1862

RIGA TECHNICAL UNIVERSITY


**Jānis Ārents**


# RESEARCH AND DEVELOPMENT OF SMART CONTROL METHODS FOR INDUSTRIAL ROBOTS

Summary of the Doctoral Thesis



RTU Press
Riga 2023

**RIGA TECHNICAL UNIVERSITY**
Faculty of Electrical and Environmental Engineering
Institute of Industrial Electronics and Electrical Engineering

**Jānis Ārents**

Doctoral Student of the Study Programme ”Computerised Control of Electrical Technologies”

# RESEARCH AND DEVELOPMENT OF SMART CONTROL METHODS FOR INDUSTRIAL ROBOTS

**Summary of the Doctoral Thesis**

Scientific supervisors
Senior Researcher Dr. sc. comp.
MODRIS GREITĀNS

Professor Dr. sc. ing.
PĒTERIS APSE-APSĪTIS

RTU Press
Riga 2023

Ārents, J. Research and Development of Smart Control Methods for Industrial Robots. Riga: RTU Press, 2023. 39 p.

Published in accordance with the decision of the Promotion Council "RTU P-14", 1st of September 2023, Minutes No. 04030-9.12.2/4.

**https://doi.org/10.7250/9789934229893**
**ISBN 978-9934-22-989-3 (pdf)**

# DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR THE PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (Ph. D.), the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on 28$^{th}$ of November 2023 10.00 at the Faculty of Electrical and Environmental Engineering of Riga Technical University, 12/1 Azenes Street, Room 212.

OFFICIAL REVIEWERS

Professor *Dr.sc.ing.* Nadežda Kuņicina,

Riga Technical University, Latvia

Managing Director *Dr.* Niki Kousi

EIT Manufacturing, Greece

Technical Domain Leader *Ph.D.* Muhammad Raheel Afzal

Flanders Make, Belgium

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for the review to Riga Technical University for the promotion to the scientific degree of Doctor of Science (Ph. D.) is my own. I confirm that this Doctoral Thesis has not been submitted to any other university for promotion to a scientific degree.

Jānis Ārents ……………………………. (signature)
Date: ………………………

The Doctoral Thesis has been written in English. It consists of an Introduction, 5 chapters, Conclusions; 57 figures, 6 tables, 2 appendices; the total number of pages is 125, including appendices. The Bibliography contains 166 titles.

# CONTENTS

# ABBREVIATIONS

**2D** – 2 Dimensional
**3D** – 3 Dimensional
**AP** – Average Precision
**CNN** – Convolutional Neural Network
**CV** – Computer Vision
**DL** – Deep Learning
**DOF** – Degrees of Freedom
**EDI** – Institute of Electronics and Computer Science
**EOAT** – End of Arm Tooling
**FN** – False Negative
**FP** – False Positive
**GRU** – Gated Recurrent Unit
**IoU** – Intersection over Union
**OMPL** – Open Motion Planning Library
**PCA** – Principal Component Analysis
**RGB** – Red, Green, Blue
**RGB-D** – Red, Green, Blue, Depth
**RNN** – Recurrent Neural Network
**ROI** – Region of Interest
**ROS** – Robot Operating System
**ROS-I** – ROS-Industrial
**Rviz** – ROS visualisation
**SMEs** – Small and Medium-sized Enterprises
**STOMP** – Stochastic Trajectory Optimisation for Motion Planning
**TCP** – Tool Centre Point
**TP** – True Positive
**YOLO** – You Only Look Once

# NOMENCLATURE

$\pi, \pi_\theta$ – policy, policy with parameters $\theta$

$s$ – state of the system or direct observation

$s_t, s'$ – $s$ in time step $t$, future $s$

$a, a_t$ – action, action in time step $t$

$o$ – observation

$\boldsymbol{r}$ – translation vector

$g_t$ – gripper actuator signal

$q$ – quaternion

# GENERAL DESCRIPTION OF THE WORK

## The Urgency of Subject Matter

Numerous sectors are currently in the midst of digital transformation, resulting in shifts within pertinent markets, strategies, and methodologies. Full automation, where the need for smart control of industrial robots would probably be absent, is possible mostly in large enterprises as it requires significant resources, however, 99 % of enterprises in the manufacturing sector, which is also one of the main practical beneficiaries of the work carried out in this Thesis, are SMEs [1].

For SMEs, resources to implement fully automated processes that would avoid unstructured and dynamic environments can be limited. So, smart approaches and novel technologies are required. However, these technologies require new skills, possibly re-skilling. But we are also in front of a skill gap where Europe's workforce is ageing [2], and there is a decrease of new students interested in STEM subjects, which is going to be a very big issue in the future in having personnel with the required skills for operating complex systems, including AI-based robotic systems. Mainly since such high-level qualifications are needed throughout many sectors with a forecast to increase [3].

There is a clear need for technologies that can achieve more human-like performance, are easy to deploy, reconfigure, and operate and are essentially more friendly for SMEs and address their challenges in a more agile way. Smart industrial robots are one of the ways how this global challenge can be tackled by being adaptive to both dynamic environments and the variety of tasks they need to complete. However, current state-of-the-art approaches still require substantial manual work for programming robots or preparing data for relevant algorithms.

## The Objectives of the Thesis

**The Thesis addresses the smart control methods for industrial robots, their applications, and learning strategies**. Particularly, efficient data preparation, computer-vision-based robot control, and knowledge acquisition from humans through demonstrations. **The primary aim of the Thesis is to research and develop novel techniques in these fields and, therefore, improve the potential deployment of industrial robots for working in dynamic environments and advancing such systems' operability.** Several tasks have been defined to reach the aim of the Thesis:

- to perform a review of the literature on smart control methods for industrial robots;
- to formalize the concept of smart control of industrial robots;
- to research and develop efficient data preparation methods for computer-vision-based robot control;
- to research and develop a methodology for human demonstration incorporation in the knowledge acquisition process;

- to construct an experimental setup and apply the proposed methods in demonstrations;
- to draw conclusions about the Thesis results.

## Research Methodology

The initial objectives of the Thesis are addressed through analytical research methods. Analytical techniques are employed to review the existing literature within the field systematically. This process serves to identify challenges while also facilitating the proposal and development of novel techniques for the smart control of industrial robots. Programming languages such as C++ and Python are utilized for implementation and validation purposes. Subsequently, rigorous testing and validation procedures are conducted to assess the attained levels of precision across various tasks. The efficacy of the developed methods is substantiated through their integration into several demonstrations. These demonstrations serve as comprehensive tests of the methods' functionality and effectiveness in real-world scenarios. By subjecting the methods to practical applications, their performance is evaluated, and any refinements are made to ensure their robustness and reliability.

## Scientific Novelty and Main Results

The Thesis results in novel methods for industrial robotic systems in the fields of efficient data preparation, computer-vision-based robot control, and knowledge acquisition from humans through indirect demonstrations.

**Smart industrial robot concept**. The identification of core functionalities within the smart control of industrial robots was fueled by the exploration of current trends in robot control leading to a formalization of the concept of smart control of industrial robots following the principle of ”see-think-act”. The concept is a broad view of smart industrial robots consisting of such functionalities as perception; high-level instructions and context-aware task execution; knowledge acquisition and generalization; adaptive planning.

**Synthetic data generation** methodology for training data acquisition demonstrates its feasibility and potential for deep-learning-based tasks. Synthetic data generation frameworks show promising results for industrial robot pick and place tasks in a bin-picking setup for simple-shaped objects (cylindrical shaped and parallelepipedic shaped objects) and can not only supplement real data when the variety of real training data is insufficient but also completely replace real data and still perform reliably in real-world settings. The use of synthetic data generation in robotic systems remarkably decreases the required time and manual efforts within the data generation process, therefore, increasing the maintainability of smart robotic systems and essentially improving the usability of modern computer vision methods in manufacturing settings.

**Computer-vision-based robot control** exploits the synthetic data generation methodology, the respective deep-learning models in computer-vision-based robot control for grasp pose estimation are trained solely on synthetic data. The achieved results show the ability to effectively tackle current manufacturing challenges efficiently, addressing maintainability and usability issues of complex AI-based robotic systems. The synthetic image generator also has a critical role in the evaluation process, forming a novel validation framework. Its versatility enables a rapid evaluation of various aspects of both the proposed techniques and other possible approaches, providing tentative precision results that demonstrate the viability of the specific use case. Thus, the synthetic data generation approach is a powerful tool in the iterative development and optimization of computer-vision-based robot control systems.

**Imitation-learning-based robot control** incorporates indirect human demonstrations in knowledge acquisition process. A methodology is proposed, and a framework is developed for turning raw data acquired by motion capture equipment into demonstrations for training neural networks for accomplishing the throwing task of waste plastic bottles. The solved task itself expresses the novelty by increasing the reach of industrial robots and potentially decreasing the cycle times by throwing objects. The proposed methodology, however, eases the programming of industrial robots. Even though some steps are application-specific, the methodology is adaptable to different scenarios, serving as a novel framework for indirect human demonstration utilization in imitation-learning-based robot control.

