University of Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Master i ingeniørfag robotteknologi og signalbehandlings	Vårsemesteret 2023 Åpen
Forfatter(e): Robin Liebert, Yuriy Yurchenko	
Fagansvarlig: Karl Skretting Veileder(e): Karl Skretting	
Tittel på bacheloroppgaven: Maskinlæring Uten Simulering: Rask Policy-Læring Gjennom Imitation Learning og Reinforcement Learning i Løfteoperasjoner Engelsk tittel: Machine Learning Beyond Simulation: Fast Policy Learning through Imitation and Reinforcement Learning in Lifting Operations	
Studiepoeng: 30	
Emneord: Machine learning Imitation Learning Reinforcement Learning Artificial Neural Networks Liquid Time Constant NN Robotics Lifting Operations Edge Computing	Sidetall: 36 + 1 GitHub Repository Stavanger 14. juni 2023

Abstract

In this research, the implementation and evaluation of a novel learning approach for an autonomous crane operation called LOKI-G (Locally Optimal search after K-step Imitation - Generalized) using Closed-form Continous-time (CfC) Artificial Neural Network (ANN) was explored. The study revolved around addressing the Sim-to-real gap by allowing the model to learn on edge with minimal examples, mitigating the need for simulators. An emphasis was placed on creating a sparse, robust, reliable, and explainable model that could be trained for real-world applications.

The research involved five experiments where the model's performance under varying conditions was scrutinized. The model's response under baseline conditions, sensory deprivation, altered environment, and object generalization provided significant insights into the model's capabilities and potential areas for improvement.

The results demonstrated the CfC ANN's ability to learn the fundamental task with high accuracy, exhibiting reliable behaviour and excellent performance during Zero-Shot Learning. The model, however, showed limitations in regard to understanding depth. These findings have significant implications for accelerating the development of autonomy in cranes, thus increasing industrial efficiency and safety, reducing carbon emissions and paving the way for the wide-scale adoption of autonomous lifting operations.

Future research directions suggest the potential of improving the model by optimizing hyperparameters, extending the model to multimodal operation, ensuring safety through the application of BarrierNet, and adopting new learning methods for faster convergence. Reflections on the importance of waiting during tasks and the quantity and quality of data for training also surfaced during the study.

In conclusion, this work has provided an experimental proof of concept and a springboard for future research into the development of adaptable, robust, and trustworthy AI models for autonomous industrial operations.

Acknowledgements

We wish to express our deep and sincere gratitude to those individuals and institutions whose contributions have been pivotal to the completion of this master's thesis. Foremost among these is the University of Stavanger, whose rigorous academic environment and extensive resources have enabled our pursuit of knowledge and research. We also appreciate UiT The Arctic University for laying a robust foundation that has fortified our work on this thesis.

Our profound thanks go to our academic supervisor, Karl Skretting, whose invaluable guidance and steadfast support have shepherded us throughout this research journey.

We also extend our warm appreciation to our industrial partners. Their willingness to share their expert knowledge, coupled with their support, has greatly enriched our understanding of the subject matter. The opportunity to collaborate with them and gain practical insights from their industry experience has been invaluable.

Furthermore, we would like to express our gratitude to the researchers at MIT CSAIL, especially Mathias Lechner and Ramin Hasani. Their groundbreaking work and contributions have served as a source of inspiration and have made it possible to perform this research.

Finally, we want to thank our families, friends, and loved ones for their unwavering support, encouragement, and understanding throughout this academic journey. Their belief in us and patience have been a necessity to accomplish everything we have.

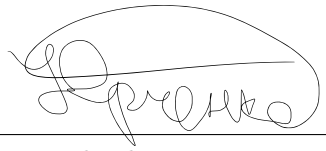
This thesis stands as the culmination of the collective efforts and support of numerous individuals and institutions. We are deeply humbled and grateful for the guidance, encouragement, and contributions provided by all those who have been part of this enriching journey.

Preface

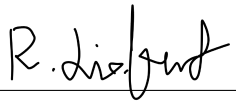
It is expected that the reader has some basic knowledge regarding various available sensors and their capabilities, is familiar with programming and has some understanding of how physical systems are modelled. No prior knowledge of Lifting Operations or Fast Policy Learning is necessary.

This report is the original intellectual product of the authors, Yuriy Yurchenko and Robin Liebert. Since the project has been done in collaboration with external partners, the groups contribution has been clearly identified and stated. All outside work used for research or citations has been referenced.

Stavanger, 14.06.2023



Yuriy Yurchenko



Robin Liebert

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CfC	Closed-form Continuous-time
IL	Imitation Learning
LTC	Liquid Time Constant
ML	Machine Learning
MLP	Multi-Level Perceptron
NCPs	Neural Circuit Policies
PPG	Phasic Policy Gradient
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
SOTA	State-of-the-Art

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traditional control of robotic systems, primarily through rule-based programming and explicit instructions, has proved inadequate in managing the growing complexity and dynamics of real-world tasks. The promise of an alternative lies in the advancements of Machine Learning (ML) and Artificial Intelligence (AI), which automate the decision-making processes in robotics. This shift towards ML and AI is particularly evident in sectors such as manufacturing, logistics, healthcare, and services, where they are driving higher levels of automation [3]. This automation revolution, empowered by the rise of sensor technology and digitization, is transforming industry operations [23].

Within Machine Learning, two particular areas of research that have shown promising results are Imitation Learning (IL) and Reinforcement Learning (RL). Imitation Learning allows robots to learn tasks by observing and mimicking human actions. While promising, it often has difficulty adapting to new and complex situations and is limited by the ability of the agent it is trying to learn from. On the other hand, Reinforcement Learning involves learning through trial and error by receiving rewards and punishments [27]. However, it suffers from long training times and the accuracy of its learning depends on how well the "reward function" is constructed [1].

To improve upon these limitations and push the boundaries of what is currently possible in robotic control, this thesis presents a novel approach that combines Behaviour Cloning (a branch of Imitation Learning) and Phasic Policy Gradient (a method within Reinforcement Learning). Specifically, in the context of crane operations, this thesis investigates a modified version of the LOKI (Locally Optimal search after K-step Imitation) algorithm [5], utilizing a Closed-form Continuous-time (CfC) Artificial Neural Network (ANN) [12]. The adoption of these algorithms and networks is anticipated to yield a model that not only advances the field of robotics control but also delivers practical solutions for automating heavy industrial applications. The end goal is to create efficient and explainable models for edge devices, addressing a pressing need in the industry, and thus driving forward the frontier of ML-enabled automation.

1.1 Motivation

This thesis is inspired by the substantial challenges associated with the training of deep neural networks, which is traditionally a time-consuming and computationally intensive process. The pursuit of more efficient learning strategies and neural network architectures has led to the examination of several pioneering works in the field.

The paper "Fast Policy Learning through Imitation and Reinforcement" [5] (2018) proposes an innovative strategy called "LOKI". This strategy utilizes the strengths of both Imitation Learning (IL) and Reinforcement Learning (RL) for more effective control of robotic systems. Notwithstanding its potential, the original LOKI algorithm operates as a randomized learning method, alternating between online IL and RL after a random number of steps. Therefore, this thesis proposes an

enhancement to the LOKI algorithm, modifying it for use with offline IL. This adaptation promises improved hyperparameter tuning and model architecture optimization over the collected training examples.

Additionally, attention is drawn to the transformative work of Ramin Hasani and Mathias Lechner at TU Wien and MIT. Their research, which emulates the neural structure of the nematode Caenorhabditis Elegans, introduced the concept of Liquid-Time Constant (LTC) neural networks [14] that are designed to model continuous-time systems. These networks employ a unique activation function that possesses a liquid-like property (due to varying time constants), thus facilitating faster and more efficient training, as well as improved accuracy on data structured as time series. This groundbreaking concept enables the development of models for machine control that bypass the inherent black box properties of deep learning.

