}\left[0, 0, -F_z^d, 0, 0\right]\\ \left[0, 0, -F_z^d, +R_x^d, 0\right]\\ \left[0, 0, -F_z^d, -R_x^d, 0\right]\\ \left[0, 0, -F_z^d, 0, +R_y^d\right]\\ \left[0, 0, -F_z^d, 0, -R_y^d\right]\end{gathered} \tag{2.9}$$

The Q-learning RL algorithm is used in this research. The state space, action space and reward function design concepts have inspired the following research. However, this research uses discrete actions to perform the PiH task with low efficiency and limited accuracy.

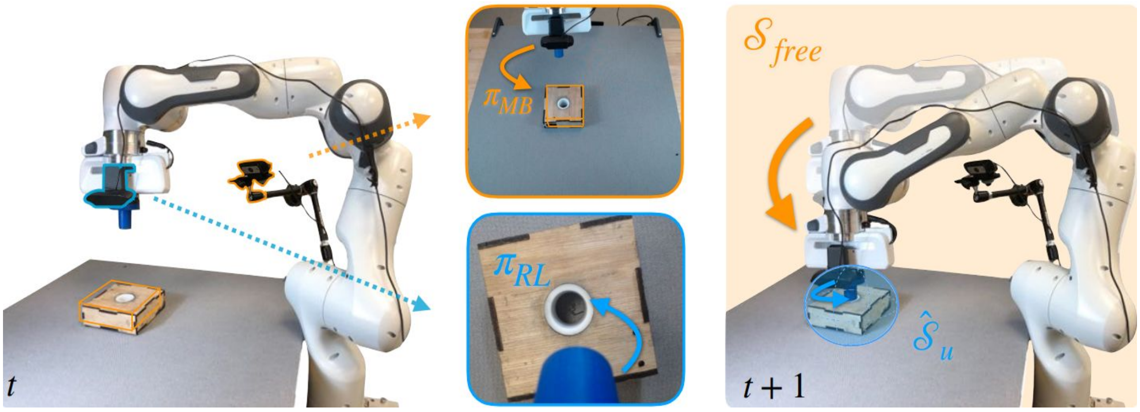


Figure 2.9: Concept and structure of Guided Uncertainty Aware Policy Optimization (GUAPO) policy: beyond the uncertainty area, the model-based method, π_{MB} is used to drive the system from free space to target; once inside the uncertainty area, a RL policy π_{RL} that learned from the raw image sensory input from the eye-in-hand camera (blue) that gives enough information to complete the insertion task [76]. Reprinted Image: ©2020 IEEE.

Lee *et al*. combined the strengths of model-based methods with the flexibility of learning-based methods to propose a GUAPO policy that is able to overcome pose uncertainties in contact-rich tasks [76].

In this method, a nonparametric distribution $\left\{\mathscr{S}_u^i\right\}_{i=1}^n$ and their associated weights $p\left(\mathscr{S}_u^i\right)$ is used to represent the uncertainty region [76]:

$$p\left(s \in \mathscr{S}_u\right)=\sum_{i=1}^n \nVdash\left[s \in \mathscr{S}_u^i\right] p\left(\mathscr{S}_u^i\right) \tag{2.10}$$

Then a function:

$$\alpha(s)=\nVdash\left[s \in \hat{\mathscr{S}}_u\right] \tag{2.11}$$

can be defined to distinguish the different region to use model-based policy π_{MB} or RL policy $\pi_{RL}(a \mid s)$. The GUAPO policy can be presented as:

$$\pi(a \mid s)=\alpha(s) \cdot \pi_{RL}(a \mid s)+(1-\alpha(s)) \cdot \pi_{MB}(a \mid s) \tag{2.12}$$

The SOTA model-free off-policy RL algorithm Soft Actor-Critic (SAC) is used in this research.

2.2.3.2 Learning by Demonstration

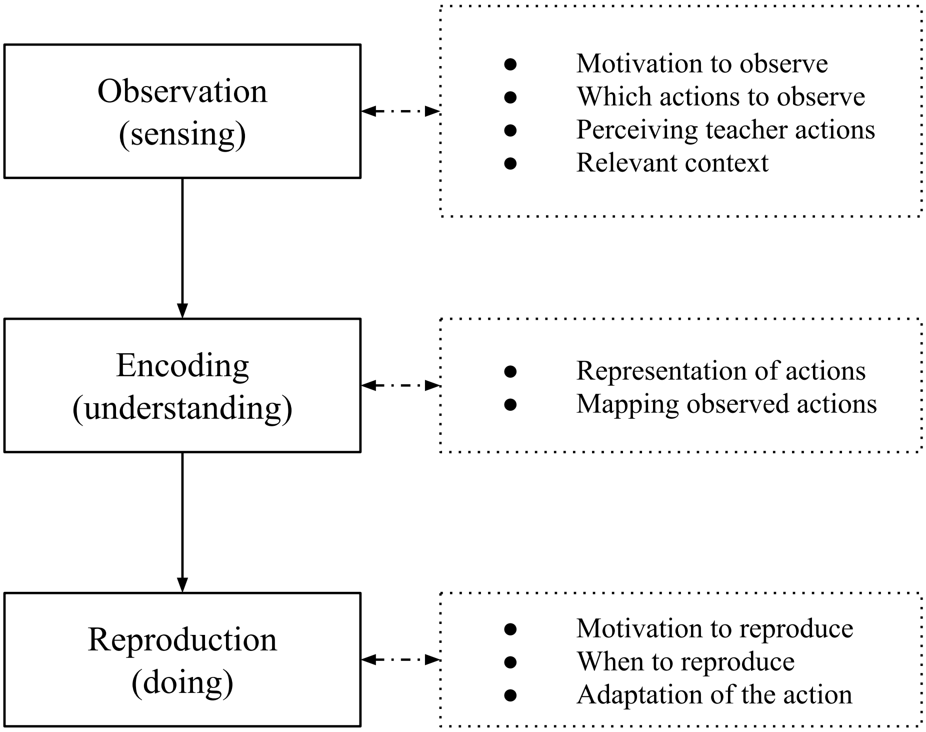


Figure 2.10: Three main phases in Learning by Demonstration (LbD) [9], [169].

According to contact model recognition, LfD techniques give a way for doing robotic PiH assembly without handmade reprogramming (Figure 2.10). LfD can be understood as a supervised learning issue as the abilities acquired from the demonstrator may be considered labeled information. LfD is a proper strategy to utilize when perfect behavior cannot be taught by standard robot programming or clearly defined as maximizing a known reward function via RL. Collecting demonstration experiences is a difficult process, but it is critical for increasing not only data efficiency but also the adaptability and generalization of the taught assembly policy. Several LbD concepts are explained as follows.

Kinesthetic Demonstration:

The kinesthetic guiding method directly records the movements of the robot, which does not need the transfer phase from a different kinematics and dynamics system.

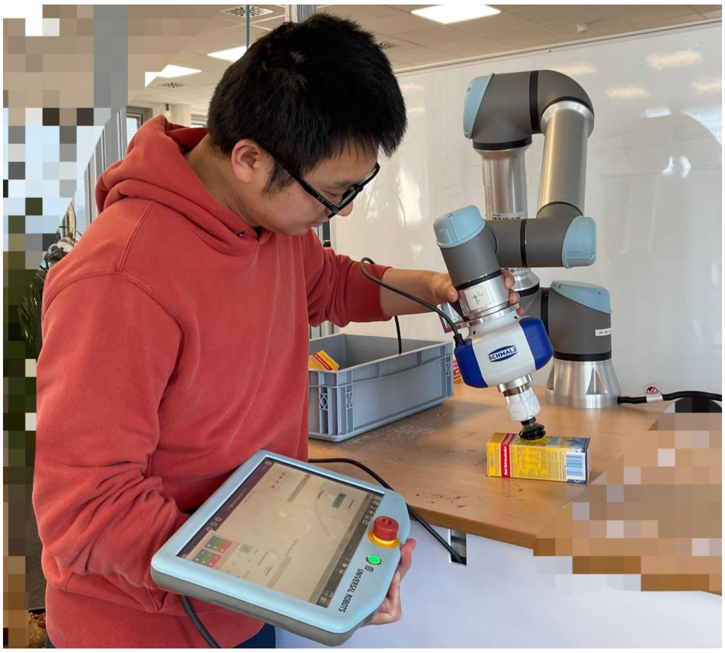


Figure 2.11: Kinesthetic guiding using a UR robot in gravity compensation mode for box picking task.

Figure 2.11 (a) demonstrates how the robot was trained in zero-gravity mode using kinesthetic demonstration. The robot joints were placed to a gravity torque compensation (passive) mode, allowing the human demonstrator to move each limb independently. During the demonstration, the kinematics of each joint motion were captured at a rate by proprioception. The robot "sensed" its own motion by recording the joint-angle data (Every DoF on the robot was equipped with motor encoders). The engagement with the robot was more fun than utilizing a graphical simulation, which allowed the user to sense the robot's limitations in the real world [1], [169].

Alejandro *et al.* [115] proposed a method that combines an LbD approach with a model-based and constraint-based task specification and control methodology as shown in Figure 2.11 (b). This research showed a complex constraint-based task with sensor interactions. However, their motion model is only sufficient for a uni-modal distribution, also the computational cost could be sensitive.

Sensor-based Demonstration

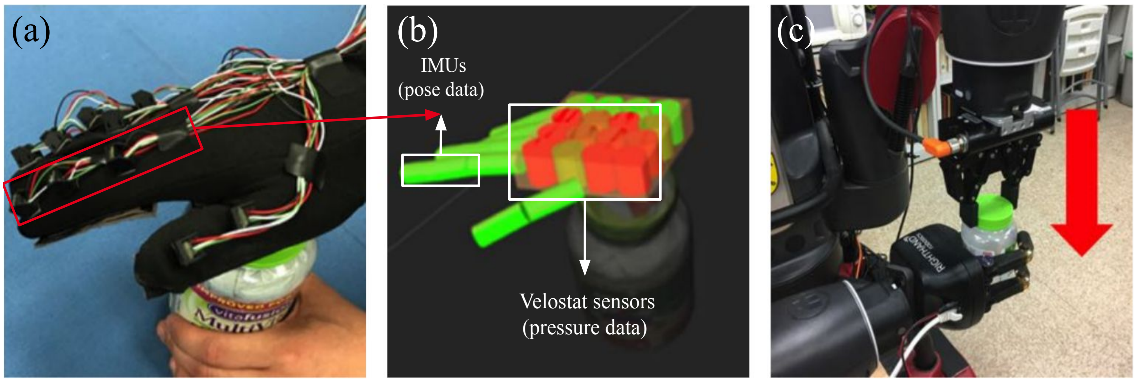


Figure 2.12: (a) and (b): a tactile glove is used to reconstruct both forces and poses from human demonstrations [32]. (c): with observed forces and trajectories, the robot learned to open a medicine bottle successfully. Reprinted Image: ©2017 IEEE.

Sensors and tracking devices can be used to record the movements and the forces of a demonstrator for manipulation tasks. Edmonds *et al.* [32] use a tactile glove to utilize both the poses and forces exerted by the demonstrator within a single demonstration. An And-Or-Graph (AOG) representation that can integrate both poses and forces is used to encode the demonstration.

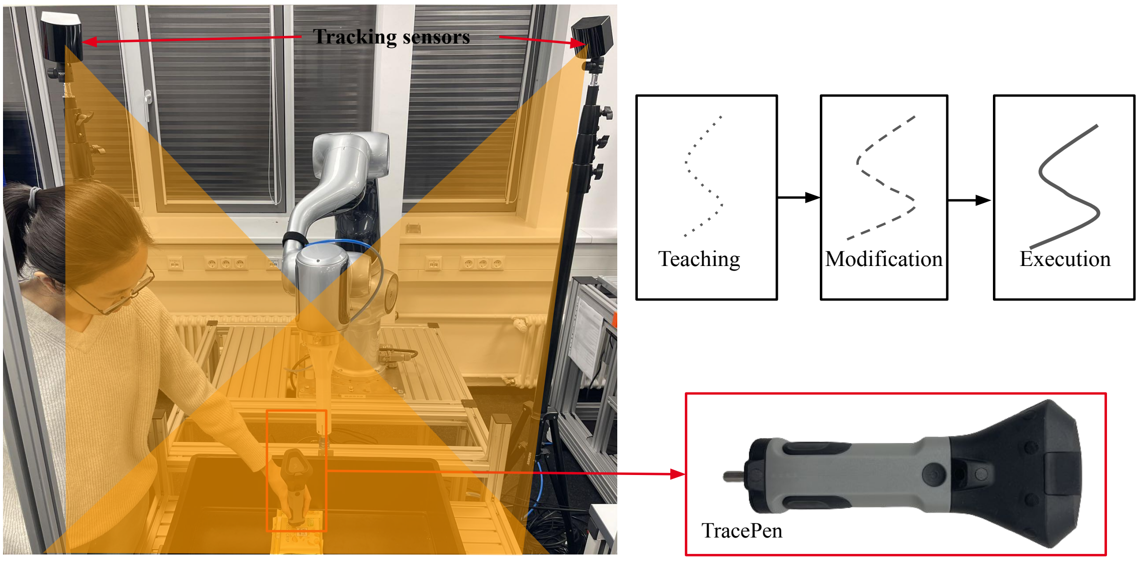


Figure 2.13: Wandelbots use a tracked pen to record the trajectories with high precision. The teaching has three steps: 1-teaching the trajectories with TracePen, 2-modifying the trajectories on programming GUI, 3-executing the trajectories in the real robot.

A German startup called Wandelbots [40] offers a commercial solution based on the notion of a motion track sensor. Active infrared sensors are used in the system to track a portable pen in 3D space. The pen can record trajectories with high precision for different procedures, the company promises a significant programming time reduction compared with other traditional robot programming methods [47].

Teleoperated Demonstration

With teleoperated demonstration approaches, the real-time tracking system will receive the demand EE pose information, then the robot can learn the behavior of the demonstrators.

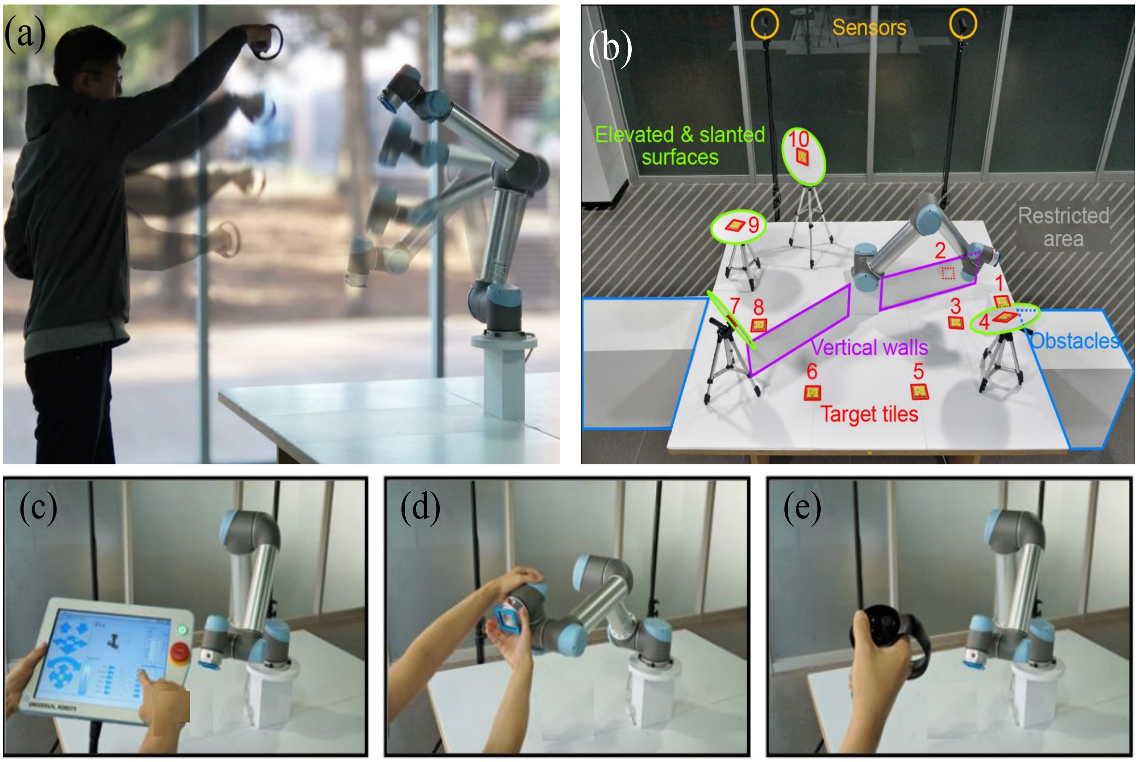


Figure 2.14: (a): Robot Telekinesis [75] allow the user to control the movement of the robot EE with hand gestures remotely. (b): the workspace setup for the teaching methods evaluation. (c): teaching pendant teaching. (d): hand-guiding teaching. (e): Robot Telekinesis teaching. Reprinted Image: ©2020 IEEE.

Lee *et al.* [75] present a novel robot interaction technique called Robot Telekinesis that allows users to control the movement of the EE of a robot arm in complex and changing environments with unimanual and bimanual hand gestures. Their method is quite fast and intuitive and does not need any physical effort. However, their method is lack force interaction with the environment, thus not feasible for the contact-rich task.

Chapter 3

Robot Force Control

3.1 Force Control Approaches

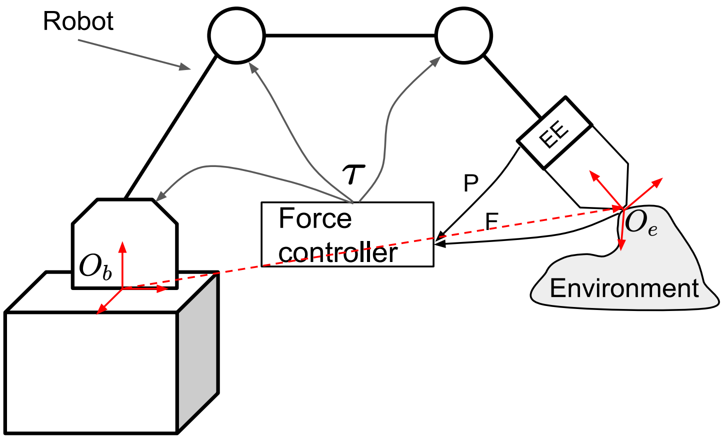


Figure 3.1: A robot force controller is used in contact space, O_b is the fixed base frame and O_e is the EE frame.

Controlling the interaction between a robot and its surroundings is critical for the effective completion of a variety of practical jobs in which the robot EE must move an object or execute a task on a surface as shown in Figure 3.1.

Many tasks such as assembly [144], polishing [102], [168], pushing [147], [25], cutting [121], scraping [20], deburring [52], grinding [143], pounding [73], excavating [57] always require the robot to interact with its environment. The environment constrains the geometric pathways that the EE can take during contact. Constrained motion is the term for this circumstance. In this situation, using a motion control method to govern interaction is doomed to fail.

In order to implement all these tasks, not only does the predisposed position need to be realized but also the resistance from the environment needs to be overcome with the necessary force. Thus, robot force control involves the integration of task goals such as modeling the environment, or adjustment of the applied torque in the robot joints according to the position, velocity and force feedback. Therefore, different kinds of robot force control algorithms were developed.

How to estimate the interaction forces and efficiently use feedback signals to synthesize the necessary input signals so that the intended motion and force may be maintained is a basic undertaking in robot force control. Position, velocity, acceleration, and force are the four essential variables in robot force control. The diverse applications of these basic variables and their correlations result in discrepancies in the existing fundamental force control algorithms [164].

3.2 Indirect and Direct Force Control

Force control strategies can be distinguished into two basic categories:

- indirect force control

  which achieves force control behavior via motion control, without explicit closure of a force feedback loop.

- direct force control

  which offers the possibility of controlling the contact force with the force feedback loop.

In order to clarify the calculations for indirect force controller and direct force controller, we define the frames (Figure 3.1) and notation as follows [139]:

- q denotes the vector of the joint angles, $\dot{q}$ donates the vector of joint velocities.
- p_e is the (3×1) position vector that characterize the position of the robot EE frame $\Sigma_e\left(O_e\right)$ with respect to a fixed base frame $\Sigma_b\left(O_b\right)$. $\dot{p}_e$ the (3×1) vector of robot EE linear velocity.
- R_e is the (3×3) rotation matrix that characterize the orientation of the robot EE frame $\Sigma_e\left(O_e\right)$ with respect to a fixed base frame $\Sigma_b\left(O_b\right)$. ω_e the (3×1) vector of robot EE angular velocity.
- J is the robot $(6 \times n)$ EE geometric Jacobian matrix.
- $v_e = \left[\begin{array}{cc} \dot{p}_e^{\mathrm{T}} & \omega_e^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}$ is the robot EE linear velocity and angular velocity.
- $h = \left[\begin{array}{cc} f^{\mathrm{T}} & \mu^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}$ where f denotes the (3×1) vector of external EE force and μ the (3×1) vector of external EE moment between the EE and the environment.
- $S(\cdot)$ is the operator performing the cross product between two (3×1) vectors.
- B is the $(n \times n)$ symmetric and positive definite inertia matrix.
- $C\dot{q}$ is the $(n \times 1)$ vector of Coriolis and centrifugal torques.
- $F\dot{q}$ is the $(n \times 1)$ vector of viscous friction torques, and g is the $(n \times 1)$ vector of gravity torques.

- K_P is suitable feedback matrix gains that same to the proportional part in the proportional-derivative controller.
- K_D is an $(n \times n)$ positive definite matrix damping gain, it provides additional control damping torque at each joint.

The kinematic model of the robot are:

$$\begin{aligned} p_e &= p_e(q) \\ R_e &= R_e(q) \end{aligned} \tag{3.1}$$

The desired EE position and the actual EE position error is given as:

$$\Delta p_{de} = p_d - p_e \tag{3.2}$$

The simple way for defining an orientation error if given as:

$$\Delta \varphi_{de} = \varphi_d - \varphi_e \tag{3.3}$$

Here, φ_d and φ_e are the Euler angles representations that extract from the orientation matrices R_d and R_e, respectively.

The angle/axis representation of the orientation error between the desired and the actual EE orientation is given as follows , ${}^eR_d = R_e^{\mathrm{T}} R_d$, ϑ_{de} and ${}^er_{de}$ are the rotation and the unit vector corresponding to ${}^eR_d$, respectively.

$${}^e\varepsilon_{de} = \sin\frac{\vartheta_{de}}{2}{}^er_{de} \tag{3.4}$$

The differential kinematics model is given as:

$$\begin{aligned} v_e &= J(q)\dot{q} \\ J &= \begin{bmatrix} J_p \\ J_o \end{bmatrix} \end{aligned} \tag{3.5}$$

The angular velocity is given as:

$$\dot{R}_e = S(\omega_e) R_e \tag{3.6}$$

In view of the partition of v_e, it is appropriate to partition the vector a into its linear and angular components, i.e. $a = \begin{bmatrix} a_p^{\mathrm{T}} & a_o^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, a_p is linear partition of v_e and a_o is angular partition of v_e, they both are (3×1) vectors:

$$\begin{aligned} \ddot{p}_e &= a_p \\ \dot{\omega}_e &= a_o \end{aligned} \tag{3.7}$$

The dynamic model of the Lagrangian form is given:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = \tau - J^{\mathrm{T}}(q)h \tag{3.8}$$

3.2.1 Indirect Force Control

Considering compliance (also named stiffness) control [124] and impedance control [50], where the contact force is connected to the position inaccuracy via mechanical stiffness or impedance of configurable parameters. An analogous mass-spring-damper system with the contact force as input may be used to describe a robot manipulator under impedance control. The resultant impedance is often nonlinear and coupled in several task space directions. Force feedback can be employed in the control law to obtain a linear and decoupled impedance controller if a force/torque sensor is available.

Compliance control is used to control the interaction's desired static behavior. In order to obtain the desired dynamic behavior, in addition to stiffness, the real–model mass and damping at the contact region must be addressed, resulting in impedance control [139].

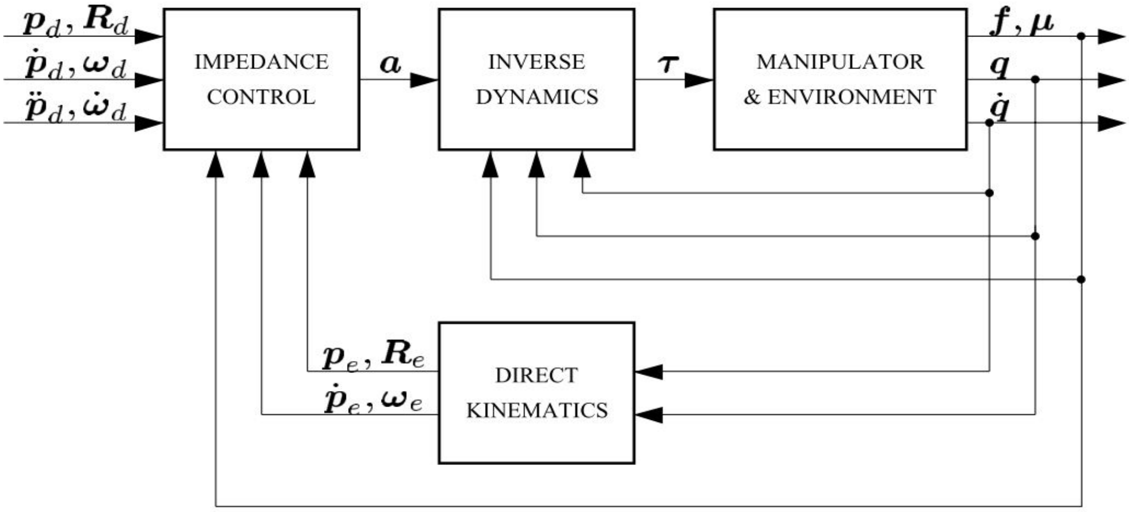

$\boldsymbol{p}_d, \boldsymbol{R}_d$
$\dot{\boldsymbol{p}}_d, \boldsymbol{\omega}_d$
$\ddot{\boldsymbol{p}}_d, \dot{\boldsymbol{\omega}}_d$
IMPEDANCE CONTROL
$\boldsymbol{a}$
INVERSE DYNAMICS
$\boldsymbol{\tau}$
MANIPULATOR & ENVIRONMENT
$\boldsymbol{f}, \boldsymbol{\mu}$
$\boldsymbol{q}$
$\dot{\boldsymbol{q}}$
$\boldsymbol{p}_e, \boldsymbol{R}_e$
$\dot{\boldsymbol{p}}_e, \boldsymbol{\omega}_e$
DIRECT KINEMATICS

Figure 3.2: A block diagram of the indirect force control (Impedance control) [156]. Reprinted Image: ©Springer Nature 2016. Reproduced with permission from Springer Nature.

According to the definition of the notation in Section 3.2, Figure 3.2 chooses

$$\tau = B(q)\alpha + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) + J^{\mathrm{T}}(q)h \tag{3.9}$$

Then the new control input α can be defined as:

$$\alpha = J^{-1}(q)(a - \dot{J}(q,\dot{q})\dot{q}) \tag{3.10}$$

Then,

$$\begin{aligned} a_p =& \ddot{p}_d + K_{M_p}^{-1}\left(K_{Dp}\Delta\dot{p}_{de} + K_{Pp}\Delta p_{de} - f\right) \\ a_o =& T\left(\varphi_e\right)\left(\ddot{\varphi}_d + K_{Mo}^{-1}\left(K_{Do}\Delta\dot{\varphi}_{de} + K_{Po}\Delta\varphi_{de} - T^{\mathrm{T}}\left(\varphi_e\right)\mu\right)\right) \\ &+ \dot{T}\left(\varphi_e, \dot{\varphi}_e\right)\dot{\varphi}_e \end{aligned} \tag{3.11}$$

Finally, with K_{Mp} and K_{Mo} positive definite matrix gains, the closed-loop dynamic behavior is given as:

$$\begin{aligned} K_{Mp}\Delta\ddot{p}_{de} + K_{D_p}\Delta\dot{p}_{de} + K_{Pp}\Delta p_{de} &= f \\ K_{M_o}\Delta\ddot{\varphi}_{de} + K_{D_o}\Delta\dot{\varphi}_{de} + K_{P_o}\Delta\varphi_{de} &= T^{\mathrm{T}}(\varphi_e)\,\mu \end{aligned} \tag{3.12}$$

3.2.2 Direct Force Control

The contact force was indirectly controlled in Section 3.2.1 by appropriately managing the EE motion. It is feasible to secure limiting contact force values for a given preliminary estimate of the environment stiffness in this way. Certain interaction activities, on the other hand, necessitate the achievement of a precise contact force value. In principle, this might be accomplished by fine-tuning the active compliance control action and selecting a suitable intended position for the EE; however, such an approach would only work if accurate modeling of the contact stiffness is known.

A thorough representation of the environment is not accessible in most realistic scenarios. In this scenario, a successful method is inner/outer motion/force control, which involves closing an outer force control loop around an inner motion control loop that is normally accessible in an industrial robot [30]. The intended EE motion can be fed to the inner loop of an inner/outer motion/force control system to integrate the capability of directing motion alongside the unconstrained task directions. The resultant parallel controller consists of a force controller and a motion controller, the former intended to prevail over the latter to ensure force control along with constrained task directives [22].

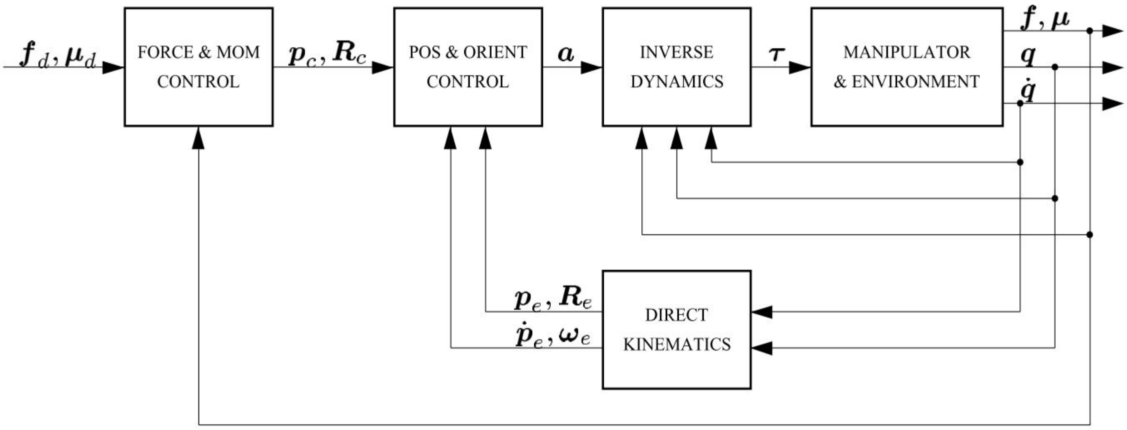


Figure 3.3: A block diagram of the direct force control [139], [156]. Reprinted Image: ©Springer Nature 2016. Reproduced with permission from Springer Nature.

A compliant frame Σ_c is introduced through a force and moment controller as shown in Figure 3.3, the pose error is given as:

$$\begin{aligned} \Delta p_{ce} &= p_c - p_e \\ \Delta\varphi_{ce} &= \varphi_c - \varphi_e \end{aligned} \tag{3.13}$$

f_d and μ_d represent a desired force and a desired moment, respectively.

A proper control action τ is designed to realize the equivalent force and moment γ that drives the EE to the desired position and orientation:

$$\tau = J^{\mathrm{T}}(q)\gamma - K_D\dot{q} + g(q) \tag{3.14}$$

According to the static model-based compensation, γ_p and γ_o can be chosen as:

$$\begin{aligned}\gamma_p &= K_{Pp}\Delta p_{ce} + f_d \\ \gamma_o &= T^{-\mathrm{T}}\left(\varphi_e\right) K_{Po}\Delta\varphi_{ce} + \mu_d\end{aligned} \tag{3.15}$$

Then the force and moment error can be created as:

$$\begin{aligned}\Delta f &= f_d - f \\ \Delta\mu &= \mu_d - \mu\end{aligned} \tag{3.16}$$

Then, a proportional-integral (PI) controller based on the force error and moment error is used to calculate p_c and φ_c:

$$\begin{aligned}p_c &= K_{Pp}^{-1}\left(K_{Fp}\Delta f + K_{I_p}\int_0^t \Delta f \mathrm{d}\varsigma\right) \\ \varphi_c &= K_{Po}^{-1}\left(K_{Fo}\Delta\mu + K_{Io}\int_0^t \Delta\mu \mathrm{d}\varsigma\right)\end{aligned} \tag{3.17}$$

Here, K_{Fp}, K_{Ip}, K_{Fo} and K_{Io} are positive definite matrix gains that can be tuned suitably. The contact force and moment may be regulated to the appropriate values if the control gains are properly adjusted to assure the closed-loop system's stability.

It is important to explore the dynamic model-based compensation to improve the system's performance during the transient, thus, the linear and angular accelerations can be chosen as:

$$\begin{aligned}a_p &= -K_{Dp}\dot{p}_e + K_{Pp}\Delta p_{ce} \\ a_o &= T\left(\varphi_e\right)\left(-K_{Do}\dot{\varphi}_e + K_{Po}\Delta\varphi_{ce}\right) + \dot{T}\left(\varphi_e, \dot{\varphi}_e\right)\dot{\varphi}_e\end{aligned} \tag{3.18}$$

The direct force control (Figure 3.3) is different with indirect force control (Figure 3.2), as the control target is to achieve force regulation, thus the feedforward linear and angular velocity and acceleration of Σ_c is not used.

3.3 Hybrid Force/Position Control

If a thorough model of the environment is provided, a common technique is a hybrid position/force control, which seeks to manage position along unconstrained task directions while managing force along limited task directions. For normally flat contact surfaces, a selection matrix operating on both desired and feedback values fulfills this goal. However, for general curved contact surfaces, explicit constraint equations must be considered [96], [97].

The parallel composition controller can compose the compliant position with the desired position by:

$$p_r = p_c + p_d \tag{3.19}$$

Then, same with the previous two controllers, the static model-based compensation control action can be chosen as:

$$\gamma_p = K_{Pp}(p_r - p_e) + f_d \tag{3.20}$$

Finally, the demand torque τ can be set as:

$$\tau = J_p^{\mathrm{T}}(q)\gamma_p - K_D\dot{q} + g(q) \tag{3.21}$$

which could allow the position control along the unconstrained task directions while generating force error in the constrained task directions [23].

In general, the indirect force control, direct force control and hybrid force/position control introduced in this chapter are the basic controllers for the whole work of this thesis.

Chapter 4

Reinforcement Learning

4.1 Robotic Reinforcement Learning Approaches

RL provides a framework and a collection of tools for designing complex and difficult-to-engineer behaviors in robots. Through trial-and-error interactions with agent's environments, RL allows a robot to automatically determine an ideal behavior [68].

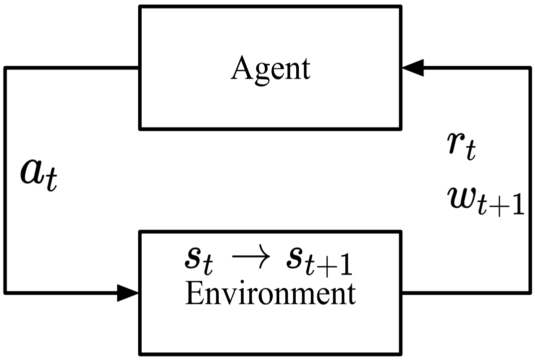


Figure 4.1: A universal model of RL.

The universal model of RL is shown in Figure 4.1, which is physiologically realistic since it is based on learning through punishment or reward as a result of changes in the environment that are either reinforcing or unreinforcing to certain behaviors/actions. The evolutionary pressure of best behavioral adaptation to environmental limitations drives natural RL.

The RL is a machine learning approach for teaching agents to solve different tasks based on trials and errors when interacting with environments. The RL agent aims to learn a policy $\pi(a_t|s_t)$, which selects the action a_t, and meanwhile the agent observes the environment s_t. The transition probability $p(s_{t+1}|a_t,s_t)$ is used to connect the state change over dynamics. The final trajectory can be represented as $\tau = (s_0,a_0,s_1,a_1,...)$. The discount factor γ controls the sum of the reward. An optimal policy π^* should maximize the cumulative reward $r(s_t,a_t)$ during interactions with the environment, as shown in Equation (4.1).