## Thesis Statements to Be Defended

1. The utilization of synthetic data in the training process for simple-shaped object detection in bin-picking setup reduces the required manual effort, whilst trained models are able to detect at least one object in every scene with an IoU threshold above 0.95.
2. For computer-vision-based robot control in the case of simple-shaped objects, at least one valid grasp is found in bin-picking setup in more than 99 % of scenes when computer-vision algorithms have been trained solely on synthetic data.
3. The learned trajectories for an object-throwing task by utilizing indirect human demonstrations achieve cosine similarity of more than 0.98 when compared to the validation dataset and, when executed on a real robot, closely resemble the expert human throwing motions.

## Practical Value and Approbation

The results presented in this thesis are mostly the result of research activities at the Institute of Electronics and Computer Science, Latvia, in Horizon 2020 Electronic Components and Systems for European Leadership (ECSEL) projects:

- “Artificial Intelligence for Digitizing Industry” (**AI4DI**, GA:826060),
- “Vision, Identification, with Z-sensing Technology and key Applications” (**VIZTA**, GA:826600),
- “Intelligent Motion Control under Industry 4.E” (**IMOCO4.E**, GA:101007311).

Horizon 2020 project:

- “Digital Technologies, Advanced Robotics and increased Cyber-security for Agile Production in Future European Manufacturing Ecosystems” (**TRINITY**, GA:825196).

Horizon Europe, Key Digital Technologies Joint Undertaking (KDT JU) project:

- “Edge AI Technologies for Optimised Performance Embedded Processing” (**EdgeAI**, GA:101097300).

National funding projects:

- Latvian Council of Science, “Smart Materials, Photonics, Technologies and Engineering Ecosystem” project No. VPP-EM-FOTONIKA-2022/1-0001, **MOTE**,
- Latvian–Lithuanian–Taiwanese scientific cooperation support fund’s project “Development of microrobot based on visual recognition and machine learning for manipulation of individual living cells”, No. LV-LT-TW/2021/8, 2020-2023, **RoVam**,
- Project No. 1.2.1.1/16/A/002 “Competency centre for Latvian Electrical and optical manufacturing industry” Research no. 11 “The research on the development of computer vision techniques for the automation of industrial processes”, **DIPA**.

The results are presented in several published papers in various scientific journals [4–10], scientific conferences [11–13], and book chapters [14, 15]. Part of the research activities of the Thesis are included in the **“Smart robot system with advanced vision, sensing, and human gesture understanding capabilities”** which was evaluated by the Latvian Academy of Sciences in the annual Science Achievements’ Competition as one of the most significant scientific achievements in Latvia in 2022[1].

## Structure of the Thesis

The Thesis consists of 125 pages, 57 figures, six tables, a list of 166 literature sources, and two appendices. It contains six chapters. Chapter 1 is an overview of smart industrial robot control methods. It begins with a formalization of the smart industrial robot concept, including the identification of the core functionalities, continues with an overview of different control strategies, and lists the identified challenges and open issues at the end. Chapter 2 shows the developed synthetic data generation techniques and usage in deep-learning-based tasks. Chapter 3 builds upon the synthetic data generation approaches for usage in robotic systems, namely as computer-vision-based robot control. Chapter 4 explores learning from human demonstrations, therefore, imitating humans and realizing imitation-based robot control. Chapter 5 describes the experimental setup and demonstrations where the developed approaches are applied and qualitatively evaluated. The conclusions about the developed methods and test results are presented in Chapter 6.

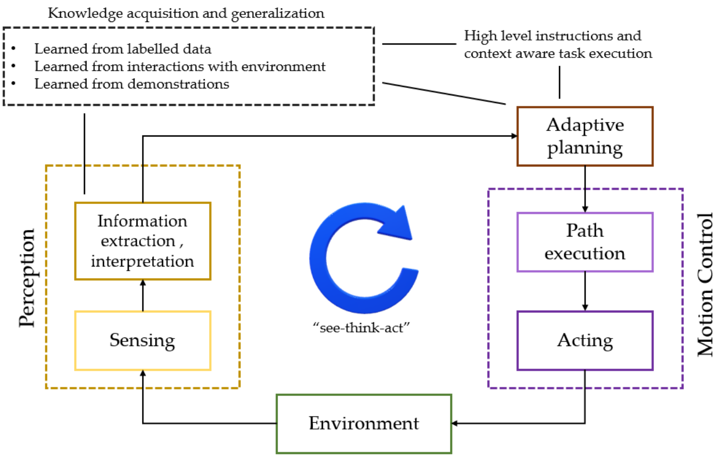
[1] *https://www.lza.lv/en/activities/news/1368-winners-of-the-annual-science-achievements-competition-2022-announced*

# 1. OVERVIEW AND TRENDS OF SMART CONTROL METHODS FOR INDUSTRIAL ROBOTS

The need for smart industrial robot control methods arises when there is uncertainty in the environment. With the environment in this context, we mainly understand the region in the environment that can be reached by the robot, e.g., the robot's workspace. Typically this workspace scales down to a region of interest (ROI) or objects of interest to which the robot is manipulating. Nevertheless, there can be different types of uncertainties in the robots' workspace, such as randomly placed objects [4], dynamic obstacles [16], and human presence, etc. Usually, such tasks cannot be preprogrammed, and the robotic system strictly depends on the information it receives from the environment and the learned strategies. Additionally, there is also a need to reduce the time spent on programming industrial robots, as traditional approaches of handcrafting trajectories [17] is a laborious, complicated and error-prone task.

## 1.1. Smart industrial robot concept

Smart industrial robot control is an important element, yet only a fraction of the smart manufacturing ecosystem. In the Thesis, and especially in this section, the focus is concentrated on components, methods and strategies that act as enablers of smart industrial robot control. The overall concept of smart industrial robot control follows the principle of "see-think-act". The diagram illustrated in Fig. 1.1 is built upon this principle and integrates such core functionalities of smart industrial robots as perception, high-level instruction and context-aware task execution, knowledge acquisition and generalization, and adaptive planning.




1.1. Fig. Smart industrial robot concept [6].

## 1.2. Computer-vision-based control

Computer-vision-based methods, in combination with the control loop of the robot, are one of the most common approaches [18] to deal with the environment's uncertainty and interact with it. Several different scenarios or combinations of conditions can be distinguished, such as working with overlapping, similar, different, multiple, complex objects with undefined positions, etc. Additionally, the objects can be known, and whether the information is available beforehand or otherwise, the objects are unknown. Some combinations of conditions make the task more complex and vice versa, and accordingly, available methods are similar for some scenarios. The required industrial robot control methods can briefly be distinguished in two different cases, respectively:

- tasks that require the robots' end-of-arm tooling (EOAT) to be precisely positioned for the entire trajectory;
- tasks that require explicit grasp planning.

The application scenarios for the second task have the most occurrence in modern industrial robot use cases. Typically this can be a region of interest that includes multiple, obstructed or complex objects requiring specific grasping approaches. One of the most common robotic grasping problems in industry settings is bin-picking [19]. Accordingly, this problem has been addressed for several decades [20]. Bin-picking is a task where arbitrarily placed objects that overlap each other in a bin must be taken out and placed in a dedicated position and orientation. Historically this problem has been addressed as one of the most significant robotic challenges in manufacturing automation [21]. As many different combinations of conditions can be present in this problem, the proposed methods that tend to solve this task also cover most of the robotic grasping issues.

Vision-based grasping approaches can be divided into analytic or data-driven methods, where analytic methods focus on analyzing the shape of the target object, whereas data-driven methods are based on machine learning [22]. Learning-based approaches tend to be more generic and flexible in dealing with uncertainties of the environment and, therefore, also more promising in the smart manufacturing context. Model-based and model-free learning-based grasping approaches can be distinguished, respectively, by working with known or unknown objects.

## 1.3. Imitation-learning-based control

The ability to utilize expert behaviour and learn about tasks from demonstrations is the basis of imitation learning [23]. The two major classes of imitation learning are behavioural cloning and inverse reinforcement learning. Behavioural cloning is often referred to as learning a policy that directly maps from the state to the control input. Alternatively, recovering the reward function from demonstrations is referred to as inverse reinforcement learning [24]. The demonstrations can be acquired directly and indirectly, such as teleoperation, kinesthetic teaching, motion capture, virtual/augmented reality and video demonstrations [25]. In contrast to direct teaching, indirect teaching does not directly control the robot. Such systems typically learn from visual demonstrations [26] and wearable devices [27] by capturing human motions to generate trajectories for robots. When it comes to the use of motion capture data, previous work in the robotics and imitation learning corner is quite sparse.

