Establishing a Robotic Environment for Reinforcement Learning Study

Bálint Károly Farkas
Alba Regia Faculty, Óbuda University, H-8000
Székesfehérvár, Hungary
farkas.balint@amk.uni-obuda.hu

Károly Széll
Antal Bejczy Center for Intelligent Robotics, Óbuda University
Alba Regia Faculty, Óbuda University, H-8000
Székesfehérvár, Hungary
szell.karoly@amk.uni-obuda.hu

Péter Galambos *Senior Member, IEEE*
Antal Bejczy Center for Intelligent Robotics, Óbuda University
Budapest, Hungary
peter.galambos@irob.uni-obuda.hu

***Abstract*—This paper aims to outline both hardware and software design to create a reliable and efficient framework for future robotics and reinforcement learning research. The primary goal of this paper is to design and establish a functional environment for a reinforcement learning pick-and-place project. This includes the integration of key hardware components, as well as the development of a robust software architecture. The software setup will be built around ROS (Robot Operating System) to provide seamless communication and control of the system. Additionally, plans for software integration will include the use of NVIDIA Isaac Manipulator for AI-driven motion planning and object detection, leveraging tools like cuMotion and FoundationPose. This environment will be a solid foundation to future researches.**

***Index Terms*—Reinforcement Learning; AI-driven motion planning; Autonomous Manipulation; Robotic Research Platform**

I. Introduction

The increasing demand for automation in industries such as manufacturing, assembly and logistics has brought robotic systems to the forefront. Among these, pick-and-place tasks are critical as they require a robot to identify, grasp and move objects from one location to another with precision. This article discusses the setup of a pick and place environment using the UR5e robotic arm, Robotiq gripper and Intel RealSense D435 camera integrated with a NVIDIA GPU-equipped PC running NVIDIA Isaac Manipulator for efficient motion planning and object recognition.

The primary advantage of this project lies in its focus on establishing a modular and adaptable system that can serve as a foundation for advanced research in Reinforcement Learning (RL) for robotic manipulation. By setting up this environment, future research can explore various RL algorithms, allowing the robot to autonomously learn and optimize its behavior in complex tasks, such as picking objects of different shapes, sizes and weights, or dynamically adapting to changes in the environment.

The environment opens up numerous possibilities for advancing RL research in robotics. Researchers can experiment with state-of-the-art RL techniques, like Deep Q-Networks (DQNs), Proximal Policy Optimization (PPO), and Deep Deterministic Policy Gradient (DDPG), to train the robot in simulated or real-world settings [1] [2]. Using the NVIDIA Isaac platform's GPU acceleration, this project can facilitate faster training times and enable the robot to learn intricate tasks through trial and error [3]. It paves the way for future research to enhance robot autonomy.

II. Hardware Setup

Setting up the hardware for a robotic pick-and-place task requires careful selection and integration of various components to ensure efficient and seamless operation. The key elements include the UR5e robotic arm, Robotiq 2F-85 gripper, Intel RealSense D435 camera and a PC. Each component plays a crucial role in the system, providing motion control, object handling, perception and data processing. Ensuring proper connectivity and calibration between these devices is essential to achieve precise and reliable robotic performance.

UR5e Robotic Arm: The UR5e from Universal Robots is a highly versatile 6-axis robotic arm commonly used in research because of its flexibility and the ability to easily integrate with other systems as Toner et al. did [4]. Its compatibility with Robot Operating System (ROS) enables smooth communication with the rest of the system, including the vision system and gripper.

Robotiq Gripper 2F-85: The Robotiq adaptive gripper is a highly adaptable gripper designed to work seamlessly with the UR5e. It supports multiple grasping strategies, including parallel, encompassing and fingertip grasps, allowing it to handle various object shapes and sizes, which is a good solution for us, since it can grip a various type of parts. Its ease of use and quick integration with ROS make it a perfect fit for our setup.

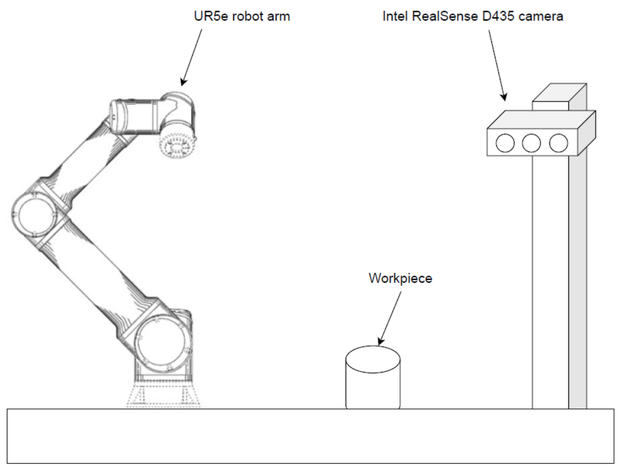

Fig. 1. Spatial arrangement of the components

Intel RealSense D435 Camera: The Intel RealSense D435 camera provides high-quality RGB-D data (color and depth), which is essential for object detection and scene understanding. The camera is capable of real-time depth sensing, making it a valuable tool for detecting objects in 3D space and providing accurate depth information to the robot.