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t (r(s_t, a_t)) \right] \tag{4.1}$$

With the development of expressive function approximation such as neural networks (e.g., Deep RL), high-dimensional inputs such as raw images can be handled [99], [149]. Great success has been gained because of the advances in RL in many fields, for instance, the development of video games such as Atari [99], dexterous hand manipulation [7], robot grasping [64], and robot manipulation [81].

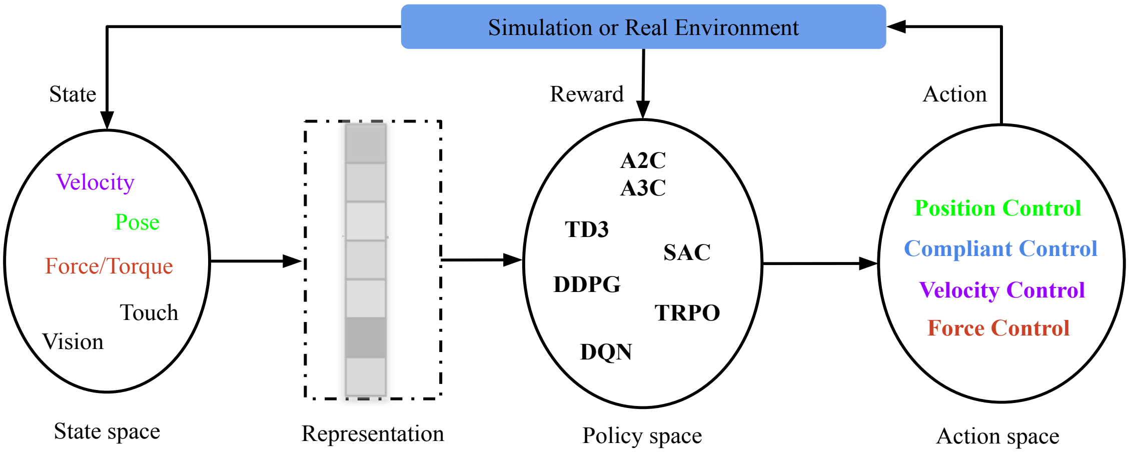


Figure 4.2: A general schematic diagram of robotic manipulation control using Deep Reinforcement Learning (DRL) [105], [87].

There are several categories in which to put RL algorithms. The RL solutions may be broken down into two main types based on their complexity: approximation solution approaches and tabular solution methods. The value function may be expressed in table form and the former techniques are appropriate for straightforward RL problems with constrained state and action areas. The value function must offer a decent estimate throughout the whole state and action set, as the situation becomes more complicated, such as when the state or action space is continuous.

The RL algorithms can be separated into model–free and model–based approaches depending on how they access the environment model, which is utilized to forecast transitions and rewards. The agent may plan the subsequent action sequence in the future using the model that is accessible. Model–based techniques provide the benefit of algorithms with higher sampling efficiency. However, obtaining a ground truth model of the environment is often difficult, which restricts the range of applications.

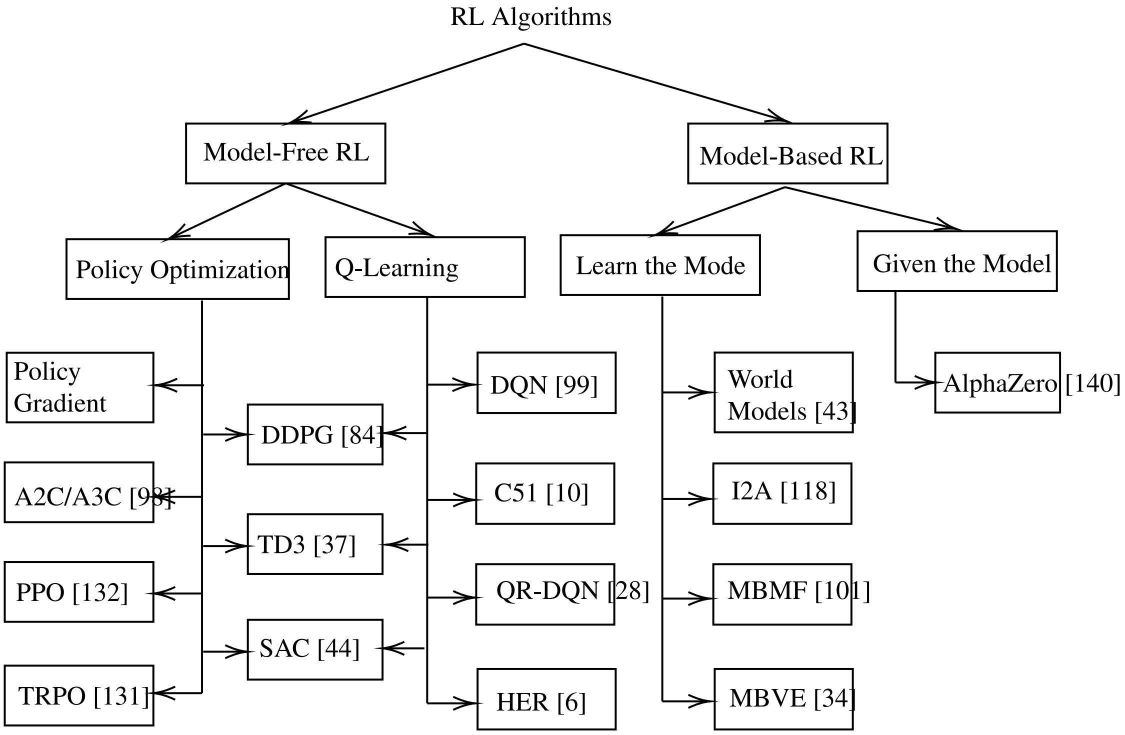


Figure 4.3: The taxonomy of RL algorithms [106].

Figure 4.3 gives a taxonomy of RL algorithms, Q–Learning, DQN, and SAC algorithms are used in this work.

Two of the most important DRL challenges of robotic manipulation are sample efficiency and generalization [87]. In the context of robotic manipulation control, the purpose of DRL is to train a deep neural network policy, to recognize the best command sequence for completing the tasks. The present state space, as shown in Figure 4.2, is the input, which can comprise the pose or the velocity of the EE, or even the force/torque. Furthermore, the current pose of target objects, as well as the status of associated sensors if any are present in the surroundings, can be tallied in the current state space. The policy network's output is an action that specifies control directives for each actuator, such as Cartesian velocity or position commands. In Figure 4.2, some widely used policies can be found as well.

4.2 RL for Contact-Rich Manipulation

Pushing [155], door opening [63], tool use [46], peg-in-hole [56] and related assembly manipulations [51] are examples of tasks that have been handled using RL.

RL offers a set of tools for the design of sophisticated robotic behaviors that are difficult to engineer. RL has been applied previously and has gained great success in solving various of problems in robotic manipulations [77], [89], [128], [56], [90].

Newman et al. [103] inverted the mapping from relative positions to observed moments and trained a neural network to guide a robotic assembly. Inoue et al. [56] used long short-term memory to learn algorithms with two threads (an action and a learning thread) for searching and inserting a peg into a tight hole; however, their methods required several pre-defined heuristics and flat searching surfaces.

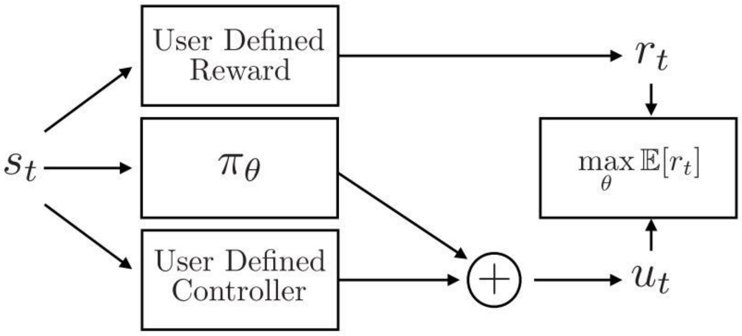


Figure 4.4: A schematic diagram of combining user-defined controllers with a residual RL controller [62]. Reprinted Image: ©2019 IEEE.

Residual RL could exploit the efficiency of conventional controllers and the flexibility of RL. The idea is to try injecting prior information into an RL algorithm to speed up the training process instead of randomly exploring from scratch [61]. In many assembly tasks, conventional controllers could optimize the priory of environment interactions whereas RL could learn fine-grained user-defined controllers. In order to take advantage of the efficiency of hand-engineered controller but also the flexibility of RL controller, the action could be chosen as [61]:

$$u = \pi_H \left(s_{\mathrm{m}}\right) + \pi_\theta \left(s_{\mathrm{m}}, s_{\mathrm{o}}\right). \tag{4.2}$$

Here, $\pi_H \left(s_{\mathrm{m}}\right)$ is the hand-engineered controller (such as a PID controller), the learned policy $\pi_\theta \left(s_{\mathrm{m}}, s_{\mathrm{o}}\right)$ is optimized by an RL algorithm to maximize expected reward on the task.

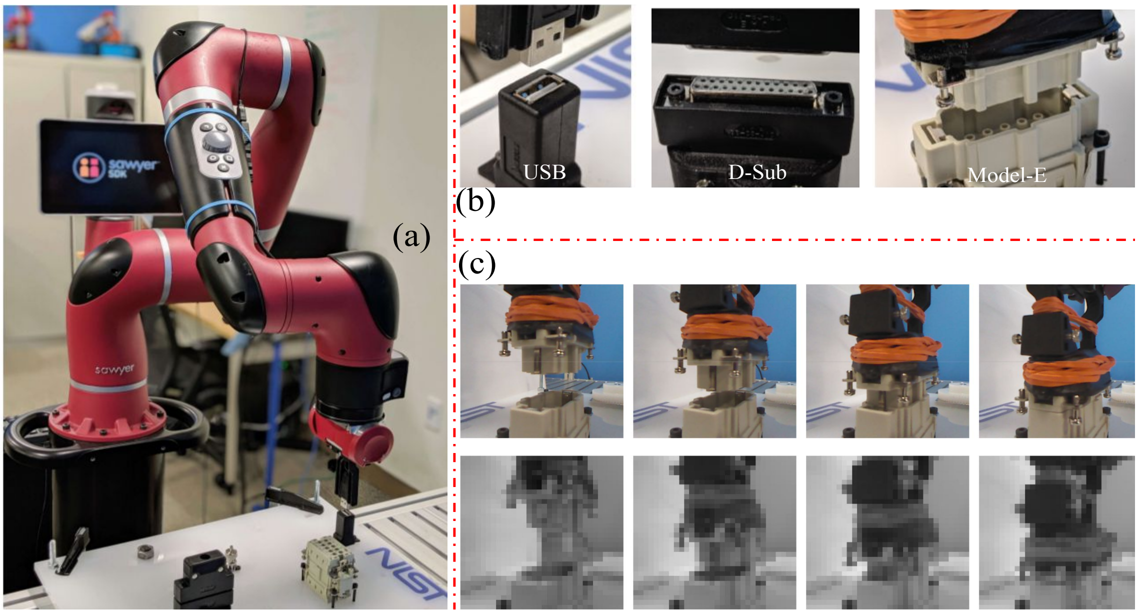


Figure 4.5: (a): A Sawyer robot setup for insertion tasks [129]. (b): Three different types of connectors are used for insertion validation. (c): Model-E connector insertion process with visual state input. Reprinted Image: ©2020 IEEE.

Specifying goals via images makes it possible to specify goals with minimal manual effort such as imaging as shown in Figure 4.5. Schoettler *et al.* consider a number of challenging industrial insertion tasks (USB, D-Sub and Model-E) with visual inputs and a range of natural reward criteria, including sparse incentives and goal pictures [129]. An anti-windup Proportional-Integral-Derivative (PID) control–based joint impedance controller is used to guaranteeing the interaction safety.

A 32 × 32 grayscale image is taken as the state provided to the learned policy as shown in Figure 4.5 (c). Sparse reward function and dense reward function are both used in the research. A dense reward based on the distance to the target position is:

$$r_t = -\alpha \cdot \|x_t - x^*\|_1 - \frac{\beta}{(\|x_t - x^*\|_2 + \varepsilon)} - \varphi \cdot f_z \tag{4.3}$$

x^* is the target location, where $0 < \varepsilon \ll 1$. The hyperparameters settings are: $\alpha = 100, \beta = 0.002, \varphi = 0.1$. When the connector is inserted, the sign of the force term flips by setting $\varphi = -0.1$.

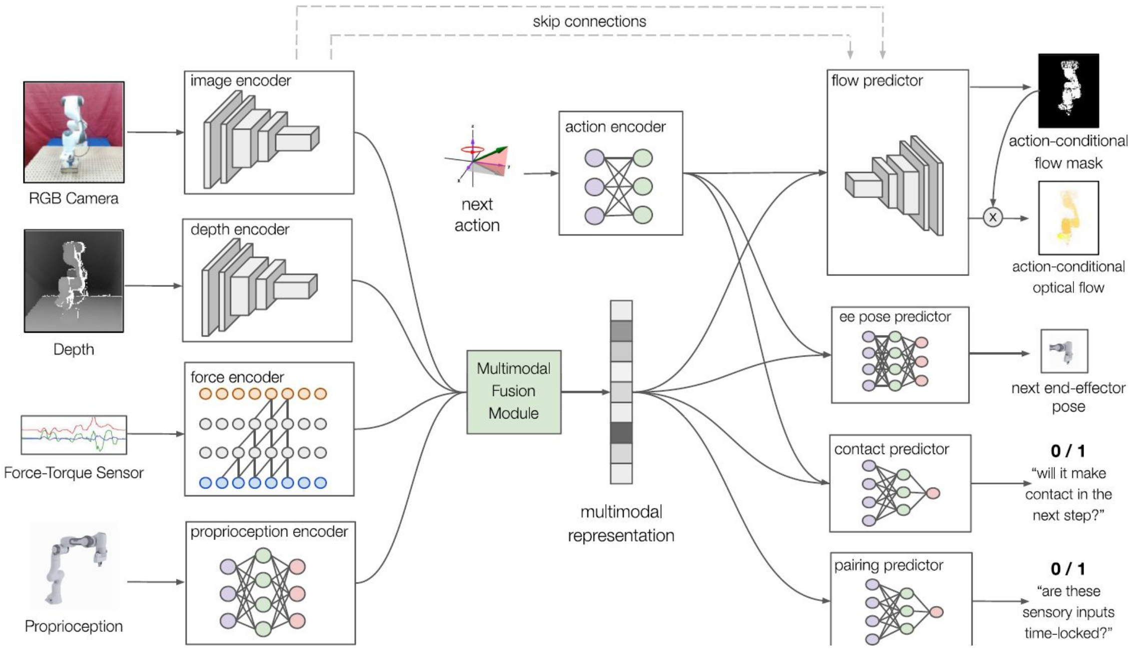


Figure 4.6: Architecture of multimodal representation learning with self-supervision. visual information including RGB and depth images, force and torques, as well as EE position, orientation and velocity are encoded into a multimodal representation using a variational encoder [78].Reprinted Image: ©2020 IEEE.

Combining the sense of vision and touch could endow robots with a similar ability as humans to facilitate the assembly tasks as shown in Figure 4.6 [78], which could provide robustness to the sensor and actuator noises [128] as well as position uncertainty. Lee *et al.* [78] demonstrate a peg insertion task with haptic and visual input for hole search, peg alignment and insertion. Moreover, a new variational representation learning technique is proposed and significantly expands the experimental evaluation of the overall methodology. In this research, a dense four-stage (reaching, aligning, inserting, and completed) reward function is designed as:

$$r(\mathbf{s}) = \begin{cases} c_r\left(1-(\tanh\lambda\,\|\mathbf{s}\|_2\right)\left(1-s_\psi\right) & \\ 1+c_a\left(1-\frac{\|\mathbf{s}\|_2}{\|\varepsilon_1\|_2}\right)\left(1-\frac{s_\psi}{\varepsilon_\psi}\right) & \text{if } \mathbf{s}\leq\varepsilon_1\&s_\psi\leq\varepsilon_\psi \\ 2+c_i\left(h_d-|s_z|\right) & \text{if } s_z<0 \\ 5 & \text{if } h_d-|s_z|\leq\varepsilon_2 \end{cases} \tag{4.4}$$

The definitions are:

- s_ψ: the current relative orientation along the z-axis between the peg and the hole.
- $s=(s_x,s_y,s_z)$: the peg's current relative position to the hole.
- $(0,0,h_d)$: target peg position, h_d is the height of the hole.
- λ is a constant factor to scale the input to the tanh function.
- c_r and c_a are constant scale factors.

However, only a few studies have focused on real industrial production contact-rich tasks, and the aforementioned methods always require a sliding surface for the hole search algorithms [77], [56], [76]. Moreover, agent training in the real world has low efficiency and always has safety issues [48].

4.3 Sim-to-real Transfer of RL

One of the most important factors limiting the use of RL in robot manipulation is sample inefficiency. Due to sampling inefficiency, even the greatest existing RL algorithms may be unfeasible. There are several causes for the issue [105]:

- many algorithms attempt to learn to accomplish a task from scratch, which necessitates a large amount of data.
- algorithms are currently insufficient for extracting relevant data from current data. For some on-policy algorithms, each update step requires new data.
- data collecting in robots can take a long time.

Another important difficulty is safety. Learning contact-rich manipulations directly in a physical robot system via RL is unsafe because the exploration required for learning might generate large contact force in high stiffness environment interactions. In physical systems, methods such as torque control [80], impedance control with restricted stiffness [46], or explicit action limitation [72] are used to reduce the potential for damages.

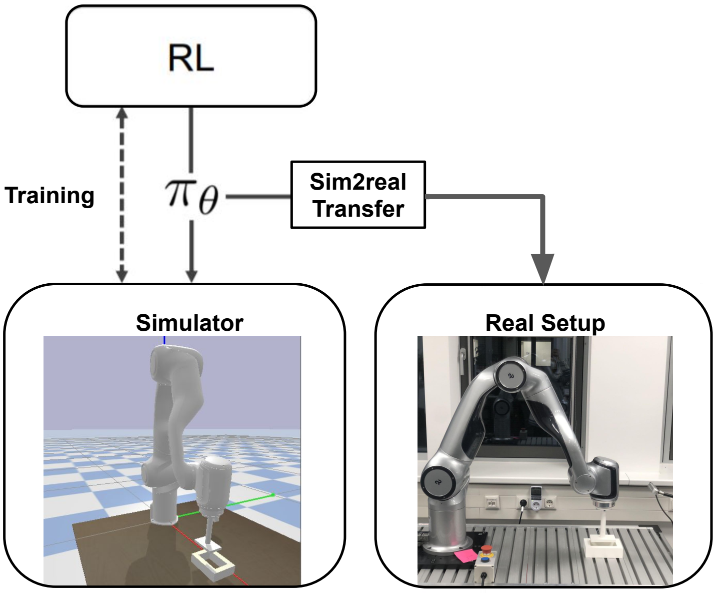


Figure 4.7: A sim2real concept [19].

Policies can be learned in a simulation environment before being executed in the actual system, as opposed to learning in a real system, as shown in Figure 4.7. This enables the investigation to be carried out securely in simulation while also providing access to a large amount of training data. The technique, however, creates a new problem in the shape of a reality gap between the simulation and the physical system.

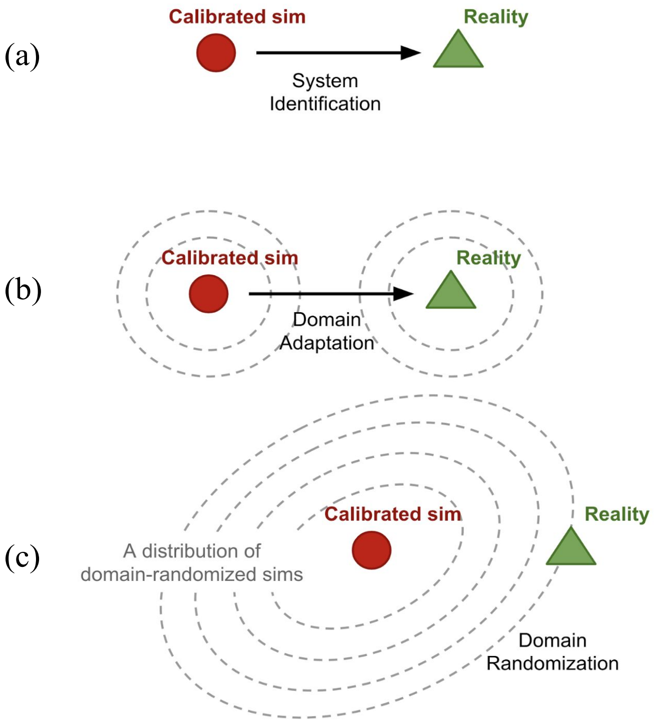


Figure 4.8: Conceptual illustrations of three approaches for sim2real transfer [160]. (a): system identification. (b): domain adaptation and (c): domain randomization. This figure has the original author's use and editing authorization.

As shown in Figure 4.8, the gap can be narrowed by

- Figure 4.8 (a): employing system identification to calibrate the simulation [66];

  The goal of system identification is to create an accurate mathematical representation of a physical system in order to improve the simulator's realism.

- Figure 4.8 (b): training them with simulated noise (domain adaptation) [155] to increase the policies resiliency, [8];

  Domain adaptation approaches employ data from a source domain to improve the performance of a learned model on a target domain with fewer data.

- or Figure 4.8 (c): with a known range of simulator parameters (domain randomization), [152].

  Domain randomization is the concept of heavily randomizing a simulation in order to cover the true distribution of real-world data.

Moreover, a simulation-trained policy can also be fine-tuned in the actual world, such as by employing meta-learning to develop flexible policies [130].

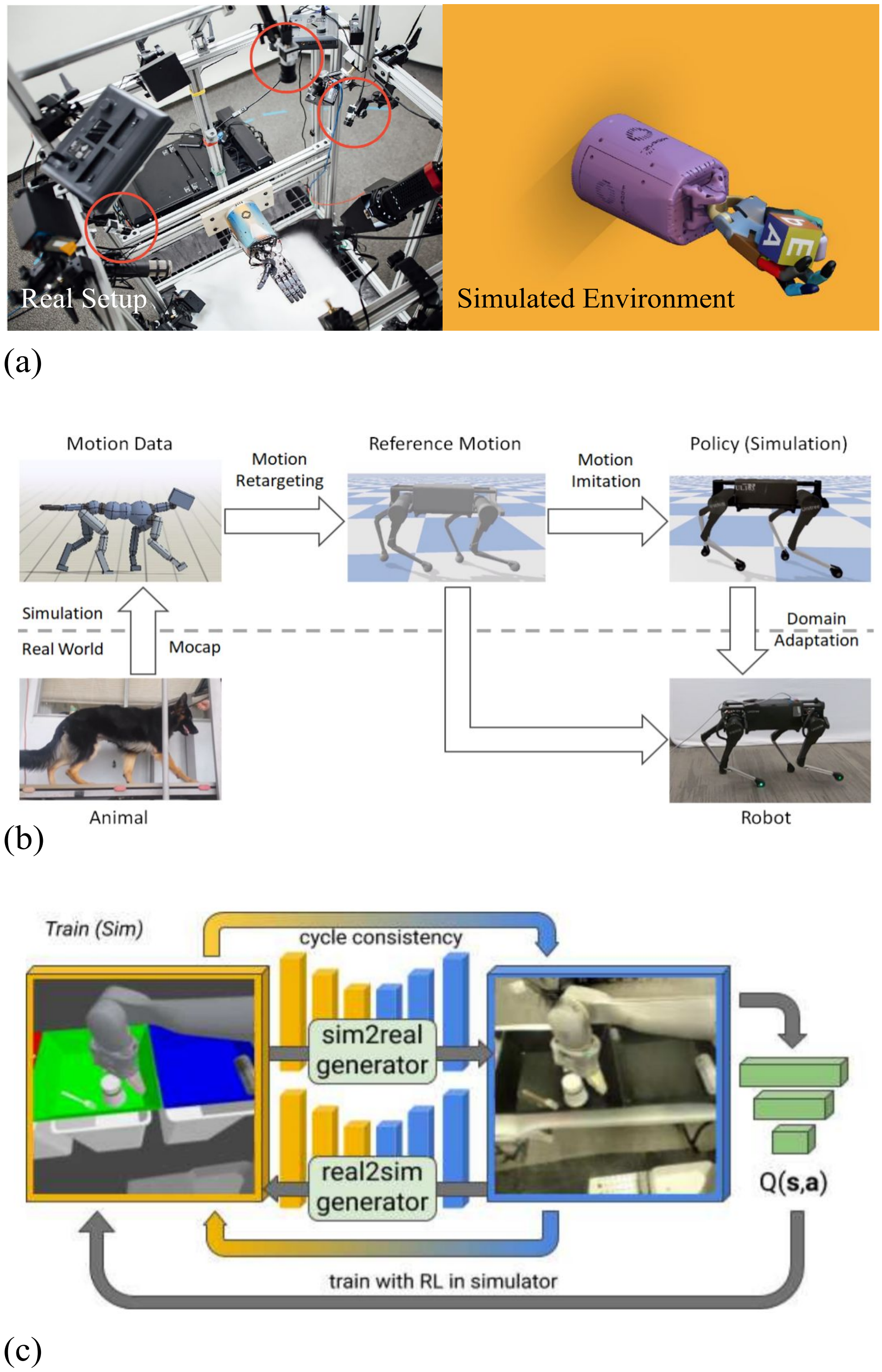


Figure 4.9: Various state-of-the-art applications of Sim2Real in robotics manipulation. (a): RL is used to learn dexterous in-hand manipulation policies to perform vision-based object reorientation on a physical Shadow Dexterous Hand [7]. (b): A single learning-based technique may automatically synthesize controllers for a varied range of actions for legged robots by using reference motion data [114]. (c): RL-CycleGAN is a method for transferring RL from simulation to the real world that eliminates the requirement for task-specific feature engineering [119]. Reprinted Image: ©2020 IEEE.

In Figure 4.9, some of the studies carried out in recent years are highlighted.

Andrychowicz *et al.* [7] used RL to train dexterous in-hand manipulation policies on a real Shadow Dexterous Hand that can execute vision-based object reorientation as shown in Figure 4.9 (a). Many physical parameters of the system, such as friction coefficients and the appearance of an item, are randomly generated throughout the training. Despite being taught exclusively in simulation, their policies are eventually translated to an actual robot.

Peng *et al.* [114] proposed a learning framework that consists of three stages (Figure 4.9 (b)):

- motion retargeting. The motion retargeting stage processes the reference motion first, employing inverse-kinematics to convert the motion clip's morphology from that of the original subject to that of the robot.
- motion imitation. In the motion imitation stage, the retargeted reference motion is used to train a policy to reproduce the motion with a simulated robot model.
- domain adaptation. The sample efficient domain adaptation procedure adapts the policy's behavior using a learned latent dynamics representation before it is transmitted to a real robot.

This framework takes motion data from an animal as input and generates a control strategy that allows an actual robot to mimic the motion.

Rao *et al.* [119] proposed RL-CycleGAN as a new approach for sim2real transfer for RL as shown in Figure 4.9 (c). This method is based on combining CycleGAN [167] with a Q-learning model [157]. RL-CycleGAN uses a jointly learned RL model to train a GAN that is encouraged to distinguish between styles and semantics. In theory, the output of the RL model should only be determined by the task's semantics, thus restricting the GAN with the RL model encourages the GAN to retain task-specific semantics.

Part II

DRL for the Force-Controlled Robotic Manipulation

Chapter 5

Force-Controlled Robots

There is a famous phrase in China: no practice, no gain in one's wit, which means what's learned from books is superficial after all; it's crucial to have it personally tested somehow. Therefore, the experience of being involved in developing a commercial robot is quite valuable. This chapter presents the main knowledge and innovations during the development of the torque-controlled robot including system design, joint design, joint space controller, Cartesian space controller, and robotic skills development and validation.

5.1 Torque-controlled Robot Development

The characteristics of traditional robots are [3]:

- high positioning accuracy (repeatability and absolute accuracy);
- high speed;
- durability;
- robustness;
- and the relatively low price.

Torque-controlled robots have been developed for unstructured environments that are fundamentally different from the environments where classical industrial robots have been used.

The DLR 7 DoF LWR systems developed in the 1990s at DLR are designed for interaction with humans and objects in unstructured environments. The LWR robots are designed for application areas that are generally not covered by industrial robots such as assembly processes, human-robot cooperation, and service robotics.

The characteristics of LWR robots are [3], [4], [5], [107]:

- compensate the effects of the robot elasticity;
- robust performance (with respect to positioning and model uncertainty);
- active vibration damping;

- compliance and force/ torque control;
- collision and failure detection;
- active safety for the human and the robot.

Agile Robots AG [1] attempts to combine the advantages of industrial robots and LWR robots. Agile Robots AG is a spin-off of the DLR who is looking to push the boundaries of robotics [2]. The mission of Agile Robots AG's is to bridge the gap between Artificial Intelligence (AI) and robotics, the company has developed SOTA full-body force sensitivity robots and world-leading vision intelligence products.

5.1.1 System Overview

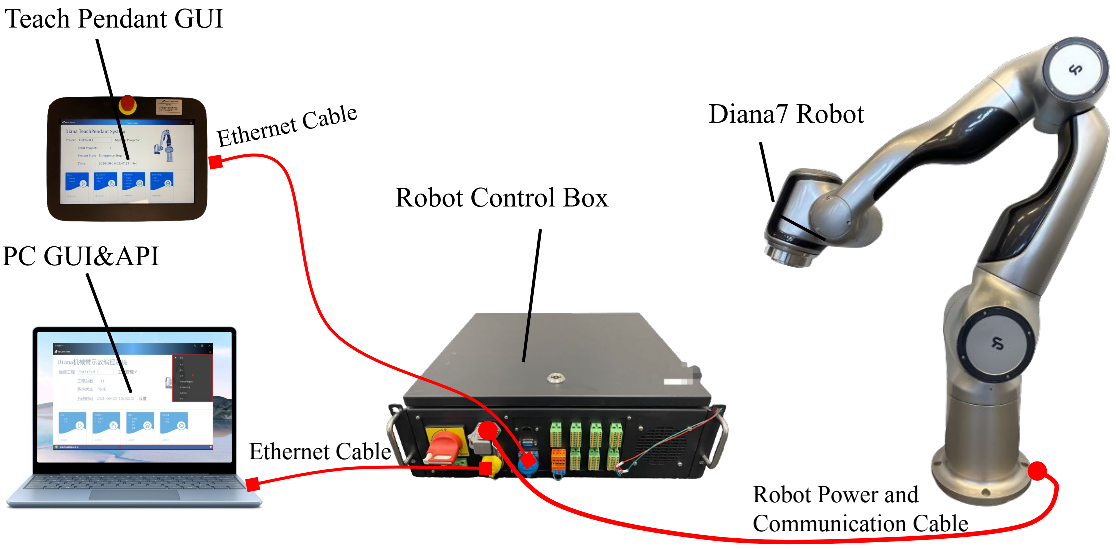


Figure 5.1: Diana7 robot system overview [3].

The Diana 7 robot system consists of a robot body and a robot control box (either the alternating current (AC) power control box CB2T or the direct current (DC) power control box CB2TD). The Diana 7 robot has seven rotating joints, which are connected by connecting rods. There are in total 7 degrees of freedom including the base (joint 1), shoulder (joint 2, joint 3), elbow (joint 4), and wrist (joint 5, joint 6, joint 7). The base connects the foundation to the robot body, and the robot head flange connects the robot head to the tool.

5.1.1.1 Software Design

The Diana7 robot control structure can be seen as Figure 5.2. The whole software system is consisted by non-realtime API and GUI, real-time robot control unit and real-time joint control units.

[1] https://www.agile-robots.com/

[2] https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-17675/

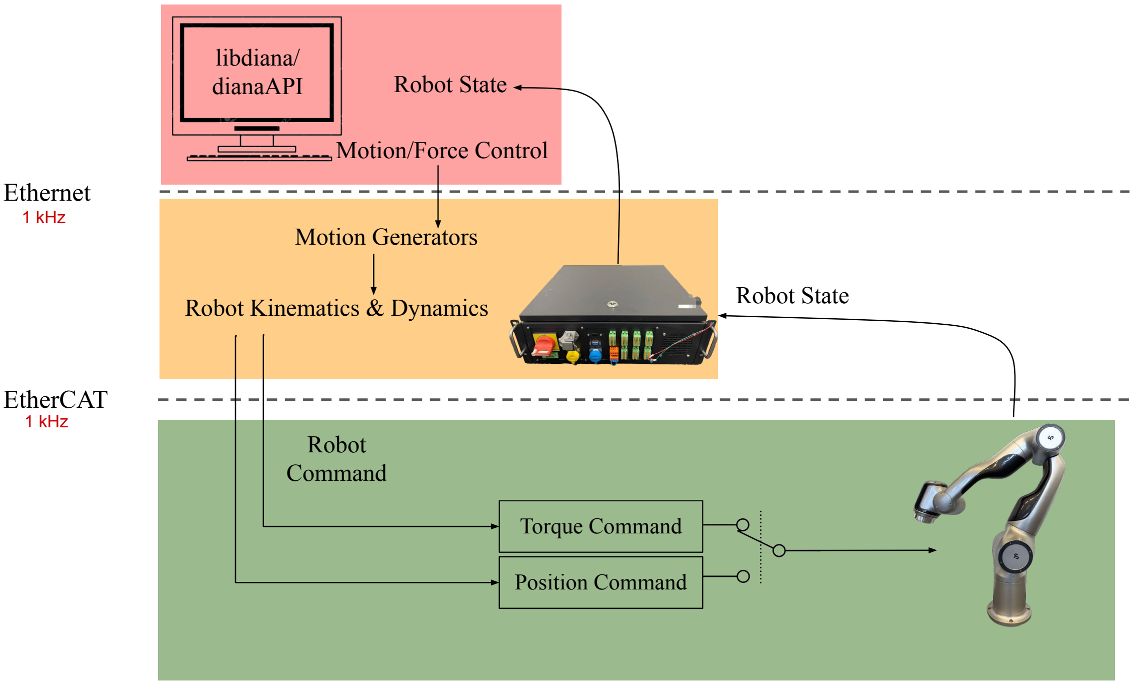


Figure 5.2: Diana7 robot control structure. The red part is non-realtime which includes the Application Programming Interface (API) and GUI. The yellow part is the real-time robot control unit which includes the motion generators as well as kinematic and dynamic functions. The green part is the joint control units that execute the torque and position commands.

The Diana7 robot API and GUI commands are TCP/IP-based that can provide robot control and configuration as:

- execute non-realtime commands to set the robot system parameters;
- perform manual movement control of the robot;
- execute real-time commands at 1 kHz control loops;
- check the robot status from sensors and internal controllers at 1 kHz;
- access the robot model library to compute the desired kinematic and dynamic model.

The Diana7 robot GUI was designed by the software team, I was not involved in the GUI development, thus more GUI details will not be presented.

In the user manual of API (version 2.5.1), there are 136 API functions in total. However, the mainly used API functions in this thesis are:

- `setJointCollision`, `setCartCollision`, `setCartImpedance` and `changeControlMode` commands are used to set the robot system safety and behavior parameters;
- `getTcpPos`, `getJointPos` functions are frequently used to get the robot kinematic state;

- `enterForceMode`, `moveJ` and `moveLToPose` can be used to control the robot in force, joint position, or Cartesian impedance mode, respectively;

In order to support the promotion of educational robots, the Robot Operating System (ROS) control interface of Diana7 was also developed.

5.1.1.2 Hardware Design

A Diana7 robot can be thought of as a combination of seven joints and links. Diana7 robot joints are designed in three sizes in order to fit different joint torque requirement while all design has similar concepts. The design concepts and details are presented as follows.

A robotic joint with harmonic drive, motor side and link side encoder, as well as torque sensor, was developed [21]. A special design of the Diana 7 robot joint is the torque sensor location. The torque sensor is fixed between the flexible spline and joint shell, which is different from the design of LWR [3] and Franka Emika robot [4] that fix the torque sensor on the output of circular spline.