## 1.4. Identified challenges and open issues

### 1.4.1. Availability of the data

The collection and processing of data for machine-learning tasks, including smart industrial robot control, are vital, with data-related tasks taking up a significant portion of time in AI projects, as indicated by [28]. Different data collection techniques are employed based on use cases, and openly available datasets can be adapted for specific model training needs in fields like autonomous driving [29], household object recognition, people detection, and machine translation [30]. However, the rapidly changing product variety in smart manufacturing leads to low re-usability of existing datasets. Manual labelling methods are impractical due to time and cost constraints and the need for expert knowledge, potentially causing human errors. On the other hand, synthetic data generation, utilizing physics and graphics engines, reduces the time required for labelling by replicating real-life environmental uncertainties; however, there is a lack of comprehensive studies in the manufacturing environment.

### 1.4.2. Maintainability and usability

The literature review also focused on identifying challenges in integrating and deploying smart control methods for industrial robots, emphasizing the importance of software quality characteristics defined in ISO/IEC 25010 [31] and corresponding measures in ISO/IEC 25023 [32]. Maintainability and usability were identified as crucial elements for successful smart industrial robot deployment and control. Maintainability involves efficiently modifying the system to adapt to changes, while usability ensures ease of use for specified goals without deep technical knowledge. The proposed techniques provide valuable insights into addressing system deployment and adaptation concerns, such as computer vision-based solutions combined with robots and the potential of imitation learning to transfer knowledge efficiently.

# 2. SYNTHETIC DATA GENERATION FOR DEEP-LEARNING-BASED TASKS

Even though modern object detection methods are effective at detecting traffic signs, pedestrians, household objects, etc., the methods fall short in industrial settings due to a lack of training data. The proposed method utilizes modern graphics engines to generate realistic data for training neural networks for deep-learning tasks in industrial-specific scenarios, where data can be use-case-specific and change over time. The focus is concentrated on generating versatile data with a vast amount of adjustable parameters. In this case, the data of unobstructed objects in the corresponding orientation is extracted to train AI models only on the objects that have the highest probability of resulting in a successful grasp when the model is deployed. The object detection described in the experiments of this section serves as the first step in the whole pick and place process, and only information about the most promising objects that could be grasped is then processed further.

The main aim of this task is to automate the systematic rendering of highly realistic synthetic pictures to generate data sets for training the object detection algorithm. The image generator obtains images by arranging any kind of objects that have 3D models within a virtual 3D scene from which it renders highly realistic images. In this case, it is a box with simple-shaped objects – white bottles. By tuning parameters of the 3D scene such as an object, camera and light positions, object colour or texture and surface properties, brightness, contrast, and saturation, a large image diversity can be generated in resolution and levels of realism depending on the user's needs. By further exporting relevant ground truth data from the 3D scene, including the location and orientation of objects within a rendered image, the generated data is fully annotated.

By sampling all possible configurations of objects and image parameters within the 3D scene (in arbitrary, user-defined granularity), a modification space is defined, allowing for the automated generation of large synthetic data sets, for which the diversity of images is controlled by the user. By systematically defining appropriate scenes and modification spaces, the image generator can be used to generate not only training and validation data sets in sufficient quantity (overcoming a lack of training data, which is often a limiting problem) but, in general, also allows for generating data sets specifically designed for specific experiments later on (e.g. to compare across different classification models or to extract specific insights of the algorithm.

## 2.1. Datasets

For the feasibility evaluation of synthetic data on corresponding experiments, the real data was gathered by randomly distributing bottles in a box. For each of the acquired images, the positions of the bottles were altered. The intensity of lightning and camera exposure time was systematically modified to acquire a high diversity of different lighting conditions. In total, for

(**a**) Generated image (**b**) Generated depthmap

2.1. Fig. Synthetic data of randomly piled, similar objects.

(**a**) Ground truth bounding box (**b**) Ground truth segmentation image

2.2. Fig. Data for learning-based computer vision algorithms [12].

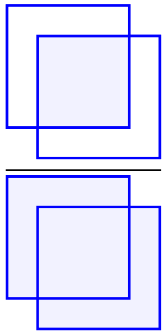
training purposes, 2200 real images were acquired and manually labelled, from which 1760 images were for training and 440 images for validation purposes. Furthermore, the training and validation datasets were rotated four times by 90 degrees, in total, resulting in 7040 training images and 1760 validation images. Two test datasets were gathered and manually labelled; the first dataset *Test 1* was captured in similar conditions as the real training dataset; however, *Test 2* was captured with a different camera and in different conditions. The synthetic dataset was generated in the same amount as the real dataset, consisting of 8800 photo-realistic high-resolution scenes.

## 2.2. Evaluation metrics

The most commonly used metric to measure the accuracy of object detection in the images is average precision (AP), which is also utilized in this section to evaluate the performance of object detection in the experiments. In this case, variously set (0.5–0.95) intersection over union

(IoU) thresholds are used to perform AP measurements.

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} = \tag{2.1}$$



$$class(IoU) = \begin{Bmatrix} Positive \rightarrow & IoU \geq & Threshold \\ Negative \rightarrow & IoU < & Threshold \end{Bmatrix} \tag{2.2}$$

With the IoU, the overlapping area is measured between the ground truth and the predicted bounding box, and the precision is evaluated over different thresholds. True positive (TP), false positive (FP), and false negative (FN) estimates for each detected object are used to calculate precision and recall to perform AP measurements.

$$precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$AP = \int_0^1 p(r)dr \tag{2.5}$$

, where $p(r)$ is the interpolated precision at each recall level $r$. The integral represents the area under the precision-recall curve.

## 2.3. Results

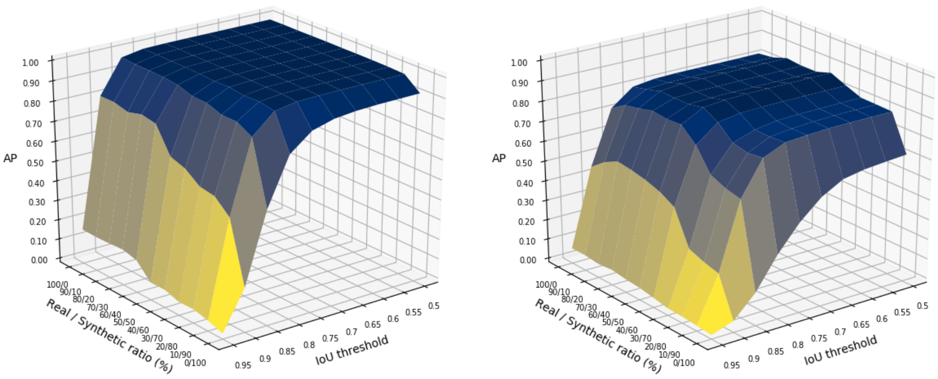
The performance and precision of the proposed synthetic data generation approach were evaluated by various aspects. First, the object detection performance was investigated with deep learning models trained using different ratios of synthetic and real data combinations and by utilizing the maximum amounts of the acquired data. Starting with 100 % of real data, the real data ratio was incrementally decreased by 10 % at the time by substituting it with the synthetic data as depicted in Table 2.1.

The evaluation was performed on the described datasets – *Test 1* and *Test 2*. Object detector evaluated on data close to real training data *(Test 1)* scored similar average precision results when real data amount was higher than synthetic data, as depicted in Fig. 2.3(**a**). This also

2.1. table

Evaluation of Object Detectors Precision [12]

| **Data distribution** | | | ***Test 1*** | | | ***Test 2*** | | |
|---|---|---|---|---|---|---|---|---|
| **Real** | **Synthetic** | **Real / Synthetic ratio %** | **Step** | **AP @0.5:0.95** | **OD %** | **Step** | **AP @0.5:0.95** | **OD %** |
| 8800 | 0 | 100 / 0 | 9100 | 88.61 | 100.00 | 9100 | 69.22 | 96.95 |
| 7920 | 880 | 90 / 10 | 7900 | 88.61 | 100.00 | 9200 | 71.04 | 98.47 |
| 7040 | 1760 | 80 / 20 | 6000 | 88.36 | 100.00 | 8000 | 73.23 | 100.00 |
| 6160 | 2640 | 70 / 30 | 6300 | **88.65** | 100.00 | 7600 | 72.83 | 100.00 |
| 5280 | 3520 | 60 / 40 | 6900 | 88.22 | 100.00 | 7400 | 72.5 | 100.00 |
| 4400 | 4400 | 50 / 50 | 4400 | 85.82 | 100.00 | 5000 | **73.84** | 100.00 |
| 3520 | 5280 | 40 / 60 | 8000 | 85.59 | 100.00 | 8700 | 70.27 | 100.00 |
| 2640 | 6160 | 30 / 70 | 7200 | 84.39 | 100.00 | 5900 | 64.57 | 100.00 |
| 1760 | 7040 | 20 / 80 | 7200 | 84.23 | 100.00 | 4500 | 63.62 | 100.00 |
| 880 | 7920 | 10 / 90 | 8200 | 82.59 | 100.00 | 7000 | 63.25 | 100.00 |
| 0 | 8800 | 0 /100 | 7700 | 70.01 | 100.00 | 5000 | 38.66 | 100.00 |




(**a**) *Test 1* (**b**) *Test 2*

2.3. Fig. Average precision of object detection models on real images over different IoU thresholds, viewed by the ratio of real to synthetic data in the training datasets [12].

holds true for higher IoU threshold values from 0.85 to 0.95. All of the trained models showed similar average precision results in the IoU threshold region from 0.5–0.8. The main difference can be seen in the case when the model is trained on purely synthetic data, as the precision remarkably decreases. A different situation can be seen when the object detector is evaluated on a test dataset that contains different environmental parameters – *Test 2*. In this case, the synthetic data supplements real data and increases average precision whilst achieving the highest precision on a 50/50 ratio. Similarly, as with the evaluation results on *Test 1* dataset, also in this case, the object detector trained on purely synthetic datasets showed the least precision.