PC: Our PC is equipped with a NVIDIA GeForce RTX 4070 GPU, which is required for the Reinforcement Learning agent and for the visual parts of the research.

After selecting and configuring the core hardware components, it is essential to ensure smooth integration for seamless communication and task execution. Figure 1 shows the spatial arrangement of the system's primary components. The UR5e robotic arm is centrally positioned, with the Robotiq gripper attached to it, communicating with the robotic arms own integrated cable. The Intel RealSense camera is placed at an optimal height and angle to have the best view to the space before the robotic arm.

III. Software Environment

The successful operation of a robotic pick-and-place system depends not only on well integrated hardware, but also on a robust and flexible software environment. This system relies on ROS, an open-source framework that enables seamless communication between hardware components.

In addition, the NVIDIA Isaac Manipulator platform enhances the system by providing powerful AI-driven tools for motion planning, perception and trajectory optimization, leveraging the processing power of an NVIDIA GPU.

The software environment ensures real-time data exchange and synchronization between the various components, allowing the system to perform complex tasks efficiently and accurately. This section outlines the key software tools and libraries involved in integrating the hardware, managing communication, and optimizing task execution.

A. ROS

ROS is a widely-used, open-source framework that provides flexible architecture for developing and integrating robotics software. In this pick-and-place system, ROS serves as the backbone for communication between the different hardware components, such as the UR5e robotic arm, the Robotiq gripper and the Intel RealSense D435 camera.

ROS allows for modular control of the robot through its nodes and topics, where each device of the process is treated as a separate node that communicates through published messages on specific topics.

Furthermore, ROS offers numerous libraries and tools for robotic control, including MoveIt for motion planning and manipulation.

With its open-source nature and extensive community support, ROS ensures flexibility and scalability in robotic applications, enabling the seamless integration of new sensors, actuators, or algorithms as needed.

B. NVIDIA Isaac Manipulator

The NVIDIA Isaac Manipulator is a powerful AI-driven platform built to accelerate the development of robotic systems, specifically focusing on tasks such as manipulation, motion planning, and perception. It integrates seamlessly with ROS and provides a suite of tools and pre-trained models that leverage GPU acceleration to optimize robot performance.

The key components of the Isaac Manipulator include **cuMotion**, a GPU-accelerated motion planner that ensures smooth and efficient trajectory planning, **FoundationPose**, a model for the estimation of the pose of objects in 6D, crucial for pick-and-place tasks.

This platform significantly reduces the development time for robotic applications by providing ready-to-use workflows and AI models, making it easier to implement complex tasks while optimizing performance through parallel processing on GPU.

C. Object Detection and Perception

In a pick-and-place robotic system, object detection and perception are crucial to allow the robot to interact with its environment effectively. Advanced object detection algorithms like SyntheticaDETR provide real-time object detection capabilities, enabling precise and adaptive manipulation. However, the robustness of perception systems is often challenged by external factors and software-related issues that may cause errors. Implementing fault-tolerant solutions, similar to those used in microcontroller-based systems, can help mitigate these risks [5]. Our camera is a RGB-D camera, which means it gives us an RGB picture plus depth information. This depth information is curcial for accurate object detection.

To interpret the camera data, the system utilizes advanced object detection algorithms like **SyntheticaDETR** for NVIDIA Isaac, which is designed for real-time object detection in complex environments.