Figure 5.3 gives the internal force and torque transmission analysis of the design. In this design, the motor output is connected with the wave generator thus the wave generator is the input of the Harmonic drive, the flexible spline is fixed, and the circular spline is the output of the Harmonic drive.

The external torque is measured as follows: first of all, the external torque from the link acts on the circular spline, and the forces generated as the teeth "slide" against each other will cause the gear teeth in the meshing zone to align and push the circular spline to rotate, circular spline gear-tooth normal force $F_{c-tooth-normal}$ and flexible spline gear-tooth normal force $F_{s-tooth-normal}$ are paired as action force and reaction force (i.e. $F_{c-tooth-normal} + F_{s-tooth-normal} = 0$); then, the external torque is transmitted to the flexible spline by the gear-teeth; as the torque sensor is fixed with flexible spline, thus the external torque is transmitted and measured, and vice versa if we analysis from the motor torque input side.

The advantages of this design concept are:

- avoid the collision during the assembly of the joints as the torque sensor is installed inside the joint;
- avoid external impact torque acting directly on the torque sensor;
- no need to consider bending torque effects, simplifying the design of torque sensors;
- separate the torque sensor power and signal cable with joint motor cables, to avoid signal disturbance.

The disadvantages are:

- torque measurement is affected by link side friction;

[4] https://www.franka.de/robot-system

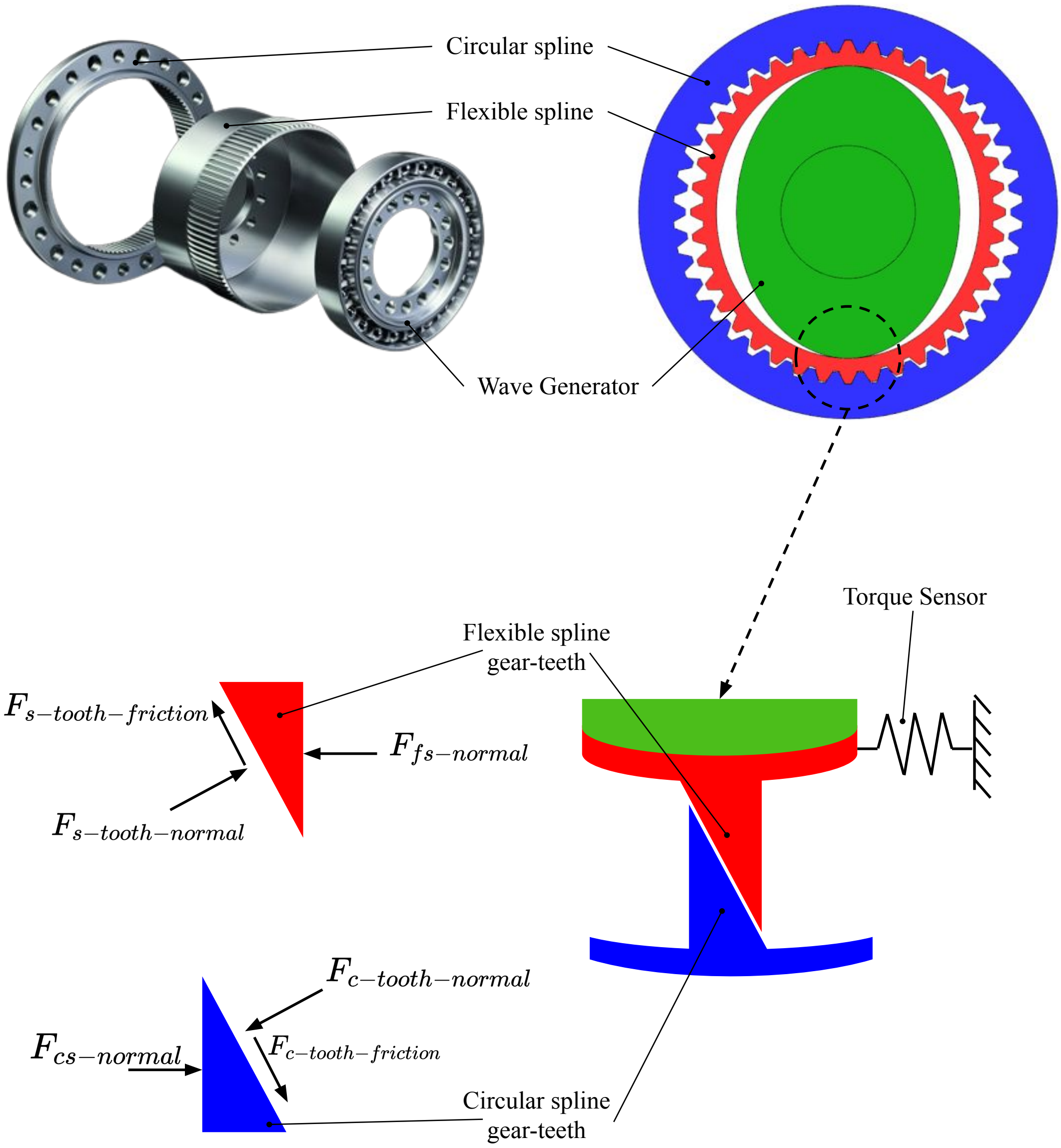


Figure 5.3: Force and torque transmission during the rotation of the Harmonic drive reducer.

- friction between flexible spline gear-teeth circular spline gear-teeth cannot avoid; the larger the torque, the greater the impact; also the faster the speed of rotation, the greater the friction [154].

The friction plays a key role in this design to attach the torque sensor solidly with the flexible spline to avoid the backlash issue. A High-speed side magnetic encoder and a Low-speed side magnetic encoder are used to measure the motor position and link position. A motor side brake with the appropriate friction torque is used to ensure that the joints are locked but can be forced to rotate when the robot is powered off.

5.1.2 Impedance Controller

For a flexible joint robot, for example, equipped with the harmonic drive reducer and torque sensor, the robot model can be assumed as [145]:

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= \tau + \tau_{ext} \\ B\ddot{\theta} + \tau &= \tau_m \end{aligned} \tag{5.1}$$

The definitions are as follows:

- $q \in \mathbb{R}^n$: the vector of link side joint angles;
- $\theta \in \mathbb{R}^n$: the vector of motor angles;
- $\tau \in \mathbb{R}^n$: the joint torques are determined by the linear relationship $\tau = K(\theta - q)$; $K \in \mathbb{R}^{n\times n}$ is a diagonal matrix containing the individual joint stiffness, which consists of a series connection of reducer stiffness K_{re} and torque sensor stiffness K_{ts};
- $B \in \mathbb{R}^{n\times n}$: the diagonal matrix, which consists of the rotor inertias B_i;
- $M(q) \in \mathbb{R}^{n\times n}$: the manipulators (link side) mass matrix;
- $C(q,\dot{q})\dot{q}$ represents the centrifugal and Coriolis-terms of the link side rigid body part of the model;
- $g(q) \in \mathbb{R}^n$: the vector of gravity torques;
- $\tau_m \in \mathbb{R}^n$: the motor torques command for the control;
- $\tau_{ext} \in \mathbb{R}^n$: the external forces and torques vector act on the robot.

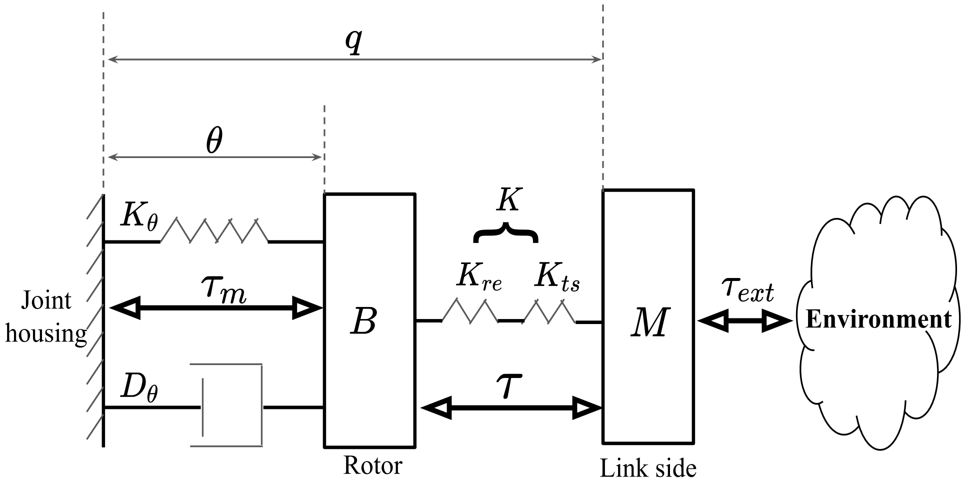


Figure 5.4: A single joint PD-Control based on motor position.

With a traditional PD-controller, the key joint structure and parameters can be seen as Figure 5.4.

However, the inherent flexibility introduced into the flexible joints by harmonic drive reducer and torque sensor can cause vibrations when a traditional PD-controller is used. In order to provide safe, reliable, and robust manipulation when in contact with unknown passive environments using the flexible joint robot, an impedance controller was introduced into robotics instead of the position controller [5].

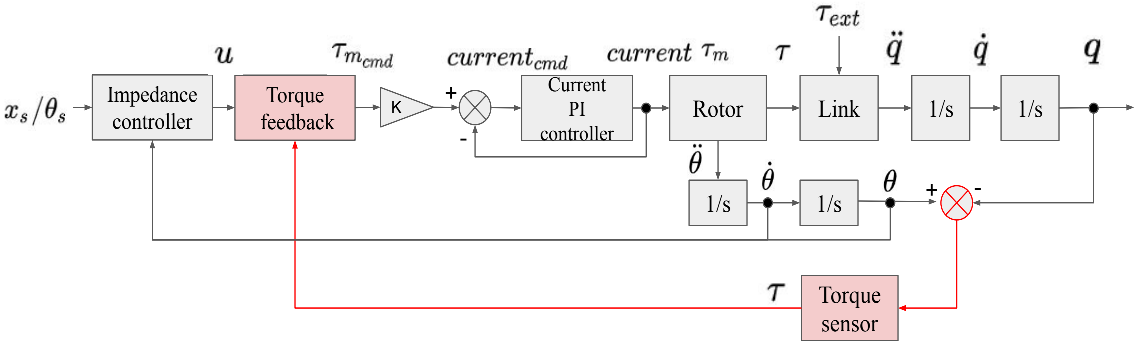


Figure 5.5: A torque feedback-based impedance control structure.

A passivity-based impedance approach [5] that only relies on the joint motor position (the link side position is estimated by joint torque and mechanical stiffness) and joint torque signals provide a high degree of robustness to unmodeled robot dynamics and in the contact with unknown environments is shown in Figure 5.5. The inner torque feedback loop (red part in Figure 5.5) is used for the design of impedance controllers, more details about impedance controllers (joint space and Cartesian space) will be explained in the next sections.

5.1.2.1 Joint Space Impedance Controller

For the joint space and Cartesian space impedance controller, the torque sensor in each joint plays a key role. The basic controller consists of a torque feedback loop, and this torque feedback controller can scale the motor inertia B to the desired value B_θ [107]:

$$\tau_m = BB_\theta^{-1}\tau_u + (I - BB_\theta^{-1})\tau \tag{5.2}$$

Where τ_u is an intermediate control input that could shape the Cartesian or joint impedance behavior [4], and τ is the joint torque data measured by the torque sensor. τ_m is the torque on demand of the motor controller. B is real rotor inertia and B_θ^{-1} is the desired apparent rotor inertia.

BB_θ^{-1} ratio is mainly determined by the torque sensors' noise level and structure frequency. For a single joint compliant behavior, BB_θ^{-1} ratio can be set up to 50, however, for the compliant joint controller on the Diana7 robot, 4 to 6 is a proper setting. Lower ratio values are chosen for high stiffness mode, and vice versa.

For the impedance behavior of joint coordinates, we have

$$\tau_u = -K_\theta(\theta - \theta_s) - D_\theta\dot{\theta} \tag{5.3}$$

wherein $\tau_u \in \mathbb{R}^n$ is a joint demand torque vector, $K_\theta = \operatorname{diag}(k_i) \in \mathbb{R}^{n\times n}$ is a positive definite stiffness matrix and $D_\theta = \operatorname{diag}(d_i) \in \mathbb{R}^{n\times n}$ is a damping matrix, $\theta_s \in \mathbb{R}^n$ is a desired robot configuration, $\overline{g}(\theta) \in \mathbb{R}^n$ is the gravity torque vector.

We combine Equation (5.1) and Equation (5.2), then we have:

$$B\ddot{\theta} + \tau = BB_\theta^{-1}\tau_u + \left(I - BB_\theta^{-1}\right)\tau \tag{5.4}$$

Replace τ_u with Equation (5.3), the new robot closed loop equations are:

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= \tau + \tau_{ext} \\ B_\theta\ddot{\theta} + D_\theta\dot{\theta} + K_\theta(\theta - \theta_s) + \tau &= \mathbf{0} \end{aligned} \tag{5.5}$$

Note: another way to consider the Equation (5.2) is to take the controller as a P controller:

$$\begin{aligned} \tau_m &= BB_\theta^{-1}\tau_u + \left(I - BB_\theta^{-1}\right)\tau \\ &= BB_\theta^{-1}(\tau_u - \tau) + I\tau \\ &= I\tau_u + (BB_\theta^{-1} - I)(\tau_u - \tau) \end{aligned} \tag{5.6}$$

Now it is clear to find that Equation (5.6) is a P controller with feedforward τ_u. This explanation would help researchers to understand Equation (5.2) when tuning the torque controller.

5.1.2.2 Cartesian Space Impedance Controller

In real scenarios, some assembly applications need desired impedance behavior in Cartesian space $x \in \mathbb{R}^n$.

For Cartesian impedance behavior, we have

$$\begin{aligned} \tau_u &= -J(\theta)^T(K_x\tilde{x}(\theta) + D_x\dot{x}(\theta)) + \overline{g}(\theta) \\ \tilde{x}(\theta) &= f(\theta) - x_{des} \\ \dot{x}(\theta) &= J(\theta)\dot{\theta} \end{aligned} \tag{5.7}$$

wherein $\tau_u \in \mathbb{R}^n$ is a joint demand torque vector, K_x and D_x are the permutation and diagonal matrices of desired stiffness and damping, respectively. $x_{des} \in \mathbb{R}^n$ is the desired EE pose, and $x(\theta) = f(\theta)$ is the EE pose computed based on the motor position. $J(\theta) = \partial f(\theta)/\theta$ is the manipulator Jacobian. θ is the measured motor positions, $\overline{g}(\theta) \in \mathbb{R}^n$ is the gravity vector.

Please notice that due to the gravity issue, for a desired link side position q_s which corresponds to the desired Cartesian position x_s should be modified as $x_s = f\left(q_s + K^{-1}g\left(q_s\right)\right)$ instead of $x_{q,s} = f\left(q_s\right)$.

Thus, the new closed-loop system is:

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= \tau + \tau_{\text{ext}}\,, \\ B_\theta\ddot{\theta} + J(\theta)^T\left(K_x\tilde{x}(\theta) + D_x\dot{x}\right) + \tau &= \mathbf{0} \end{aligned} \tag{5.8}$$

Note: due to the system design such as control theory, control frequency and delay, joint mechanical stiffness, signal noise level and so on, the maximum Cartesian space stiffness is generally less than 10000N/m.

For damping design, two methods are proposed in [4]: factorization design and double diagonalization design.

First of all, some assumptions need to be defined:

- $K \to \infty$, which means $q \approx \theta$;
- the mass matrix $M(q)$ is changing slowly, thus the derivative can be neglected.

Then, the approximately closed-loop dynamics can be used for the damping design:

$$\begin{aligned}\Lambda(\theta)\ddot{\tilde{x}}(\theta)+D_x\dot{\tilde{x}}(\theta)+K_x\tilde{x}(\theta)&=\mathbf{0}\\ \Lambda(\theta)&=\left(J(\theta)\left(M(\theta)+B_\theta\right)^{-1}J(\theta)^T\right)^{-1}\end{aligned}\tag{5.9}$$

It is obvious that the damping matrix D_x cannot be constant, because it has to be calculated as a function of $\Lambda(\theta)$. A well-defined variable damping matrix leads to significant performance improvements in practice than a constant one.

Take factorization design as an example, the key idea is set:

$$\begin{aligned}&D_x=AK_{x1}+K_{x1}A\\ &AA=\Lambda\\ &K_{x1}K_{x1}=K_x\end{aligned}\tag{5.10}$$

Then combine with Equation (5.9):

$$A\left(A\ddot{\tilde{x}}(\theta)+K_{x1}\dot{\tilde{x}}(\theta)\right)+K_{x1}\left(A\dot{\tilde{x}}(\theta)+K_{x1}\tilde{x}(\theta)\right)=\mathbf{0}\tag{5.11}$$

With the substitution $A\dot{\tilde{x}}(\theta)+K_{x1}\tilde{x}(\theta)=w$ïijŽ

$$\begin{aligned}A\dot{\tilde{x}}(\theta)+K_{x1}\tilde{x}(\theta)&=w\\ A\dot{w}+K_{x1}w&=\mathbf{0}\end{aligned}\tag{5.12}$$

Then a general damping design can be:

$$D_x=AD_\xi K_{x1}+K_{x1}D_\xi A\tag{5.13}$$

Here, $D_\xi=\operatorname{diag}\{\xi_i\},(0\le\xi_i\le1)$ is a diagonal matrix, and $\xi_i=0$ for undamped behaviour, $\xi_i=1$ for real eigenvalues.

5.1.2.3 Joint Torque Based Friction Observer and Compensation

The impacts of joint friction can have a significant impact on the system performance for robots with high gear ratios (such as 100:1) striving for low own weight and high payload. A joint friction test curve can be seen as Appendix A.1. As the joint torque sensor is installed after the gearbox, then the external force and friction can be distinguished easily. Thus, the friction observer based on joint torque measurement is a proper choice for torque-controlled robots.

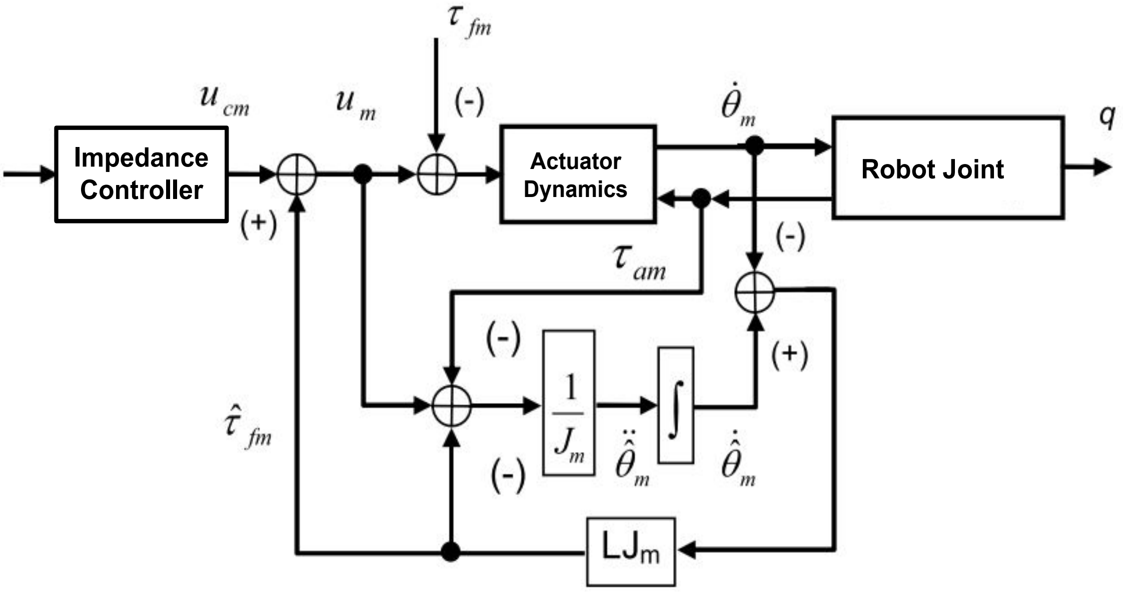


Figure 5.6: Friction observer and compensation structure for a single Joint [74]. Reprinted Image: ©2008 IEEE.

Consider the actuator dynamics:

$$u_m = J_m \ddot{\theta}_m + \tau_{am} + \tau_{fm} \tag{5.14}$$

Here,

- u_m is the motor torque;
- θ_m is the motor position;
- J_m is the motor inertia;
- τ_{am} is joint torque;
- τ_{fm} is the friction torque.

A standard friction model containing Coulomb friction is considered:

$$\tau_{fm} = \tau_{fm,c} + \tau_{fm,v} = f_c \operatorname{sign}\left(\dot{\theta}_m\right) + f_v \dot{\theta}_m \tag{5.15}$$

According to Figure 5.6, the observer dynamics is designed as:

$$\begin{aligned} u_m &= J_m \ddot{\hat{\theta}}_m + \tau_{am} + \hat{\tau}_{fm} \\ \hat{\tau}_{fm} &= -LJ_m \left(\dot{\theta}_m - \dot{\hat{\theta}}_m\right) \end{aligned} \tag{5.16}$$

$L > 0$, $\hat{\tau}_{fm}$ and $\dot{\hat{\theta}}_m$ are the estimation of the friction and the observer state, respectively. As the measurement of motor position and joint torque is easy to get, thus the friction observer can be a simple structure.

Note: after the parameter L is fixed (means the friction is estimated properly), a scale (0.5–0.9) should be used to reduce the $\hat{\tau}_{fm}$ in order to guarantee the controller passivity and avoid overcompensation. After the friction compensation is activated, the joint can be moved easier in zero-force mode compared with non-friction compensation.

5.2 Force Controller Evaluation and Implementation

5.2.1 Force Controller Evaluation

As Diana 7 is a force-controlled robot, thus the Cartesian space force control accuracy test is described as follows.

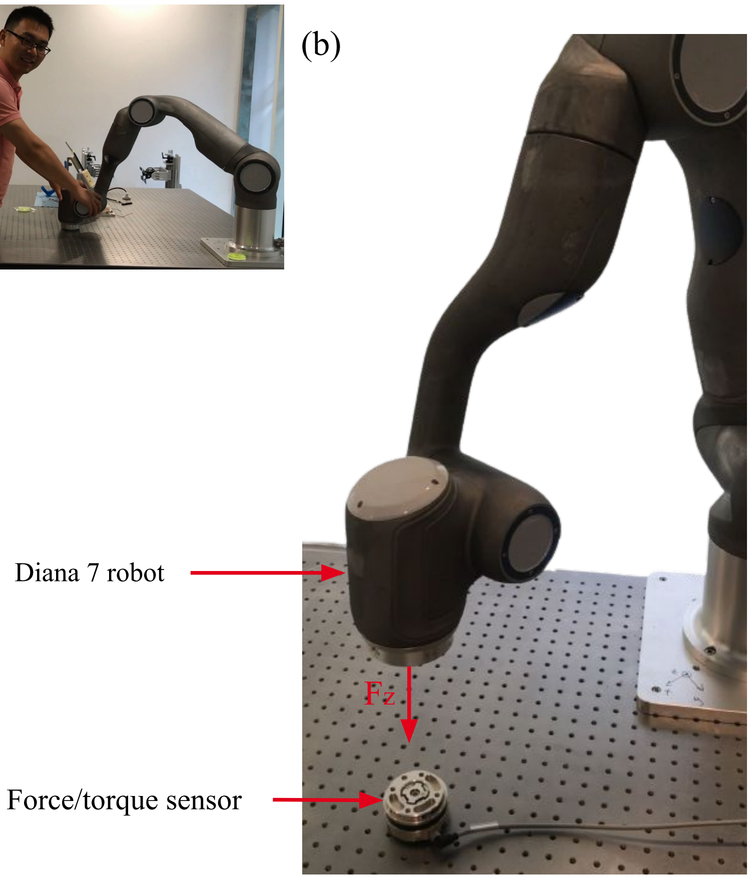


Figure 5.7: (a): Human-Robot interaction under zero gravity mode with a Diana 7 prototype (The robot shell hasn't even been painted yet). (b): Diana 7 force control accuracy evaluation setup.

A Diana 7 robot force control accuracy evaluation is executed as shown in Figure 5.7 (b). A 6 DoF force/torque sensor axia80 [5] is used as a force evaluate reference and the force data is read by TwinCAT3 [6]. The axia80 force/torque sensor is fixed on the table and the robot executes the force command $F_z = 2N, N = (1,2,3...15)$ to press the force/torque sensor for a period of time. Therefore, a step force curve can be obtained.

[5] https://www.ati-ia.com/index.aspx

[6] https://www.beckhoff.com/en-en/products/automation/twincat/

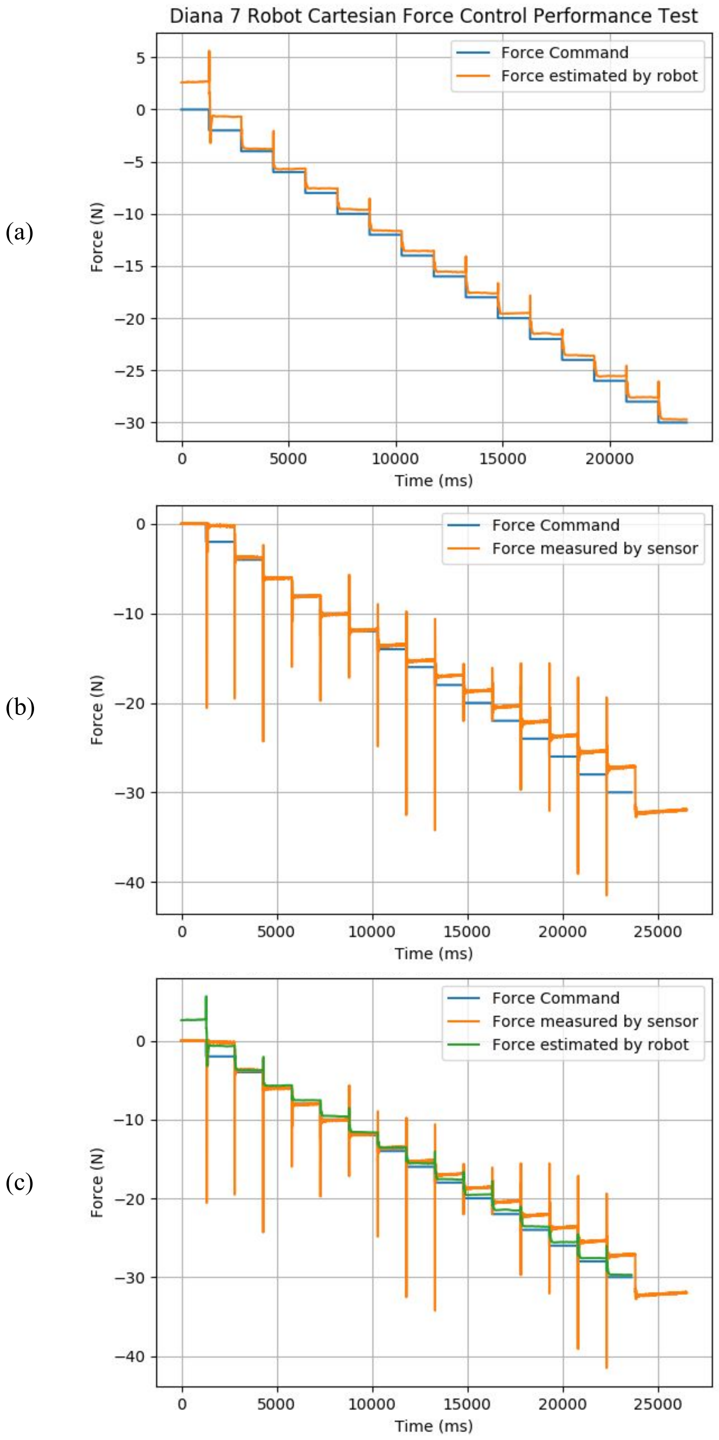


Figure 5.8: (a): the force command and robot estimated force which is quite close. (b): the force command and the sensor measured force which has a bigger error compared with robot estimated force. (c): three force curves in one figure.

A step force curve including command force, robot estimated force and sensor measured force is obtained in Figure 5.8. In Figure 5.8 (a), thanks to the force PI controller, the robot estimated force is quite close to the command force in most steps. However, the sensor-measured force gives a bigger error especially when $F_z > 15N$, and the maximum force error is 3 N (When the force accuracy test range is 30 N).

Note: since Diana 7 robot can calibrate the force controller before entering contact space, thus the force control accuracy can reach 0.5 N in real-use scenarios (UR e series also introduce this method to increase the force control accuracy).

5.2.2 Force Control Skill Implementation

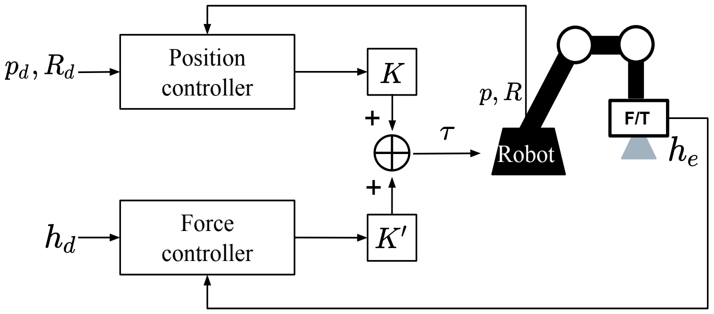


Figure 5.9: Hybrid position/force controller.

To generalize the robotic assembly capabilities, assembly skills based on hybrid position/force trajectories, such as linear, zigzag, spiral, sinus, and Lissajous trajectories, have been developed for robots equipped with force and torque sensors [138]. In this thesis, a newly designed hybrid position/force search skill named cross-search skill is developed to evaluate the assembly capability of the Diana 7 robot.

5.2.2.1 Cross-search Skill

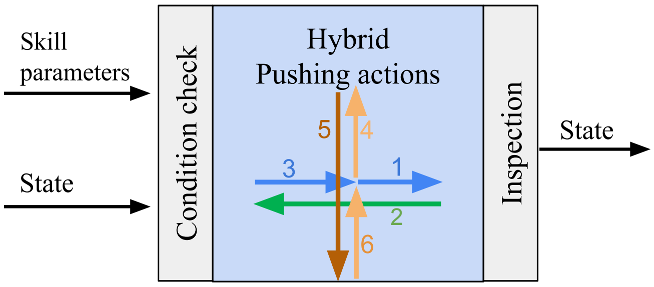


Figure 5.10: Pushing-based hybrid position/force assembly skill model.

A pushing-based hybrid position/force assembly skill was designed, as shown in Figure 5.10. Once the state is confirmed, 6 linear hybrid position/force movements are executed according to the skill parameters (i.e., position and force). An inspection was performed after the execution.

A hybrid position/force controller is implemented under the EE frame for the skill execution (Figure 5.9), where p_d and R_d are command pose, p, R are current pose; h_d

is a command force/torque, and h_d is a feedback force/torque. The diagonal matrices K and K' were used to indicate position or force control under the EE frame. In this study, force control along the Z direction was defined by Equation (5.17):

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, K' = I - K \tag{5.17}$$

As cross-search skill is a newly developed method, the illustration and analysis are given as follows.

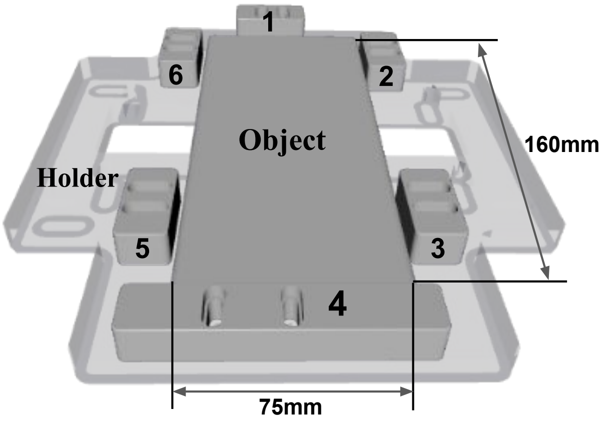


Figure 5.11: Assembly task illustration: the task is to insert the object into the holder.

The general tending task can be seen as Figure 5.11: aligning two parts (object and holder) with a certain geometric feature and maintaining the alignment using a certain operation process. Several adjustable fixtures on the holder are used to fix the object. The fixtures (numbered 1–6) are used to adjust the tension of the holder.

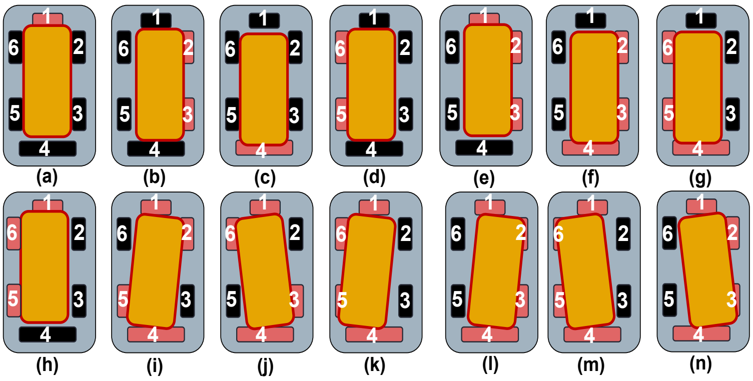


Figure 5.12: 14 possible contact states. The pink fixtures indicate contact with the object.

According to the analysis, there are several contact states that can cause object and holder misalignments (Figure 5.12).

In this work, the constraints provided by the fixtures can guide the object alignments with the holder geometry is proven and verified.

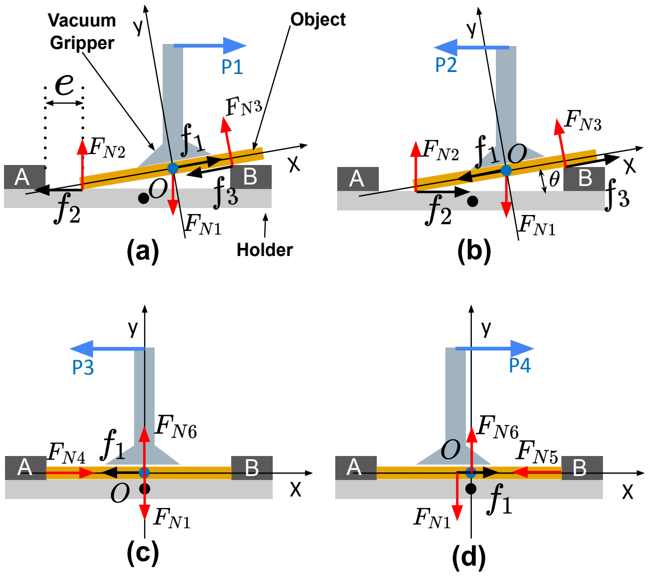


Figure 5.13: Illustration of a pushing action (in X direction) during an assembly task. (a) The vacuum gripper pushes the object along the $X+$ direction; (b) the vacuum gripper moves to the $X-$ direction, and the object can be pushed until contact occurs with the left fixture; (c) the object is pushed into the holder; and (d) the vacuum gripper continues to move along the $X+$ direction and slide on the object's surface.

Figure 5.13 was used to analyze this phenomenon according to a simplified contact situation, whereas Figure 5.13(a)–(d) illustrate the pushing action along the X direction.

e is the initial position error between the object and the holder. P is the translational movement command along the X direction. Frame O is attached to the center of the object, and θ is the angle between the object and the holder surface. **A** and **B** represent the adjustable fixtures on the holder. f_1 and μ_1 represent the maximum static friction and coefficient of static friction between the vacuum gripper and object (the vacuum gripper is released), respectively; f_2 and μ_2 represent the static friction and coefficient of static friction between the object and holder/fixtures, respectively. F_{N1} is the force applied by the vacuum gripper on the object. Additionally, F_{N2}, F_{N3}, F_{N4}, F_{N5} and F_{N6} are the normal forces exerted by the holder/fixtures on the object.