(**a**) *Test 1* (**b**) *Test 2*

2.4. Fig. Results visualized over different data sets.

The real data still outperforms synthetic data if we are solely analysing the object detectors' average precision; however, in smart control of industrial robots, it is not that crucial to have precise information about all the objects in the scene, as typically only one object can be grasped anyway, after which the scene has changed and needs to be perceived again. Respectively, the goal is to detect at least one object in the scene with an IoU threshold above 0.95. The obtained results on this aspect are depicted in Table 2.1 under columns Object Detected (OD). On both test datasets, the trained models could meet this requirement, except in the Test 2 case, when the model was trained on purely real data and in the following 90/10 ratio, proving the first statement. Overall, in most cases when the real data ratio was higher than synthetic data, the model achieved higher precision ratings. Whereas the achieved precision is sufficient in the case of simple-shaped-rigid objects for the computer-vision-based control. Thus, by diversifying the training dataset with synthetic images, a precision increase can be observed; however, when the synthetic ratio is over 50 %, the precision decreases.

# 3. COMPUTER-VISION-BASED CONTROL

To increase the applicability of industrial robots and enhance their autonomy, the integration of AI techniques and computer vision-based control is pursued. Traditional robot programming is challenging and requires expertise, but computer-vision-based control enables robots to perceive and understand the environment, performing tasks in dynamic conditions. However, the complexity of developing such systems increases due to the need for additional processing of environmental data. Motion planning is also crucial to ensure precise and collision-free movements in dynamic environments. Synthetic data is used to drive experiments and develop computer-vision-based robotic systems.

## 3.1. Grasp-pose estimation and grasp planning

Computer vision algorithms give valuable information about the environment and particularly, in this case, about the objects and where they are located in the workspace. The object detection algorithms have been trained to detect only the objects that are most likely to result in a successful grasp. Also in this matter, the results show that in every scene with a model trained on completely synthetically generated data, at least one object with an IoU threshold over 0.95 is detected.

In case when objects overlap each other in a pile, as illustrated in Fig. 2.3, the position of objects varies in three dimensions respectively the objects are positioned also in different depth levels. For scenarios when the object just needs to be grasped and placed without any additional manipulation steps or a need for precise placing, the pick-and-place cycle can be completed with object detection as the main computer-vision functionality. After object detection, the grasp pose is estimated by aligning the RGB information with respective depth information in the region typically acquired by RGB-D sensors/3D cameras. Depth information in RGB-D images is aligned with the corresponding image pixels, providing information on how far from the camera the region of interest is located. The region of interest in this case, is a circle shaped area around the centre point of the detected object. From this area, the approach angle is calculated by principal component analysis.

For precise manipulation of object or placement, the 2D object detection part itself falls short, as in this way, the object is localized only in 2D space with a bounding box that includes the object and some area around it. The detection results cannot be directly used to determine the orientation of the object, which is required for precise placement or other manipulation procedures. Therefore, the detected area is utilized further by the object segmentation algorithm. Segmentation determines which class every pixel of an image belongs to. Instance segmentation, however, is a type of segmentation that differentiates among pixels belonging to different instances of the same class. With this information, a visible shape of a specific object is acquired and used to determine the grasp pose of an object, which in turn is handy for picking up and manipulating the object. For the object segmentation task, the Mask R-CNN algorithm is utilized, where the neural network is trained on purely synthetically generated data.

(**a**) Bottles (**b**) Cans and bottles

3.1. Fig. Different scenarios that include segmented objects with calculated grasp positions.

The information returned by object segmentation allows to precisely determine the longitudinal direction of the object and other specific features of the object, for example, in the case of the bottle, also the top and bottom parts of it can be identified. After mapping the acquired information to corresponding depth information, the grasp pose is calculated, similarly as described previously. The grasp pose is also illustrated in Fig. 3.1 with the axis symbol.

## 3.2. Robot-control

The grasping process itself is usually split into several parts that typically depend on the types of gripper and object of interest. In the Thesis, the robot control is mostly connected to two grasping techniques, namely, vacuum grasping and 2-finger grasping. There are some minor differences between both, however, the main grasping phases are similar: approaching the object, coming into contact with the object, increasing the force/vacuum until the object is securely grasped and moving the object until it can be released in the place position. In bin-picking settings, a surrounding-aware visual perception system, in combination with the correct approach and retreat movements, is a remarkably important part of the navigation system. Objects are randomly distributed, and unintended collisions that can occur with the bin, other environmental elements, and the robot itself should be strictly avoided.

Firstly, the data acquisition device, in this case, a 3D camera, and the robot or robots must be operating in the same coordinate system, for which hand-eye calibration is performed. Secondly, the robot control framework needs to be chosen. The way how industrial robots are programmed strongly depends on the robot manufacturer. In terms of flexibility of the system, hardware abstraction is important so industrial robots from different manufacturers can be programmed and controlled in one way. There are multiple frameworks that offer compatibility with different industrial robot brands, but the most popular of them is ROS-Industrial [33], which is a part of the Robot Operating System (ROS). Thirdly, a crucial aspect is the motion planning part, as in applications where it is not possible to predefine how the robot must move in order to manipulate objects, the motions have to be planned before each move. For this purpose, OMPL and STOMP motion planners were experimentally analyzed and compared in two different scenarios where the STOMP motion planner was chosen according to planning time, computed trajectories and their repeatability and used in the developed robotic systems. Finally, a set of actions that are compliant with an unstructured environment was defined and implemented.

## 3.3. Experimental validation

The proposed computer vision-based control has been experimentally validated for simple-shaped rigid objects. The precision of the estimated grasp pose was measured as a validation metric. The synthetic data generation framework described in Chapter 2 serves for training and validation by creating diverse datasets. A validation pipeline was built using synthetic data with known object positions and orientations. It calculates optimal reference grasp poses for predicted grasps and uses various metrics to compare them. The synthetic data generator from Chapter 2 was enhanced to include object pose outputs, forming the basis of the validation framework. This generates varied validation datasets with different object types, counts, lighting, and placements. These datasets include virtual camera images and annotation files with object information.

The validation dataset is fed into an inference pipeline with data acquisition, object detection, and grasp pose estimation modules to predict grasps. These predicted grasps, along with annotated input images showing grasp locations and orientations, are saved. Each pickable object class has separate inferred grasp data treated independently. Cases with no pickable objects are recorded.

To compare grasps, ground truth object poses and inferred grasp information are required. After inferring on validation data, these steps compute evaluation metrics:

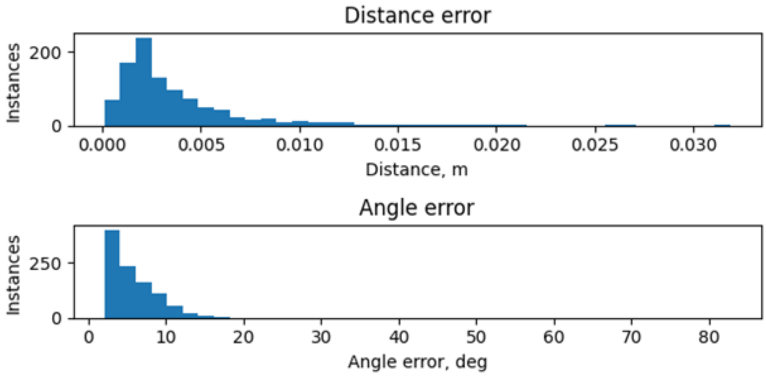
- Compare inferred grasp with ground-truth object positions, finding the closest object based on Euclidean distance. If incorrect type, it's marked as misclassified and excluded.
- Use target object and reference grasp to compute optimal grasp with type-specific heuristics. For cylindrical bottles, the reference grasp aligns with the object's central circumference on the XZ plane. For rectangular cans, 4 valid grasps are constructed.
- Save the dataset of each grasp paired with its reference and scene identifier.