Furthermore, the publication "Closed-form Continuous-Time Neural Networks" [12] (2022) introduces Closed-form Continuous-time (CfC) artificial neural networks (ANN). This revolutionary concept enables the bypass of numerical differential equation solvers, accelerating the learning process. The CfC ANN is also inherently explainable, a feature that is particularly advantageous for visual tasks like lifting operations, the primary focus of this thesis. The specific use case in this thesis requires a camera sensor for object localization and general awareness. Coupling the CfC ANN with a Convolutional Neural Network (CNN) for feature generation allows for the creation of a heatmap of the incoming image stream, enhancing the model's explainability. This visual representation aids in determining whether a model is ready for production and assists in fault detection should the model exhibit unwanted behaviour.

The ambition of this thesis is to synthesize these influences to engineer a model architecture that strikes a balance between accuracy and size, addressing limitations associated with edge computing and memory constraints. This goal is increasingly important given growing concerns about the environmental footprint associated with training and utilizing large models on the cloud [19]. As such, the thesis advocates for the deployment of compact models, trained on minimal data, directly on the machines they operate (known as edge devices), as an environmentally conscious alternative.

While the proposed approach is applicable to various types of heavy machinery, lifting operations have been selected as a relevant case study. The operations' inherent visual dimension within a 3D space allows for tangible observation of the model's output, thus providing an engaging context for the application and evaluation of the proposed methods.

1.2 Research Objective

Based on what this thesis aims to contribute, the research objective can be worded as follows:

> The primary objective of this research is to determine the potential of integrating a generalized version of the 'LOKI' fast policy learning method with Closed-form Continuous-time Neural Networks for developing an efficient, robust and transparent solution for real-world lifting operations using minimal training examples.

To accomplish the primary objective, the following list of **secondary objectives** has been identified:

- Analyzing the open-source frameworks for performing IL and RL in an industrial setting, with a focus on the eligibility of the LOKI algorithm and CfC ANN.
- Adapting the LOKI algorithm to offline IL and state-of-the-art (SOTA) RL.
- Creation of a custom environment for performing lifting operations. Create logging tools for State-Action pairs used in offline IL and reward function for RL.

- Evaluate the performance of the model when performing the lifting operation, focusing on risk minimization, efficiency and explainability.

The overall aim is to explore the feasibility of training small neural networks on edge devices that are suited for controlling heavy machinery. The validation of this approach could provide a more environmentally friendly alternative to the increasing reliance on large neural networks in the cloud, thus expanding the scope of machine learning applications in heavy machinery while reducing their environmental impact.

By focusing on a lifting operation as a case study, this research intends to provide insights into the performance of this approach in a real-world setting. The insights derived from this study would inform future research on optimizing machine learning strategies for industrial applications.

If the proposed approach is validated as a robust, efficient and transparent method for creating controllers for use in industrial machines, the general goal is to democratize the use of machine learning in heavy machinery. This would expands the possibility of what tasks could be automated and lower the cost of programming control systems.

1.3 Significance and Potential Impact

The primary aim of this study is to create a robust, efficient, and transparent learning algorithm that can learn directly from real-world examples. This investigation, drawing inspiration from significant works on Imitation Learning, Reinforcement Learning, liquid-time constant networks, and closed-form continuous-time neural networks, aspires to amalgamate the strengths of these methodologies, thus proposing a novel solution to various challenges inherent in machine learning for industrial automation.

One of the main challenges that this study seeks to address is the performance gap often observed when training a model in a simulated environment before its deployment in the real world. Simulation environments, while valuable for initial model training and safety, often fail to fully capture the complexity and variability of real-world situations. As a result, models trained in these environments do not perform as well when transferred to real-world tasks, requiring additional fine-tuning or even a complete retraining process. By learning directly from real-world examples, the proposed algorithm aims to mitigate this performance gap. For the user this would offer a seamless transition from training to deployment with minimal effort.

Another significant aspect of this study is its potential to democratize AI, particularly for industrial manufacturers. Currently, the development of machine learning models often requires substantial resources, including the creation of digital twins or simulation environments, which can be prohibitive for many smaller companies or those without substantial budgets or human resources in the field. By proposing an approach that minimizes the need for these expensive and time-consuming steps, this study could lower the barrier to entry for these companies. This would enable them to initiate pilot projects and incorporate AI into their operations without the need for substantial upfront investment in digitalization.

The use of the CfC ANN adds an element of explainability to the model at the core of the algorithm. This is particularly relevant in an industrial context, where understanding the behavior of the model could be critical for safety and efficiency.

Overall, this study is intent on designing a learning algorithm that stands out not only in terms of robustness and efficiency but also in its transparency and accessibility to a diverse range of industrial manufacturers. By utilizing advanced techniques in machine learning and neural network architectures, focusing on training with real-world examples, and prioritizing reliability, safety, and explainability in the model design, this work could bring about substantial transformations in the domain of industrial automation.

Chapter 2

Literature Review

This chapter discusses general theoretical principles relevant to this thesis, specific related studies and their influence on our approach. There are various ways to utilize machine learning techniques in the control of robotic manipulators, which also hold promise for crane control scenarios. These approaches fall into three main categories: Imitation Learning, Reinforcement Learning, and Interactive Learning. While each method and its subsets have their own advantages and drawbacks, they all share the capability to tackle problems that are formulated as sequential decision-making processes.

2.1 Background and Overview

Before developing a specific approach for machine learning-enabled crane control, a thorough understanding of available machine learning techniques is vital. The recent surge in the use of neural networks, largely enabled by increasingly accessible and user-friendly machine learning frameworks such as TensorFlow and PyTorch, has significantly influenced this field. A majority of these algorithms are trained, evaluated, or both, within simulated environments, which although convenient and safe for initial model training, present challenges in the transfer of learned models to real-world applications. These challenges, often termed as the 'sim-to-real' gap, are a significant point of consideration in this work and will be discussed in greater detail in the Discussion section of this thesis (see Chapter 5).

The main focus of this work involves the application and validation of the LOKI algorithm and the Closed-form Continuous-time (CfC) Artificial Neural Network (ANN), which include Neural Circuit Policies (NCPs). Therefore, this section will provide a comprehensive introduction and background to these techniques. In order to create an ANN capable of functioning effectively in a cyber-physical environment, understanding principles of generalization and Zero-shot learning is crucial. These principles will be discussed initially, followed by an overview of existing approaches. The application of these principles and methods to the current study will be explored in later sections of the thesis.

2.2 Generalization

Generalization in machine learning refers to a model's ability to make accurate predictions or take suitable actions based on input data that it has not encountered during training. For machine control, this ability is critical. Real-world environments are incredibly diverse and unpredictable. It is impossible to expose a model to all potential scenarios during training. Hence, models must generalize from the training data to unseen situations to work effectively in real-world applications.

A machine learning model that can generalize well can handle new scenarios that it encounters, increasing the utility and safety of the machine control system. For instance, an autonomous vehicle

trained on a specific set of traffic scenarios must still be able to operate safely when presented with a situation not included in its training data.

However, the challenge is to balance a model's ability to generalize without overfitting to the training data. Overfitting occurs when a model learns the training data too closely, including the noise or outliers, and performs poorly on unseen data. A well-generalized model avoids overfitting, capturing the underlying patterns in the training data without being overly influenced by noise or outliers.