The general force analysis is performed on the object along the X and Y directions of frame O. In Figure 5.13(a), the object was assumed to be stationary. The force analysis performed along the Y direction is represented by the following equation:

$$F_{N1}\cos\theta + F_{N2}\cos\theta + F_{N3} + f_2\sin\theta = 0 \tag{5.18}$$

F_{v-o} represents the force between the vacuum gripper and object in X direction:

$$\begin{aligned} F_{v-o} &= f_1 + F_{N1}\sin\theta \\ &= \mu_1 F_{N1}\cos\theta + F_{N1}\sin\theta \end{aligned} \tag{5.19}$$

F_{o-h} represents the force between the object and holder along the X direction:

$$\begin{aligned} F_{o-h} &= f_2 \cos\theta + f_3 \\ &= \mu_2 F_{N2} \cos\theta + \mu_2 F_{N3} \\ &= \mu_2 (F_{N2} \cos\theta + F_{N3}) \end{aligned} \tag{5.20}$$

Based on Equation (5.18),

$$F_{o-h} = -\mu_2 F_{N1} \cos\theta - \mu_2 f_2 \sin\theta \tag{5.21}$$

Considering angle $\theta \approx 0$, negligible terms were removed from Equation (5.19) and (5.21) (when $\theta \approx 0$, $\sin\theta \approx 0$), Equation (5.22) was obtained as follows:

$$\begin{aligned} F_{o-h} &= -\mu_2 F_{N1} \cos\theta \\ F_{v-o} &= \quad \mu_1 F_{N1} \cos\theta \end{aligned} \tag{5.22}$$

From Equation (5.22) and Figure 5.13, we can infer that the coefficient of static friction determines the object's direction of motion. The same conclusion can be obtained from Figure 5.13(b). The vacuum grippers are mainly made of rubber [7] with a high coefficient of static friction, thus $\mu_1 > \mu_2$ is an easily obtainable condition.

The following results are obtained from Figure 5.13(c) and (d):

$$max(F_{N4}) = max(F_{N5}) \gg f_1 = \mu_1 F_{N1} \tag{5.23}$$

Thus, the entire operation was performed based on the following condition (Figure 5.13):

$$max(F_{N4}) = max(F_{N5}) \gg F_{v-o} > F_{o-h}. \tag{5.24}$$

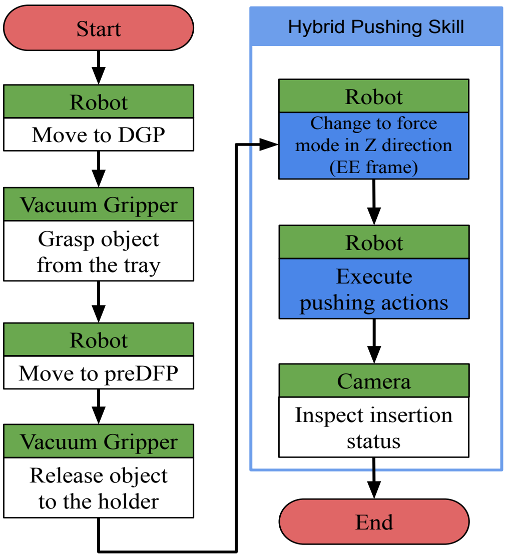


Figure 5.14: Workflow of a machine tending task. Before executing our hybrid pushing skill, the object was released by the vacuum gripper. DGP: desired grasping pose; DFP: desired final pose; EE: end-effector.

[7] https://www.festo.com/net/supportportal/files/216340/10783

Table 5.1: Comparison of the success rates for different baselines

Baseline	**Perfect**	**Uncertainty**
Baseline	69/100	47/100
Our method	**100/100**	**100/100**

A workflow was designed to evaluate the cross-search skills as shown in Figure 5.14. The skill parameters for the 6 actions (the sequence is shown in Figure 5.10) are set as:

$$\begin{aligned}
1 :&[+P^d_{\sigma x}, 0, +F_z]\\
2 :&[-2 * P^d_{\sigma x}, 0, +F_z]\\
3 :&[+P^d_{\sigma x}, 0, +F_z]\\
4 :&[0, +P^d_{\sigma y}, +F_z]\\
5 :&[0, -2 * P^d_{\sigma y}, +F_z]\\
6 :&[0, +P^d_{\sigma y}, +F_z].
\end{aligned} \tag{5.25}$$

$P^d_{\sigma x}$ and $P^d_{\sigma y}$ denote the amplitudes of the discrete actions. $P^d_{\sigma x}$ and $P^d_{\sigma y}$ are recommended to set twice bigger than the error e to ensure that the environmental constraints are fully explored. The demand F_z is set to 5 to 10 N to guarantee that the contact force is close to the human's operation force.

In this experiment, contrary to the **perfect** group, an error of $e \in [2, 4]$ mm was added in a random direction to the desired final pose to simulate the pose uncertainties in the **uncertainty** group. We evaluated our proposed method compared with the following baseline:

- **Baseline: spiral search [59].** A spiral search path was used to survey the entire environment surface. Here, the maximum search radius was set to 10mm;
- **Our cross-search method.** $P^d_{\sigma x}$ and $P^d_{\sigma y}$ is set to 8 mm, The demand F_z is set to 5 N.

Overall, the results of 400 group robot assembly tasks were recorded, as presented in Table 5.1. Baseline 1 (spiral search) always ended when stopped by one or two fixtures and thus insertion failed, moreover, it always generates sufficiently strong contact force between the objects and the holder.

To verify the generalization of our method, we also tested it on two other type holders using our method and obtained a 100% success rate (100/100 trials).

Chapter 6

Visual Residual Reinforcement Learning

Torque-controlled robots often serve computers, communication, and consumer electronics (3C) product lines, which usually involve small but complex assembly tasks, and need to be adjusted quickly and frequently. Currently, there are a few 3C assembly factory lines [120], but they require a long time to build and set up with high precision, which is unsuitable for small- and medium-sized enterprises that have automation needs but cannot afford to upgrade the entire production line. Position uncertainties are quite normal in human-based traditional production lines. Some studies used simply fixed curves for exploring [109] but they have low robustness against positional and angular errors for insertion tasks, especially when targets are not fixed accurately. Schimmels and Peshkin [116], [126] designed an admittance matrix for force-guided assembly in the absence of friction, and after two years, they improved the admittance control law. However, there still existed a maximum limit requirement of friction value [127]. Stemmer et al. [146] proposed the region of attraction method using vision and force perception to assemble specified-shape objects, while the geometry of the parts is required.

In this chapter, a visual residual policy that combines multimodal feedback from vision and touch was proposed, two modalities with different frequencies, dimensionality and value range. This method greatly enhances the robustness and efficiency of the RL algorithm.

6.1 Visual Guided Assembly

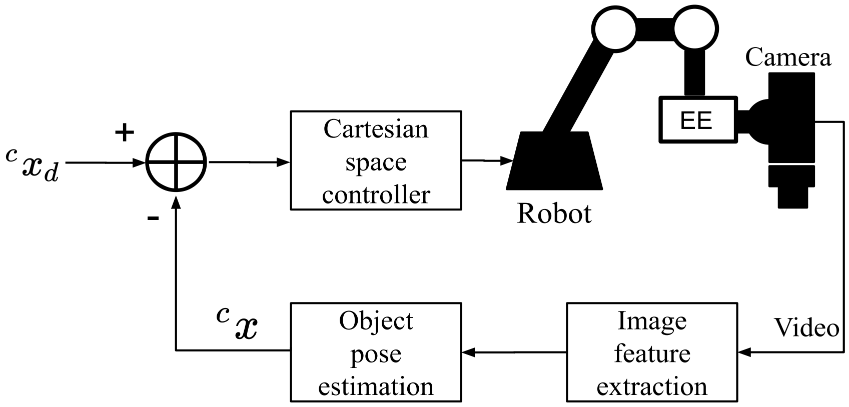


Figure 6.1: A position-based visual servo (PBVS) control system. ${}^{c}x_d$ is the desired EE pose relative to a target, while ${}^{c}x$ is the current estimated EE pose relative to a target.

A vision sensor allows a robot to measure the environment with a noncontact method. Shirai and Inoue [137] described an idea on how to use visual feedback to correct the position of a robot to increase assembly task accuracy. Position-based visual servo (PBVS) systems and image-based visual servo (IBVS) systems are the two major classes of visual servo control systems. The typical control structure of PBVS can be found in [54].

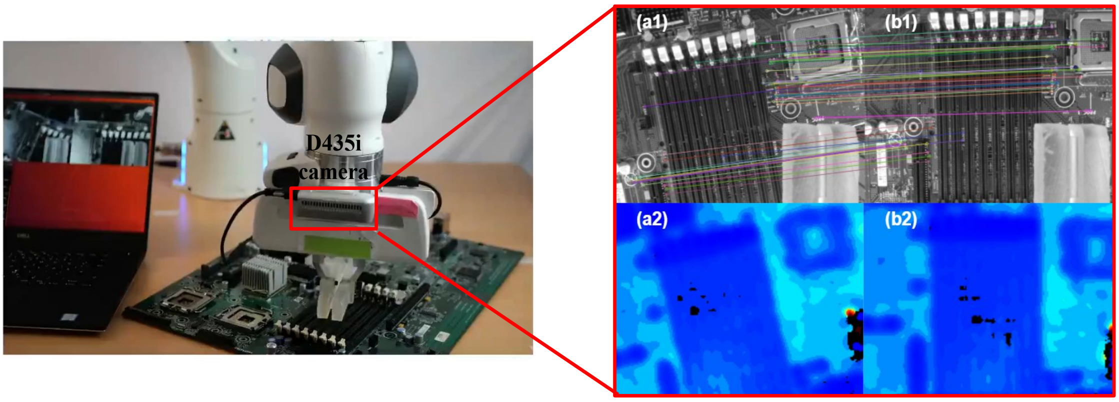


Figure 6.2: A visual servo work scene. a1 and b1 are from the RGB camera sensor, and it can be seen that the feature points have been matched with each other. a2 and b2 are from the depth camera sensor.

An EE mounted camera could acquire the target depth and orientation information that can be used directly for PBVS [151], [36]. However, the lens and imaging sensors, calibration of intrinsic/extrinsic parameters, reflection, shadow and occlusion will exert a strong influence on the precision of the visual guidance [83].

6.2 Visual Residual Learning Framework

6.2.1 Framework Overview

An eye-in-hand camera helps solve the problem of position uncertainty in unstructured environments in contact-rich tasks. The camera could try to align the characters of the target and compensate for the position error of the robot. Visual feedback control could provide geometric object properties for the pre-reaching target phase, whereas the camera aligning accuracy would always be disturbed by the target material or light.

Force feedback control is quite helpful for providing contact information between the object and environment for accurate localization and control under occlusions or bad vision conditions, and force information could be obtained easily from the proprioceptive data in the torque-controlled robot controller.

Visual feedback and force feedback are complementary and sometimes concurrent during contact-rich manipulation. In this chapter, we implemented the visual-based fixed policy combined with a contact-based parametric policy for a peg-in-hole operation (see Figure 6.3) as follows:

- For roughly locating the target hole, we use one global image taken from the teach mode with the RGB-D camera and rely only on the PBVS method [54] (i.e., the visual-based fixed policy) in this phase, because in free space, the contact-based parametric policy cannot receive proper contact information;
- After the rough location phase, the robot will move to the target hole according to the prerecorded transformation ${}^{g}x_d$ from global image pose to detailed image pose, where ${}^{g}x_d$ is recorded in the teaching phase. When the peg (for example, a RAM) contact with the target hole, the detailed image that has more accuracy for locating the hole will be used to insert the peg into the hole.

To exploit the high flexibility of RL and high efficiency of conventional controllers, we introduce an idea of residual RL from [61] with vision information; the proposed method is expected to outperform the original residual RL in a variable environment due to the position uncertainty problem. In residual RL, the policy is chosen by additively combining a fixed policy $\pi_H(s_v)$ with a parametric RL policy $\pi_\theta(s_t)$. The fixed policy can help the agent move to the target, but prevent the agent from exploring more states. To balance the exploration and exploitation between the fixed policy and parametric RL policy, we design the weighted residual RL as follows:

$$u_t = (1-\alpha)\pi_H(s_v) + \alpha * \pi_\theta(s_t) \tag{6.1}$$

Here, α is the action weight between the fixed policy and the parametric RL policy; the parametric policy is learned in the RL process to maximize the expected returns on the task. We use a P-controller as the hand-designed controller $\pi_H(s_v)$ in the experiments for the visual-based fixed policy.

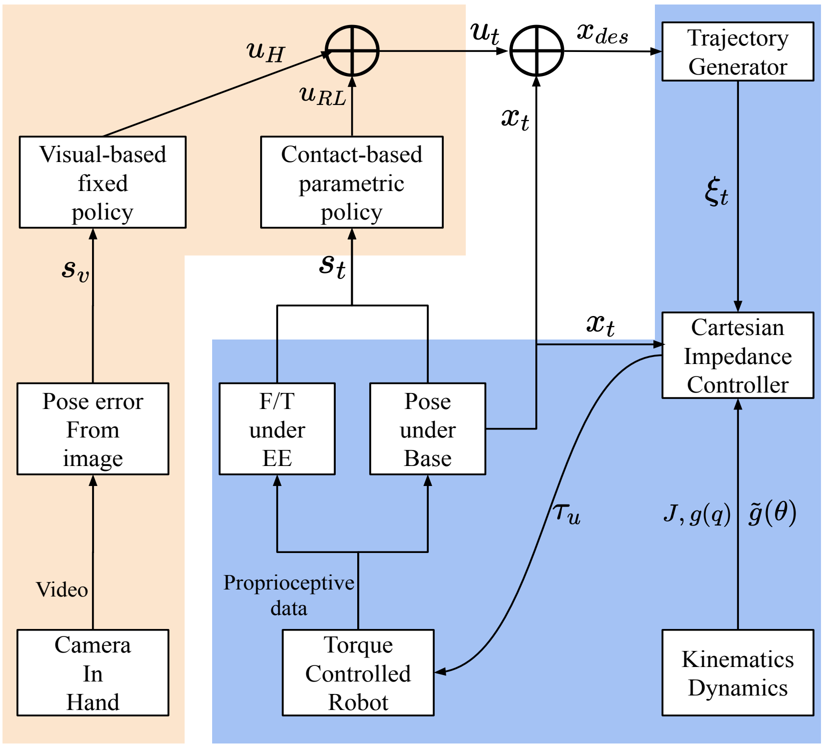


Figure 6.3: Representation of visual residual learning framework. The blue region is the real-time controller, and the wheat region is the non-real-time trained policy.

6.2.1.1 Visual Residual Policy Design

First, we explain the detailed design of $\pi_H(s_v)$. s_v represents a geometric relationship of robot states which is a Euclidean distance calculated by visual and estimated depth information. We introduce the method from [91] that used depth information in PBVS. Combined feature extraction with depth information Z_N, we could obtain the estimated target feature set:

$$^cP^* = (X_1^*, Y_1^*, Z_1^*, \ldots, X_N^*, Y_N^*, Z_N^*) \tag{6.2}$$

and current feature set:

$$^cP = (X_1, Y_1, Z_1, \ldots, X_N, Y_N, Z_N) \tag{6.3}$$

whose coordinates are expressed with respect to the camera coordinate frame c following the perspective projection method [54]:

$$\begin{bmatrix} X_N \\ Y_N \end{bmatrix} = \frac{Z_N}{f} \begin{bmatrix} u_N \\ v_N \end{bmatrix}. \tag{6.4}$$

Here, f is the focal length of the camera lens. $[u_N, v_N]^T$ represents the coordinates of the image feature set expressed in pixel units. Iterative closest point (ICP) [11] could be used to get the coordinate transformation $^{c*}x_c$ by the feature set ^cP and $^cP^*$.

$$^{c*}x_c = \begin{pmatrix} ^{c*}R_c & ^{c*}t_c \\ 0 & 1 \end{pmatrix} \tag{6.5}$$

Here, we set $s_v = ({}^{c*}t_c, \theta u)$ depending on Equation (6.5), where ${}^{c*}t_c$ is the translation error vector, and θu gives the angle/axis representation for the rotation error [138]. Then a velocity control scheme is designed by using an exponential and decoupled decrease of the error (i.e., $\dot{e} = -\lambda e$) as:

$$\begin{aligned} v_c &= -\lambda ({}^{c*}R_c)^T {}^{c*}t_c \\ w_c &= -\lambda \theta u \end{aligned} \tag{6.6}$$

Equation (6.6) is used in the rough location phase. $[v_c, w_c]^T$ is the camera frame velocity command under current camera frame $\mathscr{F}_c$, which could be transferred to robot EE frame $\mathscr{F}_e$ easily. In this paper, we calculate robot movement commands under robot EE frame $\mathscr{F}_e$ first and then transfer them to the base frame before inputting them to Equation (5.7).

We directly use $s_v = ({}^{c*}t_c, \theta u)$ as the states of fixed policy in accurate location phase,

$$\pi_H(s_v) = -k_p \cdot s_v, \tag{6.7}$$

which is quite convenient to implement.

6.2.1.2 Contact Policy Design

In this work, we use a value-based RL called Q-learning algorithm as the contact-based parametric RL policy $\pi_\theta(s_t)$, the Q-function is implemented as a table with states as rows and actions as columns, then we can update the table by using the Bellman equation:

$$Q^\pi(s_t, u_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r_t + \gamma \mathbb{E}_{u_{t+1} \sim \pi}[Q^\pi(s_{t+1}, u_{t+1})]] \tag{6.8}$$

The estimated 6-DoF external forces and moments along the X, Y, and Z axis under the EE frame are read from the Franka controller. The contact force and the moments between the robot's EE (i.e., the peg) and the hole the states as follows:

$$s = [F_x, F_y, F_z, M_x, M_y, M_z] \tag{6.9}$$

In order to simplify the state's design, we set a threshold T to clarify the contact status. We assume that the EE contacts the slot when the external force $|F| > T1$ N or the external moments $|M| > T2$ Nm, a value of ±1 means that contact is made, whereas 0 means that there is no contact with the encoding states.

The visual residual RL structure and training process can be seen as follows:

Algorithm 1 Visual Residual RL

```
Require: RL policy $\pi_\theta$, fixed policy $\pi_H$.
for iteration=1 to M episodes do
    Copy latest policy $\pi_\theta$ from learning thread
    Sample initial state $s_0$
    for step=1 to N do
        Get action $u_{RL}$ by greedily picking from $\pi_\theta(s_t)$
        Get action $u_H$ from $\pi_H(s_v)$
        Output policy action: $u_t = (1-\alpha)u_H + \alpha * u_{RL}$
        Get next state $u_t \rightarrow s_{t+1}$
        Optimize $\pi_\theta$ with Equation (6.8)
        if EpisodeEnd == true then
            break
        end if
    end for
end for
```

The increment equation $x_{des} = x_t + u_t$ was used to avoid the potential “far away” problem for safety concerns; x_{des} is the desired EE pose, and x_t is the current EE pose; u_t is the increment action command from the agent.

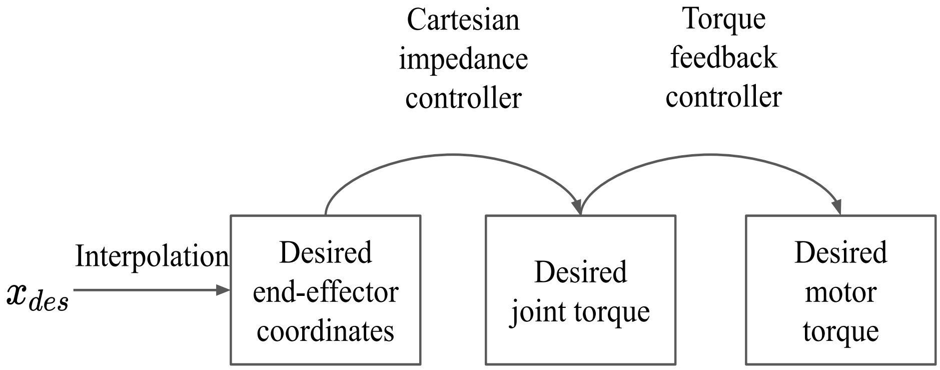

Cartesian impedance controller
Torque feedback controller
Interpolation
x_{des}
Desired end-effector coordinates
Desired joint torque
Desired motor torque

Figure 6.4: Illustration of the robot’s low-level control scheme. The actions x_{des} are computed at a low frequency, and the desired joint torques calculated directly by the Cartesian impedance controller at 1000 Hz. The joint controller runs the torque feedback controller at 3000 Hz.

The evaluation of the visual residual policy is described in Chapter 9 with a RAM insertion task setup.

Chapter 7

Region-limited Residual Reinforcement Learning

The ability to rapidly setup and reprogram newly-introduced products in factories is an increasingly essential requirement for adaptive robotic assembly systems [70], [141]. Position-controlled robots have the ability to handle known objects in well-structured assembly lines with high efficiency and achieve highly accurate position control. However, they require considerable setup time and tedious reprogramming to fulfill new tasks, and cannot adapt to any unexpected variations in assembly processes [169].

Collaborative robots offer the promise of closing the gap between onerous reprogramming and unexpected variations by combining the capabilities of position-controlled robots with dexterity and flexibility. For example, the hand-guiding method enables unskilled users to interact with collaborative robots and facilitates quick programming [122]. However, during assembly line reconfiguration, a long time is still required to remove and reinstall the robot arms and various attachments. Mobile manipulators (where robotic arms are mounted on mobile bases) were introduced to expand the productivity and adaptive capacity of manufacturing automation, particularly during the setting up phase when production lines must be reconfigured [92]. Because mobile manipulators can only be placed beside production lines and cannot be installed on production lines as collaborative robots, which occupy space previously provided for human workers. However, programming a robot in a constrained space is very difficult [75]. Overall, ease-of-programming has been identified as an open challenge in robot assembly [141], [93].

Additionally, collaborative robots equipped with force control functions can perform certain hybrid position/force operations for contact-rich tasks [5], [56], [38], [77]; however, their effectiveness and variation adaptive capacity in assembly processes are still unsatisfactory [90], [128]. Herein, an intuitive programming method was proposed to decrease the setup time of mobile manipulators and a RL algorithm was introduced to overcome unexpected variations in assembly tasks.

7.1 LbE based on RRRL policy

7.1.1 Problem Statement

Visual sensors can be used for target recognition, pose estimation, measurement, and positioning using traditional methods [83]. However, visual sensors, lenses, imaging sensors and the calibration of intrinsic/extrinsic parameters considerably influence the precision of visual guidance, as well as reflection, shadow, and occlusion may fail to extract the edges and features of objects owing to lights changes and object textures [83].

7.1.2 Method Overview

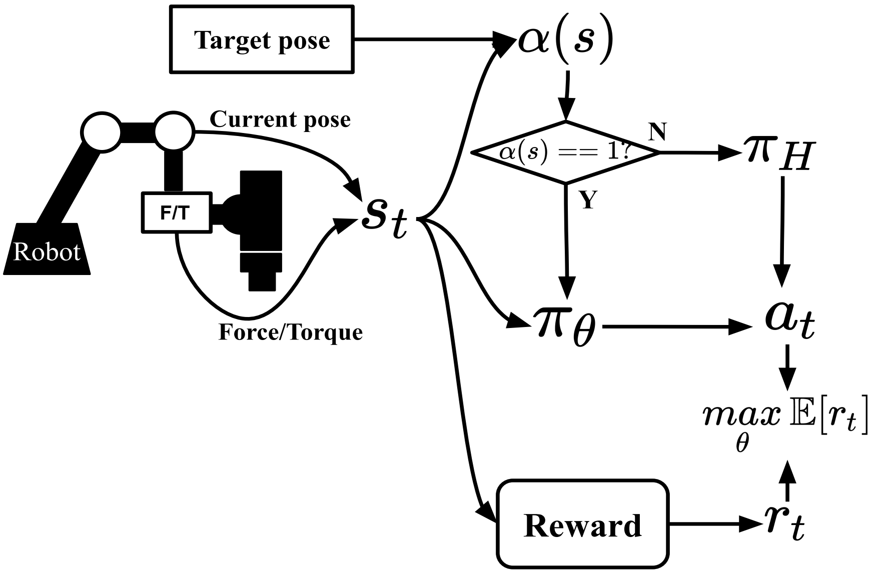


Figure 7.1: The RRRL policy structure.

Residual RL [128], [62] is a novel method that exploits the efficiency of conventional controllers and the flexibility of RL. Residual RL attempts to introduce prior information in an RL algorithm to accelerate the training process, rather than performing random explorations from scratch. For example, the estimated position can be set as prior information and even can have errors. The GUAPO [76] showed better performance than residual RL, SAC, and pure model-based method such as Deep Object Pose Estimator (DOPE). However, the force and torque information is not considered in the GUAPO policy. This information can provide observations regarding current contact conditions between objects and their environments for accurate localization [78]. Moreover, it can ensure manipulation safety [90]. However, pure force-based learning policies may lead to substantial deviations from goals and reduce learning efficiency. Thus, we combine the “region limitation” idea from GUAPO and the residual RL policy to develop a force-based approach called **RRRL** (Figure 7.1). In the RRRL policy, the rough target pose is obtained using by the teaching phase as the residual part and a function $\alpha(s) = 1[s \in \mathbb{S}_u]$ is used to switch between the fixed $\pi_H(s)$ and the parametric $\pi_\theta(s)$ policies [76]:

$$\pi(a|s) = (1 - \alpha(s)) \cdot \pi_H(a|s) + \alpha(s) \cdot \pi_\theta(a|s). \tag{7.1}$$

$\mathbb{S}_u$ is the region containing the goal position with uncertainty. Because force control is more safe and reliable in the fine motion/manipulation phase than position control and impedance control in assembly tasks [56], the RRRL policy $\pi_\theta(s)$ takes the operational force controller as the desired force/torque in operational spaces. thus, our goal is to maximize expected return (i.e., $\max_\theta \mathbb{E}[r_t]$) through the RRRL policy. The fixed policy u_H is used to move the object back to the initial target pose when the function $\alpha(s) = 0$. A double Deep Q Network (DQN) with proportional prioritization [125] was selected as the learning policy $\pi_\theta(s)$ in this study.

Algorithm 2 RRRL

Require: Model based policy π_H, learning frequency C_1, and target action-value update frequency C_2.

```
Initialize replay memory $\mathscr{H}$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$
Initialize target action-value function $Q_{target}$ with weights $\theta^- = \theta$
for episode = 1 to $M$ do
    Sample state $s_0$
    while NOT EpisodeEnd do
        Calculate $\alpha(s)$ using Equation (7.8)
        Select action $a_H$ from $\pi_H(s_t)$
        With probability $\varepsilon$, select a random action $a_{RL}$
        Otherwise select $a_{RL} \sim \pi_\theta(s_t)$
        Obtain action $a_t = (1 - \alpha) * a_H + \alpha * a_{RL}$
        Execute $a_t$, and observe reward $r_t$ and state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $\mathscr{H}$ with priority $p_t = max_{i<t} p_i$
        for $j$ = 1 to $C_1$ do
            Sample minibatch of transitions with priority from $\mathscr{H}$
            Update transition priority
            Update $\theta$ using the method proposed in [125]
        end for
        In each $C_2$ step, reset $Q_{target} = Q$
    end while
end for
```

7.2 LfD based on Visual Servoing

7.2.1 Problem Statement

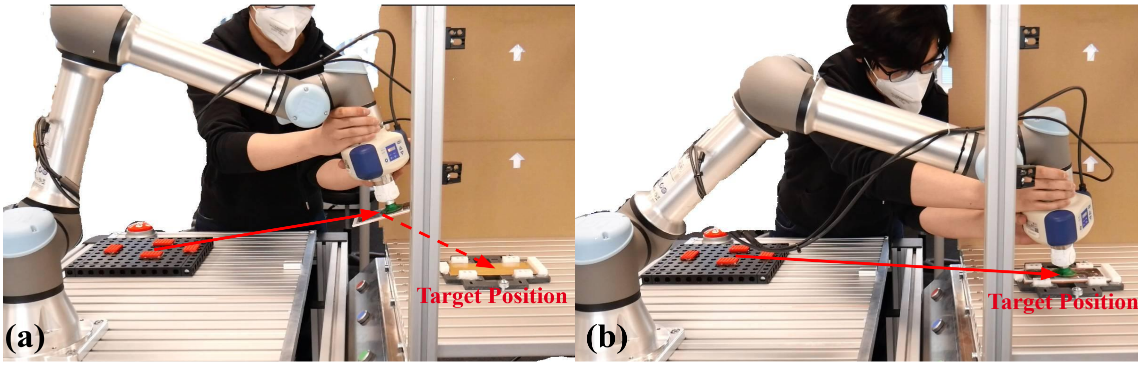


Figure 7.2: Examples of machine tending task in a heavily constrained space.

Mobile manipulators can considerably reduce robot and device's installation time [94]. Programming based on demonstration approaches has been proposed to address variations in geometry and configurations for assembly, placement, handling, and picking tasks [94], which can reduce the programming time and user training requirements [169]. The use of mobile manipulators introduces a **positioning error at the ±5 mm level** [148], and errors as small as ±1 mm can induce large **huge contact forces and consistent failures** in typical assembly tasks [128]. In the present study, we address the more typical cases of mobile bases that involve the repositioning of mobile manipulators according to task requirements.

Teach pendants are still used for precision positioning (position and orientation of the EE in many tasks [122]. However, these devices limit the intuitiveness of teaching processes and are time-consuming. Hand-guiding is a typical physical contact kinesthetic teaching solution, where programming is embodied using demonstration concepts, enabling users to quickly and intuitively program robots. However, it has drawbacks in terms of accuracy, locational separations, and operations involving dangerous objects [169]. Moreover, neither the hand-guiding nor the teach pendant programming methods can compensate for the positioning errors that accompany mobile units [148], and can result in the generation of a huge contact force that can damage objects.

Additionally, the base of a mobile manipulator requires considerable space within work cells. Figure 7.2 shows some real-world factory examples of constrained spaces. Sometimes, a user must teach a robot with a highly **awkward body posture** owing to anthropometric limitations [75]. Moreover, the delicate movement required by a user may be difficult to realize due to the resistance of robots in the drag mode [75]. Hence, because of these two issues, the use of hand-guiding for accurate assembly is rather difficult and yields low quality (e.g., excessive contact force and low accuracy).

7.2.2 Method Overview

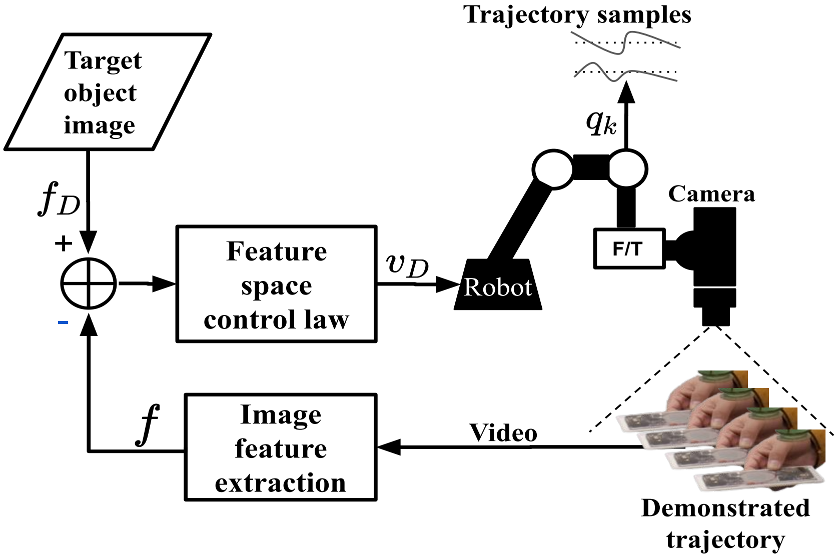


Figure 7.3: The LfD policy structure.

We propose a method that is simple and fast to implement and reduce the physical contact force to solve the aforementioned problems described in Section 7.2.1. We introduce visual guiding into the teaching phase as shown in Figure 7.3. The proposed visual guiding teaching is performed using two steps:

- **Grasping pose definition:** First, a user guides the robot to achieve a grasping pose under the target object frame. Then, the robot moves up and uses an eye-in-hand camera to capture a photo of the object as a reference.
- **Trajectory generation:** Second, the robot follows the moving object (the object can be moved by the user) to achieve a new target pose using a vision-based control algorithm (e.g., visual servoing [69], [54]), and records the entire moving trajectory.

Compared to utilizing a global camera [159], the method proposed in this paper uses an eye-in-hand camera to effectively avoid the occlusion of the target, and ensure that the robot follows the object to achieve the proper trajectory, thus achieving better performance in real industrial scenarios. Trajectory errors have little effect on final assemblies because uncertainties can be generally ignored in gross motion planning, and a fine motion planner can solve uncertainties during the assembly process [42].

7.3 Combine RRRL with LfD

This section focuses on combining visual servoing-based LfD and force-based LbE to enable the fast and intuitive programming of contact-rich tasks with minimal user efforts. Two learning approaches were developed and integrated into a framework, one relying on human-to-robot motion mapping (visual servoing approach) and the other relying on force-based RL. The developed framework can implement the noncontact demonstration teaching method based on the visual servoing approach and optimize the demonstrated robot target positions according to the detected contact state.

7.3.1 Framework Design

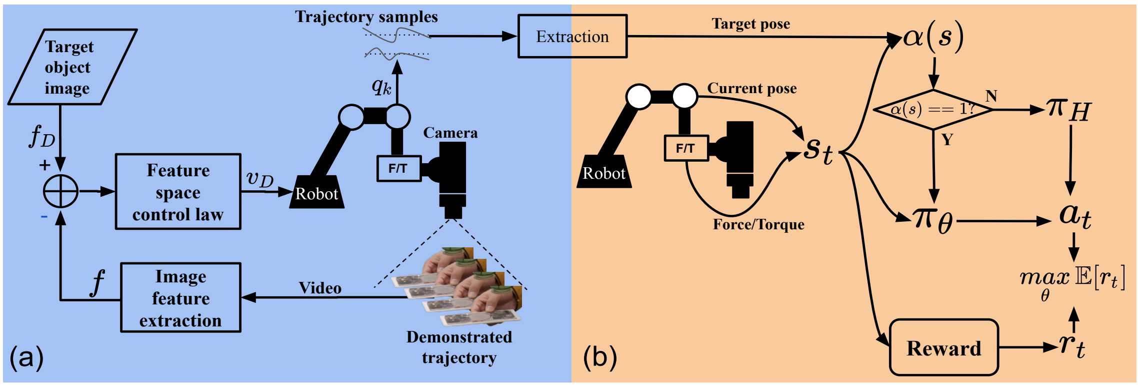


Figure 7.4: Combination of (a): LfD policy and (b): RRRL policy.

The processes in LfD can be divided into three steps: observing, representing, and reproducing an operation [169]. We perform our entire policy following these processes, and then add the RRRL policy at the end of the operations. Our method comprises two learning policies:

- **LfD policy**: The robot learns gross motions via human demonstrations, wherein a human demonstrates the target object images and grasping position to the robot.
- **RRRL policy**: The robot learns fine motions based on the RRRL policy, as described in Algorithm 2. The RRRL policy can be trained in advance to save the setup time.

First, we define the terminology and notation required to represent the coordinate transformations. We represent the task space of the robot as $\mathscr{T}$, which constitutes a set of positions and orientations that can be attained by the robot EE (i.e., suction gripper). $\mathscr{T}$ is a smooth m-manifold in which $m = 6$ and $\mathscr{T} = SE^3 = \mathscr{R}^3 \times SO^3$. The superscripts/subscripts of the coordinate frames are listed in Table 7.1.

Table 7.1: Coordinate frames

e	Coordinate frame attached to the robot EE
c	Coordinate frame of the camera
b	Base coordinate frame of the robot
o	Coordinate frame attached to the target object

In our policy, we equip an eye-in-hand camera to avoid the occlusions caused by the robot's links and other industrial devices [169], [85] during the demonstration. The relative homogeneous transformation ${}^{e}x_c$ and intrinsic camera parameters are determined by means of the hand/eye calibration method [153].