The evaluation included generating 1000 scenes with varying object counts (10, 20, 50, 100) per preset. Each preset contained 250 randomized combinations of bottles and cans; an example is illustrated in Fig. 3.4.

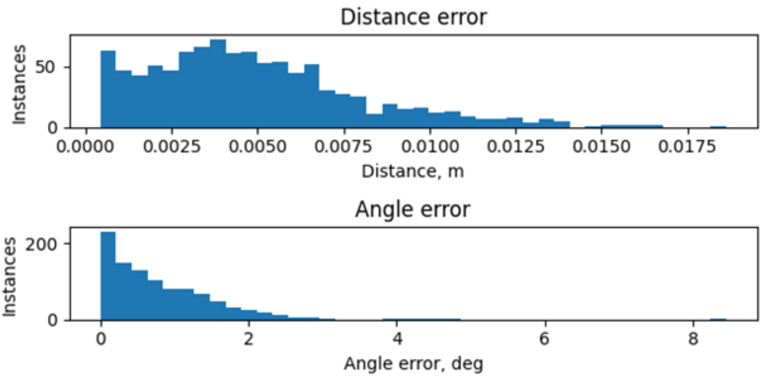
3.1. table

Object Detection and Grasp Pose Estimation Results

| Scene count | Object type | Position error | | Rotation error | | Valid grasps (%) | Classified correctly (%) |
|---|---|---|---|---|---|---|---|
| | | Mean average (mm) | Stdev (mm) | Mean average (degrees) | Stdev (degrees) | | |
| 1000 | Bottle | 4.926 | 3.156 | 0.802 | 0.736 | 99.9 | 100.00 |
| | Can | 3.722 | 3.446 | 4.637 | 4.432 | 99.8 | 100.00 |




3.2. Fig. Histogram of distance and angle errors for grasp pose estimation of a can.




3.3. Fig. Histogram of distance and angle errors for grasp pose estimation of a bottle.

The experimental results are summarized in Table 3.1. Both object types show accurate classification with no errors in identifying the object type. Valid grasps were obtained in 99.9 % of bottle cases and 99.8 % of can cases, proving the second statement. Some corner cases may

3.4. Fig. Example of estimated grasp poses within a preset of 50 objects.

arise due to dynamic environmental features and object position randomness. Occasionally, all objects could be positioned in a way that does not meet grasp requirements, potentially due to collision with the container. Though this likelihood is low, it is reflected in the results.

The second validation aspect relates to the grasp pose precision. Figs. 3.2 and 3.3 illustrate distance and orientation error distributions for cans and bottles, respectively. On average, errors for both object types stay below +/– 5 mm for position and +/– 5 degrees for rotation. Even with a higher error, the grasp remains valid as long as the point is on the object's graspable surface. Higher errors might affect post-grasp steps like object placement, depending on the application. The validation framework can recommend an additional repositioning system to ensure accurate object placement.

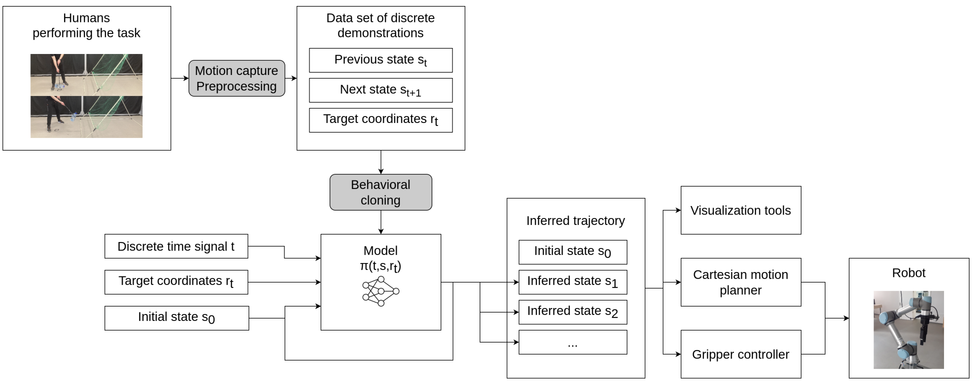
Real-life deployment could yield differences from synthetic results due to randomly dropped objects, making direct precision evaluation challenging. Therefore, real data testing focused on the system's ability to grasp at least one object in each scene. Tests with different presets in real-life scenarios confirmed the approach's success in grasping at least one object per scene without classification errors.

# 4. IMITATION-LEARNING-BASED CONTROL

In the context of industrial robotics, where tasks mirror human actions, imitation learning is adopted to incorporate human demonstrations into knowledge acquisition for robots. Human actions are recorded, forming explicit demonstration-based training data augmented with indirectly observed variables and control signals. Artificial neural network models are trained to replicate these trajectories as Cartesian motion plans. Motion capture is chosen as an intermediate solution. It eliminates the need for simulated environments, operates in Cartesian coordinates instead of robot configuration space, and avoids complex image processing layers [34–36]. An example use case guides development and evaluation: enhancing a recycling plant robot's capabilities by enabling object throwing. Although its basic mechanics can be explicitly programmed, it is chosen as a benchmark for robot programming via demonstrations. The task is non-trivial under limited information conditions, suitable for intuitive human evaluation, and allows quantitative assessment of model performance via extrapolated throw accuracy. While the author primarily contributes to conceptualization and methodology, the implementation is done by Pēteris Račinskis, assisted by the author.

## 4.1. Proposed approach

A high-level overview of the proposed approach is illustrated in Fig. 4.1. Demonstrations are recorded with motion capture equipment, a split and cropped data set augmented with target coordinates and a gripper actuation signal is created. This is used to train neural network models in behavioural cloning. When operating in the open-loop regime, prior model outputs are used to autoregressively predict subsequent states. The resulting sequence can then be fed into a Cartesian motion planner to control a robot.




4.1. Fig. A high-level overview of the proposed approach [8].

The system can be broken down into three main sections – the collection of observations in the physical environment, a pipeline for turning raw observation data into structured demonstra-

4.1. table

Model Hyperparameters [8]

| Parameter | Feedforward | RNN |
|---|---|---|
| Architecture | 2 dense hidden layers, ReLU | GRU, dense linear output |
| Parameter counts | 128–1024 perceptrons per layer | 128–512 perceptrons in the unit |
| Training epochs | 20–100 | 300–1200 |
| Batch size | 64 | 32 |
| Optimizer | Adam | Adam |
| Learning rate | $10^{-4}$ | $10^{-3}, 10^{-4}$ |

tions with additional control signals, and neural network model implementations. System performance was evaluated, and design feedback was obtained in two main ways. First, qualitative observations of the generated trajectories were made using spatial visualization in a virtual environment, followed by execution on simulated and real robots. When satisfactory performance was attained, a series of quantitative metrics were computed for comparing system outputs with training and validation datasets on different model architectures and hyperparameter sets.

Two classes of parametric models were studied as part of this task – simple feedforward neural networks and RNNs, operating autoregressive. After an initial hyperparameter discovery process, the values in Table 4.1 were arrived at for both model types respectively.

The basic model footprint is given as follows:

$$(\mathbf{r}_{t+1}, \mathbf{q}_{t+1}, g_{t+1}) = \pi_{\boldsymbol{\theta}} \left( \frac{t}{f_{sample}}, \mathbf{r}_t, \mathbf{q}_t, g_t, \mathbf{r}_t^{target} \right) \tag{4.1}$$

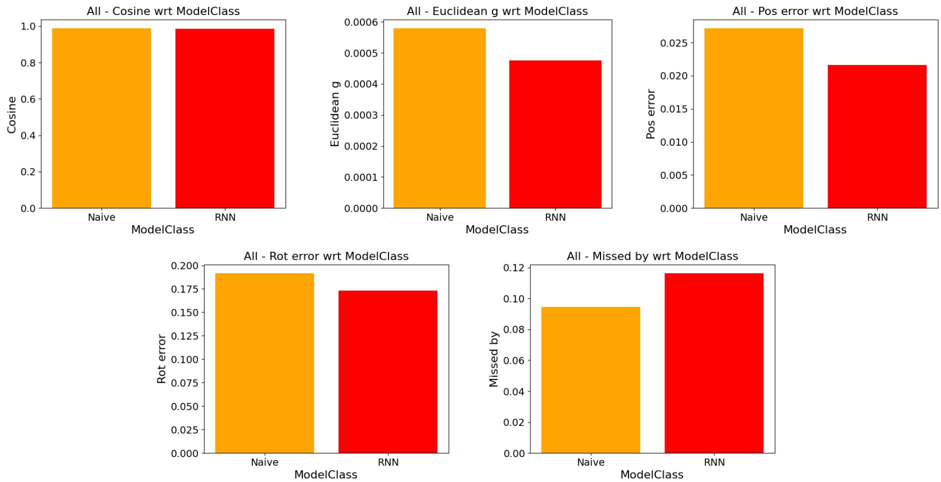
, where $\mathbf{r}_t, \mathbf{q}_t, g_t$ represent the end effector translation vector, orientation quaternion and gripper actuator signal, respectively, at time step $t$ in both the input and output. The input is augmented with a time/phase signal (discrete time step divided by the sampling frequency, in this case, $100Hz$) relative to the start of the demonstration or generated trajectory, as well as the target coordinate vector $\mathbf{r}_t^{target}$ which corresponds to the extrapolated target coordinates in the demonstration data set and commanded throw coordinates at inference. The time signal was added to the input, as it was found that trajectories generated by feedforward networks were liable to diverge without it, and the data sets thus modified were used for all training thereafter.