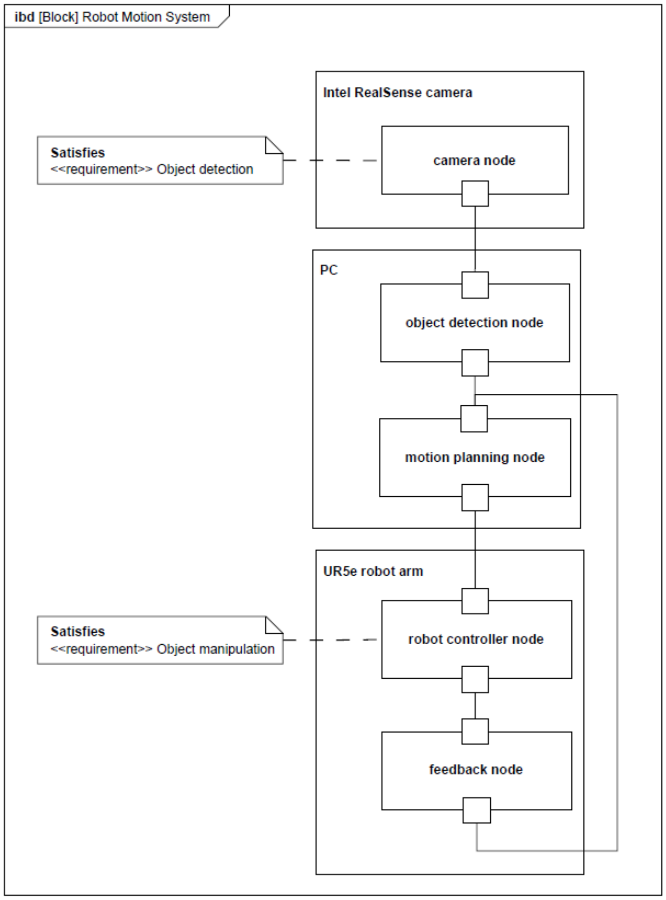

Fig. 2. Communication pathways

This perception tool allows the robot to dynamically adjust its actions based on real-time data ensuring precise grasping and manipulation of objects.

The overall software architecture of the system, including its communication pathways and data flows, is depicted in figure 2, providing a clear visualization of how the components are interconnected and how data are processed throughout the workflow.

D. Implementing Software Environment

The software integration plan involves setting up and configuring the software components needed for communication, perception, and control of the robotic system.

ROS2 Jammy: ROS is installed on the main PC and the Raspberry Pi 5, with necessary ROS packages for the UR5e, Robotiq gripper, and RealSense D435 camera. This setup ensures that each hardware component is treated as a ROS node capable of publishing and subscribing to topics for data exchange. Nodes for motion control, gripper operation, and sensor data processing are launched to create a cohesive system.

NVIDIA Isaac Integration: The NVIDIA Isaac Manipulator software stack is installed on the PC equipped with an NVIDIA GPU. The integration process involves setting up the cuMotion module for motion planning and the FoundationPose model for object detection. These components are integrated into the ROS ecosystem, allowing them to interact with other ROS nodes seamlessly. Isaac's perception tools are configured to process the depth and RGB data from the RealSense camera to identify and localize objects in the workspace.

To set up the software environment, I followed these steps:

1) **Windows Subsystem for Linux (WSL) Installation**
2) **NVIDIA Drivers Installation**
3) **CUDA Installation**
4) **Docker Setup**
5) **NVIDIA Container Toolkit Installation**
6) **Creating an Isaac ROS Workspace**
7) **Cloning Isaac ROS Common**
8) **Cloning RealSense ROS**
9) **Cloning Isaac Manipulator**
10) **Running "run_dev.sh" Script**

IV. Future Work and Research Opportunities

The setup and design of this pick-and-place robotic environment open up a wide array of future research directions, particularly in the field of RL and autonomous robotic manipulation. By establishing a robust hardware and software foundation, this system can be used to explore and develop advanced algorithms that push the boundaries of current robotic capabilities.

A. RL for Adaptive Manipulation

One of the primary opportunities for future work involves implementing and experimenting with various RL algorithms to enable adaptive and intelligent manipulation. With the existing environment, researchers can train the UR5e robotic arm to autonomously learn optimal grasping strategies for objects of varying shapes, sizes and materials. Advanced algorithms such as PPO and DQN can be utilized to enable the robot to adapt its behavior dynamically in response to changes in the environment, such as moving targets or unpredictable obstacles [6]. This research can lead to the development of robots capable of performing complex tasks in unstructured environments, such as logistics, e-commerce warehouses, medical laboratories.

B. Transfer Learning and Sim-to-Real Transfer

Another potential area of research is the use of transfer learning and sim-to-real transfer techniques. By leveraging the NVIDIA Isaac Simulator in conjunction with the real-world environment, the robot can be trained in a simulated environment where it can learn efficiently without the risk of hardware damage [7]. The learned policies can then be transferred to the real robot, minimizing the sim-to-real gap. This approach can significantly reduce the time and resources required for training complex behaviors, providing a more efficient pathway for deploying robots in real-world scenarios [8].

C. Advanced Perception and Object Recognition

Future work can also focus on enhancing the perception capabilities of the system. By integrating more sophisticated object detection algorithms, such as Convolutional Neural Networks (CNN), the robot can be trained to recognize and manipulate a wider variety of objects with greater accuracy. Additionally, incorporating scene understanding and semantic segmentation could allow the robot to distinguish between different types of object and make context-aware decisions during pick-and-place tasks.