In the demonstration phase, teaching was categorized into the three following steps:
1) Robot EE was moved to the **desired grasping pose (DGP)**, which was then recorded as ${}^{b}x_{DGP}$.
2) Robot EE was moved to the **desired visual servoing pose (DVSP)**, which was recorded as ${}^{b}x_{DVSP}$; here, the object must be kept in view of the camera. The **first reference photo (RF1)** was captured, and a fixed **relative pose (RP)** was calculated as follows:

$$
{}^{c}x_o = ({}^{c}x_e)({}^{DVSP}x_{DGP}) = ({}^{c}x_e)({}^{b}x_{DVSP})^{-1}({}^{b}x_{DGP}), \tag{7.2}
$$

where ${}^{c}x_o$ is the coordinate transformation of the object frame o with respect to the camera frame c.
3) The visual servoing strategy [54] was activated, and the following system constraints [17] were applied to the robot during the teaching process:

$$
\begin{aligned}
&\mathbf{q} \in \mathbb{Q}_c, && && (7.3)\\
&\mathbf{q} \in [\mathbf{q}^{min}, \mathbf{q}^{max}], && \mathbf{q}^{min}, \mathbf{q}^{max} \in \mathbb{R}^N, && (7.4)\\
&\dot{\mathbf{q}} \in [\dot{\mathbf{q}}^{min}, \dot{\mathbf{q}}^{max}], && \dot{\mathbf{q}}^{min}, \dot{\mathbf{q}}^{max} \in \mathbb{R}^N, && (7.5)\\
&\mathbf{q}_{k+1} = \mathbf{q}_k + \delta\dot{\mathbf{q}}_k, && && (7.6)
\end{aligned}
$$

where $\mathbb{Q}_c$ is the set of configurations that do not cause any part of the arm to collide with obstacles that are difficult to model. Equation (7.4) and Equation (7.5) describe the robot's joint positions and velocity constraints, respectively. The object was then moved from the **DGP** to the **desired final pose (DFP)** by the user, the robot EE followed the trajectory $\mathbf{q}_k$ from the **DVSP** to the **DFP** under the aforementioned constraints. Further, the trajectory could be recorded. At **DFP**, the camera automatically captured the **second reference photo (RF2)**, and the **DFP** could be easily calculated at the end of the trajectory by Equation (7.7):

$$
DFP = ({}^{b}x_e)({}^{e}x_c)({}^{c}x_o) \tag{7.7}
$$

In this study, the image-based visual servoing method was introduced based on specified observed feature positions (Figure 7.4(a)). With the human expert (user) in the teaching loop, the trajectory-based representation output using LfD could avoid collisions even if the robot operated in a heavily constrained operation space. Additionally, our noncontact guiding method can avoid the resistance force of the robot in the drag mode, making it easier and more accurate for the users to perform teaching.

In most assembly tasks, the aim is to minimize the distance between objects and their goal positions [90]. The limited region is used to constrain the exploration area. Many methods can describe the uncertainty region with respect to a nonparametric distribution [76] or a parametric equation. In this study, the Euclidean distance was used to indicate the switch signal between the fixed $\pi_H(s)$ and parametric $\pi_\theta(s)$ policies in the hybrid RRRL policy:

$$\alpha(s) = \begin{cases} 1, & \text{if} \, \|CP - DFP\|_2 < D \\ 0, & \text{otherwise,} \end{cases} \tag{7.8}$$

CP and DFP denote the current and desired final poses, respectively. D is an engineering hyperparameter determined based on experience, and we suggest that D should be at least twice the final positioning error introduced by the fixed policy $\pi_H(s)$. When $\alpha(s) = 1$, the **force-based** learning policy $\pi_\theta(s)$ is used, otherwise the **position-based** fixed policy $\pi_H(s)$ is used to the object back to its initial uncertain target pose. The same switch algorithm was used during the training and execution phases. A double DQN with proportional prioritization [125] was selected as the learning policy $\pi_\theta(s)$ in this study.

7.3.1.1 Action Design

The actions in the assembly task can be either a position [78], [76] or force/torque command [56], [90]. Because we aim to reduce the contact force between the object and environment to ensure safety, the force/torque command action in the operational space (i.e. under frame x_e) was selected [90].

Here, we set the orientation space as the position mode to utilize the flexibility of the suction cup. Then, all the torque commands in the operation space were set to 0 Nm, and all the force amplitudes were set to 10 N, which is half of the force amplitudes set in [56].

7.3.1.2 State Design

Forces and torques feature the most direct information that characterizes contact states during an operation.Thus, the 6-dimensional force-torque vector $s = [F_x, F_y, F_z, M_x, M_y, M_z]$ under the robot EE's frame x_e were sent to the RRRL network as the input state.

7.3.1.3 Reward Design

We employed the precise target position of the hole as a reference for the reward during the learning phase. Unlike the execution phase, the precisely desired position was easy to obtain because the robot exhibited high positional repeatability.

$$r = \begin{cases} 1 - k_{steps}/k_{max}, & \text{success} \\ -\|CP - DFP\|_2, & \text{otherwise.} \end{cases} \tag{7.9}$$

The evaluation of the combined RRRL and LfD framework is described in Chapter 10 with a 3C machine tending task setup.

Chapter 8

Sim-to-Real Transfer Learning

Industrial robots are commonly used in structured environments, such as car manufacturing factories and phone assembly lines. The requirement to push the border of the "Robot Zone" [163] toward the manual manufacturing domain is increasing rapidly. Humans can execute manual manufacturing tasks easily using visual and force feedback, whereas robotic conventional methods, such as position control or visual servoing, are difficult to accomplish. RL shows the potential to solve complex robot manipulation problems because it allows an agent to interact with the environment for trial-and-error learning and accepts high-dimension feedback as the input [78], [150], [68].

For contact-rich manipulations, it is nontrivial to establish a robotic system that can learn a task with a safety guarantee and avoid wear and tear problem. Thus, sim-to-real methods are proposed to address the aforementioned concerns [113]. Recently, style transfer methods based on GAN [41] have been proposed recently in the computer vision field, enabling the use of vision-based manipulation tasks for deploying visual sim-to-real methods.

RL has shown some progress in robotic contact-rich tasks in unstructured environments; however, sample efficiency and safety concerns are two main problems when performing policy training. Many RL algorithms require millions of steps to train policies for performing complex tasks [82], [78]. In other words, human supervision is always needed in resetting experiments, hardware status monitoring, and safety assurance, which is quite time-consuming and tedious [55].

The sim-to-real approach shows the potential to solve the aforementioned problems; however, one significant difficulty associated with this approach is bridging the reality gap to address the mismatch in distinct distributions of rendered images and real-world counterparts. Another challenge is ascribed to force modeling in simulation as the force interactions will inevitably occur between the target object and environments when performing contact-rich tasks. Moreover, it is expensive to apply the system calibration due to the limitation of the simulation domain expert's ability [144] and accurate requirements [48].

In this chapter, the sim-to-real framework was proposed to solve the aforementioned problems. The training system was built in a Bullet simulator [33], the robot and task environment are modeled based on OpenAI Gym [14].

8.1 Soft-Actor-Critic based RL Framework

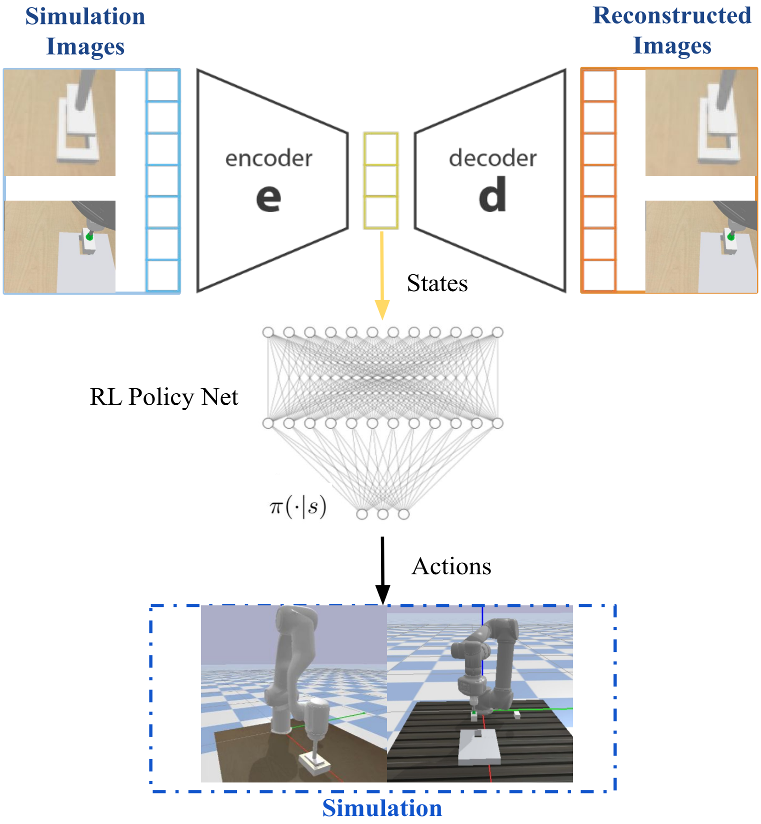


Figure 8.1: The SAC based RL framework.

We proposed a sim-to-real learning-based framework for contact-rich PiH operation by utilizing CycleGAN approach and robot force control.

Our approach is divided into two steps:

Step 1: a visual-based PiH insertion policy is trained in the simulation environment.

Step 2: the CycleGAN approach to transfer the real-world observations to the simulation observations, moreover, a force controller is used to solve the contact-rich issue during the assembly.

8.1.1 Policy

The SAC algorithm [44] was employed in our framework. SAC introduces an entropy H in its objective function (Equation (8.2)), which is a significant characteristic, where α denotes a temperature parameter that determines the importance of the entropy term.

$$H(P) = \mathbb{E}_{x \sim P}[-\log P(x)] \tag{8.1}$$

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H(\pi(\cdot|s_t))\right)\right] \tag{8.2}$$

Similarly, the value functions Equation (8.3) is also modified by the additional entropy:

$$\begin{aligned} V^{\pi} &= \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\left(R\left(s_t,a_t,s_{t+1}\right)+\alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right) \mid s_0=s\right] \\ Q^{\pi}(s,a) &= \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}R\left(s_t,a_t,s_{t+1}\right)+\alpha\sum_{t=1}^{\infty}\gamma^{t}H\left(\pi\left(\cdot \mid s_t\right)\right) \mid s_0=s,a_0=a\right] \end{aligned} \tag{8.3}$$

The entropy is used to measure the randomness of a given policy. In this study, the policy is trained to maximize a value that relies on the expected return value as well as the entropy. It helps to reach a good trade-off between exploration and exploitation.

SAC will update one policy π_θ and two Q-functions $Q_{\phi 1}, Q_{\phi 2}$ during the training. SAC policy uses Mean-squared Bellman Error (MSBE) loss for each Q network, which is shown as:

$$L\left(\phi_i,\mathscr{D}\right)=\mathbb{E}_{(s,a,r,s',d)\sim\mathscr{D}}\left[\left(Q_{\phi_i}(s,a)-y\left(r,s',d\right)\right)^2\right] \tag{8.4}$$

$$y\left(r,s',d\right)=r+\gamma(1-d)\left(\min_{j=1,2}Q_{\phi_{\text{targ},,}}\left(s',\tilde{a}'\right)-\alpha\log\pi_\theta\left(\tilde{a}' \mid s'\right)\right), \quad \tilde{a}'\sim\pi_\theta\left(\cdot \mid s'\right) \tag{8.5}$$

$\mathscr{D}$ represents the replay buffer. To prevent the overestimation of Q value, SAC chooses the minimum Q-value in two Q approximates. The policy aims to maximize the new objective function including the expected return and expected entropy. In order to allow stochastic descent on the loss function, a reparameterization trick is utilized as [1]:

$$\tilde{a}_\theta(s,\xi)=\tanh\left(\mu_\theta(s)+\sigma_\theta(s)\odot\xi\right),\xi\sim\mathcal{N}(0,I). \tag{8.6}$$

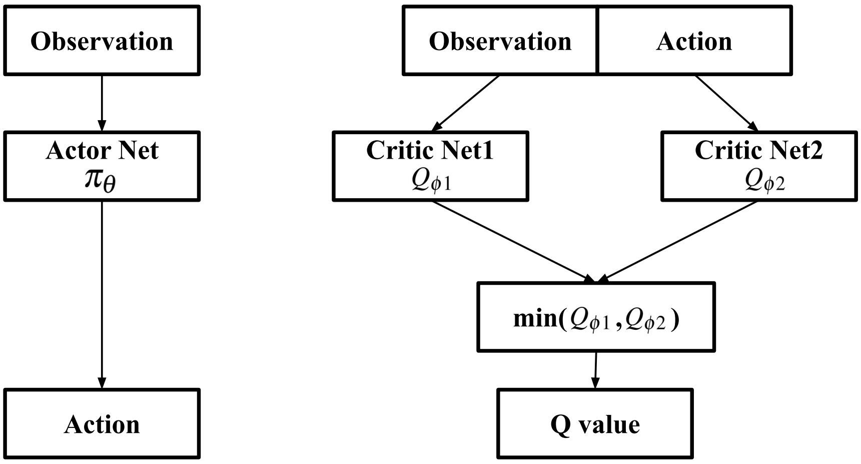


Figure 8.2: The general SAC architecture.

The general SAC architecture is shown as Figure 8.2.

[1] https://spinningup.openai.com/en/latest/algorithms/sac.html

8.1.2 States

For a vision-based learning policy, the commonly used observation states are the RGB, grayscale, and latent representation [78], [128]. In this work, we select observation spaces as follows:

- RGB observation space: $3 \times 64 \times 64$ tensor
- Grayscale observation space : $1 \times 64 \times 64$ tensor
- Latent representation observation space: 128×1 vector.

For the RGB and grayscale observation spaces, the network conducts end-to-end learning; in other words, raw images are inputted to the network and the output command is obtained. For the latent representation observation space, an Variational Autoencoder (VAE) [117] is employed as a part of the network. This autoencoder comprises an encoder and a decoder, we exploit the encoder to compress the input image and generate the latent representation observation space. Different from an autoencoder, a VAE can provide a probabilistic manner for describing an observation in latent space instead of an encoding vector with specific values. Each latent attribute for a given input is represented as a probability distribution.

8.1.3 Actions

Inspired by the literature [77], the necessary translation movement along the X-, Y-, and Z axes are considered and the orientation of the EE is fixed. We define a three-dimensional (3D) vector that contains the translation movement information of the robot. We use a position controller in the simulation, and the robot will move along a relative distance with respect to the current pose. The continuous 3D displacement action space ΔP

$$\Delta P = [\Delta x,\ \Delta y,\ \Delta z], \tag{8.7}$$

which considers translation movement along the X-, Y-, and Z-axes. The value in each axis is strictly in the interval of $[-0.02, 0.02]$ m.

8.1.4 Rewards

Some researchers set reward functions based on the different insertion phases such as reaching, alignment and insertion [77], [56], making the reward function hard to design; and need to distinguish the different phases. We only design one normal reward function that combines L1 and L2 distances for reaching, alignment and insertion phases and one reward for successful insertion:

$$R(\mathbf{s}) = \begin{cases} 50, & \text{(Success)} \\ -(flag * 10 + 0.4 * (\|p_{obj} - p_{goal}\|) & \\ +0.6 * (|p_{obj} - p_{goal}|)) & \text{(Otherwise),} \end{cases}$$

where p_{obj} and p_{goal} represent the positions of the peg and hole, respectively, and *flag* is set to 1 if the robot moves to a distance exceeding a certain threshold (i.e., 15 cm away from the hole center); otherwise, it is set to 0. Here, *flag* works as a punishment when the robot makes unexpected movements.

8.2 CycleGAN and Force Control based Sim-to-Real Transfer Learning

The sim-to-real learning-based framework for a PiH insertion task is presented in Figure 8.3. Sim part (in blue) is used to train the encoder (Frozen Net) and RL policy net in a simulator. In real pipeline (red part), $G_{RS} : R \to S$ is a mapping function generated using a cycle-consistent generative adversarial networks (CycleGAN) to transfer an image from a real-world style to a simulator style. In the simulation, a position controller is used to execute the policy actions while a force controller is called when transferring the policy to the real world.

8.2.1 Observation Space Transfer

To transfer our policy from the simulator to the real world, we must transfer the images from the domain of the real world to their counterparts in the simulator. Conventionally, training an image-to-image translation model requires a paired dataset. The requirement for paired examples is a limitation, it is challenging and expensive to prepare these datasets.

A successful approach for unpaired image-to-image translation is the Cycle-consistent Generative Adversarial Networks (CycleGAN) [167]. CycleGAN aims to update the data distribution in simulation to match the real one through mapping or regularization enforced by the task model, i.e., it is an approach to transfer the source data distribution to the distribution in the target domain. In order to avoid the drawback of GAN, CycleGAN introduces a novel approach called cycle-consistency, which can be used to calculate the reconstruction error of the images. There are two mapping functions: $G : X \to Y$ and $F : Y \to X$ and associated adversarial discriminators D_Y and D_X.

For function $G : X \to Y$ and its discriminator D_Y:

$$\mathscr{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))] \quad (8.8)$$

For function $G : Y \to X$ and its discriminator D_X:

$$\mathscr{L}_{GAN}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))] \quad (8.9)$$

For the cycle consistency loss function:

$$\mathscr{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \quad (8.10)$$

Finally, the full objective is:

$$\mathscr{L}(G, F, D_X, D_Y) = \mathscr{L}_{GAN}(G, D_Y, X, Y) + \mathscr{L}_{GAN}(F, D_X, X, Y) + \lambda \mathscr{L}_{cyc}(G, F) \quad (8.11)$$

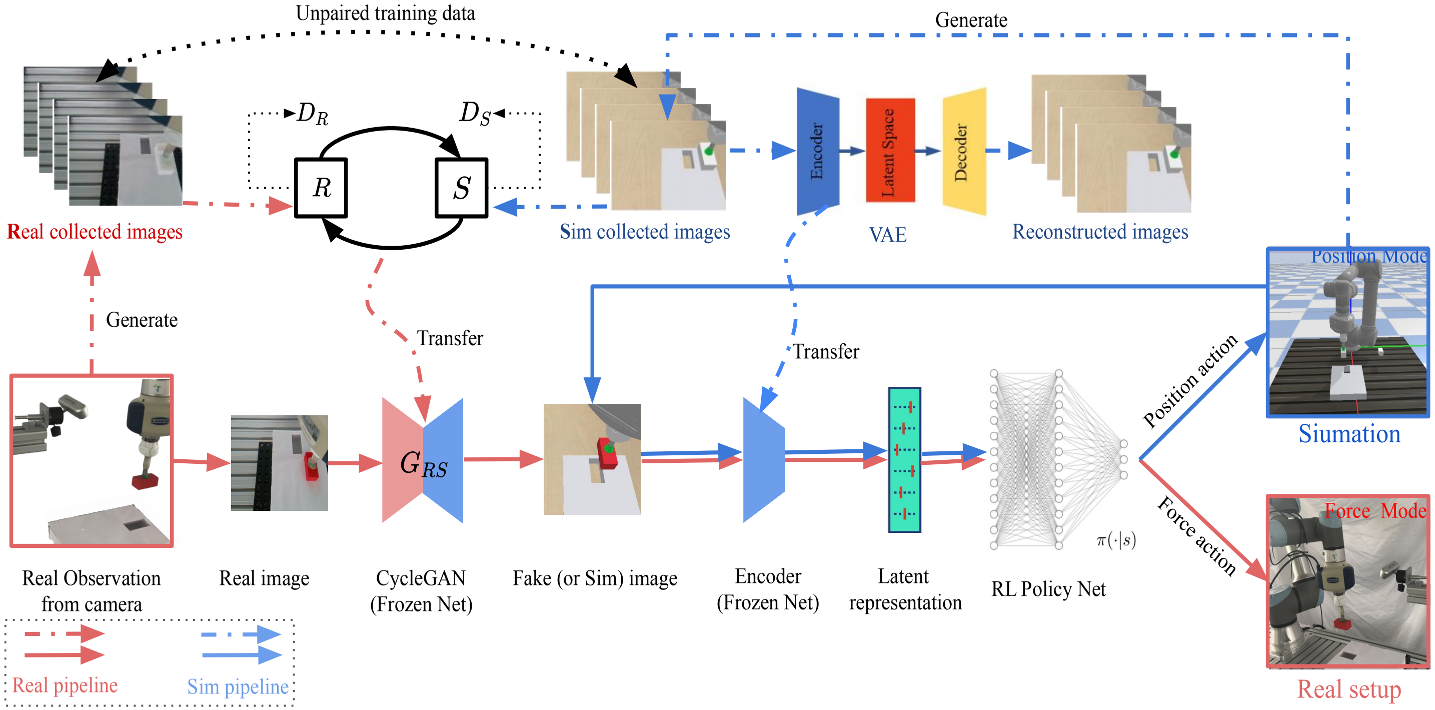


Figure 8.3: Sim-to-real learning-based framework for PiH insertion task.

where λ controls the relative importance of the two objectives, and function:

$$G^*, f^* = \arg\min_{G,F}\max_{D_x,D_Y} \mathscr{L}\left(G, F, D_X, D_Y\right) \tag{8.12}$$

should be solved.

In our framework, we command the robot to move randomly in the view of the camera and captured its random state each time. Approximately 200-300 images can be effortlessly obtained for training the model. Using the style transfer based on the CycleGAN, we map the view of the camera in the real-world environment to its counterpart, which we use to train our policy.

8.2.2 Action Space Transfer

In a study [144], researchers incorporated force augmentations by multiplying a random constant α with the force and the moment because they arrived at the conclusion that **the direction of the vectors** (F, M), but not the magnitude, **is the most important factor** in insertion operations. We extended this conclusion to our sim-to-real transfer process using a new method: we multiply gain K and the original position action output ΔP and then use the product C_{real} as the control command for the real robot force controller:

$$\begin{aligned} C_{real} &= [F_x, F_y, F_z] \\ &= K\Delta P \\ &= K[\Delta x,\ \Delta y,\ \Delta z], \end{aligned} \tag{8.13}$$

For instance, if $K = 100$ N/m, then the force command values along the X-, Y-, and Z-axes are in the range $[-2, 2]$ N.

Part III

Evaluation

Chapter 9

Visual Residual Reinforcement Learning for RAM Insertion

9.1 Experiment Setup

9.1.1 Setup Description

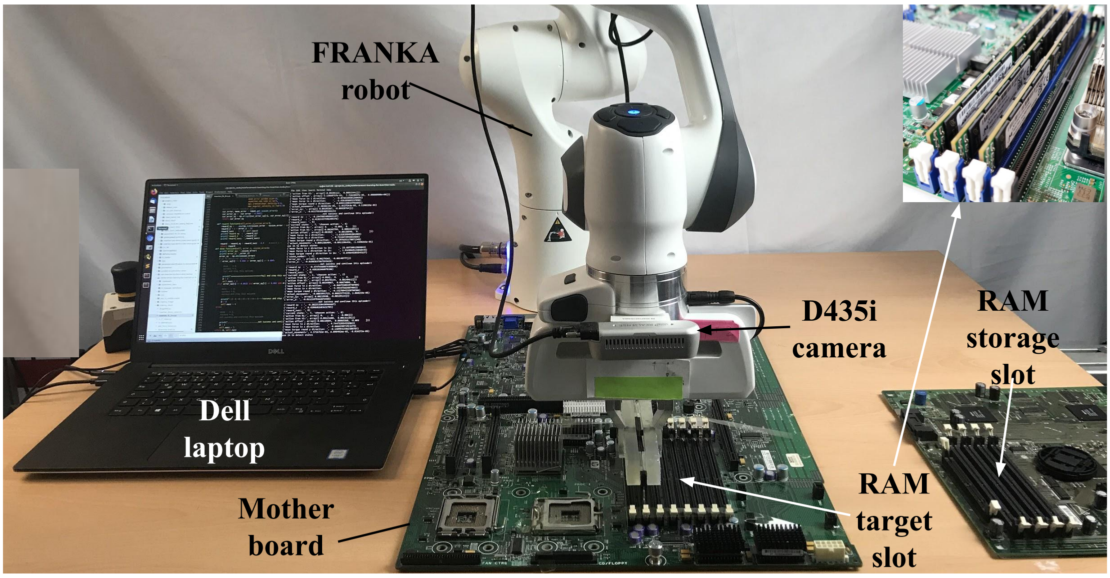


Figure 9.1: A contact-rich task scenario: RAM insertion.

We consider the experiment for the insertion task here. The task can be described as moving the already-grasped parts to their goal pose (Figure 9.1). This is the most common setting in manufacturing. The success of such tasks can be measured by minimizing the distance between the objects and their goal pose especially in the Z direction (see Figure 9.1).

We used the Franka robot [38] [1] for real robot experiments and set the translational Cartesian stiffness as 3000 N/m and stiffness for the rotations as 300 Nm/rad (Recommended upper limit).

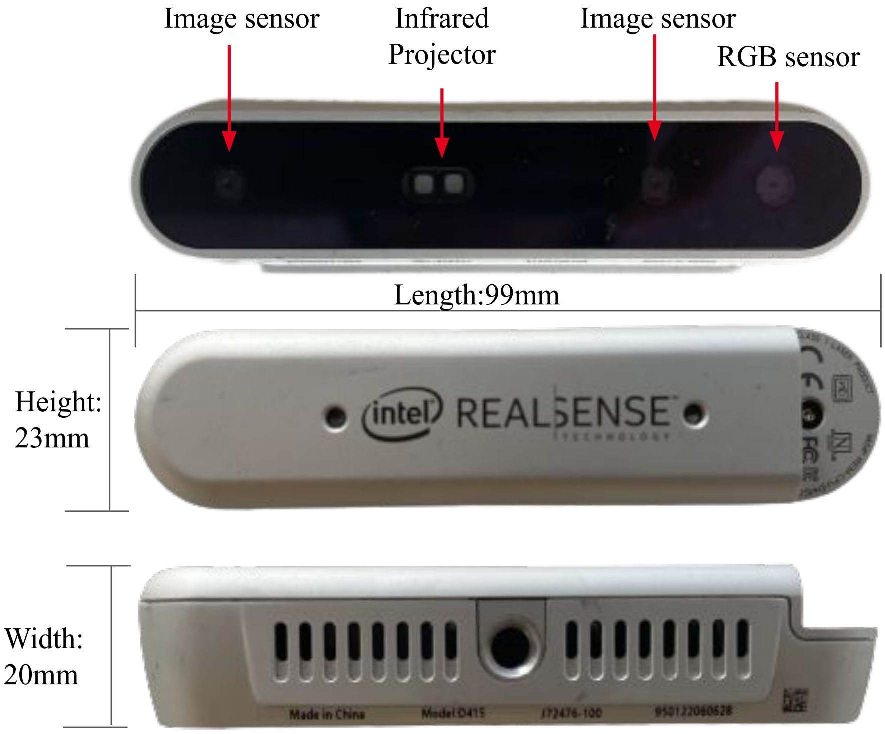


Figure 9.2: Intel RealSense D435 camera sensor configuration, appearance, and dimensions.

Two sensor modalities were available in the real hardware, including proprioception and red-green-blue (RGB) depth camera as shown in Figure 9.2 (This figure is redrawn according to [2]). The RGB and depth information was recorded using the eye-in-hand Intel RealSense Depth Camera D435i. The policy ran on a Dell Precision 5510 laptop and sent the updated position to the real-time controller, which calculated the joint torque command and sent it to the robot controller at 1000 Hz. We used a CORSAIR DDR3 RAM and a motherboard as the training and testing environment.

[1] https://frankaemika.github.io/docs

[2] https://www.intelrealsense.com/depth-camera-d435/

9.1.2 Task Analysis

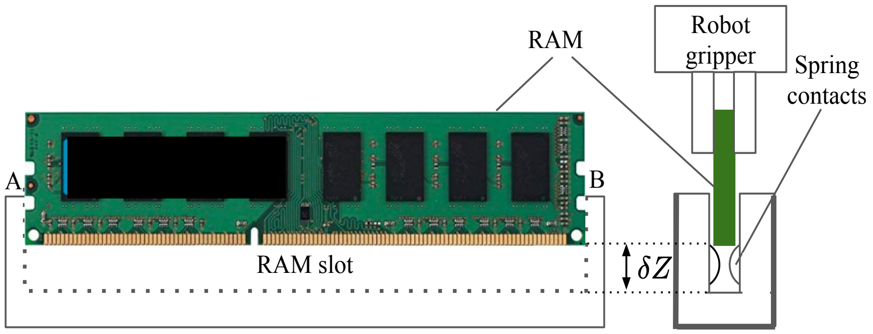


Figure 9.3: A contact-rich task scenario: RAM insertion. Such tasks always have stuck problems due to tight clearance and narrow space.

9.1.2.1 Position Uncertainty in Unstructured Environments

Position uncertainties are quite normal in human-based production lines as the operation objects are not fixed. Workers could perform high-precision robotic assembly tasks with their strong intelligence, excellent visual ability, and dexterous hands. Whereas these tasks are challenging to robots, especially in unstructured production environments.

In addition, the friction and obstruction in contact-rich tasks introduce large positional errors due to the low-stiffness design concepts of torque-controlled robots. The limited control stiffness combined with the friction and obstruction in contact-rich tasks gives the position control error at a millimeter level. Torque-controlled robots are expected to achieve a desired dynamic relationship between environmental forces and robot movements to avoid breaking the environments or targets, thus the desired position and contact force cannot be satisfied in the same DoF simultaneously. Moreover, the location of the targets is uncertain sometimes due to the insufficient accuracy of industrial assembly lines.

Using the visual method to correct the positions of the targets is an intuitive solution, while we still have position control problems when the robot contacts with targets due to the reason as we explained in Section 6.1, even though we have implemented some explore actions (e.g., the spiral explore method [109]).

In 3C production lines, the insertion scenarios are different from the typical simplification settings of peg-in-hole [77], [56]. For example, the random-access memory (RAM) insertion task has the following problems:

1. The RAM slot or other slots do not have proper surfaces for the sliding behavior of a robot in the alignment stage [77], [89] (Figure 9.1), which makes sliding-type algorithms not to work anymore;
2. The objects (like the RAM or hard disk) would be easily stuck by the structure near the slot or the slot itself in the explore/alignment stage;

3. Compared with previous studies, the slot has a long and narrow shape with tight clearance, which is difficult to insert by random and traditional search algorithm [109], [112].

9.1.3 Uncertainty of POMDP States

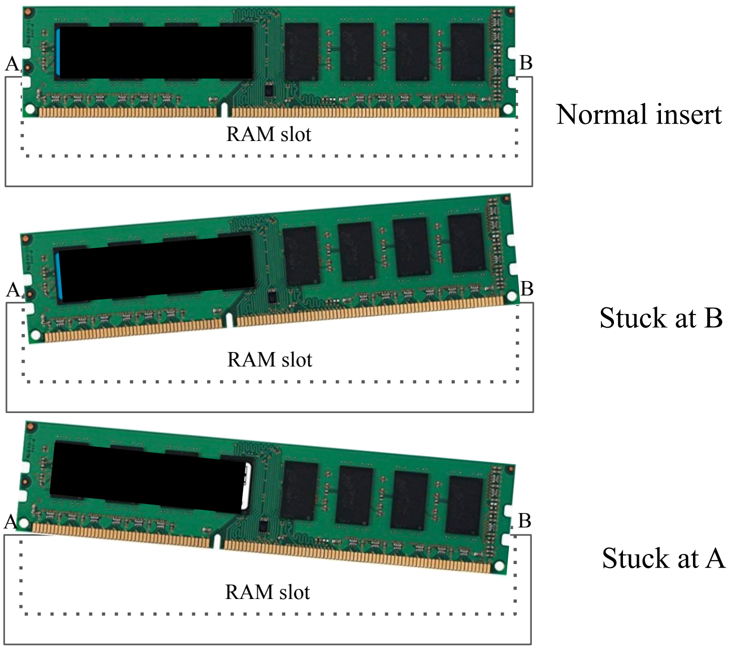


Figure 9.4: Success and failure insertion cases

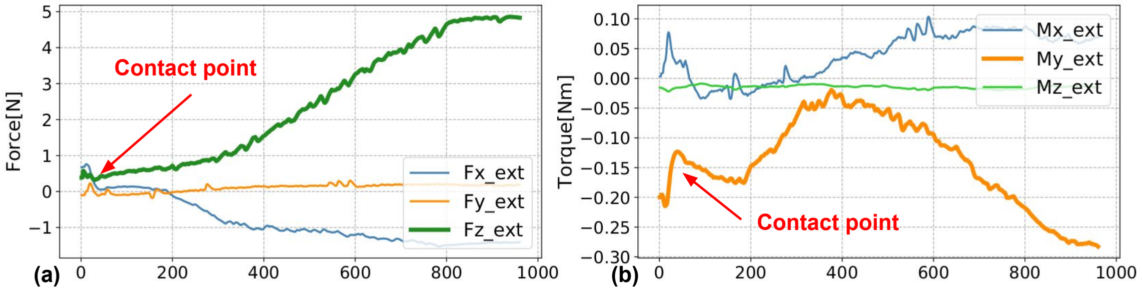


Figure 9.5: (a) RAM contacts with one slot side in **movement action** with 5 N force feedback in the Z direction. (b) External moment data M_y which is difficult to detect the torque contact status (goes up first and then down during the contact force increase).

The main challenge of the traditional policy is to design adaptable, yet robust algorithms when faced with inherent difficulties in modeling all possible interaction behaviors. RL enabled us to find new control policies automatically for contact-rich problems where traditional heuristics had been used, but the results were unsatisfactory.

Contact states are hard to estimate due to the sensor noise and robot modeling error, changing the Markov decision process (MDP) to POMDP, making it significantly harder to find an optimal policy [104], and it requires more training time. Belief state tracking is one way to handle the POMDP problem [86], [161], [88], but this method takes too much time to find an optimal policy.

9.2 RAM Insertion Task Implementation

9.2.1 Proactive Action

Most studies [77], [90], and [61] have modeled the robot manipulation task as a finite-horizon discounted Markov Decision Process (MDP) $\mathscr{M}$ in an environment E, with a state space $\mathscr{S}$, an action space $\mathscr{A}$, state transition dynamics $\mathscr{T} : \mathscr{S} \times \mathscr{A} \to \mathscr{S}$, a discount factor $\gamma \in (0,1]$, and a reward function $r : \mathscr{S} \times \mathscr{A} \to \mathscr{R}$ to determine an optimal stochastic policy π.

In practice, many contact states s_t cannot be observed directly in the manipulation tasks that are close to a POMDP problem. However, the POMDP problem is confined to the modeling error of the torque-controlled robot, which makes it difficult to detect the contact states.

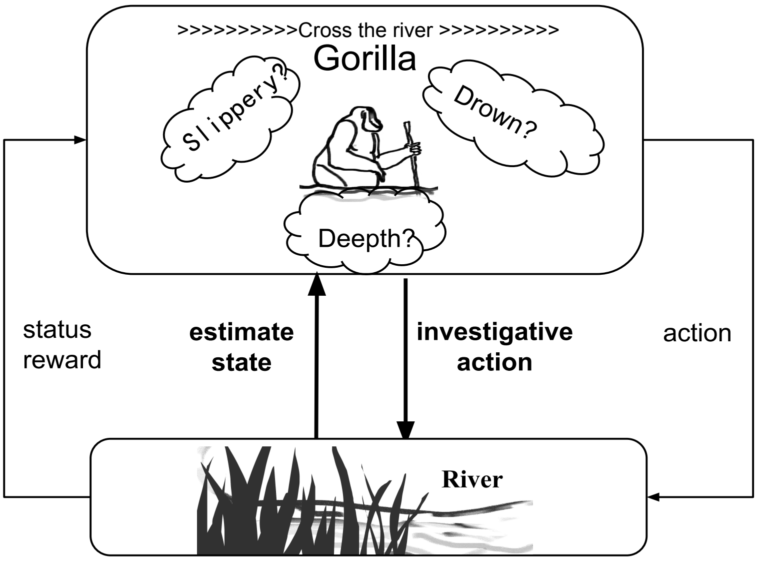


Figure 9.6: Process modeling of a gorilla cross river task. An investigative action is used to clarify the state and to avoid punishment (i.e. the dangerous states).

Inspired by wild gorillas, who tried crossing a pool of water using a walking stick to test the water depth [13], the process is modeled as Figure 9.6.