2.3 Zero-shot learning

Zero-shot learning takes generalization one step further. It refers to a model's ability to handle tasks or make decisions about which it has received no explicit training. This capability is particularly vital in control systems because it allows these systems to extend their operations beyond their training environments or tasks.

Zero-shot learning is particularly useful for resource-constrained systems. Training machine learning models on all possible scenarios is requires significant computational resources and time. For many scenarios it would not even be safe to provide a demonstration. If a model can effectively perform tasks without explicit training, this could lead to substantial resource savings.

This means that if the model performs well on zero-shot learning metrics it allows for greater adaptability. In real-world applications, machines may need to handle tasks or scenarios that were unforeseen during model development. For instance, a crane may need to interact with a new type of object or navigate in a previously unexplored environment. With zero-shot learning capabilities, the crane can still function effectively in these scenarios.

In summary, generalization and zero-shot learning are crucial for machine control systems, enabling them to handle diverse and unpredictable real-world scenarios. They increase the system's utility, safety, adaptability, and efficiency, making them more robust and reliable [16].

2.4 Imitation Learning

Imitation Learning (IL) is a supervised machine learning technique. In IL, an algorithm tries to optimize a controller based on expert examples. There are various methods to implement IL, with offline and online IL being the most common.

Offline IL uses Behaviour Cloning (BC), which leverages a set of state-action pairs logged from an expert. Despite its simplicity and wide usage, BC has notable limitations. These include never exceeding the performance of the expert, potential bias in the training set, poor generalization in dynamic environments, and susceptibility to covariate shifts in data distribution [7].

In contrast, online IL has access to the expert policy during training, allowing for faster convergence time and higher accuracy. It has also shown efficiency against covariate shifts with algorithms like DAgger [22]. However, online IL's need for interaction with an expert often restricts its usage primarily to simulated environments [31].

Interactive Imitation Learning (IIL), a field that intersects with Interactive Machine Learning, closely mimics human learning patterns. The algorithm learns from demonstrations and receives performance feedback from the expert. This focused learning method improves areas needing enhancement and reduces random loss-searching, making it a promising approach [4].

2.5 Reinforcement Learning

Reinforcement Learning (RL) is a distinct learning approach based on trial and error. RL requires a reward function to evaluate the actions taken, with the algorithm performing random actions until

it identifies a pattern that maximizes rewards and minimizes penalties. Historically, RL required numerous experiments to recognize desirable behavior. However, given enough time, it tends to exceed human performance [9].

The randomness of actions during the training period has historically limited the use of RL in scenarios where simulator training is not possible due to the potential for damage and costly training time [18].

Two primary RL practices are model-free and model-based. Model-free RL associates an observation with an action, while model-based RL stores the dynamics of the model/system it optimizes against. This model can either be known to the algorithm from the start (known model) or learnt during the trial and error phase (learnt model) [21].

2.6 Hybrid Approaches

One solution for combining both Imitation Learning (IL) and Reinforcement Learning (RL) is the Locally Optimal search after K-step Imitation (LOKI) algorithm, originated from Georgia Tech's School of Interactive Computing [5]. LOKI is designed for pre-training a neural network using IL before employing RL to perform and refine a given task. The algorithm uses first order methods in both phases, however, the oracles for estimating the gradients (g_n) differ.

During IL, g_n is an estimate of equation 2.1, where $\tilde{c}$ is the surrogate loss and the per-round cost is defined as $ln(\pi) = \mathbb{E}d_{\pi_n}\mathbb{E}_{\pi}[\tilde{c}]$.

$$\nabla_\theta ln(\pi_n) = \mathbb{E}d_{\pi_n} \left(\nabla_\theta \mathbb{E}\pi\right) [\tilde{c}]|\pi = \pi_n \tag{2.1}$$

During the RL phase, the gradient g_n is an estimate of $\nabla_\theta J_n(\pi)$, as expressed in equation 2.2. Here, γ denotes the discount factor, and $\mathbf{A}$ represents the (dis)advantage function.

$$(1-\gamma)\nabla_\theta Jn(\pi)|\pi = \pi_n = \mathbb{E}d\pi_n \left(\nabla_\theta \mathbb{E}\pi\right) [\mathbf{A}d_{\pi_n}]|_{\pi=\pi_n} \tag{2.2}$$

LOKI randomly samples a number $K \in [\mathbf{N_{min}}, \mathbf{N_{max}}]$ using the probability distribution given by:

$$P(K = n) = \frac{n^d}{\sum_{m=N_m}^{N_M} m^d} \tag{2.3}$$

Subsequently, it performs online IL using mirror descent for K iterations before transitioning to Trust Region Policy Optimization (TRPO). By selecting a random K iterations for IL training to create a sub-optimal expert and not expecting expert performance, the need for training examples during the IL phase is reduced, significantly reducing the computational resources and time required for training. While there is proof of the sub-optimal experts performance in the paper introducing LOKI, this was in the specific setting of using first order oracles and online IL. Other research in the field shows that IL algorithms in self-supervised tasks has demonstrated resilient behaviour to the use of sub-optimal experts compared to optimal experts. For offline IL the sub-optimal experts has exhibited better generalization properties than expert policies [11].

Algorithm 1 LOKI

Parameters: d, N_m, N_M
Input: π^*
Sample K with probability in equation 2.3.
for $t = 1 \ ... \ K$ **do** ▷ Imitation Phase
    Collect data D_n by executing π_n
    Query g_n from equation 2.1 using π^*
    Update π_n by mirror descent with g_n
    Update advantage function estimate $\hat{A}_n$ by D_n
end for
for $t = K + 1 \ ... \ \infty$ **do** ▷ Reinforcement Phase
    Collect data D_n by executing T_n.
    Query g_n from equation 2.2 using $\hat{A}_\pi$
    Update π_n by mirror descent with g_n
    Update advantage function estimate $\hat{A}_n$ by D_n

2.7 Artificial Neural Network design

Many existing approaches are benchmarked against each other in simulations, using standardized artificial neural network (ANN) designs based on multi-layer perceptrons (MLPs). The introduction of new techniques provides opportunities for improving the sample efficiency, interpretability, and robustness of ANNs. These factors are highly valued when the algorithm is implemented in production. This research employs the current state-of-the-art artificial neurons and wiring for robotics. Following is a brief introduction.

2.7.1 Closed-form Continuous-time Neural Networks

In our research, we employ an innovative, state-of-the-art neural network model, the Closed-form Continuous-time Neural Network (CfC). Developed at the MIT Computer Science and Artificial Intelligence Lab (CSAIL) [12], the CfC advances traditional neural networks by offering notable improvements in speed and efficiency.

The CfC's main strength lies in its ability to handle sequential decision-making problems by formulating them as differential equations. This approach significantly differs from conventional discrete-time models, setting the CfC apart with its unique approach of modeling systems' changes over time, also known as dynamical system representation. As a result, the CfC excels in tasks that involve simulating complex physical dynamics, such as lifting operations. It has also demonstrated its superiority over Transformers, a popular machine learning model known for its performance in tasks involving sequential data, by an impressive 18% margin while drastically reducing computational overheads. Moreover, the CfC utilizes only a tenth to half the time per epoch compared to Transformers, emphasizing its superior computational efficiency.

A defining feature of the CfC is its use of Liquid Time-Constant Networks (LTCs) [14]. Originally designed for tasks that require a temporal relationship between input and output, LTCs have been recognized for their capability to model intricate systems that change over time. However, the conventional implementations of LTCs have faced limitations due to the necessity of a numerical solver to handle the differential equations inherent in the model, limiting the scalability of LTC-based networks.