D. Collaborative and Multi-Robot Systems

The current setup provides a single-robot environment, but future research could expand this to collaborative or multi-robot systems. By integrating additional robotic arms or mobile robots into the environment, researchers can explore how multiple agents can work together to accomplish complex tasks, such as cooperative assembly, sorting or task allocation [9].

E. Real-Time Adaptation and Fault Tolerance

Another potential area of exploration is the development of algorithms for real-time adaptation and fault tolerance. In real-world scenarios, robots must be capable of handling unexpected situations, such as hardware malfunctions, sensor noise, or changes in object properties. Implementing fault detection and recovery strategies, combined with RL-based adaptive learning, can enable the robot to continue operating effectively even in the face of unforeseen challenges, increasing the robustness and reliability [10].

By exploring these avenues, the current environment can serve as a versatile platform for advancing the state-of-the-art in robotic manipulation, RL. Each research direction offers the potential to significantly enhance the capabilities of robotic systems, driving progress in both academic research and practical industrial applications.

V. Conclusion

This project focused on the comprehensive design and setup of an environment tailored for a robotic pick and place system using the UR5e robotic arm, Robotiq gripper and Intel RealSense D435 camera. The main objective was to establish a solid foundation for the system, ensuring that all hardware and software components were carefully selected, configured, and integrated to facilitate future development and implementation of pick and place tasks.

Throughout the project, critical aspects such as hardware positioning, software integration through ROS, and the incorporation of NVIDIA Isaac Manipulator tools were addressed to ensure seamless communication and data processing within the system. Special attention was paid to the calibration and configuration of each component to establish a synchronized environment capable of supporting complex robotic operations.

Although the system is not yet fully operational, the groundwork laid during this project is crucial for future developments. By thoroughly designing and setting up the environment, we have created a robust platform that will allow further exploration, testing, and optimization of pick-and-place algorithms. This environment serves as a foundational step towards a fully functional robotic system, allowing future work to focus on refining task execution, improving efficiency, and addressing real-world automation challenges.

Acknowledgment

Péter Galambos and Károly Széll thankfully acknowledge the financial support of this work by project no. 2019-1.3.1-KK-2019-00007 implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.3.1-KK funding scheme. Péter Galambos is a Bolyai Fellow of the Hungarian Academy of Sciences. Péter Galambos is supported by the UNKP-22-5 (Bolyai+) New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

Bálint Farkas is supported by the 2024-2.1.1 university research scholarship program of the ministry for culture and innovation from the source of the national research, development and innovation fund.

The authors acknowledge the use of ChatGPT, a language model developed by OpenAI, in an assistive and supervised manner for generating and refining content in this article. ChatGPT was utilized to draft and enhance sections such as the Abstract, Introduction, and Conclusion. The AI-generated content was carefully reviewed and edited by the authors to ensure accuracy and alignment with the research objectives.

References

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8103164

[2] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, "Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations," *IEEE Access*, vol. 8, pp. 178 450–178 481, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9210095

[3] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "RVT-2: Learning Precise Manipulation from Few Demonstrations," Jun. 2024, arXiv:2406.08545 [cs]. [Online]. Available: http://arxiv.org/abs/2406.08545

[4] T. Toner, M. Saez, D. M. Tilbury, and K. Barton, "Opportunities and challenges in applying reinforcement learning to robotic manipulation: An industrial case study," *Manufacturing Letters*, vol. 35, pp. 1019–1030, Aug. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213846323001128

[5] G. Györök and B. Beszédes, "Fault-tolerant Software Solutions in Microcontroller Based Systems," 2017.

[6] Z. Zhou, R. Zuo, B. Ying, J. Zhu, Y. Wang, X. Wang, and X. Liu, "A Sensory Soft Robotic Gripper Capable of Learning-Based Object Recognition and Force-Controlled Grasping," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 844–854, 2024, conference Name: IEEE Transactions on Automation Science and Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/9999176

[7] "Closing the Sim-to-Real Gap: Training Spot Quadruped Locomotion with NVIDIA Isaac Lab," 2024, publication Title: NVIDIA Technical Blog. [Online]. Available: https://developer.nvidia.com/blog/closing-the-sim-to-real-gap-training-spot-quadruped-locomotion-with-nvidia-isaac-lab/

[8] D. Horváth, K. Bocsi, G. Erdős, and Z. Istenes, "Sim2Real Grasp Pose Estimation for Adaptive Robotic Applications," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5233–5239, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896323004676

[9] X. Lan, Y. Qiao, and B. Lee, "Application of Multi Agent Reinforcement Learning to Robotic Pick and Place System," in *2024 10th International Conference on Automation, Robotics and Applications (ICARA)*, 2024, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10552987

[10] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing," Sep. 2024, arXiv:2409.15610 [cs]. [Online]. Available: http://arxiv.org/abs/2409.15610