## 4.2. Results

Numerous models were trained in both of the architectures. After the initial trial and error process, a systematic training regimen produced the following:

- feedforward networks – a total of 48 models, varying the training data set, presence of a time signal in the features, perceptron counts per hidden layer, and training duration;
- RNNs – 36 models with data set, learning rate, model size, and training length being the

variable parameters.




4.2. Fig. Results – comparison between model architectures [8].

The qualitative aspects of the achieved results – the visual representations of trajectories and their characteristics when executed on a real robot – closely resemble throwing motions as executed by the human experts, suggesting that the selected approach to data collection is adequate for encoding the key features of this task. The same can be said for the fact that a large minority of the feedforward models and a majority of the recurrent ones were able to reproduce a complete throw trajectory for every set of initial conditions in the training and validation data sets. The most important takeaway regarding this aspect of the proposed approach is that augmenting the data with time-step information is important to achieve good performance. Comparing the two proposed model architectures by their best-attained values, in terms of distribution similarity measures such as cosine similarity and Euclidean distance, the results are generally quite good for both, with the RNN having the edge in most distance measures but notably one of the feedforward models demonstrating the highest cosine similarity, proving the third statement.

# 5. EXPERIMENTAL SETUP AND DEMONSTRATIONS

The developed methods within the Thesis are integrated into several demonstrations serving both as showing the practical use of them and as the final demonstrations in three Horizon 2020 projects:

- "Artificial Intelligence for Digitizing Industry" (**AI4DI**), described in Section 5.2.
- "Vision, Identification, with Z-sensing Technology and key Applications" (**VIZTA**), described in Section 5.3.
- "Digital Technologies, Advanced Robotics and Increased Cyber-security for Agile Production in Future European Manufacturing Ecosystems" (**TRINITY**), described in Section 5.4.

The experimental setup described in the upcoming section, however, serves as the basis for all the listed demonstrations.

## 5.1. Hardware and software building blocks

The experimental setup is built in a laboratory environment in EDI and within the respective hardware and software infrastructure. The hardware and software infrastructure, except for the high-performance computer (HPC), has been carefully researched and mostly built during the development of the Thesis, including the digital versions of it. The main hardware and common software components used in the experimental setup and demonstrations are described in the following subsections.

### 5.1.1. Hardware components

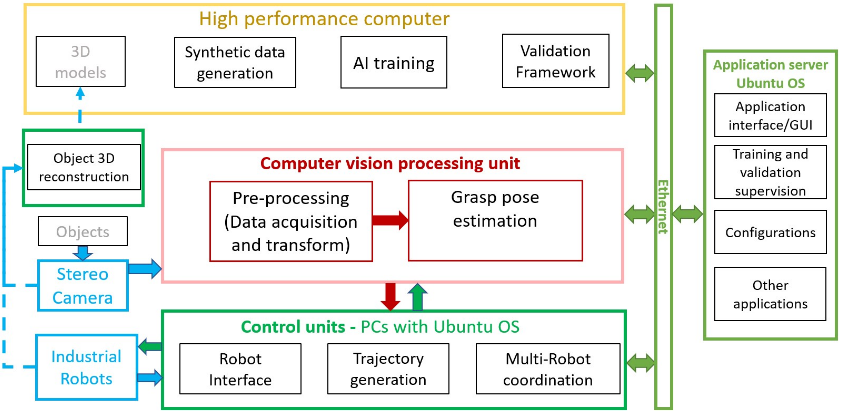
**Grippers**. The two main gripping strategies are vacuum gripping and parallel gripping. Both gripping strategies are included in the hardware infrastructure of demonstrators, specifically for the vacuum gripping Schmalz Rob-Set UR [37] is used. For parallel gripping, Robotiq two-finger gripper 2f-140, with a stroke of 140mm, grip force up to 125N and payload of 2.5kg.
**Depth sensors**. The hardware infrastructure for demonstrators contains two depth sensors, Zivid One M and Intel RealSense D415, that vary in parameters like precision, field of view, working range, weight, dimensions, etc.
**Industrial robots**. Industrial collaborative robots (Universal Robots UR5 [38] and UR5e) are used in demonstrators. These robots are ideal for automating low/medium-weight processing tasks. They have a maximum payload of 5kg, an 850mm reach radius, which can be improved with gripper modifications, and a pose repeatability of +/– 0.1mm for UR5 and +/– 0.03mm for UR5e.

### 5.1.2. Software components

**Robot Operating System**. The whole system communication is done using ROS [39] Melodic Morenia, as it is the ROS version compatible with Ubuntu Linux 18.04 configured on the processing units. All the further described packages are also compatible with this ROS version. ROS-I is being used in the demonstrators as a high-level controller in conjunction with a low-level controller provided by industrial robots.
**State Machine**. The state machine control of the system is done using the SMACC library for ROS. Conceptually, SMACC defines each individual state machine component as an *Orthogonal*, which also includes a *Client*. *Clients* define connections outside of the state machine. In the state machine, transitions between the states are triggered by *Events* [40].
**Data acquisition**. For unified data acquisition from depth sensors and required data transformations such as 3D coordinate computation from depth data.
**MoveIt!**. The path planning and execution.
**Universal Robot driver**. All the necessary tools for connecting, communicating, and controlling the robots.

## 5.2. Multi-robot collaboration for bin-picking task

A lot of industrial processes involve operations with a large number of different objects with an arbitrary location. It is hard to automate these kinds of processes because sometimes it is impossible to predetermine the positions of these objects. To overcome this issue, the developed methods within this Thesis are applied in the demonstration of muti-robot collaboration for the bin-picking task. Main functions, including the preparation details for data generation and training, hardware/software components and partitioning, are depicted in Fig. 5.1.




5.1. Fig. Hardware and software partitioning [14].

The most crucial robot movements are based on the information given by the computer vision system as described in Section 3.1 for rigid objects. It is launched separately from the two robot control programmes. When the robot state machines are launched, they connect to the computer vision system over an IP/TCP connection. When a robot is in a pick state, the first task is detecting the object. This is done by sending a request to the computer vision and artificial intelligence node. All the data is transferred via a standard ROS transport system with request/response semantics. The grasp position (x, y, z) and orientation in quaternion format (qw, qx, qy, qz) is a response to ROS service request. The returned information is an object's grasp position in reference to the camera coordinates. This then needs to be transformed in reference to the robot base coordinates. At this point, the given object grasp position and orientation are easily understandable by the robot, which means it can safely move on to the next tasks, which are computing the pick poses, motion planning, trajectory generation and execution.

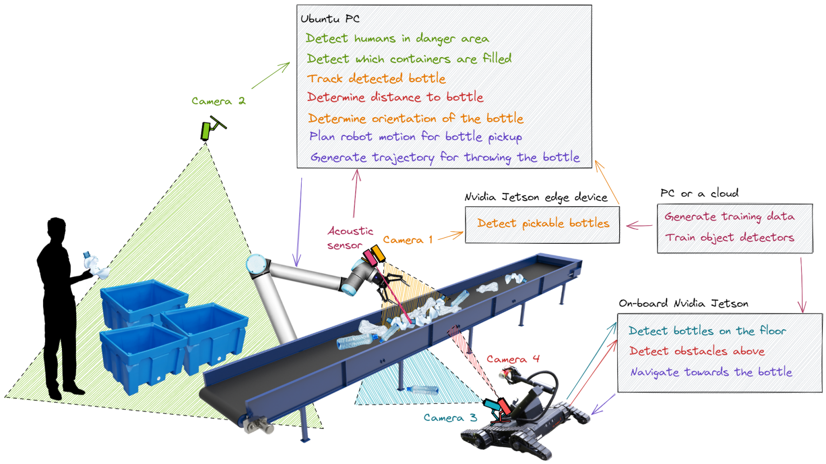
Overall, the system works reliably and succeeds at the integration of the proposed methods developed within the Thesis. For further illustrative purposes, the demonstration video can be watched by following the link: `https://youtu.be/OezRBPVZ1xU`.