Addressing these limitations, the CfC introduces an ingenious solution by approximating the exact, or closed-form, solution of LTCs. This significant advancement substantially accelerates the network, achieving between one to five orders of magnitude faster execution during both training and inference compared to ordinary differential equation (ODE)-based continuous networks. As a result, the CfC effectively eliminates the scalability bottleneck seen in previous models, offering a

more robust solution for controlling industrial machines.

Given the remarkable scalability, efficiency, and performance of the CfC, it paves the way for novel applications of machine learning models, particularly in edge devices. This makes the CfC an excellent choice for tasks such as lifting operations.

2.7.2 Neural Circuit Policies

Neural Circuit Policies (NCPs) were utilized to structure the connections of the CfC neural network. NCPs present a novel and efficient approach for establishing connections within neural networks, significantly enhancing both the interpretability and efficiency of the model. The fundamental concept behind NCPs is the emulation of the tap-withdrawal neural circuit found in the nematode Caenorhabditis elegans, in which a mere 302 neurons manage all of the nematode's motor functions. This innovation was pioneered by researchers at the Technische Universität Wien (TU Wien), who also made notable contributions to machine learning in the robotics field with the development of the CfC artificial neural network [20].

Adopting strategies from nature contributes to a model that is more interpretable and offers various practical benefits. One significant advantage of the NCPs structure is that it requires fewer weights than conventional designs. This reduced complexity translates into less memory consumption on the hardware, rendering NCPs a more efficient solution. This efficiency is particularly beneficial in environments with limited computational resources or in scenarios where the goal is to reduce environmental impact.

To better understand the unique architecture of NCPs, consider a simplified wiring diagram as shown in Figure 2.1. This example comprises a network of ten neurons, including two output neurons, and provides a visual representation of the unique NCPs structure. Not only does the diagram highlight the design, but it also demonstrates how this layout facilitates differentiable properties. Essentially, differentiable properties enable the model to learn and adjust itself over time, and in this layout, each neuron's state at a given moment is influenced by its state in the previous time-step. This is clearly seen with the neuron at the top in Figure 2.1. Thus, incorporating NCPs enhances the efficiency and interpretability of neural networks, making it particularly beneficial for Artificial Neural Networks (ANNs) in the Liquid Time-Constant (LTC) family, such as the CfC used in our work.

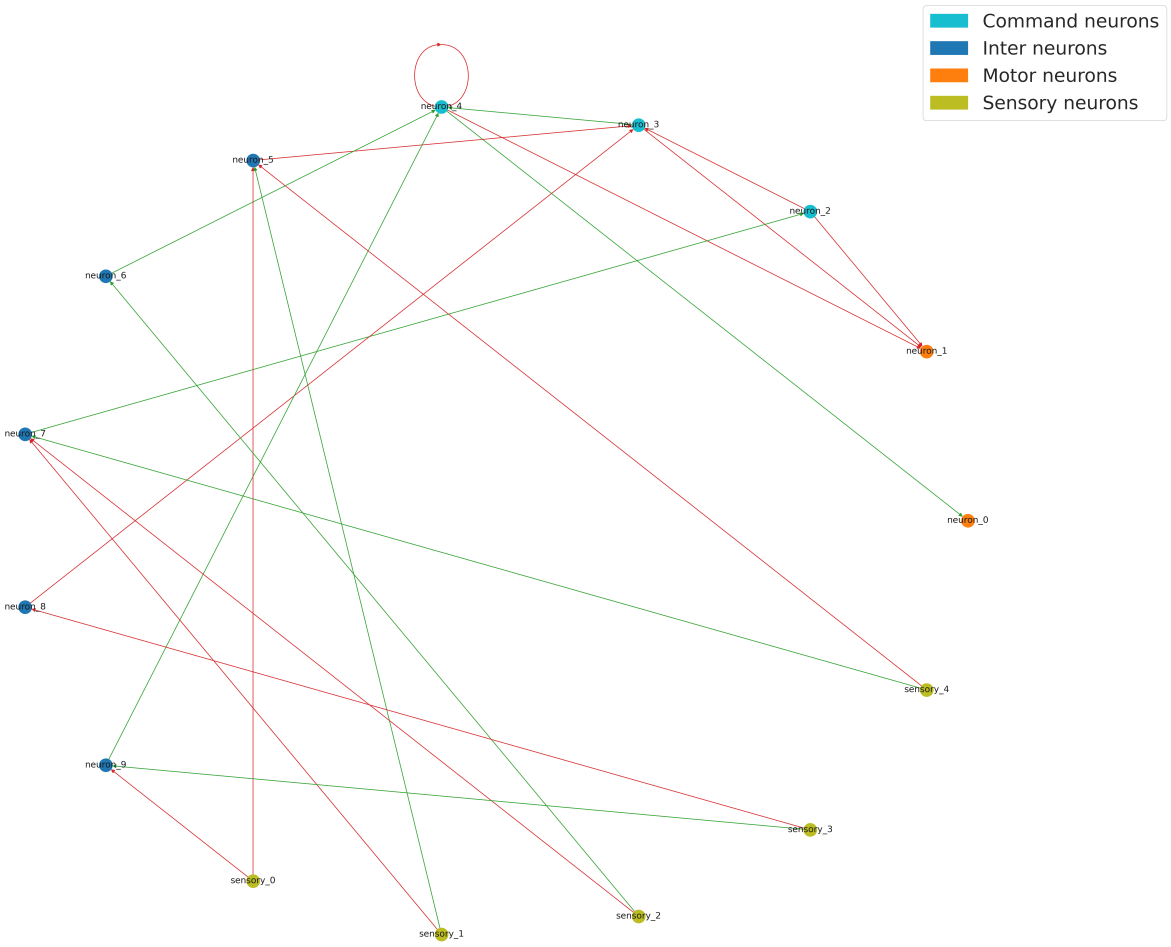


Figure 2.1: Illustration of ANN following NCPs design

Chapter 3

Methodology

This chapter aims to describe the methods utilized in this study, as well as the setup for the experiment.

3.1 Hardware and Software

Describe the hardware and software setup we will be using: robotic arm, computer for data collection and analysis, software for programming and controlling the arm, and software for training the imitation learning model.

3.1.1 Hardware

This section describes the physical setup for the experiment performed for this thesis. The hardware used in the experiment consists of a robotic arm, a camera, and a computer system.

The core hardware components for the experimental setup comprise the following:

- **Robotic Arm**: The experiment utilized an ABB IRB 140 robotic arm, renowned for its compactness and high payload capacity. Equipped with six joints and a single arm, the ABB IRB 140 offers flexibility in movement and object manipulation. For the purposes of this study, the robot arm was outfitted with a rope and a hook to facilitate the movement of various objects.
- **RGB Camera**: The robotic system was augmented with a standard RGB camera. The camera served as the robot's primary sensor, capturing images of the robot's workspace and providing visual data that was critical for training the imitation learning model. The camera was strategically placed to capture a clear view of the objects and the designated placement zone in the robotic arm's environment.
- **Computer System**: A computer system was employed to manage data collection, analysis, and training of the learning model. In addition, the computer served as the primary interface for programming and controlling the robotic arm's operations.

3.1.2 Software

The following software tools were leveraged to support the experimental execution and data analysis:

- **Python and TensorFlow**: Python was the primary programming language for this experiment, chosen for its readability and extensive support for scientific computing. The deep learning model was implemented using TensorFlow, a Python library offering comprehensive capabilities for machine learning and deep learning.

- **Robot Studio**: The RobotStudio Augmented Reality Viewer enables you to visualize robots and solutions in a real environment. It was used to upload the necessary modules to the robot arm, which allowed the control to be a blend of Python files communicating with RAPID code.
- **OpenAI Gym**: OpenAI's Gym was used as the simulation environment to test the imitation learning model. Gym provides a wide variety of pre-defined environments that simplify the development and comparison of reinforcement learning algorithms.