We improved our RL process by adding a proactively investigative action (a_I) that could detect the clear states (es_t) involved in the RL process (Figure 9.7), which is different with [56] that continues to push the target to obtain a detectable moment; the

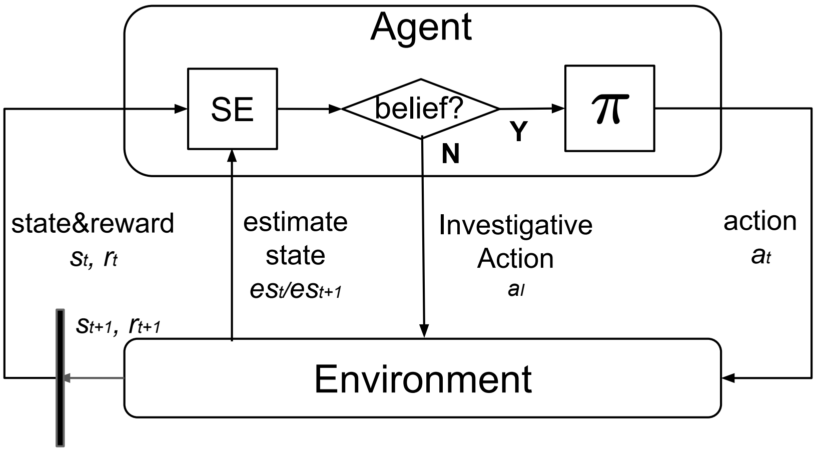


Figure 9.7: Investigative action idea for solving POMDP problem. SE: state estimator. The states will be estimated by the SE function, then the policy receives the clear states and outputs an action.

investigative action space $\mathscr{T}^I$ is a smooth m-manifold, where m = 6 and $\mathscr{T}^I = SE^3 = \mathscr{R}^3 \times SO^3$.

We use the investigative action a_I combined with u_t to construct a new policy $u_t^I(s_t)$ instead of the original $u_t(s_t)$, which can be written as $a_I, u_t \to E \to s_{t+1}^I$, where s_{t+1}^I is determined by adding an investigative action a_I of the torque-controlled robot to the environment. Consequently, the heuristic design of the investigative action prevents the learning process from falling into multiple unclear states.

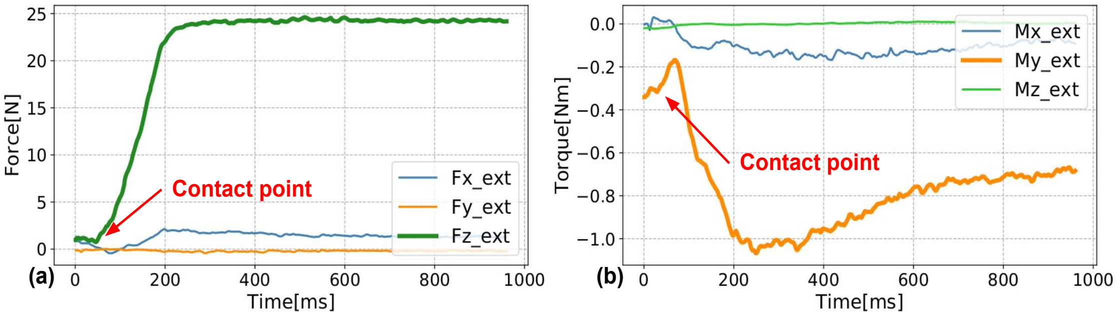


Figure 9.8: (a) RAM contacts with the side of the slot using **investigative action** with 25 N press force. (b): External moment M_y reaches -1 Nm which could clearly detect contact status.

In particular, the torque-controlled robot outputs either the movements or the forces. In our experiments, the movements are considered as the actions in the action space $\mathscr{A}$, and the forces are considered as the investigative actions. Instead of using 20 N force continuously to detect the values of the moments in the search phase [56], we only command the controller to exert a force (10–25 N) in some directions in a short time (0.5–1 s) as the investigative action, whereas the feedback movements or forces/moments are used to verify the contact states when the states are vague. Our investigative action method can markedly reduce the friction and probability of being stuck when the robot performs movement actions.

9.2.2 Experiment Algorithm Design

In the weighted residual RL in this chapter as shown in Figure 9.9, actions u_t are designed by adding the fixed policy $u_H = \pi_H(s_v)$ with the parametric policy $u_{RL} \sim \pi_\theta(s_t)$:

$$u_t = (1-\alpha)u_H + \alpha * u_{RL}. \tag{9.1}$$

The fixed policy output u_H is calculated by a hand-designed controller as given in Equation (6.7); α helps to adjust the balance between exploration and exploitation. We set k_p to (1,1,0.3,0,0,0) when calculating the fixed policy. To identify a reasonable weight between the two components, we initially experimented with the weighted residual RL by introducing a group of action weight parameters, such as 0.3, 0.5, and 0.7. The training experiments suggested an optimum policy output with a weight of 0.5, whereas the weight could increase or decrease around 0.5 according to the visual condition in the implementation phase. We used the algorithm to detect states and implemented its slightly-modified version, where the trained policies were constructed by the two aforementioned components. Here the flag belief is set to 0 or 1, according to the moment threshold settings, a detectable moment (over threshold) always gives the true belief state. Combined with the investigative action mentioned in Section 9.2.1, the modified Q-learning algorithm was trained at a high speed, and it easily resulted in optimization.

9.2.2.1 Action Design

We design Cartesian movement actions for this experiment. Each Cartesian movement dimension was set to $+1$ for a positive movement and -1 for a negative movement; therefore, we had $6*2 = 12$ discrete actions. We set λ as the scale parameter to adjust the amplitude of the discrete actions similar to [56] as

$$a = \lambda[P^d_{\sigma x}, P^d_{\sigma y}, P^d_{\sigma z}, R^d_{\sigma x}, R^d_{\sigma y}, R^d_{\sigma z}]. \tag{9.2}$$

Here, P and R are positional and orientational movements under EE frame, respectively. λ is easy to choose because it is closely related to assembly clearance and visual accuracy, normally we set $\lambda = 0.002$, then we have movement resolution at 0.002 mm and 0.002 rad level. We found that orientational movement accuracy was enough by using the fixed policy u_H, so we only output positional movement actions in our RL idea, this is a normal setting because of the visual feedback and force feedback are complementary during contact-rich manipulation.

The *investigative action* was designed as the force action ${}^eF_z = 25N$ under robot EE frame $\mathscr{F}_e$ for 1 s. The robot will try adding force but will stop moving if the force is greater than 25 N or the movement is greater than 3 mm. Then, the agent will obtain clear state feedback because of the large contact force and torque amplitude (Figure 9.8).

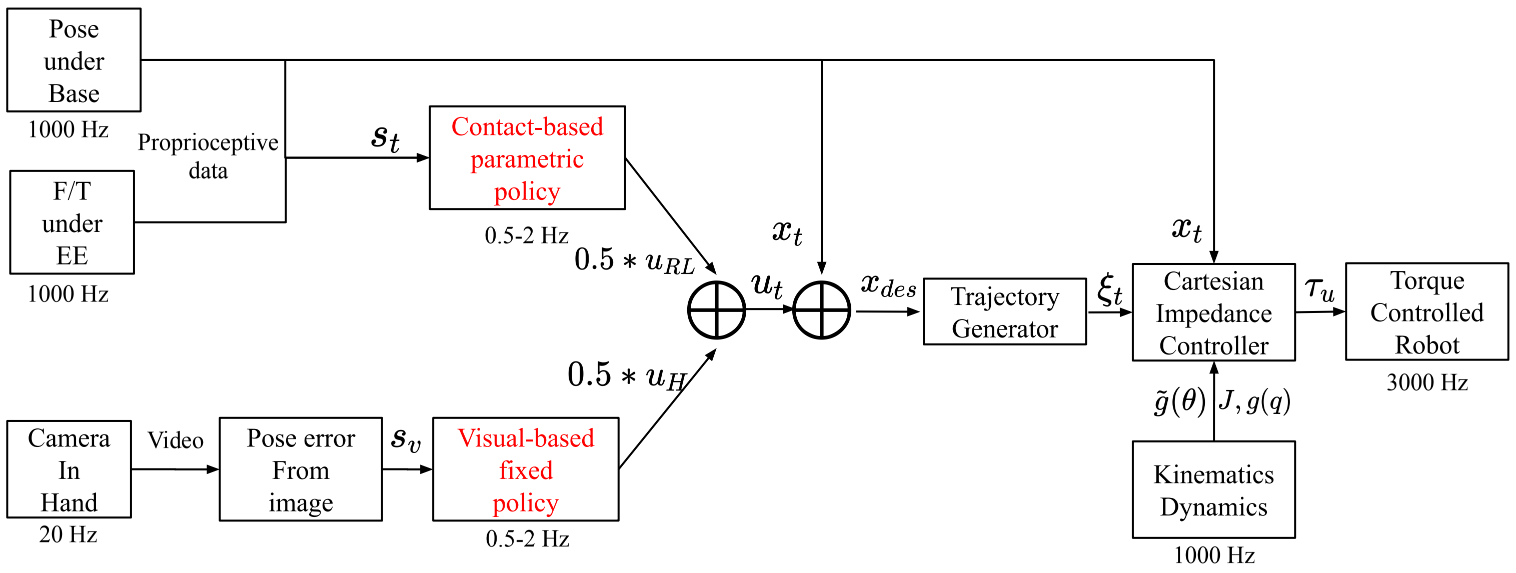


Figure 9.9: The control frame design for the RAM insertion task.

9.2.2.2 Reward Design

Depending on the pose error between the current and the target pictures, the reward function was set as follows:

$$r = \begin{cases} 1, & \text{(success)} \\ -2, & \text{(failed).} \\ 1 - 150\|s_{xy}\|_2 - s/s_{max}, & \text{(otherwise).} \end{cases}$$

Here, s_{xy} is the norm of the x and y errors of the images, s is the number of steps in one episode, and s_{max} is the maximum steps in one episode.

9.2.2.3 State Design

The estimated 6-DoF external forces and moments along the X, Y, and Z axis under the EE frame were obtained from the Franka controller. The contact force and the moments between the robot's EE (i.e., the RAM) and the slot were considered as the Markov Decision Process (MDP) states as follows:

$$s = [F_x, F_y, F_z, M_x, M_y, M_z] \tag{9.3}$$

We assume that the EE contacts the slot when the external force $|F| > 4$ N or the external moments $|M| > 0.4$ Nm, a value of ±1 means that contact is made, whereas 0 means that there is no contact with the encoding states.

9.2.2.4 Experiment Algorithm

Combine proactive action with policy framework as shown in Figure 6.3, the Algorithm 1 was improved as:

Algorithm 3 Visual Residual RL with Proactive Action

```
Require: RL policy $\pi_\theta$, fixed policy $\pi_H$.
for iteration=1 to M episodes do
    Copy latest policy $\pi_\theta$ from learning thread
    Sample initial state $s_0$
    for step=1 to N do
        Get action $u_{RL}$ by greedily picking from $\pi_\theta(s_t)$
        Get action $u_H$ from $\pi_H(s_v)$
        Output policy action: $u_t = (1-\alpha)u_H + \alpha * u_{RL}$
        if belief ==true then
            Get next state $u_t \rightarrow s_{t+1}$
        else
            Get next state $a_I, u_t \rightarrow s_{t+1}$
        end if
        Optimize $\pi_\theta$ with Equation (6.8)
        if EpisodeEnd == true then
            break
        end if
    end for
end for
```

9.3 Experimental Evaluation

9.3.1 Tasks Setup

In the ablation study experiment, the trained policy was evaluated by masking different modalities as four baselines given below:

1. *No vision*: masks out the visual part action; $\alpha = 1$;
2. *No RL policy*: masks out the RL part action; $\alpha = 0$;
3. *Random policy*: generates a random Q table
4. *No investigative action*: masks out the investigative action and chooses random action when the state is not clear.

The maximum steps was set as 10 and initial random errors ($|error| \in [2,3]mm$) was added in the x and y directions for each baseline in the ablation study experiment.

In the comparison study experiment, we compared the task success rates of our method with the other four baselines in the real scenarios (no maximum steps limit and no initial random errors for each baseline) by moving the motherboard, which are as follows:

1. Baseline 1: For normal teaching and direct insertion;
2. Baseline 2: For normal teaching with spiral exploration;

3. Baseline 3: For teaching with vision and direct insertion;
4. Baseline 4: For teaching with vision and spiral exploration.

9.3.2 Results and Discussion

Table 9.1: Ablation study of policy evaluation statistics

Baselines	**Result(success/total)**	**Total Time Cost**
No vision	92/200	1.09 h
No RL policy	112/200	0.65 h
Random RL policy	77/200	2.59 h
No investigative action	66/200	0.85 h
Our method	**179/200**	1.18 h

The policy was trained with 500 episodes, and each episode lasted a maximum of 50 steps. The training time for the exploration was approximately 150 min, which is much less than [77]. We specified discrete actions in this experiment, and the action execution had errors. The policy can increase the probability of success and decrease the cost steps but cannot guarantee success every time. The random errors were set for the initial pose of the robot; sometimes, the robot will successfully insert by chance and obtain a high reward in the early stage of training.

Table 9.1 shows the ablation study result of the policy evaluation statistics. *Random RL policy* and *No investigative action* had poor performances with success rates of 38.5% and 33%, respectively. *No vision* had a 46% success rate because of discrete overshooting actions whereas *No RL policy* had a 56% success rate because the RAM was always stuck by the short side of the slot. The proposed method had a success rate of 89.5%. Notably, the success rate of our method is limited by the maximum steps in the experiment.

The absence of either visual or correct forces/moments information negatively affected the task success rate, and wrong policy performance was even worse than without RL policy. Therefore, the *Random RL policy* and *No investigative action* had similar performances because the RL policy is always in conflict with the visual output action. None of the four baselines reached the same level of performance as the final method. With visual input alone, the robot sometimes cannot overcome the last small distance because of either the limited movement accuracy of the robot or contact friction, whereas the RL policy is capable of recovering from such issues, which could be proven in our method. Without the visual input, the robot will require more steps to find the proper pose for insertion and will always overshoot for some actions (i.e., drop out of the slot).

Table 9.2 shows a comparison of the success rates of different traditional method baselines. To simulate an industrial scenario, the additional random error and maximum step limit in the ablation study are removed. Obviously, baselines 1&2 work well only

Table 9.2: Comparison of success rates for different baselines

Baselines	**Fix motherboard**	**Move motherboard**
Baseline 1	97/100	0/20
Baseline 2	100/100	0/20
Baseline 3	98/100	81/100
Baseline 4	100/100	88/100
Our method	**100/100**	**100/100**

when the motherboard is fixed in the same position as in the teaching phase, so we only test 20 times in the “move motherboard” case for baselines 1&2 for saving time. The success rates for baselines 3&4 increased with vision correction, but still have failure cases due to the visual error. Our method shows a strong ability to tolerate environmental variations and resilience from stuck with full success, which really meets the requirements of industrial scenarios. Notably, in the comparison study, the increase of success rates is also related to the removal of initial errors and removal of the limit of the maximum steps.

In this chapter, RL with an operational space visual controller was introduced to solve position uncertainty problems in high-precision assembly tasks, and a proactive action idea was proposed to solve the POMDP problem using an investigative action. This proactive action idea could also be extended to other POMPD algorithms to predict and clarify the unclear states.

The proposed method could solve the shortage of traditional visual servoing methods by using our visual residual RL algorithm, which inherits some traditional controller parameters that make the setting up not fast enough. The SSD insertion scenario with our policy achieves full success with 100 episodes.

At the end of this chapter, I would like to thank the gorilla for the inspiration during my research (Figure 9.10).

Figure 9.10: Thanks to the clever gorilla for the inspiration during this research. This figure has the original author’s use and editing authorization.

Chapter 10

Combine LfD and LbE for 3C Machine Tending

10.1 Experiment Setup

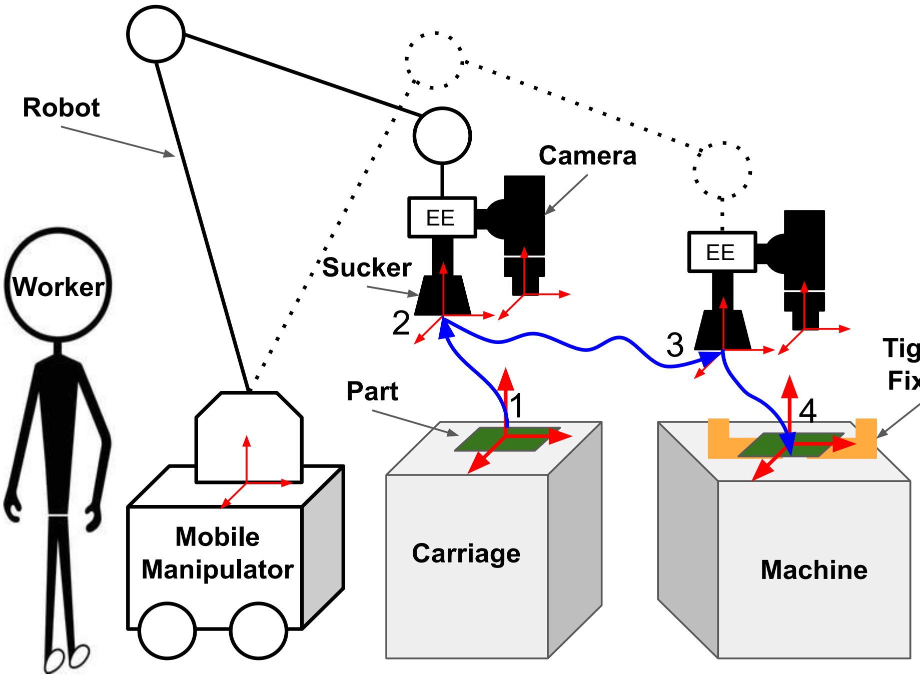


Figure 10.1: A mobile manipulator system performing a contact-rich tending task using a vacuum gripper.

In Figure 10.1, a contact-rich tending task is performed by a mobile manipulator system, the mobile manipulator system has one mobile manipulator and one robot, also a vacuum gripper is equipped at the robot EE. Moreover, a camera is installed at the robot EE to ensure the pose correction.

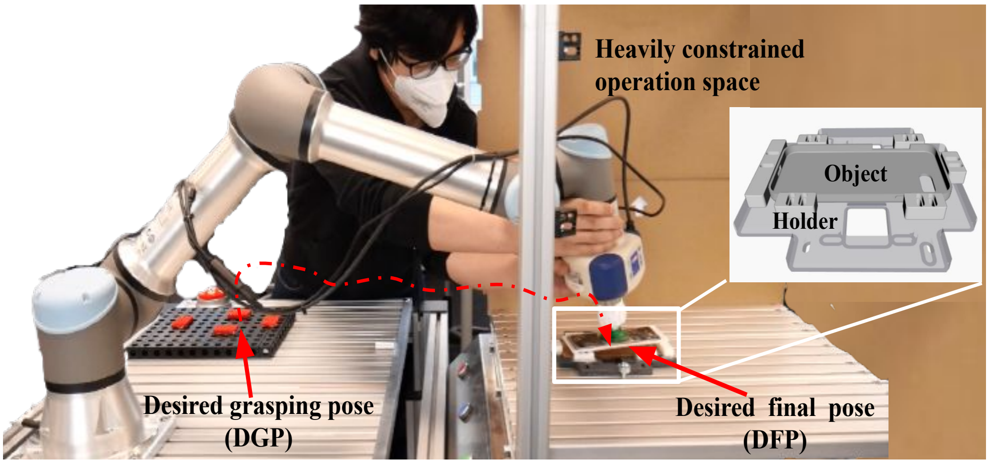

Heavily constrained operation space
Object
Holder
Desired grasping pose (DGP)
Desired final pose (DFP)

Figure 10.2: Hand-guiding teaching in a heavily constrained operation space to implement a tending task.

A UR5e robot [1] was used to implement our novel approach to facilitate the tending task Figure 10.1. The UR5e features a 6-axis and 5 kg payload, a working radius of 850 mm. It is equipped with a 6-DOF force/torque sensor on the EE. UR5e robot uses admittance controller [50] to achieve operational space force control. A Schmalz CobotPump ECBPi suction cup was installed beyond the force/torque sensor in order to ensure the detection of the contact force with the environment. An Intel RealSense Depth Camera D435i was attached to the EE to conduct the visual servoing process. Our policy was run using a Dell Precision 5510 laptop, and the updated position was sent to the UR5e controller. We used ur-rtde [2] as the Python interface for controlling and receiving data from the UR robot. A 6-DoF ATI Axia80 force sensor [3] was mounted under the holder in order to measure the operating force, and a low-pass filter with a cutoff frequency of 9.37 Hz was used to mitigate the force noise.

[1] https://www.universal-robots.com/products/ur5-robot/

[2] https://pypi.org/project/ur-rtde/

[3] https://www.ati-ia.com/Products/ft/sensors.aspx

10.2 Machine Tending Skill Implementation

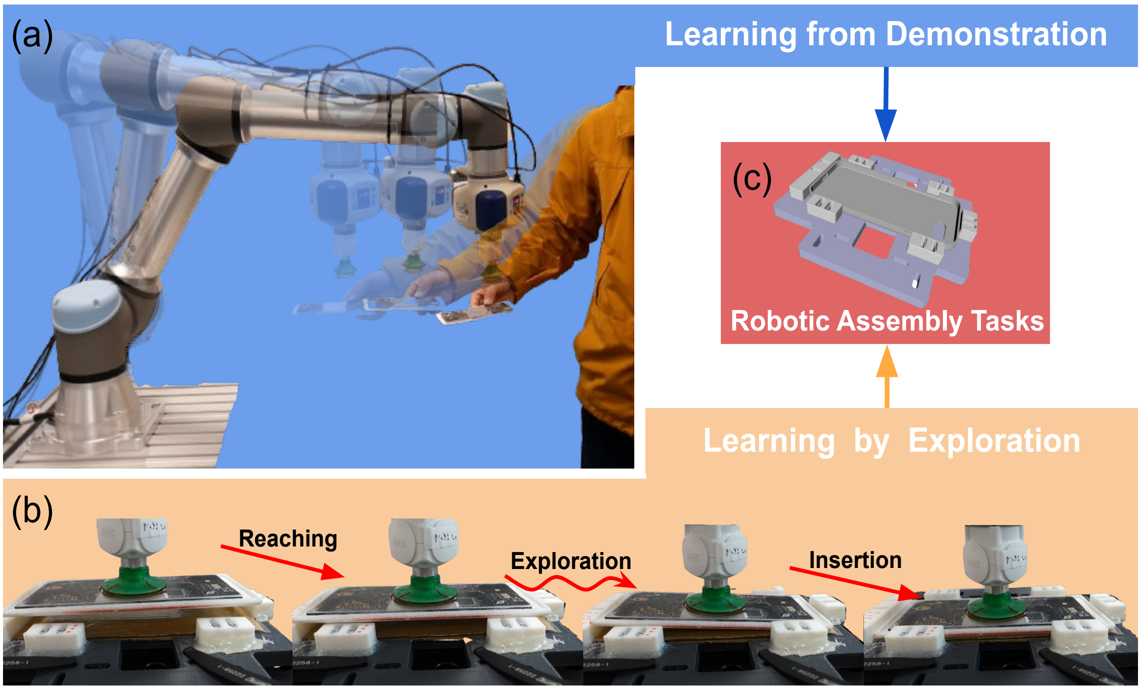


Figure 10.3: A robot arm and a suction gripper performing a machine tending skill consist of LfD and LbE.

In Figure 10.3, a machine tending skill consist of LfD and LbE is shown as Figure 10.3 (a) and (b), respectively. The gross motion is learned from human demonstration (LfD) and the fine motion is learned from exploration (LbE). An example of a contact-rich tending task is shown in Figure 10.3 (c).

10.2.1 Parameters Setup

In order to gain more contact experience, a positional error $\delta P \in [2,4]$ mm was added in a random direction during the training phase. The transitions (s_t, a_t, r_t, s_{t+1}) sampled from the environment were stored in a replay buffer [125]. The size of the experience's replay memory P_{replay} was 20, 000, the maximum number of training episodes M was 200, and the maximum number of steps k_{max} for the search phase was 50. The batch size P_{batch} was set to 64 to select random experiences from P_{replay}, and the discount factor λ was 0.5.

Carefully exploiting the natural constraints in the design of the learning policy was essential for the assembly tasks considered herein. It is obvious that the task is simplified if the motion is constrained in "wrong" directions. We utilized comparative experiments to investigate the ways in which different force-based actions utilize natural constraints. We set the same initial positional error of $\delta P \in [2,4]$ mm in a random direction, and then performed a random strategy to select the actions. We tested each action 200 times with a maximum step of 20 and a maximum force amplitude of 10N, the success rates of which are as follows:

Table 10.1: Success rate of different random actions

	Force control action	Success rate
1	Operational space controller [90]	32%
2	Fz with the force of another one dimension [56]	60%
3	Fz with the forces of other two dimensions	69%

The third discrete action as Equation (10.1) performs the best in the comparison experiment:

$$\begin{aligned} 1 :&[+^{e}F_{x}, +^{e}F_{y}, +^{e}F_{z}, 0, 0, 0] \\ 2 :&[+^{e}F_{x}, -^{e}F_{y}, +^{e}F_{z}, 0, 0, 0] \\ 3 :&[-^{e}F_{x}, +^{e}F_{y}, +^{e}F_{z}, 0, 0, 0] \\ 4 :&[-^{e}F_{x}, -^{e}F_{y}, +^{e}F_{z}, 0, 0, 0] \end{aligned} \tag{10.1}$$

Here, we set the orientation space as the position mode to fix the orientation of the suction cup. All the force amplitudes were set to 10 N. The state sampling frequency was 10 Hz, which could guarantee the observation of the contact states.

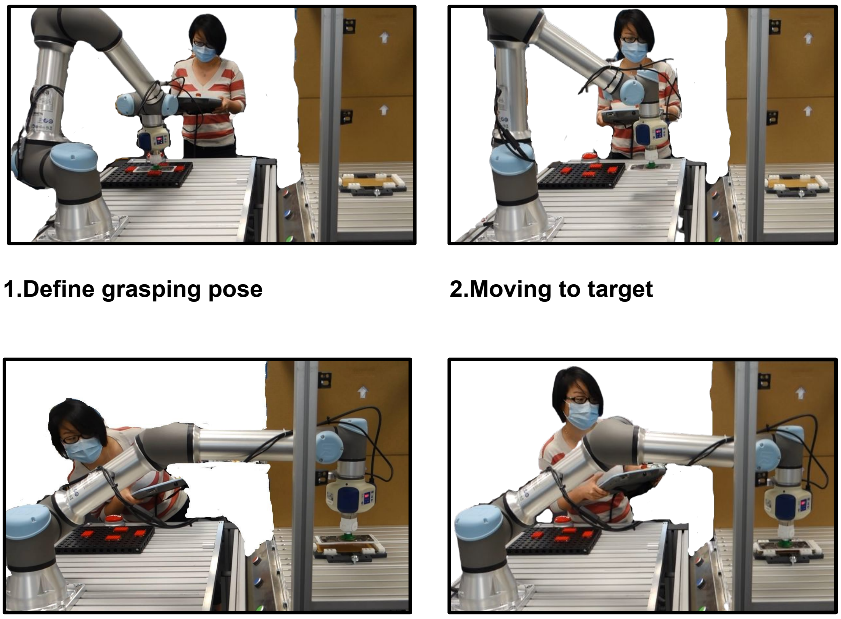
10.2.2 Task Setup

A tight clearance machine tending task (similar to the assembly task) was used to evaluate our method and the two baselines. One holder was installed in an opaque box to simulate a situation in which the field of view in the production line is obscured and an object is inserted into the holder. The users were provided a brief tutorial and allowed to practice until they felt ready. A group of four able-bodied volunteers (1 female, 3 male, aged: 24 to 35) participated in the experiment.

The commonly available methods for collaborative robots were selected as baselines [75]. We compared our proposed method with the following baselines:

1. **Teach-pendant + spiral searching.** The UR5e teach pendant with a UR PolyScope GUI was held in one hand by a user, who pressed the on-screen buttons to map the rate control of the EE's translation and rotation in the task space $\mathscr{T}$ with the other. At the DFP, a spiral search function similar to that outlined in the literature [111] was added;
2. **Hand-guiding + spiral searching.** The UR5e "Freedrive" mode was utilized, and users physically grabbed and exerted force to move the robot arm using one or two hands. At the DFP, a spiral search function was also added.

10.3 Experimental Evaluation



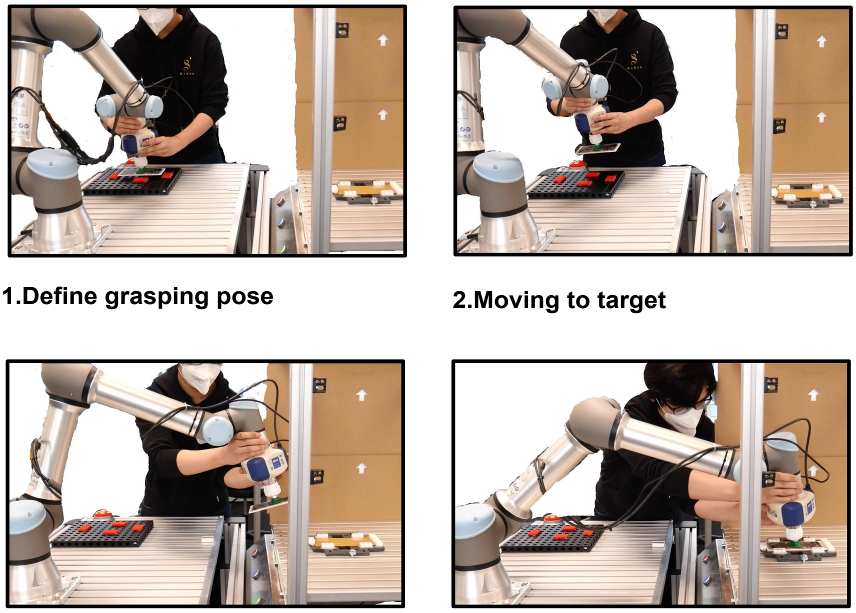
1.Define grasping pose

2.Moving to target

3.Check alignment

4.Insertion

Figure 10.4: Teach-pendant teaching: aligning the object with the target holder by eye is difficult.



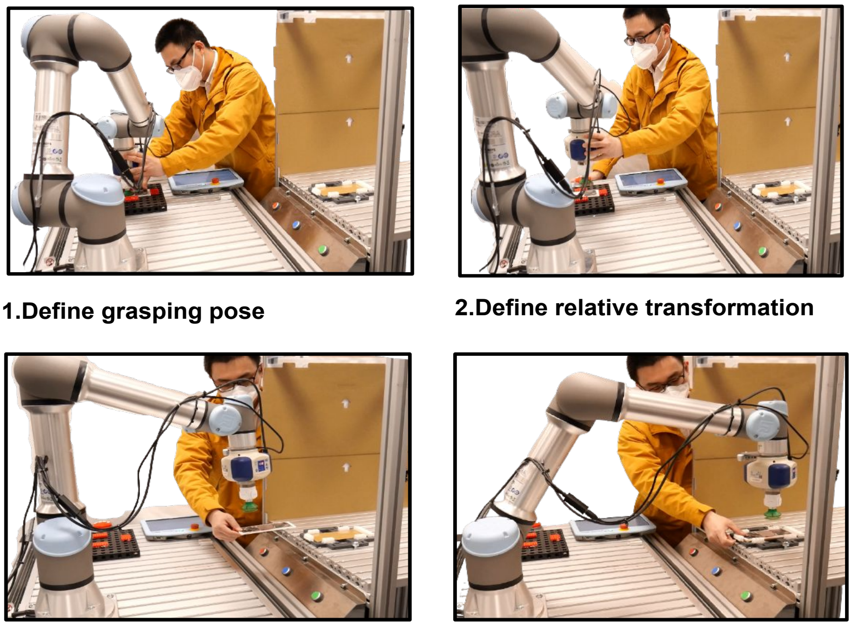
1.Define grasping pose

2.Moving to target

3.Moving to target

4.Alignment and insertion

Figure 10.5: Hand-guiding teaching: moving the robot EE with singular configurations and aligning with the target holder are difficult.



1.Define grasping pose

2.Define relative transformation

3.Guiding to target

4.Insertion

Figure 10.6: Visual servoing based LfD method teaching: contact-free guiding that is not physically demanding, and is easy to align without robot resistance.

We evaluated our methods in the **teaching** and **execution** phases. In our experiments in both phases, we aimed to answer the following questions: (**Q1**) Can our method maintain fast and easy programming abilities even in constrained operation spaces? (**Q2**) Can our method retain the execution success rate against positional uncertainty? (**Q3**) Can our method reduce the risk of damage during the operation of an object?

The following data were used to compare the advantages and disadvantages of the different methods:

1. **Teaching time:** The task's completion time was measured as the time cost by the user to move the robot from the DGP to the DFP (Figure 10.2);
2. **Execution success rate:** After the teaching phase was completed, the trajectory of the demonstration was executed and the insertion success rate was tested. Contrary to the **"perfect"** group, an error of $\delta P \in [2,4]$ mm in a random direction was added on DFP to simulate the pose uncertainties in the **"uncertainty"** group (Figure 10.2);
3. **Risk of damage:** The risk of damage caused by the operating force was ignored in previous studies [77], [89], [90], [76], [111]. In this study, the maximum absolute contact force during contact operation was employed to evaluate the risk of damage.

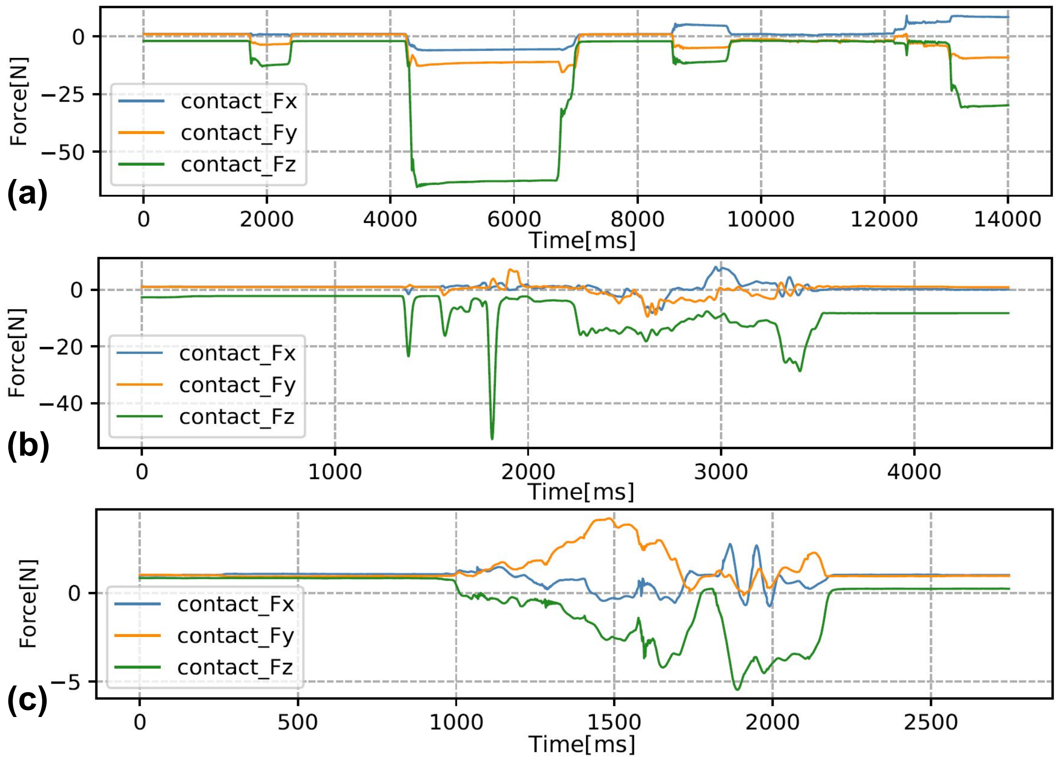


Figure 10.7: Contact force during teaching. (a) This curve shows the contact characteristics of the teach-pendant teaching method: the user uses an "observe-move" strategy and only makes adjustments after finding unsuccessful insertions, hence, the contact force is maintained during the observation. (b) The hand-guiding method: the object frequently collides with the holder when it is not inserted (impact force in the figure), and there is a continuous contact force after inserting the object. (c) Our method produces a small contact force (less than 5 N) in the teaching phase.