## 5.3. Handling of waste plastic bottles

The developed demonstration showcases the use of the proposed methods within this Thesis in an Industry 4.0 application, specifically for automating the sorting of waste plastic bottles. The overall demo is shown in Fig. 5.2 and consists of multiple 3D cameras, an industrial robot arm, and a mobile robot. The main author's contribution is connected to the synthetic data generation methodology and implementation of computer-vision-based robot control and methodology for object-throwing functionality. The main software components are also depicted.

At actual sorting plants, waste bottles are piled on a moving conveyor belt. However, the demo in the laboratory environment is simplified and uses a static table. The demonstration addresses several challenges simultaneously. It is capable of picking objects from piles, even when the objects are identical or highly varied, randomly deformed, and transparent. While the system may be slower than the commonly used jet-of-air type automated sorters, it offers greater versatility and can handle more complex situations and piles. The current gripper-equipped sorters available are only demonstrated in simplified cases, such as picking well-separated objects. Moreover, this demonstrator aims to achieve more human-like performance by showcasing the throwing functionality typically executed by human workers in manual sorting lines. Although not fully integrated, this demonstration shows the potential of the proposed methods in an Industry 4.0 application.

The first task is to detect those bottles in a random pile that can be picked up by the industrial robot. Most often, those bottles are located on the top of the pile and are unobstructed by other objects. The chosen object detectors are based on deep artificial networks and trained in a




5.2. Fig. Overview of the demonstrator for sorting waste plastic bottles.

supervised fashion. That means a training dataset is needed. It began by acquiring and manually labelling 2030 images of plastic bottle piles. The labelling consisted of drawing bounding boxes around the pickable objects and attaching one of the four class labels to each such object. As this is a long and monotonous labour, synthetic data generation is implemented.

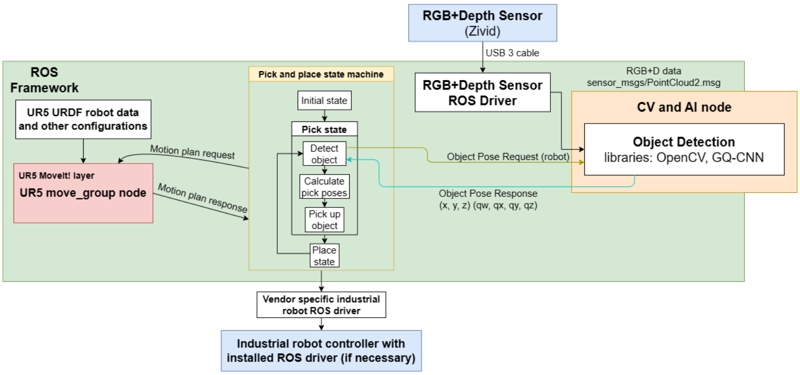
Due to a high amount of variations in object shape and transparency of the object, the approach for estimating the grasp pose was different than in the previous demonstrator and steps described for rigid objects in Section 3.1. Overall, the main task in grasp pose estimation is determining the object's orientation and depth position. As the objects are transparent, the depth readings are rather ambiguous, and the object can become invisible to the 3D camera from different viewpoints. Also, due to the deformation of objects and possible holes, the grasping of such objects is limited to parallel gripping.

To grasp a detected object, the industrial robot needs to get precise 3D information. Unfortunately, transparent bottles are sometimes also transparent for depth cameras. To tackle this challenge for grasp planning, a narrow beam acoustic sensor was attached to the robotic gripper. By tracking the detected object while moving the industrial robot, it can be placed directly above the bottle. In such a position, the acoustic sensor aims directly under the gripper, giving reliable information about the distance to the object beneath, even if the depth camera does not provide complete information about the object of interest. Even though the grasping process includes additional functionality to grasp transparent plastic bottles, by utilizing the object tracking and acoustic sensor, the construction of robot motions was done by MoveIt! and the state machine similar to the one in the Section 5.2. Additionally, in order to extend the reach of the robot and decrease cycle times, the throwing functionality was incorporated. The video of the demonstration can be watched by following the link: `https://youtu.be/UFkkfvVdrpI`

## 5.4. Post office parcel sorting

Internet shopping has led to a surge in postal parcels, necessitating efficient parcel sorting in logistic centres. While high throughput sorting lines automate most tasks, manually singulating and placing parcels on conveyor belts remains a bottleneck. Automating this seemingly simple task poses challenges due to the variety of parcel sizes, shapes, and colours, requiring a flexible computer vision algorithm. The mix of soft and hard objects also presents mechanical design challenges. The parcels are piled and occluded, necessitating careful picking order consideration. A demonstration is presented in this section, using a Universal Robot UR5 to automate the pick-and-place operation. AI-based grasp detection ensures precise handling of arbitrarily shaped objects without additional training. The system integrates easily within any workflow using the ROS framework and addresses the mentioned challenges, emphasizing the reusability of robot control modules and the integration of different sensory data processing packages.

The proposed robotic system for post office package handling consists of multiple modules and is based on the ROS framework. It includes a 6-DOF UR5 robot arm with a vacuum gripper, an RGB-Depth sensor (Zivid), and two processing units for computer-vision-based and robotic motion-based computations. The workflow begins with capturing a 3D workspace image with the RGB-Depth sensor and using a grasp quality convolutional neural network (GQ-CNN) Dex-Net 4.0 to find a grasp pose. The pose is then sent to a robot control state machine, which manages the entire process and robot functions like moving to a desired pose, picking, and releasing objects. The robot control is implemented using the MoveIt! motion planning framework, which handles path planning and hardware communication. After successfully executing a pick and place cycle, the process repeats for the next object.




5.3. Fig. Architecture [10].

The described robotic system for post office parcel handling was implemented and validated in a lab environment for multiple scenarios with different numbers of packages in the workspace, as can be seen in the video demonstration at `https://www.youtube.com/watch?`

v=djxrbQZtiKk. The system demonstrated a 100 % success rate in clearing the workspace and successfully handling all packages. However, on average, there was a 4.37 % increase in additional grasps compared to the ideal one attempt per package. The system achieved a 94.1 % successful grasp rate when measuring only one attempt per package. Most of the failed attempts were attributed to grasp poses located on uneven parts where the vacuum gripper could not achieve the required vacuum. This issue could be addressed with a more flexible vacuum cap or increasing the vacuum. In a real use case, a conveyor belt would replace the container for handling the packages.

# 6. CONCLUSIONS

The Thesis addresses the smart control methods for industrial robots, their applications, and learning strategies. The primary aim of the Thesis was to research and develop novel techniques in the fields of efficient data preparation methods, computer-vision-based robot control and knowledge acquisition from humans through demonstrations and therefore improve the deployment of industrial robots for working in dynamic environments and advancing the operability of such systems. Six tasks were defined to achieve the set aim.

**1. To perform a review of the literature on smart control methods for industrial robots.** This task was accomplished in Chapter 1. The literature review acknowledged the current state-of-the-art approaches for the smart control of industrial robots. Different control methods and learning strategies were overviewed in detail, which also led to the identification of current challenges and open issues described in Section 1.4, being a basis for the proposed methods in the Thesis. Even though the methods related to smart control of industrial robots have gone through remarkable developments in the last decade, there still are several challenges and unsolved issues, namely within efficient data preparation methods for computer-vision-based robot control and knowledge acquisition process.

**2. To formalize the concept of smart control of industrial robots.** The task was accomplished during the extensive literature review in Chapter 1. The identification of core functionalities within the smart control of industrial robots was fuelled by the exploration of current trends in robot control, leading to a formalization of the concept of smart control of industrial robots following the principle of "see-think-act", as illustrated in Fig. 1.1. The concept is a broad view of smart industrial robots consisting of such functionalities as perception, high-level instructions and context-aware task execution, knowledge acquisition and generalization, and adaptive planning. Accordingly, not all of the functionalities are addressed within the Thesis, but only the ones that were recognized as having the highest potential to address the challenges and open issues identified in Section 1.4.

**3. To research and develop efficient data preparation methods for computer-vision-based robot control.** This task was accomplished in Chapter 2 by developing a methodology for synthetic data generation and further built on in Chapter 3 accomplishing a computer-vision-based robot control. Synthetic data generation frameworks show promising results in deep learning-based object detection tasks and can not only supplement real data when the variety of real training data is insufficient but also completely replace real data and still perform reliably in real-world settings. The use of synthetic data generation in robotic systems remarkably decreases the required time and manual efforts within the data generation process while sustaining the levels of performance, which confirms Thesis Statement 1. Thus the versatile data that can be synthetically generated can be used for validation purposes, therefore evaluating the system's performance before deployment. The tentative results can show the viability of the proposed algorithms in the specific use cases, and if the achieved results are not satisfactory, the conclusion of the need for improvements or additional training can be achieved without running the system in real life. The achieved results of computer-vision-based robot control prove Thesis Statement 2.