These hardware and software configurations collectively formed the foundation for the execution of this study's experimental procedures, ensuring an effective environment for model training, testing, and performance evaluation.

3.2 LOKI-G algorithm

The pursuit of effectively utilizing Machine Learning (ML) as a controller in lifting operations presented several crucial considerations. Academic research often places the utmost importance on achieving state-of-the-art performance on benchmarks. However, in the industrial context, aspects such as robustness and up-time are paramount, and a slight performance loss is a tolerable trade-off to enhance these features [15]. An Imitation Learning (IL) / Reinforcement Learning (RL) hybrid approach, termed as the LOKI-G algorithm, emerged as a promising solution, which forms the crux of this section.

3.2.1 Comparative Overview of IL/RL Algorithms

The IL/RL hybrid approach offers a unique pathway to enhancing robustness and reducing down-time. IL, not requiring digital twins or simulators, becomes accessible to all manufacturers, irrespective of their budget or level of digitalization. However, the original LOKI algorithm demonstrated certain limitations. Notably, its use of online IL necessitated a high degree of expert involvement and restricted the algorithm to first-order methods only. This prerequisite not only excluded numerous modern algorithms but also demanded the architecture of the neural network to be finalized before training.

The LOKI-G algorithm is inspired by LOKI in regard to the use of an sub-optimal changing from the IL phase to RL phase after K numbers of training examples sampled randomly. By disregarding the requirement of first-order methods and online IL, the guarantee of convergence is not valid. The advantages is possibility of using offline IL with no involvement from the expert, hence no down-time. While the use of a sub-optimal expert would create a greater disadvantage in offline IL, there are arguments that favor it over an optimal experts in a hybrid approach. IL have shown issues concerning over fitting to the expert, the use of a sub-optimal expert has improved the generalization of the algorithm [11]. As the output of the IL phase is not a final product, higher performance can be sacrificed for a more robust, generalizing model.

3.2.2 Implementation of LOKI-G Algorithm

In the implementation of the LOKI-G algorithm, Behaviour Cloning (BC) is employed as the offline IL algorithm owing to its simplicity and maturity. This, however, doesn't limit the choice of IL algorithms; any efficient offline IL method could be suitably adopted.

By generalizing away from the requirement of first order methods in LOKI, a broader range of applicable RL methods. We use a Single-Network PPG (Phasic Policy Gradient) in our experiments due to the high sample efficiency and lower memory use than PPG [6]. When introduced PPG used PPO (Proximal Policy Optimization) [24], while our implementation uses the newer truly-PPO [29]. Originally LOKI used TRPO (Trust Region Policy Optimization) for the RL phase, PPG is two

generation newer which is the reason it is used in our implementation. There no limitation to the choice of model-free RL algorithms using our approach, but the use of model-based RL has not been tested.

As for the architecture, instead of the standard Multi-Layer Perceptrons (MLPs) used by the original LOKI, the Closed-form Continuous-time (CfC) Artificial Neural Network (ANN) is implemented, as introduced in 2.7.1. This architecture, structured following the Neural Circuit Policies (NCPs) framework (2.7.2), treats the optimization problem as a differentiable equation. This approach makes the ANN more interpretable and delivers better generalization properties with compact ANNs [12].

In the LOKI-G implementation, an ANN of the CfC type is employed. This architecture adheres to the NCPs framework, making the ANN interpretable and resulting in better generalization properties. The architecture is designed to be larger than necessary to explore if this would induce delay during real-time control. If used, the Tensorflow framework may issue warnings, which can be safely ignored as the value function layer will be trained during the RL phase. The architecture of this ANN as used in the experiments is shown in Figure 3.1.

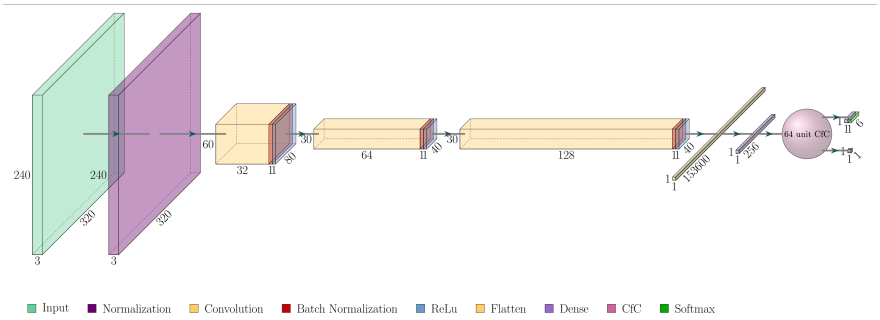


Figure 3.1: Artificial Neural Network architecture during experiments

Figure 3.1 illustrates the ANN architecture implemented in the LOKI-G algorithm during the experiments. The structure includes two output values where only the logits are trained during the IL phase, with the value function layer being trained during the RL phase.

3.2.3 Guidelines for LOKI-G Algorithm Implementation

As our approach is a more general implementation of LOKI, we have named it LOKI-G Ẇhile LOKI had three parameters (d, N_m, N_M), it's advised to set $N_m = \frac{N_M}{2}$ [5]. To increase simplicity of implementation it is no parameters in LOKI-G outside designing the ANN. N_M is defined to the number of training examples, $N_m = \frac{N_M}{2}$ and $d = 2$. To fully take advantage of using offline IL it's advised to use a hyperparameter tuner for setting the size of the ANN, this step is not made a part of the LOKI-G algorithm as memory constraints could make it infeasible in some applications. As there is no recommended number of examples, this is something that should be adapted to the task. It is recommended to approach the gathering of examples with the same mindset as with ML in general, by diversifying the examples and keep a majority of the expected normal use case.

3.2.4 LOKI-G Pseudocode

The following pseudocode provides a summary of the LOKI-G algorithm:

Algorithm 2 LOKI-G

Input: $\mathcal{D}$

Sample K with probability in equation 2.3.
for $t = 1 \ ... \ K$ **do** ▷ Imitation Phase
    Collect random example D_t from $\mathcal{D}$
    Perform offline IL to optimize π
end for
for $t = K + 1 \ ... \ \infty$ **do** ▷ Reinforcement Phase
    Collect data D_t by executing π on the environment.
    Update π according to chosen RL algorithm

3.3 Experimental Design

The experimental design consists of two main stages: (1) data collection and behaviour cloning model training, and (2) further training using Single-Network Phasic Policy Gradient (PPG).

3.3.1 Data Collection and Behaviour Cloning Model Training

The initial phase of the experimental design centred around the implementation of a custom Python script to commandeer a robotic arm equipped with a rope and hook. The robot arm, under this setup, was engineered to accomplish a specific task - lifting and relocating objects to a designated area, distinctively marked with a white colour scheme.

To facilitate data acquisition for subsequent model training, a strategically positioned camera was installed adjacent to the robot arm. The camera's function was twofold: capture visual input synchronously with the robot's operations and foster a more interactive and intuitive environment for the human operator.

During the data collection phase, a human operator manually manipulated the robot arm via the Python script, undertaking a variety of prescribed actions. While the human operator's performance was inevitably sub-optimal, this approach served a crucial function. Each action executed by the operator prompted the camera to capture corresponding frames. These visuals, coupled with action data, were meticulously catalogued to formulate the training dataset for the behaviour cloning model.

The behaviour cloning model, leveraging the power of supervised learning, was trained to emulate the expert demonstrations showcased by the human operator. By mapping observed states (camera frames) to the corresponding actions, the model could effectively learn an initial policy. This process marked the initial stride towards a more autonomous and efficient robotic system, laying a solid foundation for further iterative refinements and learning.