Table 10.2: Evaluation in the teaching phase

Teaching phase	Time cost	Maximum contact force
Teach-pendant	60–120 s	15–50 N
Hand-guiding	**15–42 s**	30–60 N
Our method	**23–30 s**	**3–10 N**

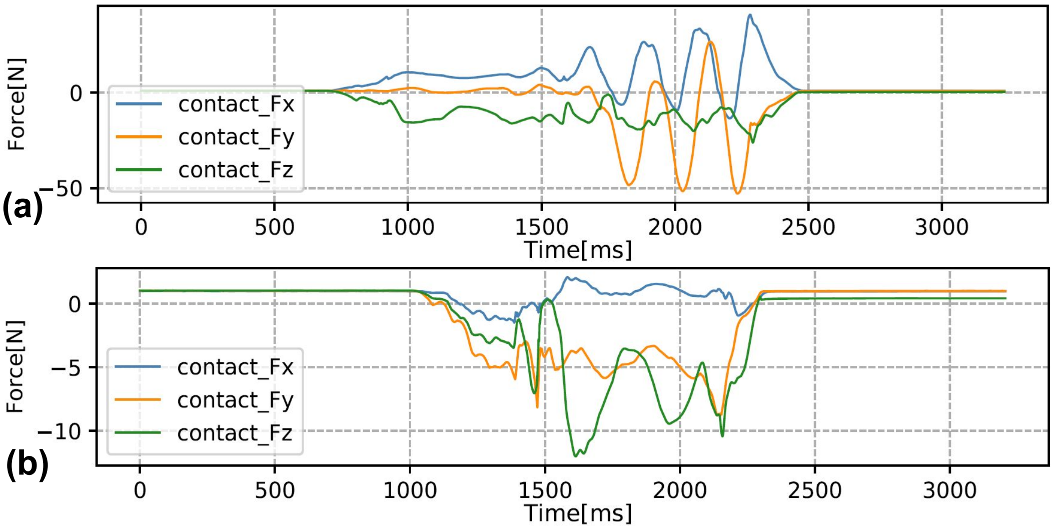


Figure 10.8: Contact force during execution. (a) Spiral exploration method: the first half of the curve (700-1500ms) is not yet constrained by the holder, and the contact force is small and stable. After 1600ms, the object is inserted into the holder, immediately generating a larger contact force. Due to the accuracy issue of the UR5e force sensor, the spiral movement generates a larger contact force than the stopping threshold. (b) RRRL method: the maximum contact force is around 10N because of the force action amplitude limitation.

Table 10.3: Evaluation in the execution phase

Execution phase	**Success rate**		**Maximum contact force**
	Perfect	**Uncertainty**	
Only teach-pendant	55/100	17/100	15 N
Only hand-guiding	33/100	5/100	15 N
Teach-pendant + spiral searching	69/100	47/100	35 N
Hand-guiding + spiral searching	51/100	33/100	35 N
Our method	**95/100**	**91/100**	**15 N**

Overall, the results of 12 group robot teaching and 1000 group robot execution were obtained.

The teaching phase test scenarios can be observed in Figure 10.4, Figure 10.5, Figure 10.6 and the results are presented in Table 10.2. Although it is more generalized, our method features a similar time cost to the hand-guiding method. Using the hand-guiding method, male volunteers always required less teaching time than females owing to physical demands. All the volunteers noted that the robot “required excessive force to move, particularly near the boundaries”. Our method does not require physical contact with the robot, therefore, it is not physically demanding. All the volunteers required considerably longer teaching times when the teach-pendant teaching method was used. A considerable amount of time was spent aligning the object with the target, and the tricky sight angle made this even more difficult. In contrast, our method does not require a human to align the object with the target, but enables the robot arm to actively track the object to attain the target position. Thus, the setup is performed quickly. Hence, the answer to **Q1** is yes. Furthermore, our method produces minimal contact force on the environment (Figure 10.7), as our schematic approach is identical to the human tending one. The other two methods resulted in considerably less transparency during the interactions with the environment owing to the resistance of the robot arm itself or the inability to interact with the environment in terms of force.

The execution phase results are presented in Table 10.3. The success rate of our method far exceeded those of the other reference baselines. We found that **the elasticity of the suction cup** had a major influence on the accuracy of the demonstrated target position. The contact force at the end of the demonstration could induce the **deformation of the suction cup** and thus affect the actual target position. Using our method, the robot EE did not contact the target environment at the end of the teaching phase; hence, there was no contact force and therefore no deformation of the suction cup. Finally, high target position accuracy was achieved. Additionally, the RRRL policy could determine force actions based on the contact state of the object and holder, thus greatly improving the insertion success rate. Our method can guarantee small contact forces owing to the amplitude limitation of the force actions (Figure 10.8(b)). Therefore, the answers to **Q2** and **Q3** are also yes.

In this chapter, we combined visual servoing based LfD and force-based LbE to facilitate the rapid and intuitive execution of the assembly tasks that require minimal user expertise, involvement, and physical exertion. The efficiency of the proposed method was validated via a series of experiments that involved the execution of a tending task using a robot arm and a suction cup system.

In a challenging setting designed to simulate heavily constrained operation spaces, which is very common in actual factories, experiments that compared our method with two commonly used baselines, namely teach pendant-based and hand-guiding teaching, were performed. Our method realized the best feedback in terms of both subjective and objective evaluations.

Chapter 11

Sim-to-Real Learning for Peg-in-Hole Manipulation

In order to validate the framework and explore its generalization, two experiments are implemented in this work. The first framework validation experiment uses a UR5e robot and focuses on the different observation states, also the different colors of the peg and background are investigated. Based on the result of the first experiment, the second generalization experiment uses a Diana7 robot and tests the different peg shapes.

11.1 Sim-to-Real Learning Validation Experiment

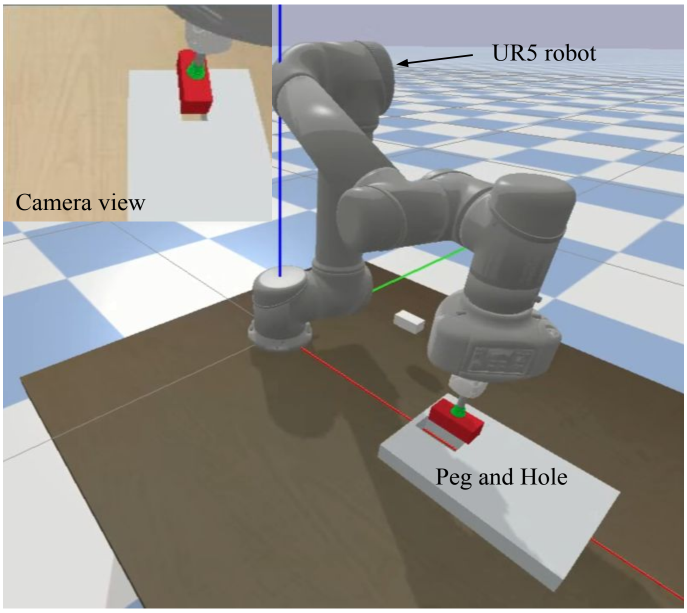


Figure 11.1: A PiH setup with a UR5e robot in simulation.

11.1.1 Simulation Setup With a 6 DoF Robot

In this section, we introduce the PiH task to validate our framework and explain the experimental results for both the simulated and real-world environments, in which we address the following questions:

1. Will all observation spaces work well in our framework?
2. Can our trained policy be transferred to a real-world environment successfully?
3. How does our framework perform compared to other insertion methods in terms of the success rate?
4. What is the robustness of our framework under external perturbations and target uncertainties?

Regarding the first question, to compare the performances of different observation spaces with our framework, we test the scene of a white block with a metallic texture. We use the success rate of a complete insertion in the simulation as a criterion to evaluate the performance of different observation spaces. A convolutional neural network is utilized as a part of the SAC network for training using the inputs from the RGB and grayscale observation spaces. A VAE is used to obtain a latent representation of the input image. The encoder part allows the compression of the original image to a lower-dimension vector that contains the important information. We first generate a series of images of the robot state by executing random actions in the simulator as the training dataset and then train the VAE using this dataset. Thereafter, we extract the encoder as a part of the SAC network. We use the generated simulated images of the RGB observation space (size=$3 \times 64 \times 64$) to train the VAE.

Table 11.1: Success rates of three observation spaces

Total episodes	**Observation space**	**Success rate**
3000	Gray $1 \times 64 \times 64$	0%
	RGB $3 \times 64 \times 64$	0%
	Latent 128×1	**96%**

We train the agent using cumulative episodes, and the results are shown in Table 11.1. With the latent representation as policy input, the policy converged and the success rate could reach 96% at checkpoints 3000. Even increasing the episodes to 10000, the results remain the same. Thus, we choose latent representation observation space: 128×1 vector, as the states, to perform the remaining experiments.

Although we demonstrate the policy performance before using a latent representation observation space in the scene of a white block with a metallic texture, it is unclear

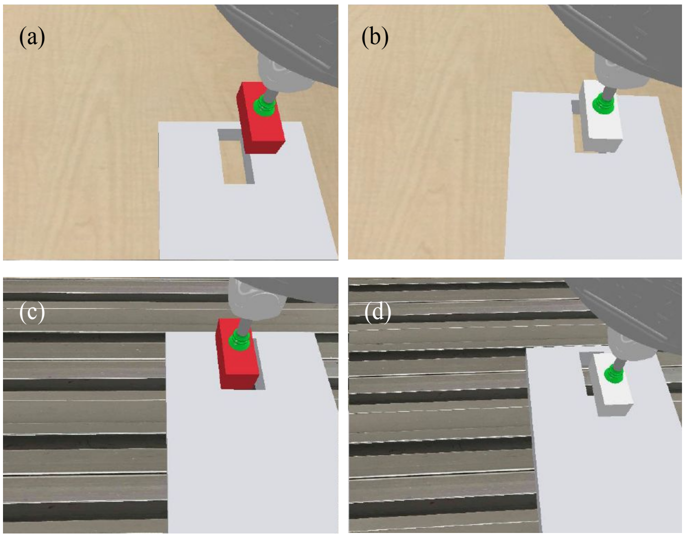


Figure 11.2: Four different scenes. (a): a red block with a wooden texture. (b): a white block with a wooden texture. (c): a red block with a metallic texture. (d): and a white block with a metallic texture.

whether the difference in the scene will influence the performance. A high success rate must be achieved in the simulation environment to perform further real experiments. Additionally, to verify the generalization of the framework, we consider the permutation of four environmental scenes with two blocks and two textures: a red block with a wooden texture, a white block with a wooden texture, a red block with a metallic texture, and a white block with a metallic texture (Figure 11.2). Every scene is trained 3000 episodes in the simulation environment with a Dell Precision 5510 laptop IntelCore i7–6700HQ CPU.

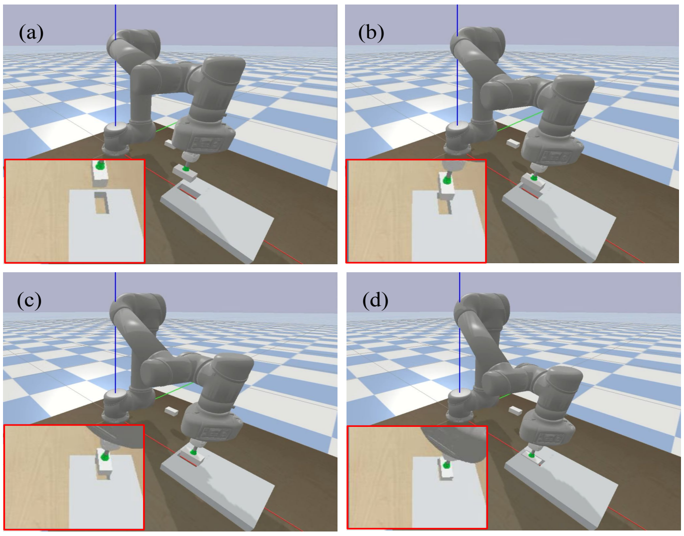


Figure 11.3: A example of UR5e PiH sequence in the simulation.

Table 11.2: Success rates of different scenes (simulation)

Evaluate Trials	Scene	Success Rate
500	red block with wooden texture	96%
	red block with metal texture	70.5%
	white block with wooden texture	99%
	white block with metal texture	96%

Table 11.2 shows the success rate of the framework obtained under different scenes. All scenes achieved a success rate higher than 90%. The white block with a wooden texture reached a 99% success rate. We compare the performance of this approach in different scenes, and one example of PiH execution sequence is presented in Figure 11.3.

11.1.2 Sim-to-Real transfer Setup With a 6 DoF Robot

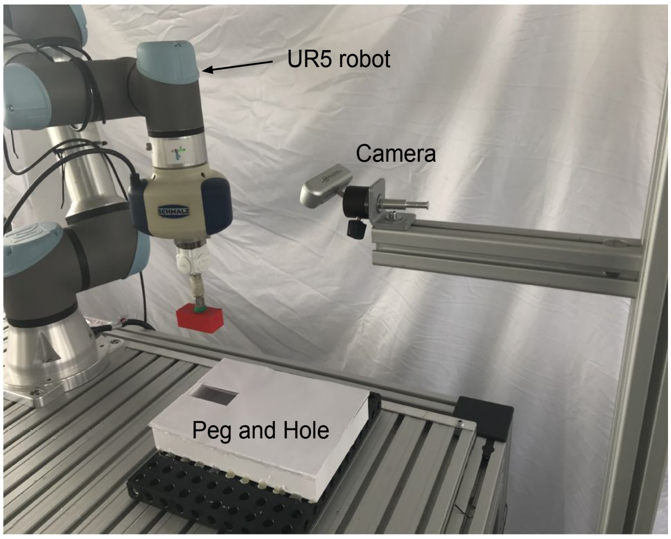


Figure 11.4: A PiH setup with a UR5e robot in the real world.

In this real-world environment setup (Figure 11.4), a UR5e robot[1] is used to perform a peg-in-hole insertion task. This 6-axis robot features a 5 kg payload and a working radius of 850 mm. It is equipped with a 6 degree-of-freedom force/torque sensor on the EE. The robot uses an operational space admittance controller [50] with a 500 Hz control

[1] https://www.universal-robots.com/products/ur5-robot/

rate. The blocks are mounted behind the force/torque sensor to ensure the detection of the contact force with the environment.

An Intel RealSense D415 camera[2] is fixed on the platform to observe the operation. The position and orientation of the camera are selected to ensure the block and hole are visible during most of the training time. Figure 11.4 shows our hardware setup in the experiment. In our experiment, we use a white and a red block with the same dimensions of $65 \times 30 \times 25$ mm, and a white block with a hole size of $70 \times 35 \times 30$mm. The clearance in each direction (i.e., length and width) is 5 mm. We select this setup because we aim to establish a potential scenario in which a packed data cable is inserted into a phone box in the mobile phone assembly line [142].

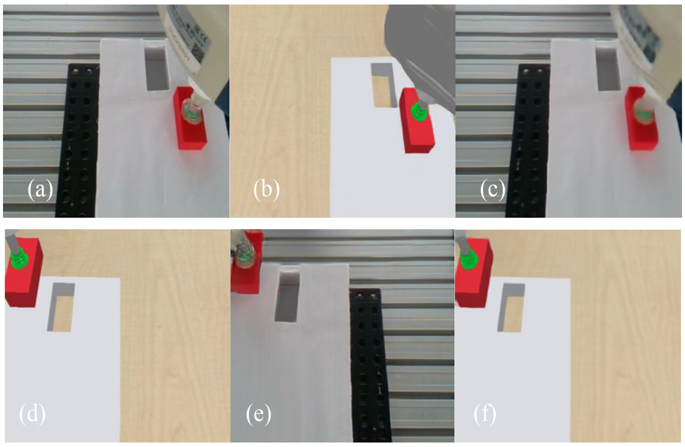


Figure 11.5: Domain adaption with CycleGAN approach. (a) is from the real domain and (b) is a fake image belonging to the simulation domain mapped by the function G, (c) is a reconstructed image mapped by the function F from (b). Vice versa for (d), (e) and (f).

The CycleGAN is introduced to perform the domain adaptation process to transfer the image distribution from the real world to the simulation. We capture 200 images of the robot state in the real world and then generated a training dataset along with 3000 simulated images to train the CycleGAN. We train the CycleGAN model on four Nvidia 1080 Ti GPUs. The domain adaptation results of the trained CycleGAN with our setup inputs are shown in Figure 11.5.

In this work, a modified admittance force controller based on the direct force control concept ([139]) is used, and the stiffness is set to zero. Thus the dynamic model is as follows:

$$M\ddot{x}_e + B\dot{x}_e = f_d - f \tag{11.1}$$

M is the desired EE inertia and B is desired EE damping. f_d is the command force and f is the contact force between robot EE and the environment. Then the demand acceleration is obtained:

$$\ddot{x}_e = M^{-1}(f_d - f - B\dot{x}_e) \tag{11.2}$$

[2] https://www.intelrealsense.com/zh-hans/depth-camera-d415/

By introducing the differentiation term and calculation cycle T into the equation, the demand velocity and position can be calculated by several integration operations:

$$\ddot{x}_e^{t+1} = M^{-1}\left(f_d - f - B\dot{x}_e^{t}\right) \tag{11.3}$$

$$\dot{x}_e^{t+1} = \dot{x}_e^{t} + \ddot{x}_e^{t+1}T \tag{11.4}$$

$$x_e^{t+1} = x_e^{t} + \dot{x}_e^{t+1}T \tag{11.5}$$

$$x_c^{t+1} = x_d^{t} + x_e^{t} \tag{11.6}$$

In this experiment, we only need a force controller, thus the new desired compliant position x_c is calculated based on the desired force f_d, desired positionx_d with equations from Equation (11.1) to Equation (11.6), M and B can be adjusted according to the behavior requirement.

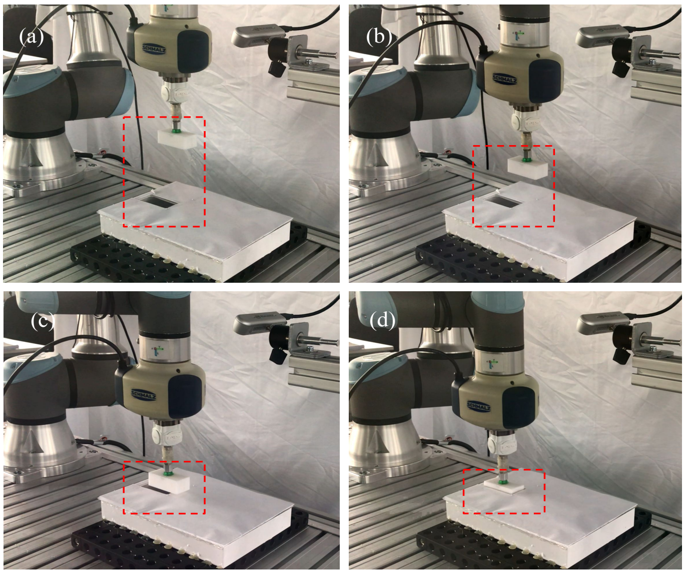


Figure 11.6: UR5e execute the PiH operation successfully.

Based on the previous results listed in Table 11.2, we can conclude that the scene with wooden texture achieves the highest success rate with our policy. Hence, we transfer the real-world image to a scene of a block with a wooden texture using the DA method to evaluate our framework in a real-world setup. We define three situations when testing the policy in the real world as [78].

Table 11.3: Performances in real environment setups

Scene	Complete insertion	Touched the box	Failed
Red block	86/100	10/100	4/100
White block	88/100	12/100	0/100

Complete Insertion means that the robot accomplishes the insertion task completely. *Touched the box* implies that the peg was moved in the right direction, but the insertion is not completed. *Failed* indicates a situation in which the robot moves far away from the target in the wrong direction or performs unexpected movements.

During the execution (Figure 11.6), we randomly occlude the camera's field of view for several seconds and push the robot in the wrong direction to the target hole to evaluate the system robustness to external perturbations. The performance of the physical robot in the real-world setup is summarized in Table 11.3. We obtain an average success rate equal to the method reported in the literature [78] with a safer sim-to-real framework as we limit the force command amplitude during the control.

11.2 Sim-to-Real Learning Generalization Experiment

In the previous Section 11.1, we verified that our framework works well with a block insertion task, In this section, a seven DoF Diana7 robot (as shown in Figure 11.8) and 6 different pegs (as shown in Figure 11.7) are used to evaluate the generalization of our framework.

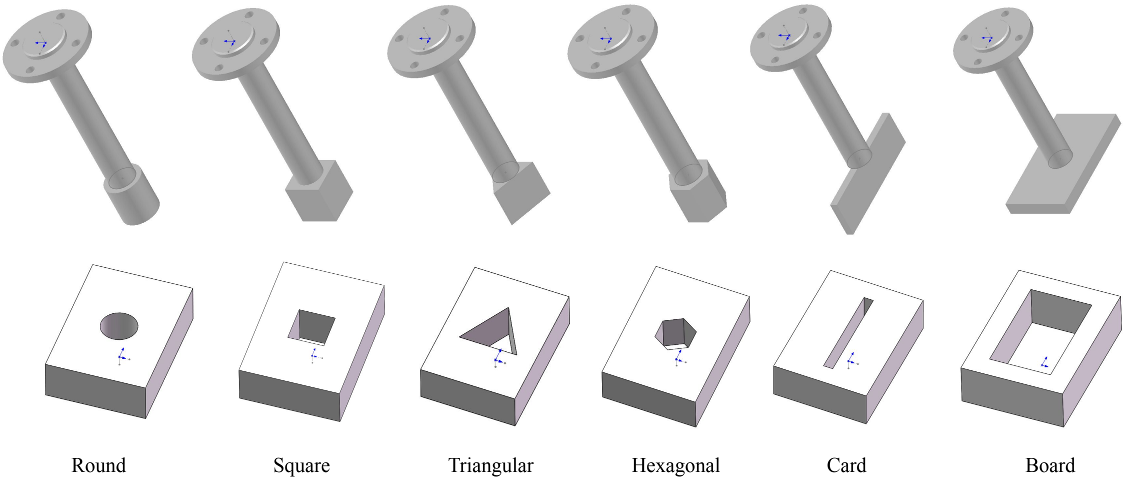


Figure 11.7: Six kinds of pegs and holes.

11.2.1 Simulation Setup With a 7 DoF Robot

In the validation experiment, we proved that the latent representation spaces (128×1 vector) perform the best in the training in simulation, thus we directly introduce the latent representation as the observation space in this section.

A URDF file is used to describe the simulation robot and its environment for each kind of peg and hole. The same collision and visual mesh files are

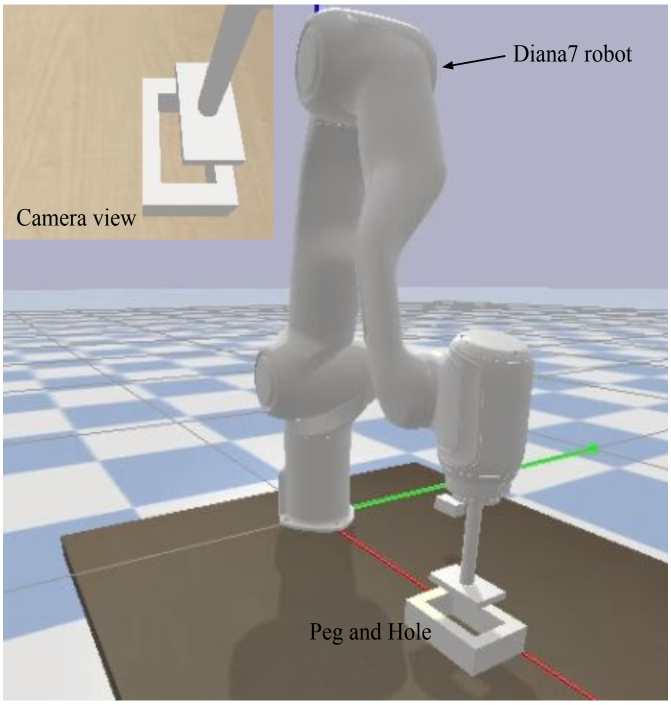


Figure 11.8: A PiH setup of board insertion with a Diana7 robot in simulation.

used to guarantee the visual input and collision feedback are aligned. Three Pybullet functions `createVisualShape`, `createCollisionShape` and `createMultiBody` are used to build a high-precision visual and collision model. According to our experiment, the collision model is necessary for the training of the policy, a collision-free model achieves a 0% success rate in the training process.

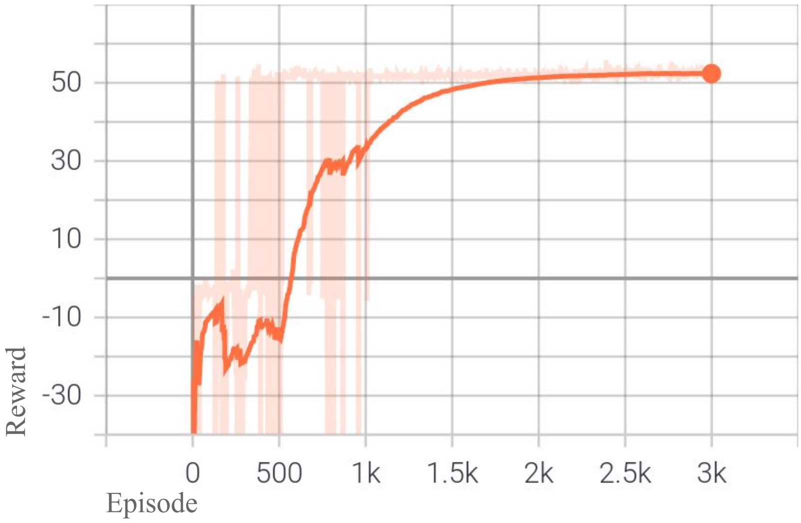


Figure 11.9: A training reward curve with Diana7 simulation setup.

We choose six different kinds of pegs and holes, namely, round shape, square shape, triangular shape, hexagonal shape, card shape and board shape, to validate the framework generalization ability. Every peg and hole has the same clearance of 1 mm. An example of a board insertion task with a Diana7 robot in simulation is shown in Figure 11.8.

A PiH example of the board insertion task training reward curve of the SAC agent is shown in Figure 11.9. We trained the policy with 3k episodes and each lasting 50 steps. It is clear that the agent converges to a policy that allows it to successfully complete the task after nearly 1000 episodes. Due to the entropy regularization concept, the agent mainly explores its environment until the 1000th episode. The smoothing weight coefficient is set to 0.99 in order to show the orange solid learning curve clearly with TensorBoard's built-in function. Moreover, the ActorLoss and CriticLoss both go up first and then go down which means the recommended actions by the actor are maximizing the rewards during the training.

As images are less capable of showing small pose errors, we also found that the robot struggles with alignment when pegs are close to holes as stated in [79], while our policy learned a more complex manipulation method than just sliding around the surface as described in [79], with our trained policy, the agent can move up and realign itself again after stuck around the hole.

Table 11.4: Six kinds of PiH insertion success rate in simulation.

3-D printed pegs	Round	Square	Triangular	Hexagonal	Card	Board
Success rate	100/100	100/100	100/100	100/100	100/100	100/100

In Figure 11.10, a Diana7 robot is performing the board insertion task. First of all, the robot starts to move the peg at the random initialized position up to 20 cm high from the target hole as shown in Figure 11.10 (a); then the peg is moved to the pose that is closer to the hole as shown in Figure 11.10 (b); lastly, the robot performed several exploration actions and find the hole as shown in Figure 11.10 (c) and (d).

11.2.2 Sim-to-Real transfer with a 7 DoF Robot Setup

In this part, an impedance-based force controller is developed and implemented in the PiH task as shown in Figure 11.11. Based on Equation (11.1) and in order to limit the maximum velocity, the damping parameter B is set to:

$$B = \frac{f_d}{\dot{x}_{\max}}. \tag{11.7}$$

The maximum velocity $\dot{x}_{\max}$ is set manually. In this experiment, the maximum velocity is set to 1 cm/s.

The same method as described in Section 11.1.2 is used to perform the domain adaptation process to transfer the image distribution from the real world to the simulation. Even with all-white color for peg and hole, the domain adaptation still performs well with clear relative positions and sharp edges as shown in Figure 11.12.

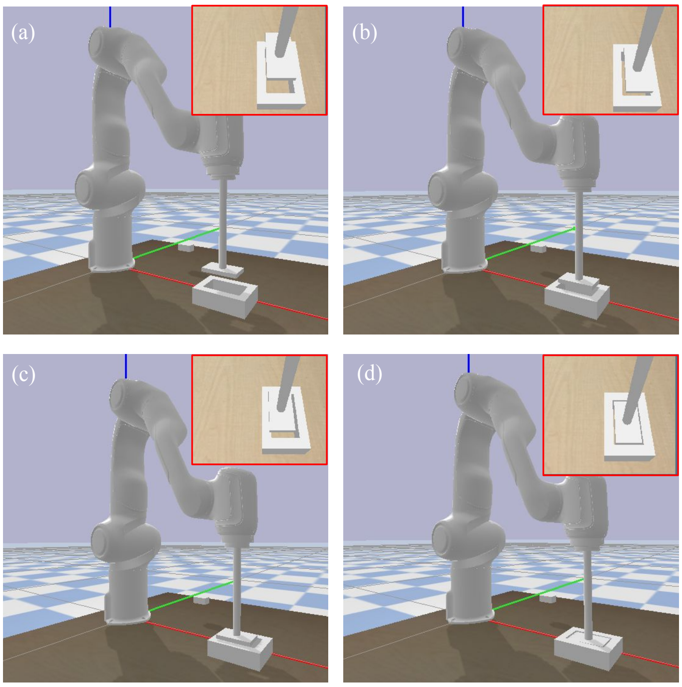


Figure 11.10: From (a) to (d): a board PiH task sequence is performed by Diana7 robot in simulation.

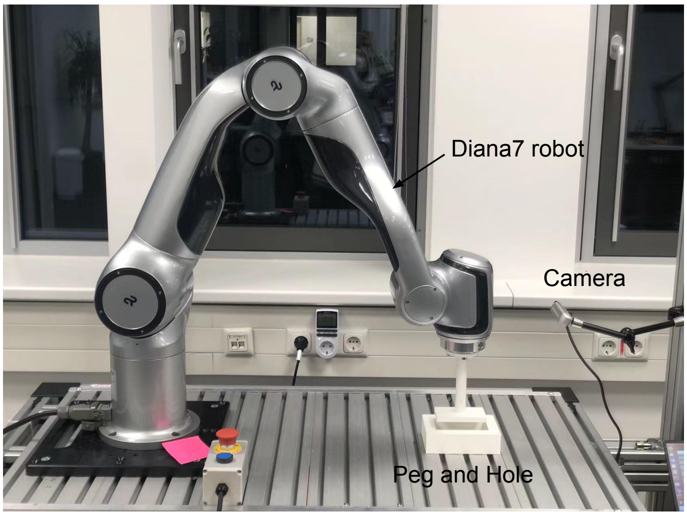


Figure 11.11: A PiH setup with a Diana7 robot in the real world.

A board peg insertion task is performed in real setup and a 80% success rate is achieved in the experiment as shown in Figure 11.13 (a)–(b), the up left corner shows the camera view of insertion status. As peg and hole are both white, it is not easy to distinguish their relative positions in the overall scene camera. According to our analysis, the success rate is limited by the real robot behavior, as the Diana 7 robot only supplies one direction of force control, thus we have to use an impedance controller in the other two directions, and the contact force is not able to control. We believe that our

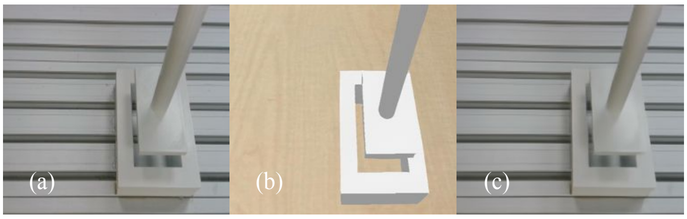


Figure 11.12: (a) to (b): from the real domain to fake image belonging to the simulation domain mapped by the function G; (b) to (c): from the fake simulation domain to a reconstructed image mapped by the function F.

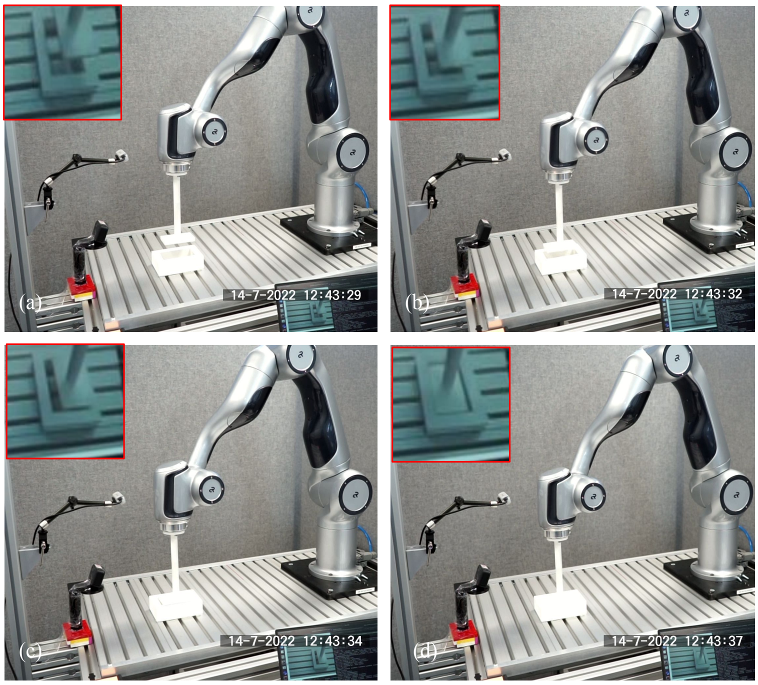


Figure 11.13: Diana7 insertion sequence in real setup.

next work will focus on the policy as well as robot performance improvement, thus not all the shapes (as shown in Figure 11.7) are used in this sim-to-real transfer experiment.

Part IV

Summary

In the previous parts, the main contributions details regarding the DRL for the force-controlled robotic manipulation is presented. In this part, we will give a conclusion in Chapter 12. Then, the limitations will be discussed in Section 13.1. In the end, the potential future work directions will be touched in Section 13.2.

Chapter 12

Conclusion

This thesis mainly focuses on the research of using visual and force information to guide the contact-rich assembly operation as described in Part II, the skills and frameworks are mainly used in PC and phone assembly lines as described in Part III. During my four years of research on this topic, I deeply understand the importance of force and vision information in assembly tasks: for example, in the teaching phase, the hand-guiding function needs high transparency in order to avoid the huge collision force with the target or environment; in the execution phase, visual information is very helpful to overcome the pose uncertainty while force information is always used for exploration. Moreover, we found that RL algorithm is quite useful, especially where traditional policies do not work. More details about contributions of this thesis are as follows:

- **Visual Residual RL:** In this method, we combined RL with an operational space visual controller to solve position uncertainty problems in high-precision assembly tasks, and we proposed a proactive action idea to solve the POMDP problem using an investigative action. The proposed method could solve the shortage of traditional visual servoing method by using our visual residual RL algorithm, which inherits some traditional controller parameters that make the setting up not fast enough. As shown in Chapter 9, our method shows a strong ability to tolerate environmental variations and resilience from stuck with full success, which really meets the requirements of industrial scenarios. This work also inspired our follow-up research.
- **Visual Servoing based LfD:** With this newly developed LfD based on visual servoing method, the fast, easy, and accurate robot setup in heavily constrained spaces was successfully implemented. Experiments that compared our method with two commonly used baselines, namely teach pendant-based and hand-guiding teaching, were performed. Our method realized the best feedback in terms of both subjective and objective evaluations. Moreover, our method produces minimal contact force on the environment owing to the high transparency during the teaching (Our method does not require physical contact with the robot, but can let the user feel contact force with the target or environment).
- **RRRL Policy:** With our LbE based RRRL policy, the force-torque information

is used in the net to overcome pose uncertainty in contact-rich tending operation. The success rate of our method far exceeded those of the other reference baselines. The reason is our RRRL policy could determine force actions based on the contact state of the object and holder, thus greatly improving the insertion success rate. Moreover, our method can guarantee small contact forces owing to the amplitude limitation of the force actions.

To our knowledge, we are the first to judge learning-based assembly strategies based on contact force as this is an important issue for contact-rich operation.