**4. To research and develop a methodology for human demonstration incorporation in the knowledge acquisition process.** The human demonstration incorporation in the knowledge acquisition process, namely imitation learning-based control, was accomplished in Chapter 4. The original goal set out to accomplish was to devise a framework for recording demonstrations by human actors, and using these to execute an object-throwing task as a stand-in for a variety of similar future applications was achieved. By using motion capture as the data collection mechanism, it proved possible to employ a series of analytic pre-processing steps to turn raw recording data into demonstration data sets suitable for the knowledge acquisition process for industrial robots. The data collection step with motion capture equipment proved to be straightforward to customize for the throwing task. Whereas the precision of learned trajectories achieves high similarity to the validation dataset and closely resembles throwing motions as executed by human experts proving Thesis Statement 3.

**5. To apply the proposed methods in demonstrations.** In total, the proposed methods were applied in three demonstrations described in Chapter 5 addressing current challenges in various robotic grasping applications. In the first demonstrator: "Multi-robot collaboration for the bin-picking task", the synthetic data generation and grasp pose estimation methods proposed have been successfully applied and demonstrated for reliable grasping and sorting of different simple-shaped, rigid objects mixed in a pile. The second demonstrator, "Handling of waste plastic bottles", however, applied the data generation of deformable objects for computer-vision-based robot control and imitation-learning-based robotic object throwing learned from human demonstrations described in Chapter 4. Last but not least, the demonstrator: "Post office parcel sorting", shows the modularity and the reusability of the developed robot control modules and the viability of integrating different sensory data processing packages.

**6. To draw conclusions about the results of this Thesis.** The final task of the Thesis is done in the current chapter. The main conclusion about smart control methods for industrial robots developed in this Thesis is that their properties and test results demonstrate that the aim of this Thesis is successfully achieved – the developed methods improve the deployment of industrial robots for working in dynamic environments and advancing the operability of such systems. In addition, the accomplished tasks prove three statements defined at the beginning of the Thesis.

# Bibliography

[1] "Businesses in the manufacturing sector." https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Businesses_in_the_manufacturing_sector. Accessed: 2023-08-04.

[2] "Population structure and ageing." https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Population_structure_and_ageing. Accessed: 2023-08-04.

[3] "Employment and social developments in europe 2023, european commission, 2023." https://ec.europa.eu/social/BlobServlet?docId=26989&langId=en. Accessed: 2023-08-04.

[4] J. Arents, R. Cacurs, and M. Greitans, "Integration of computervision and artificial intelligence subsystems with robot operating system based motion planning for industrial robots," *Automatic Control and Computer Sciences*, vol. 52, no. 5, pp. 392–401, 2018.

[5] J. Arents, V. Abolins, J. Judvaitis, O. Vismanis, A. Oraby, and K. Ozols, "Human–robot collaboration trends and safety aspects: A systematic review," *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, 2021.

[6] J. Arents and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *Applied Sciences*, vol. 12, no. 2, p. 937, 2022.

[7] P. Torres, J. Arents, H. Marques, and P. Marques, "Bin-picking solution for randomly placed automotive connectors based on machine learning techniques," *Electronics*, vol. 11, no. 3, 2022.

[8] P. Racinskis, J. Arents, and M. Greitans, "A motion capture and imitation learning based approach to robot control," *Applied Sciences*, vol. 12, no. 14, 2022.

[9] V. Feščenko, J. Ārents, and R. Kadiķis, "Synthetic data generation for visual detection of flattened pet bottles," *Machine Learning and Knowledge Extraction*, vol. 5, no. 1, pp. 14–28, 2023.

[10] O. Vismanis, J. Arents, K. Freivalds, V. Ahluwalia, and K. Ozols, "Robotic system for post office package handling," *Applied Sciences*, vol. 13, no. 13, 2023.

[11] E. Buls, R. Kadikis, R. Cacurs, and J. Ārents, "Generation of synthetic training data for object detection in piles," in *Eleventh International Conference on Machine Vision (ICMV 2018)*, vol. 11041, p. 110411Z, International Society for Optics and Photonics, SPIE, 2019.

[12] J. Arents, B. Lesser, A. Bizuns, R. Kadikis, E. Buls, and M. Greitans, "Synthetic data of randomly piled, similar objects for deep learning-based object detection," in *Image Analysis and Processing – ICIAP 2022*, (Cham), pp. 706–717, Springer International Publishing, 2022.

[13] D. Duplevska, M. Ivanovs, J. Arents, and R. Kadikis, "Sim2real image translation to improve a synthetic dataset for a bin picking task," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–7, IEEE, 2022.

[14] J. Arents, M. Greitans, and B. Lesser, *Construction of a Smart Vision-Guided Robot System for Manipulation in a Dynamic Environment*, pp. 205–220. 09. 2021.

[15] G. Urlini, J. Arents, and A. Latella, *AI in Industrial Machinery*, pp. 179–185. 09. 2021.

[16] K. Wei and B. Ren, "A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved rrt algorithm," *Sensors*, vol. 18, no. 2, p. 571, 2018.

[17] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.

[18] V. Kakani, V. H. Nguyen, B. P. Kumar, H. Kim, and V. R. Pasupuleti, "A critical review on computer vision and artificial intelligence in food industry," *Journal of Agriculture and Food Research*, vol. 2, p. 100033, 2020.

[19] M. Fujita, Y. Domae, A. Noda, G. Garcia Ricardez, T. Nagatani, A. Zeng, S. Song, A. Rodriguez, A. Causo, I.-M. Chen, *et al.*, "What are the important technologies for bin picking?," *Advanced Robotics*, vol. 34, no. 7–8, pp. 560–574, 2020.

[20] B. Horn and K. Ikeuchi, "The mechanical manipulation of randomly oriented parts," *Scientific American – SCI AMER*, vol. 251, pp. 100–111, 08 1984.

[21] J. A. Marvel, K. Saidi, R. Eastman, T. Hong, G. Cheok, and E. Messina, "Technology readiness levels for randomized bin picking," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pp. 109–113, 2012.

[22] K. Kleeberger, R. Bormann, W. Kraus, and M. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, p. 239–249, 12 2020.

[23] A. Attia and S. Dayan, "Global overview of imitation learning," *arXiv preprint arXiv:1801.06503*, 2018.

[24] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *arXiv preprint arXiv:1811.06711*, 2018.

[25] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *International Journal of Intelligent Robotics and Applications*, vol. 3, no. 4, pp. 362–369, 2019.

[26] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141, IEEE, 2018.

[27] M. Edmonds, F. Gao, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, and S.-C. 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)*, pp. 3530–3537, IEEE, 2017.

[28] Cognilytica, "Data engineering, preparation, and labeling for AI," 2019.

[29] C. Éric Noël Laflamme, F. Pomerleau, and P. Giguère, "Driving datasets literature review," 2019.

[30] J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo, "Openml: networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.

[31] "Systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models.." `https://iso25000.com/index.php/en/iso-25000-standards/iso-25010`. Accessed: 2021-04-17.

[32] "Systems and software engineering – systems and software quality requirements and evaluation (square) – measurement of system and software product quality." `https://www.iso.org/standard/35747.html`. Accessed: 2021-04-25.

[33] "Ros-industrial." `https://rosindustrial.org/about/description/`.

[34] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5628–5635, IEEE, 2018.

[35] J. S. Dyrstad, E. R. Øye, A. Stahl, and J. R. Mathiassen, "Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual reality," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7185–7192, IEEE, 2018.

[36] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125, IEEE, 2018.

[37] "Schmalz." https://www.schmalz.com/en/vacuum-technology-for-automation/vacuum-components/vacuum-generators/vacuum-generators-end-of-arm/handling-sets-ecbpi-308297/10.03.01.00504/. Accessed: 2023-04-17.

[38] "Ur5 collaborative robot arm." https://www.universal-robots.com/products/ur5-robot/. Accessed: 2023-04-17.

[39] M. Quigley, "Ros: an open-source robot operating system," in *IEEE International Conference on Robotics and Automation*, 2009.

[40] "Smacc – state machine asynchronous c++." https://smacc.dev/. Accessed: 2023-04-17.

**Jānis Ārents** was born in 1994 in Ērgļi. He received a professional Bachelor's degree in Electrical Engineering in 2017 and a professional Master's degree in Electrical Engineering in 2018 from Riga Technical University. Since 2016, he has been an electronics engineer, later a scientific assistant at the Institute of Electronics and Computer Science, where he is currently a researcher, contributing to various national and European projects. He was involved in the development of the "Smart robot with advanced vision, sensing, and human gesture understanding capabilities", which was evaluated by the Latvian Academy of Sciences as one of the most significant scientific achievements in Latvia in 2022. His scientific interests include smart robotic systems and their usage in the automation of various processes.