3.3.2 Single-Network Phasic Policy Gradient (PPG) Training

Following the initial training of the behaviour cloning model, the second phase of the experiment centred on honing the model through Single-Network Phasic Policy Gradient (PPG) training. This is a reinforcement learning algorithm designed to refine the rudimentary policy acquired from the behaviour cloning model, enabling the robotic arm to engage and assimilate lessons from its real-world environment actively.

PPG was judiciously selected for its intrinsic merits, most notably stability and sample efficiency. These traits are paramount in the realm of real-world robotic systems, where a high degree of precision is expected, and the opportunity for repeated trial and error is limited. In essence, PPG serves as a mechanism for the model to navigate its environment, extract valuable insights from its experiences, and continually evolve its policy.

In more detail, the PPG training phase commences with the behaviour cloning model as a foundational blueprint. The model, equipped with this preliminary understanding of the task at hand, proceeds to explore various strategies within the bounds of this understanding. This exploration allows the model to discern successful strategies from the less effective ones, informing its policy updates.

Each iterative cycle of exploration, learning, and policy update under PPG contributes to the model's maturing proficiency in lifting operations, bolstering its overall performance. This fine-tuning phase is vital for the model's transition from a novice learner mimicking human operators to an adept learner capable of independent decision-making based on past experiences.

3.4 Evaluation Criteria

The effectiveness of the imitation learning model was gauged through a combination of qualitative and quantitative evaluation criteria, designed to offer a comprehensive perspective on the model's performance.

3.4.1 Trajectory Analysis

The foremost evaluation criterion was the trajectory of the robot arm in performing the given task. This was qualitatively analyzed by comparing the movement paths generated by the model with the expert demonstrations and expected trajectory. The primary goal was to evaluate the model's ability to replicate the human operator's actions accurately and consistently.

3.4.2 Task Completion

Task completion served as a crucial quantitative measure. This was assessed by verifying if the robot arm could successfully move the object to the designated location. Metrics included the number of successful task completions and the rate of task completions over a set number of trials.

3.4.3 Generalization Capability

Generalization was another significant evaluation criterion. This was measured by subjecting the model to previously unseen scenarios or configurations, such as a changed object location or a different object type. The model's performance in these scenarios was analyzed to assess its ability to generalize the learned behaviours and adapt to novel conditions.

3.4.4 Sensitivity to Environmental Changes

An essential aspect of real-world robotic operations is the model's sensitivity to changes in the environment. Trials like the "Moved Sheet" and "No Camera" (further described in Section **??**) were specifically designed to test this attribute. Here, the evaluation metric was the model's ability to adjust its behaviour according to changes in the environment.

3.4.5 Stability of Learning

The stability of the learning process was evaluated by monitoring the model's performance across different training epochs. A stable learning process would indicate a consistent improvement in the model's performance with increasing training, reflected in the form of converging loss values.

Through these evaluation criteria, we aim to assess not only the model's immediate performance but also its potential for effective application in diverse, real-world scenarios.

Chapter 4

Results

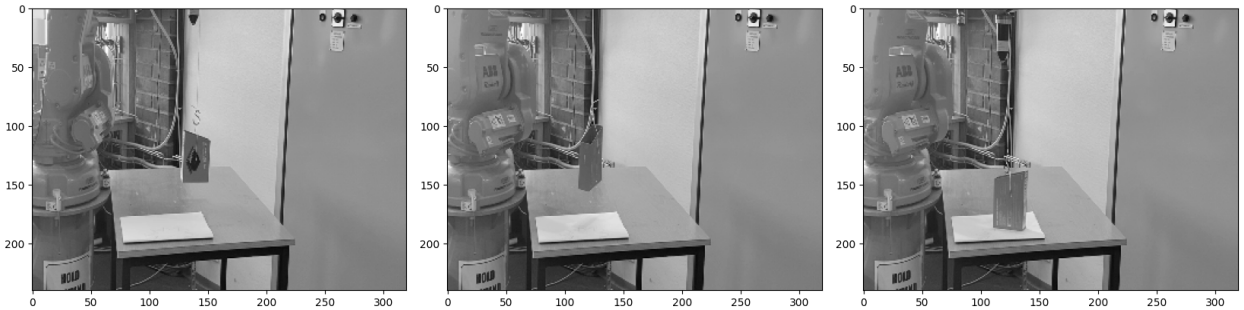
The overall experiment has been performed in two parts, first, the Manual Control run was performed to collect the data necessary for Behaviour Cloning (BC). The second part involved running the Single-Network Phasic Policy Gradient (PPG) algorithm for minor, incremental improvement of the BC. All of the code used for the experiments can be found in the repository for the project [1].

4.1 Behaviour cloning

4.1.1 Manual Control and Data Acquisition

During the manual control stage, an operator controlled the robotic arm to perform the desired object manipulation. Each movement was carefully performed, taking into consideration the task requirements and the limitations of the physical setup. An observation vector containing snapshots of 3 images for each action, along with an action vector containing all control commands were logged and saved for later use in the behaviour cloning model. This data, also referred to as expert demonstration data, comprises a sequence of state-action pairs that represent the operator's actions and their corresponding outcomes.

Figure 4.1 illustrates the initial, middle, and final configurations of the experiment. Image (a) Start displays the robot arm at the initial position (Z coordinate of 500) with an object attached to its hook. Image (b) Mid shows the robot arm in a halfway state where it has moved from its initial position but hasn't reached its final position. Lastly, in image (c) End, the robot arm is shown to have successfully placed the object on the white sheet.



(a) Start (b) Mid (c) End

Figure 4.1: Progress of a Manual Control run

The motion of the robot arm as controlled by the human operator can be visualized in Figure 4.2. The figure represents a 3D plot of the path traced by the arm in the Cartesian space during the

[1] Repository for the project: https://github.com/R-Liebert/LOKI-G

task execution.

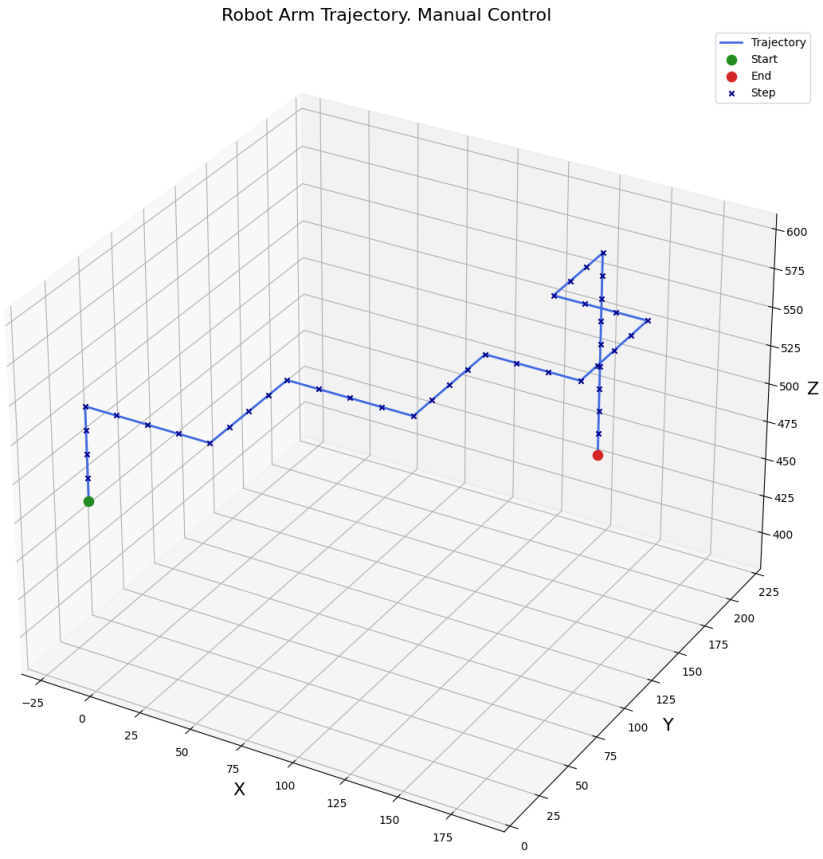


Figure 4.2: Trajectory of the Manual Control run

Each axis in the figure corresponds to one dimension of the workspace: X and Y represent the horizontal plane of the table, while Z corresponds to the vertical elevation from the table surface. The Start point of the path corresponds to the arm's initial position, while the End point of the trajectory marks the arm's final position when the task is complete. Each cross corresponds to a discreet command that moves the arm by a magnitude of 15 in one of the 6 directions.