- **CycleGAN and Force Control based Sim-to-Real Transfer of Robotic Assembly:** A vision-based sim-to-real learning framework is proposed to perform assembly tasks. In this work, we proved that our sim-to-real framework is a valid approach to solving the peg-in-hole task both in simulated and real-world environments. By employing DA and force controller, we can directly transfer the policy that was trained in a simulator to a real-world setup. Moreover, we evaluated different observation spaces and proved that the latent representation (i.e., low dimension) can accelerate the convergence of policy learning and afford a higher success rate for the task than end-to-end learning using raw image input. The importance of force control is shown by the fact that in real-world experiments

- **Pushing-based Hybrid Position/force Assembly Skill:** In this method, we present a pushing-based hybrid position/force assembly skill that can maximize environmental constraints during task execution. To the best of our knowledge, this is the first work that considers using pushing actions during the execution of the assembly tasks. We have proved that our skill can maximize the utilization of environmental constraints using mobile manipulator system assembly task experiments, and achieve a 100% success rate in the executions.

Chapter 13

Limitations and Outlook

13.1 Limitations

Despite the promising results presented in the previous chapters, in this chapter, the drawbacks of the different components of the proposed concepts are discussed.

The main limitation of this work is the algorithm generalization. For example, in the methods of visual residual RL and RRRL policy, we try to increase our method's localized generalizability with visual and force information which makes similar scenarios can import our skills directly without retraining. However, when the environment changes a lot, the policies always need to retrain and it wastes lots of time and resources.

The other limitation of the visual servoing based LfD method is the teaching trajectory accuracy. As the visual servoing is used to track the object and it inevitably has errors, thus the recorded trajectory can also have some errors. In the experiment in this thesis, we slow down the moving velocity of the object in order to reduce the following error, however, a better solution should be designed.

Furthermore, we noted the negative effect of the elasticity of the suction cups on the accuracy of the position demonstrations. We have investigated ways to utilize this elasticity using a traditional approach (in Section 5.2.2.1), however, a learning based approach to utilize the EE elasticity is not being achieved.

Lastly, the CycleGAN and force control based Sim-to-Real transfer framework still has a huge reality gap in contact-rich operations including but not limited to friction model, contact model, and sample efficiency.

13.2 Outlook

Although several issues and problems are solved in this study, however, the research never ends.

The generalizability and efficiency of the visual and force based assembly skills should be further improved based on the limitation summarized in the last section. Moreover, several possible improvements and new directions can be studied in the future.

- **Partial General Robotic Assembly Framework:** We plan to analyze more contact-rich tending tasks and more types of grippers to refine our method and improve its generalizability. While assembly skills comparable to those of human hands are impossible to achieve in the short term, thus partial generalization is our next target that robots can do 95 % operations (now around 40–60 %) in 3C production lines. We believe more sensor modalities are necessary for future research.

- **DMPs based Visual Servoing LfD:** We plan to introduce DMPs to the current visual servoing LfD, thus the trajectory can be automatically modified when a new target is given, moreover, a compensation method of the visual servoing follow error will be also invested, thus the trajectory will be more accurate.

- **Sim-to-Real Transfer Learning:** Our method can be optimized further in terms of performance and generalization ability. For example, the scene of a red block with a metallic texture in the simulation achieves a success rate of only 70.5% considerably worse than those achieved using the other three scenes. Moreover, a more complex action space with both translation and rotation, $[\Delta x,\ \Delta y,\ \Delta z, \Delta r_x, \Delta r_y, \Delta r_z]$, can be designed for training and implementation. In the end, investigating the application of the sim-to-real approach to more industrial robotic tasks will be interesting.

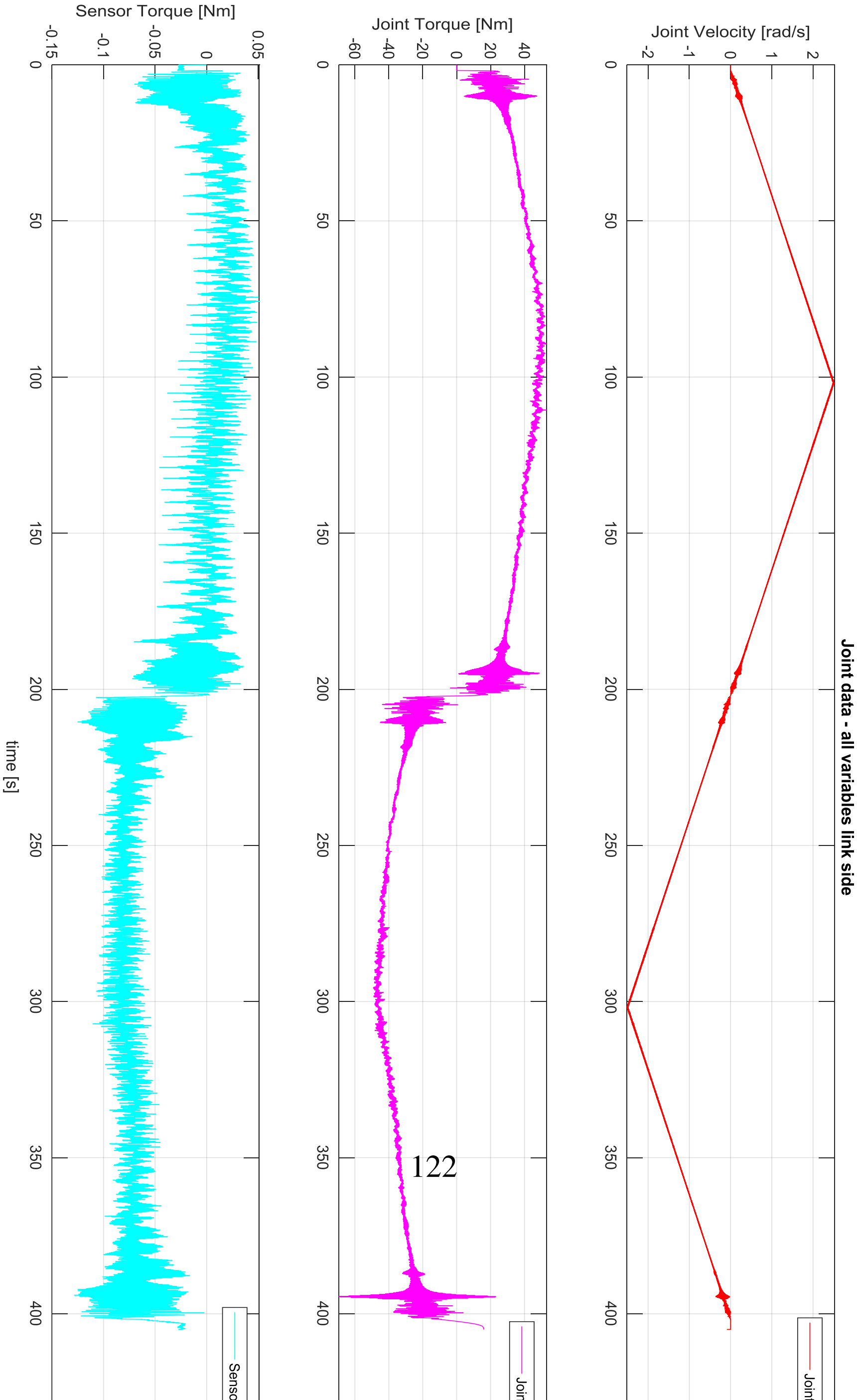
- **Elastic Structure Preserving Assembly Learning:** We have investigated a cross-search method to utilize the robot EE elasticity, however, the learning based approach to utilize the EE elasticity or the robot activate elasticity in order to learn human operation will be more interesting.

- **Tiny FPC connectors PiH operation:** The tiny FPC connectors assembly in the phone production line is an open challenge. The manipulation accuracy of tiny parts is a great challenge for the human hands, moreover, the operation force should be controlled properly to avoid damage to the connectors. Integrating haptic technology and precise manipulation technology into robotics will be a potential research direction.

Appendix A

Reference Information

A.1 Single joint test data



A joint made by Sensodrive[1] Figure A.1 is tested under quasi-static (very low accelerations) situation, which means that the joint torque represents purely the frictional and viscous losses, the maximum friction reach 50 Nm. While the sensor torque represents the friction at flange side, the maximum friction is around 0.12 Nm which close to zero and change directions according to the directions of velocity.

[1] https://www.sensodrive.de/

Appendix B

Chapter of Abbreviations

LWR Light-Weight Robots

DLR German Aerospace Center

EE End-Effector

CNN Convolutional Neural Network

DoF Degree of Freedom

DRL Deep Reinforcement Learning

GAN Generative Adversarial Network

MDP Markov Decision Process

RL Reinforcement Learning

SAC Soft Actor-Critic

VAE Variational Autoencoder

GUI Graphical User Interface

API Application Programming Interface

ROS Robot Operating System

RRRL Region-limited Residual Reinforcement Learning

LfD Learning from Demonstration

LbE Learning by Exploration

DOPE Deep Object Pose Estimator

DQN Deep Q Network

RAM Random-access memory

PCB Printed Circuit Boards

PLC Programmable Logic Controller

SARA Safe Autonomous Robotic Assistant

RCC Remote-Center-Compliance

PID Proportional-Integral-Derivative

PiH Peg-in-Hole

LbD Learning by Demonstration

GUAPO Guided Uncertainty Aware Policy Optimization

MSBE Mean-squared Bellman Error

CycleGAN Cycle-consistent Generative Adversarial Networks

VAE Variational Autoencoder

RRRL Region-limited Residual RL

RAM Random Access Memory

SSD Solid State Drives

POMDP partially observable Markov decision process

SOTA State of the Art

ROA Region of Attraction

AI Artificial Intelligence

Appendix C

Acknowledgements

I want to express my gratitude to my supervisor Prof. Dr. Jianwei Zhang who provided me an opportunity to start a doctoral program. He is a full member of the German Academy of Science and Engineering and the Academy of Sciences in Hamburg, he showed me an endless pursuit of improvement. During my four years of study, he guided me to think and understand from a wider perspective, not only in terms of science and theory, but also in terms of learning and understanding German and European culture.

I want to thank Dr. Zhaopeng Chen for his constant guidance in robot control and for founding my study. He was a research scientist and deputy head of Modex Lab at DLR/RMC from 2010 to 2020. He is the founder of two companies, one of which has become the first robotics unicorn in Germany. I saw and learned from his bravery and fearlessness, As the German philosopher Friedrich Nietzsche said: to treat life, you might as well be bold, because we have to lose it all the time.

I want to thank Antonio Gonzales Marin, Sebastian Riedel, Dr. Lok Man Ho and Dr. Bernd Henze, they gave me very effective and warm guidance and help with control, vision, and robotics aspects during my doctoral program. And thanks to Dr. Georg Stillfried, Dimitri Henkel, Oliver Porges, Samantha Stoneman, Dr. Jun Deng for the support in my work. Moreover, I would like to thank Hongxu Liu, Qian Feng, Chengjie Yuan, Yansong Wu, Chunyang Chang, Kaixin Bai, Lei Zhang, Ningxin Lu, Martin Craiu-Müller for the help in my paper writing and experiments. As the COVID-19 pandemic started at end of 2019, I don't have many chances to visit TAMS group in Hamburg during my study, but I still received a lot of help from TAMS colleagues. Thanks to Dr. Hongzhuo Liang and Dr. Shuang Li, they can help me in any issue. Thanks to Dr. Cong Lin, we have discussed many topics together and have collaborated on several papers. Thanks to Tatjana Lu Tetsis and Yihong Hu for their selfless assistance. I am also grateful to Dr.Ge Gao, Dr. Norman Hendrich, Dr. Chao Zeng, Niklas Fiedler, Michael Görner, Yannick Jonetzko, Jianzhi Lyu, Yuyang Tu, Philipp Ruppel, German Junca, Dr. Di Zhang, as we have conducted numerous online research and paper sharing workshops together, inspiration always comes from the collision of ideas. Stay negative, keep positive, this is for all of us who are doing research in corona-pandemic.

In the end, I want to thank my parents, Xinzhong Shi and Sucai Ding, their selfless love has supported me in all times and places from birth to now. Also, thanks to my younger brother Yuntian Shi for his care and support. I want to thank my wife, Chunhui Gao, for the constant love, I sincerely believe since the first sight that we have been a couple in a past life, thus we can understand and support each other in all matters. Moreover, I would like to thank my parents-in-law, Zhenduo Gao and Li Xiang, for their support and encouragement.

Thanks to everyone who helped me. It is a wonderful world.

Bibliography

[1] Fares J Abu-Dakka, Bojan Nemec, Jimmy A Jørgensen, Thiusius R Savarimuthu, Norbert Krüger, and Aleš Ude. Adaptation of manipulation skills in physical contact with the environment to reference force profiles. *Autonomous Robots*, 39(2):199–217, 2015.

[2] Abdullah Al-Zabt and Tarek A Tutunji. Robotic arm representation using image-based feedback for deep reinforcement learning. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 168–173. IEEE, 2019.

[3] Alin Albu-Schäffer, Sami Haddadin, Ch Ott, Andreas Stemmer, Thomas Wimböck, and Gerhard Hirzinger. The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: an international journal*, 2007.

[4] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A passivity based cartesian impedance controller for flexible joint robots-part II: Full state feedback, impedance design and experiments. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2666–2672. IEEE, 2004.

[5] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The international journal of robotics research*, 26(1):23–39, 2007.

[6] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[7] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[8] Aleksandra Anna Apolinarska, Matteo Pacher, Hui Li, Nicholas Cote, Rafael Pastrana, Fabio Gramazio, and Matthias Kohler. Robotic assembly of timber joints using reinforcement learning. *Automation in Construction*, 125:103569, 2021.

[9] Paul Bakker, Yasuo Kuniyoshi, et al. Robot see, robot do: An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, volume 5. Citeseer, 1996.

[10] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 449–458. PMLR, 2017.

[11] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[12] Robert Bogue. The role of artificial intelligence in robotics. *Industrial Robot: An International Journal*, 2014.

[13] Thomas Breuer, Mireille Ndoundou-Hockemba, and Vicki Fishlock. First observation of tool use in wild gorillas. *PLoS Biology*, 3(11), 2005.

[14] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[15] F Caillot and M Kerlidou. Air stream compliance. In *5th Int. Conf. on Assembly Automation*, pages 225–233, 1984.

[16] Michael E Came, Tomás Lozano-Pérez, and Warren P Seering. Assembly strategies for chamferless parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 472–477, 1989.

[17] Ambrose Chan, Elizabeth A Croft, and James J Little. Constrained manipulator visual servoing (cmvs): Rapid robot programming in cluttered workspaces. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2825–2830. IEEE, 2011.

[18] Chunyang Chang, Kevin Haninger, Yunlei Shi, Chengjie Yuan, Zhaopeng Chen, and Jianwei Zhang. Impedance adaptation by reinforcement learning with contact dynamic movement primitives. *arXiv preprint arXiv:2203.07191*, 2022.

[19] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[20] Yevgen Chebotar, Oliver Kroemer, and Jan Peters. Learning robot tactile sensing for object manipulation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3368–3375. IEEE, 2014.

[21] Zhaopeng CHEN, Xuebin SU, Yuechao ZAHO, Qian WANG, and Georg STILLFRIED. "MECHANICAL ARM JOINT", European Patent, no.WO2021104948A1, 06 2021.

[22] Stefano Chiaverini and Lorenzo Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.

[23] Stefano Chiaverini, Bruno Siciliano, and Luigi Villani. Force/position regulation of compliant robot manipulators. *IEEE Transactions on Automatic Control*, 39(3):647–652, 1994.

[24] Hyung Suck Cho, Hans-Jürgen Warnecke, and Dae-Gab Gweon. Robotic assembly: a synthesizing overview. *Robotica*, 5(2):153–165, 1987.

[25] Lin Cong, Michael Görner, Philipp Ruppel, Hongzhuo Liang, Norman Hendrich, and Jianwei Zhang. Self-adapting recurrent models for object pushing from learning in simulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5304–5310. IEEE, 2020.

[26] Lin Cong, Hongzhuo Liang, Philipp Ruppel, Yunlei Shi, Michael Görner, Norman Hendrich, and Jianwei Zhang. Reinforcement learning with vision-proprioception model for robot planar pushing. *Frontiers in Neurorobotics*, 16, 2022.

[27] Lin Cong, Yunlei Shi, and Jianwei Zhang. Self-supervised attention learning for robot control. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1153–1158. IEEE, 2021.

[28] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[29] Todor Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration: adapting dynamic movement primitives for contact-rich insertion tasks. *arXiv preprint arXiv:2008.07682*, 2020.

[30] Joris De Schutter and Hendrik Van Brussel. Compliant robot motion ii. a control approach based on external control loops. *The International Journal of Robotics Research*, 7(4):18–33, 1988.

[31] SH Drake. High speed robot assembly of precision parts using compliance instead of sensory feedback. In *Proc. 7th Int. Symp. Industrial Robots*, 1977.

[32] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In *2017*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3530–3537. IEEE, 2017.

[33] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4397–4404. IEEE, 2015.

[34] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

[35] Qian Feng, Zhaopeng Chen, Jun Deng, Chunhui Gao, Jianwei Zhang, and Alois Knoll. Center-of-mass-based robust grasp planning for unknown objects using tactile-visual sensors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 610–617. IEEE, 2020.

[36] Hiroshi Fujimoto. Visual servoing of 6 dof manipulator by multirate control with depth identification. In *42nd IEEE International Conference on Decision and Control (CDC)*, volume 5, pages 5408–5413. IEEE, 2003.

[37] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning (ICML)*, pages 1587–1596. PMLR, 2018.

[38] Claudio Gaz, Marco Cognetti, Alexander Oliva, Paolo Robuffo Giordano, and Alessandro De Luca. Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154, 2019.

[39] AM Girel. Using a rotating magnetic-field for grippers of industrial robots. *Russian Engineering Journal*, 57(6):43–45, 1977.

[40] Wandelbots GmbH. no-code-robotics everyone can work with robots, 2022.

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[42] Susan Gottschlich, Carlos Ramos, and Damian Lyons. Assembly and task planning: A taxonomy. *IEEE Robotics & Automation Magazine*, 1(3):4–12, 1994.

[43] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

[44] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pages 1861–1870. PMLR, 2018.

[45] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: Detection, isolation, and identification. *Submitted to IEEE Transactions on Robotics*, 2015.

[46] Murtaza Hazara and Ville Kyrki. Reinforcement learning for improving imitated in-contact skills. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 194–201. IEEE, 2016.

[47] Oliver Heimann and Jan Guhl. Industrial robot programming methods: a scoping review. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 696–703. IEEE, 2020.

[48] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Florian Golemo, Melissa Mozifian, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Perspectives on sim2real transfer for robotics: A summary of the r: Ss 2020 workshop. *arXiv preprint arXiv:2012.03806*, 2020.

[49] Brian D Hoffman, Steven H Pollack, and Barry Weissman. Vibratory insertion process: A new approach to non-standard component insertion. In *The Electronics Assembly Handbook*, pages 115–119. Springer, 1988.

[50] Neville Hogan. Impedance control: An approach to manipulation: Part iâĂŤtheory. 1985.

[51] Zhimin Hou, Zhihu Li, Chenwei Hsu, Kuangen Zhang, and Jing Xu. Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly. *IEEE Transactions on Automation Science and Engineering*, 2020.

[52] Feng-Yi Hsu and Li-Chen Fu. Intelligent robot deburring using adaptive fuzzy hybrid position/force control. *IEEE Transactions on Robotics and Automation*, 16(4):325–335, 2000.

[53] Yanjiang Huang, Xianmin Zhang, Xunman Chen, and Jun Ota. Vision-guided peg-in-hole assembly by baxter robot. *Advances in Mechanical Engineering*, 9(12):1687814017748078, 2017.

[54] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[55] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[56] Tadanobu Inoue, Giovanni De Magistris, Asim Munawar, Tsuyoshi Yokoya, and Ryuki Tachibana. Deep reinforcement learning for high precision assembly tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 819–825. IEEE, 2017.

[57] Keita Isaka, Kazuki Tsumura, Tomoki Watanabe, Wataru Toyama, Makoto Sugesawa, Yasuyuki Yamada, Hiroshi Yoshida, and Taro Nakamura. Development of underwater drilling robot based on earthworm locomotion. *Ieee Access*, 7:103127–103141, 2019.

[58] Maged Iskandar, Oliver Eiberger, Alin Albu-Schäffer, Alessandro De Luca, and Alexander Dietrich. Collision detection, identification, and localization on the dlr sara robot with sensing redundancy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3111–3117. IEEE, 2021.

[59] Ibrahim F Jasim, Peter W Plapper, and Holger Voos. Position identification in force-guided robotic peg-in-hole assembly tasks. *Procedia Cirp*, 23:217–222, 2014.

[60] Jingang Jiang, Zhiyuan Huang, Zhuming Bi, Xuefeng Ma, and Guang Yu. State-of-the-art control strategies for robotic pih assembly. *Robotics and Computer-Integrated Manufacturing*, 65:101894, 2020.

[61] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. *arXiv preprint arXiv:1812.03201*, 2018.

[62] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[63] Mrinal Kalakrishnan, Ludovic Righetti, Peter Pastor, and Stefan Schaal. Learning force control policies for compliant manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4639–4644. IEEE, 2011.

[64] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[65] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau. Tactile sensing in dexterous robot hands. *Robotics and Autonomous Systems*, 74:195–220, 2015.

[66] Manuel Kaspar, Juan D Muñoz Osorio, and Jürgen Bock. Sim2real transfer for reinforcement learning without dynamics randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4383–4388. IEEE, 2020.

[67] Robin Jeanne Kirschner, Nico Mansfeld, Saeed Abdolshah, and Sami Haddadin. Experimental analysis of impact forces in constrained collisions according to iso/ts 15066. In *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pages 1–5. IEEE, 2021.

[68] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[69] Danica Kragic, Henrik I Christensen, et al. Survey on visual servoing for manipulation. *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, 15:2002, 2002.

[70] Aljaž Kramberger, Bojan Nemec, Matjaž Gams, Aleš Ude, et al. Learning of assembly constraints by demonstration and active exploration. *Industrial Robot: An International Journal*, 2016.

[71] Norbert Krüger, Aleš Ude, Henrik Gordon Petersen, Bojan Nemec, Lars-Peter Ellekilde, Thiusius Rajeeth Savarimuthu, Jimmy Alison Rytz, Kerstin Fischer, Anders Glent Buch, Dirk Kraft, et al. Technologies for the fast set-up of automated assembly processes. *KI-Künstliche Intelligenz*, 28(4):305–313, 2014.

[72] Cheng-Yu Kuo, Andreas Schaarschmidt, Yunduan Cui, Tamim Asfour, and Takamitsu Matsubara. Uncertainty-aware contact-safe model-based reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):3918–3925, 2021.

[73] Safoura Rezapour Lakani, Antonio J Rodríguez-Sánchez, and Justus Piater. Exercising affordances of objects: A part-based approach. *IEEE Robotics and Automation Letters*, 3(4):3465–3472, 2018.

[74] Luc Le Tien, Alin Albu-Schäffer, Alessandro De Luca, and Gerd Hirzinger. Friction observer and compensation for control of robots with joint torque measurement. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3789–3795. IEEE, 2008.

[75] Joon Hyub Lee, Yongkwan Kim, Sang-Gyun An, and Seok-Hyung Bae. Robot telekinesis: application of a unimanual and bimanual object manipulation technique to robot control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9866–9872. IEEE, 2020.

[76] Michelle A Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7505–7512. IEEE, 2020.

[77] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and

touch: Self-supervised learning of multimodal representations for contact-rich tasks. *arXiv preprint arXiv:1810.10191*, 2018.

[78] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.

[79] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[80] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. *Advances in neural information processing systems*, 27, 2014.

[81] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[82] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[83] Rui Li and Hong Qiao. A survey of methods and strategies for high-precision robotic grasping and assembly tasks–some new trends. *IEEE/ASME Transactions on Mechatronics*, 24(6):2718–2732, 2019.

[84] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[85] Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani. Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration. *IEEE Transactions on Robotics*, 23(1):73–86, 2007.

[86] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Efficient dynamic-programming updates in partially observable markov decision processes. 1995.

[87] Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp-Langley. Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics*, 10(1):22, 2021.

[88] William S Lovejoy. A survey of algorithmic methods for partially observed markov decision processes. *Annals of Operations Research*, 28(1):47–65, 1991.

[89] Jianlan Luo, Eugen Solowjow, Chengtao Wen, Juan Aparicio Ojea, and Alice M Agogino. Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2062–2069. IEEE, 2018.

[90] Jianlan Luo, Eugen Solowjow, Chengtao Wen, Juan Aparicio Ojea, Alice M Agogino, Aviv Tamar, and Pieter Abbeel. Reinforcement learning on variable impedance controller for high-precision robotic assembly. *arXiv preprint arXiv:1903.01066*, 2019.

[91] Philippe Martinet, Jean Gallice, and Khadraoui Djamel. Vision based control law using 3d visual features. In *World Automation Congress, WAC'96, Robotics and Manufacturing Systems*, volume 3, pages 497–502, 1996.

[92] Jeremy Marvel and Roger Bostelman. Towards mobile manipulator safety standards. In *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 31–36. IEEE, 2013.

[93] Jeremy A Marvel, Roger Bostelman, and Joe Falco. Multi-robot assembly strategies and metrics. *ACM Computing Surveys (CSUR)*, 51(1):1–32, 2018.

[94] Eloise Matheson, Riccardo Minto, Emanuele GG Zampieri, Maurizio Faccio, and Giulio Rosati. Human–robot collaboration in manufacturing applications: A review. *Robotics*, 8(4):100, 2019.

[95] Vincent Mayer, Qian Feng, Jun Deng, Yunlei Shi, Zhaopeng Chen, and Alois Knoll. Ffhnet: Generating multi-fingered robotic grasps for unknown objects in real-time. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 762–769. IEEE, 2022.

[96] N Harris McClamroch and Danwei Wang. Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419–426, 1988.

[97] James K Mills and Andrew A Goldenberg. Force and position control of manipulators during constrained motion tasks. *IEEE Transactions on Robotics and Automation*, 5(1):30–46, 1989.

[98] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning (ICML)*, pages 1928–1937. PMLR, 2016.

[99] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[100] Nicky Mol, Jan Smisek, Robert Babuška, and André Schiele. Nested compliant admittance control for robotic mechanical assembly of misaligned and tightly toleranced parts. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 002717–002722. IEEE, 2016.

[101] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[102] Fusaomi Nagata, Tetsuo Hase, Zenku Haga, Masaaki Omoto, and Keigo Watanabe. Cad/cam-based position/force controller for a mold polishing robot. *Mechatronics*, 17(4-5):207–216, 2007.

[103] Wyatt S Newman, Yonghong Zhao, and Y-H Pao. Interpretation of force and moment signals for compliant peg-in-hole assembly. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 1, pages 571–576. IEEE, 2001.

[104] Andrew Y Ng. *Shaping and policy search in reinforcement learning*. PhD thesis, University of California, Berkeley Berkeley, 2003.

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

[106] OpenAI. [Blog] OpenAI Spinning Up. `https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html`, 2020.

[107] Christian Ott, Alin Albu-Schäffer, Andreas Kugi, S Stamigioli, and Gerd Hirzinger. A passivity based cartesian impedance controller for flexible joint robots-part I: Torque feedback and gravity compensation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2659–2665. IEEE, 2004.

[108] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. Omnitact: A multi-directional high-resolution touch sensor. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 618–624. IEEE, 2020.

[109] Hyeonjun Park, Ji-Hun Bae, Jae-Han Park, Moon-Hong Baeg, and Jaeheung Park. Intuitive peg-in-hole assembly strategy with a compliant manipulator. In *IEEE ISR 2013*, pages 1–5. IEEE, 2013.

[110] Hyeonjun Park, Peter Ki Kim, Ji-Hun Bae, Jae-Han Park, Moon-Hong Baeg, and Jaeheung Park. Dual arm peg-in-hole assembly with a programmed compliant system. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 431–433. IEEE, 2014.

[111] Hyeonjun Park, Jaeheung Park, Dong-Hyuk Lee, Jae-Han Park, and Ji-Hun Bae. Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture. *IEEE Robotics and Automation Letters*, 5(3):4447–4454, 2020.

[112] Hyeonjun Park, Jaeheung Park, Dong-Hyuk Lee, Jae-Han Park, Moon-Hong Baeg, and Ji-Hun Bae. Compliance-based robotic peg-in-hole assembly strategy without force feedback. *IEEE Transactions on Industrial Electronics*, 64(8):6299–6309, 2017.

[113] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[114] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[115] Cristian Alejandro Vergara Perico, Joris De Schutter, and Erwin Aertbeliën. Combining imitation learning with constraint-based task specification and control. *IEEE Robotics and Automation Letters*, 4(2):1892–1899, 2019.

[116] Michael A Peshkin. Programmed compliance for error corrective assembly. *IEEE Transactions on Robotics and Automation*, 6(4):473–482, 1990.

[117] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, 29, 2016.

[118] Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[119] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

[120] Lyric Robot. Desktop computer host automatic assembly line.

[121] Dmytro Romanov, Olga Korostynska, Odd Ivar Lekang, and Alex Mason. Towards human-robot collaboration in meat processing: Challenges and possibilities. *Journal of Food Engineering*, page 111117, 2022.

[122] Mohammad Safeea, Richard Bearee, and Pedro Neto. End-effector precise hand-guiding for collaborative robots. In *Iberian Robotics conference*, pages 595–605. Springer, 2017.

[123] Amr Salem and Yiannis Karayiannidis. Robotic assembly of rounded parts with and without threads. *IEEE Robotics and Automation Letters*, 5(2):2467–2474, 2020.

[124] J Kenneth Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *1980 19th IEEE conference on decision and control including the symposium on adaptive processes*, pages 95–100. IEEE, 1980.

[125] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[126] Joseph M Schimmels and Michael A Peshkin. Admittance matrix design for force-guided assembly. *IEEE Transactions on Robotics and Automation*, 8(2):213–227, 1992.

[127] Joseph M Schimmels and Michael A Peshkin. Force-assembly with friction. *IEEE Transactions on Robotics and Automation*, 10(4):465–479, 1994.

[128] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. *arXiv preprint arXiv:1906.05841*, 2019.

[129] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555. IEEE, 2020.

[130] Gerrit Schoettler, Ashvin Nair, Juan Aparicio Ojea, Sergey Levine, and Eugen Solowjow. Meta-reinforcement learning for robotic industrial insertion tasks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9728–9735. IEEE, 2020.

[131] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[132] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[133] Yunlei Shi, Zhaopeng Chen, Lin Cong, Yansong Wu, Martin Craiu-Müller, Chengjie Yuan, Chunyang Chang, Lei Zhang, and Jianwei Zhang. Maximizing the use of environmental constraints: A pushing-based hybrid position/force assembly skill for contact-rich tasks. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 496–501. IEEE, 2021.

[134] Yunlei Shi, Zhaopeng Chen, Hongxu Liu, Sebastian Riedel, Chunhui Gao, Qian Feng, Jun Deng, and Jianwei Zhang. Proactive action visual residual reinforcement learning for contact-rich tasks using a torque-controlled robot. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 765–771. IEEE, 2021.

[135] Yunlei Shi, Zhaopeng Chen, Yansong Wu, Dimitri Henkel, Sebastian Riedel, Hongxu Liu, Qian Feng, and Jianwei Zhang. Combining learning from demonstration with learning by exploration to facilitate contact-rich tasks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1062–1069. IEEE, 2021.

[136] Yunlei Shi, Chengjie Yuan, Athanasios Tsitos, Lin Cong, Hamid Hadjar, Zhaopeng Chen, and Jianwei Zhang. A sim-to-real learning based framework for contact-rich assembly by utilizing cyclegan and force control. *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[137] Yoshiaki Shirai and Hirochika Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern recognition*, 5(2):99–108, 1973.

[138] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. springer, 2016.

[139] Bruno Siciliano, Luigi Villani, and N Federico. From indirect to direct force control: A roadmap for enhanced industrial robots. *Invited Paper, Robótica*, 2000.

[140] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[141] Christoffer Sloth, Aljaž Kramberger, and Iñigo Iturrate. Towards easy setup of robotic assembly tasks. *Advanced Robotics*, 34(7-8):499–513, 2020.

[142] Rui Song, Fengming Li, Tianyu Fu, and Jie Zhao. A robotic automatic assembly system based on vision. *Applied Sciences*, 10(3):1157, 2020.

[143] Yixu Song, Wei Liang, and Yang Yang. A method for grinding removal control of a robot belt grinding system. *Journal of Intelligent Manufacturing*, 23(5):1903–1913, 2012.

[144] Oren Spector and Dotan Di Castro. Insertionnet-a scalable solution for insertion. *IEEE Robotics and Automation Letters*, 6(3):5509–5516, 2021.

[145] Mark W Spong. Modeling and control of elastic joint robots. 1987.

[146] Andreas Stemmer, Günter Schreiber, Klaus Arbter, and A Albu-Schaffer. Robust assembly of complex shaped planar parts using vision and force. In *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 493–500. IEEE, 2006.

[147] Jochen Stüber, Claudio Zito, and Rustam Stolkin. Let's push things forward: A survey on robot pushing. *Frontiers in Robotics and AI*, 7:8, 2020.

[148] Shijian Su, Xianping Zeng, Shuang Song, Mingqiang Lin, Houde Dai, Wanan Yang, and Chao Hu. Positioning accuracy improvement of automated guided vehicles based on a novel magnetic tracking approach. *IEEE Intelligent Transportation Systems Magazine*, 2018.

[149] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[150] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[151] Céline Teulière and Eric Marchand. Direct 3d servoing using dense depth maps. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1741–1746. IEEE, 2012.

[152] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[153] Roger Y Tsai, Reimar K Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Transactions on robotics and automation*, 5(3):345–358, 1989.

[154] Timothy Douglas Tuttle. *Understanding and modeling the behavior of a harmonic drive gear transmission*. PhD thesis, Massachusetts Institute of Technology, 1992.

[155] Eugene Valassakis, Zihan Ding, and Edward Johns. Crossing the gap: A deep dive into zero-shot sim-to-real transfer for dynamics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5372–5379. IEEE, 2020.

[156] Luigi Villani and Joris De Schutter. Force control. In *Springer handbook of robotics*, pages 195–220. Springer, 2016.

[157] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[158] PC Watson. The remote center compliance system and its application to high speed robot assemblies. *SME paper No. AD77–718*, 1977.

[159] Qianxiao Wei, Canjun Yang, Wu Fan, and Yibing Zhao. Design of demonstration-driven assembling manipulator. *Applied Sciences*, 8(5):797, 2018.

[160] Lilian Weng. Domain randomization for sim2real transfer. *lilianweng.github.io*, 2019.

[161] Chelsea C White. A survey of solution techniques for the partially observed markov decision process. *Annals of Operations Research*, 32(1):215–230, 1991.

[162] Chengjie Yuan, Yunlei Shi, Qian Feng, Chunyang Chang, Michael Liu, Zhaopeng Chen, Alois Christian Knoll, and Jianwei Zhang. Sim-to-real transfer of robotic assembly with visual inputs using cyclegan and force control. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1426–1432. IEEE, 2022.

[163] Stefan Zeiß. Manipulation skill for robotic assembly. Master's thesis, Technical University of Darmstadt, 6 2014.

[164] Ganwen Zeng and Ahmad Hemami. An overview of robot force control. *Robotica*, 15(5):473–482, 1997.

[165] Lei Zhang, Kaixin Bai, Zhaopeng Chen, Yunlei Shi, and Jianwei Zhang. Towards precise model-free robotic grasping with sim-to-real transfer learning. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–8. IEEE, 2022.

[166] Yanglong Zheng, Xianmin Zhang, Yanlin Chen, and Yanjiang Huang. Peg-in-hole assembly based on hybrid vision/force guidance and dual-arm coordination. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 418–423. IEEE, 2017.

[167] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2223–2232, 2017.

[168] Wu-Le Zhu and Anthony Beaucamp. Compliant grinding and polishing: A review. *International Journal of Machine Tools and Manufacture*, 158:103634, 2020.

[169] Zuyuan Zhu and Huosheng Hu. Robot learning from demonstration in robotic assembly: A survey. *Robotics*, 7(2):17, 2018.

List of Figures

List of Tables

Erklärung der Urheberschaft

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, 11.04.2023
Ort, Datum

Yunlei Shi
Unterschrift