4.1.2 Model Training

The BC model was trained on the expert demonstration data obtained from the manual control stage. The aim of this model was to emulate the operator's control over the robot arm, enabling the arm to perform the object manipulation task autonomously.

The results of the BC were evaluated in the early stages of the PPG algorithm as it became apparent that the performance of BC didn't undergo almost any changes at that point.

4.1.3 Model Intuition

To visualize the models intuition a saliency map highlights the most important regions of the input. The saliancy map overlayed the observation in Figure 4.3 is from a random observation with the BC model.

Figure 4.3: Saliency map of a random observation using BC model.

As can be seen in the Figure, a lot of the model's attention is on the landing area (table), the robot arm and the bottom right corner. Ideally, through running PPG, the attention should shift away from areas that bear no relevance to the final result of the trial.

4.2 Single Network Phasic Policy Gradient Result

The PPG part of the experiments has been performed across 5 different trials. Each trial's relevance is tied to various aspects of the model's performance, adaptability, and generalization capabilities, all of which are important for validating the current approach and demonstrating its potential in real-world lifting operations.

4.2.1 Trial Run 1: Original (Baseline)

Conditions

The conditions described in the methodology chapter are applied in this experiment. No changes have been made to the setup of the environment.

Relevance

The initial experiment serves as a foundational reference, establishing the baseline performance of the robotic arm under standard operating conditions. This baseline is crucial as it facilitates comparative analysis of the results derived from subsequent experimental trials, thereby enabling a comprehensive understanding of the impact of various modifications on the model's performance and adaptability. Such a comparative analysis is instrumental in validating the efficacy of our proposed approach.

Data

Figure 4.4 illustrates the trajectory followed by the robot arm in the baseline trial run.

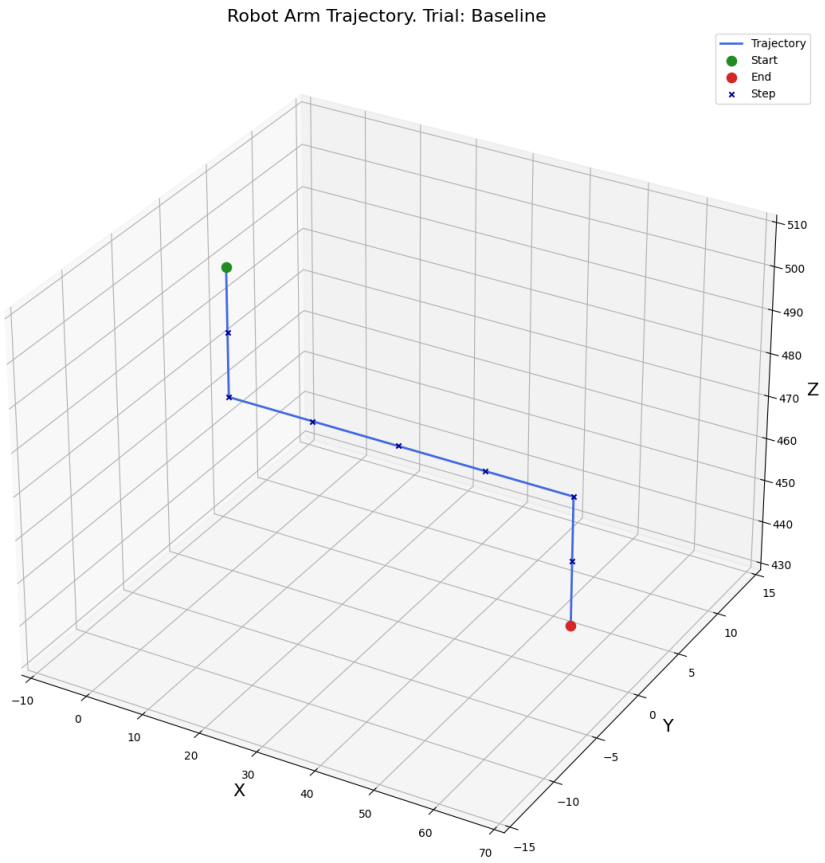


Figure 4.4: Robot Arm Trajectory. Trial: Baseline

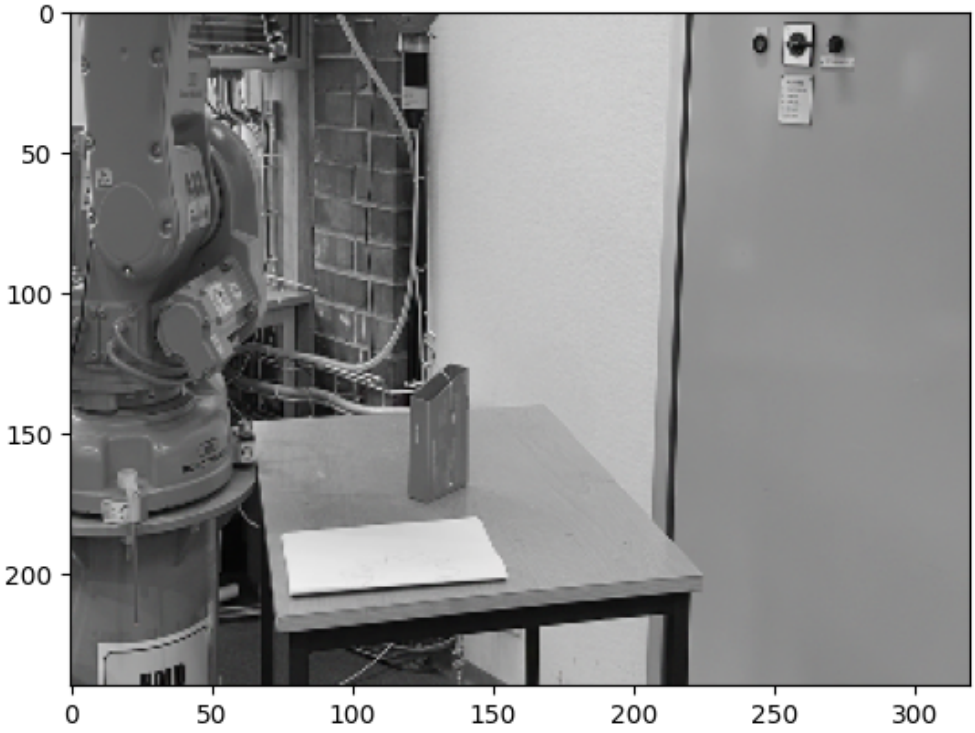


Figure 4.5: Endpoint position. Trial: Baseline

The arm's movement path is not a clear reflection of the state-action pairs learned during the behaviour cloning phase. While the robot arm recognised its task of placing the object down on the table, it didn't capture the significance of the white sheet as the designated placement zone.

The path originates at the Start point and moves towards a different Endpoint than expected. The final position of the object can be seen in Figure 4.5. The X coordinate at the end of the manual control run was 125 ± 15, whereas the baseline for PPG is 60. The Y axis is not recognised as significant as the robot arm does not move along it. The only correct estimation was the Z coordinate.

4.2.2 Trial Run 2: Moved Sheet

Conditions

The conditions only differ from the baseline in that the white sheet (designated placement zone) has been moved.

Relevance

This trial probes the model's adaptability to environmental shifts, achieved by repositioning the white sheet. This alteration challenges the model's ability to discern the importance of the designated placement zone, and subsequently, adapt its operational behaviour to the new location. This adaptability is paramount in ensuring that the model can efficiently perform under real-world conditions, where environmental consistency is not guaranteed.

Data

Figure 4.6 presents the robot arm's trajectory during the second trial, where the white sheet was relocated. Notably, despite the environmental change, the robot arm's trajectory remains largely unaltered from the baseline trial.

This path suggests that despite the model's training, it has not recognized the significance of the white sheet as the designated placement zone. Instead, the robot arm follows a trajectory similar to the one in the baseline trial, moving towards an Endpoint distinct from the new location of the white sheet.

The final position of the object can be seen in Figure 4.7. Exactly as in the baseline trial, the robot arm continues to neglect movement along the Y-axis. The final X coordinate in this trial mirrors that of the baseline trial. The Z coordinate, which was correctly estimated in the baseline trial, remains accurate in this trial run as well.

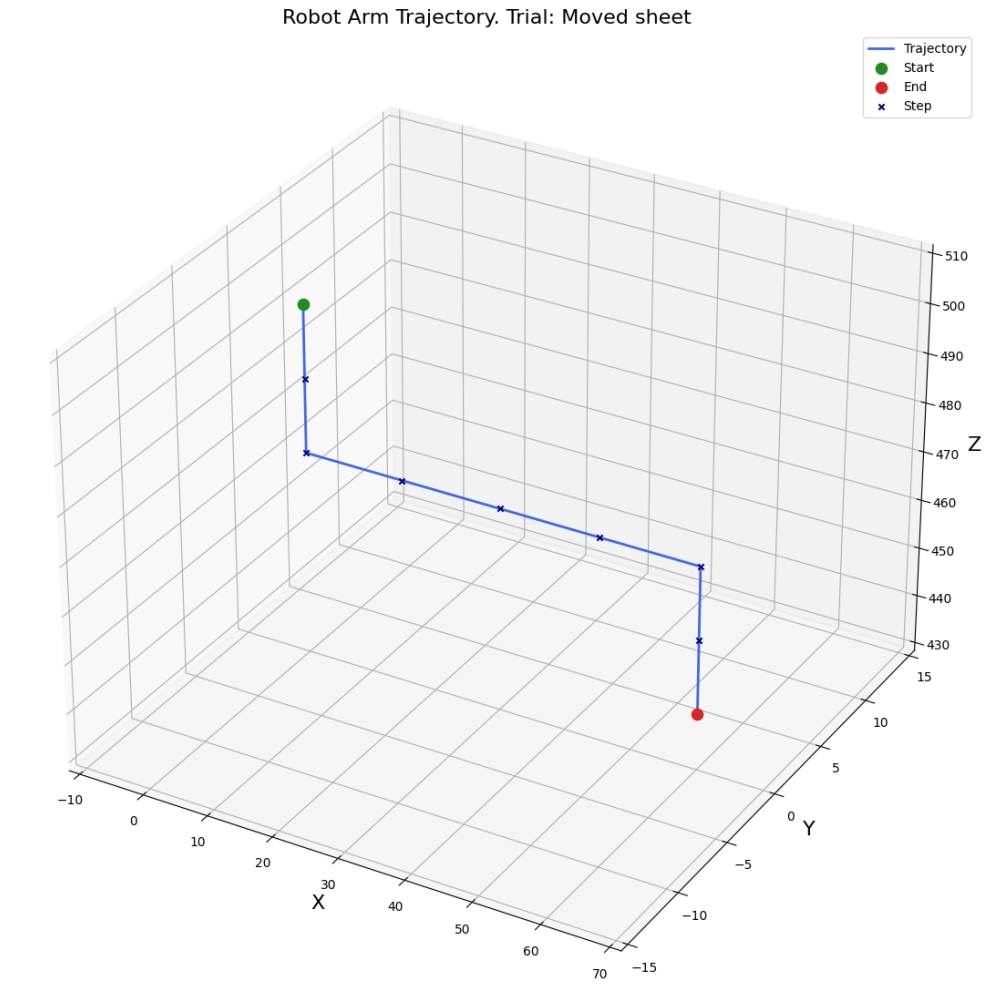


Figure 4.6: Robot Arm Trajectory. Trial: Moved Sheet

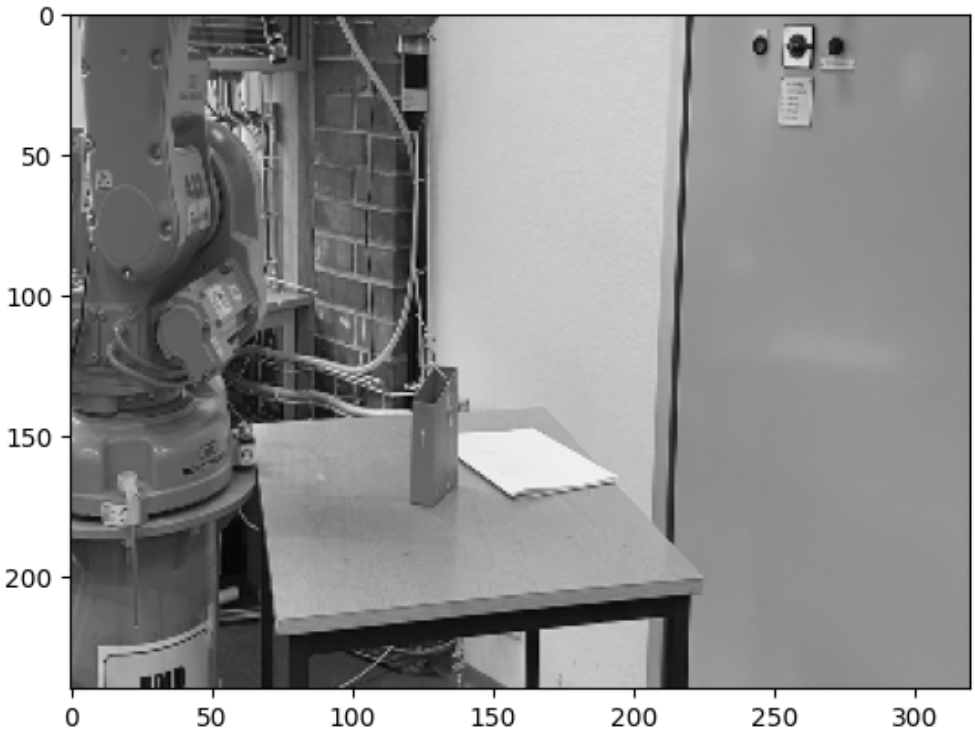


Figure 4.7: Endpoint position. Trial: Moved Sheet

4.2.3 Trial Run 3: No Camera

Conditions

The only difference from the baseline is that the camera has been covered so that it only produces images with only black pixels.

Relevance

The experiment underscores the significance of visual inputs within the model's decision-making paradigm. This is manifested by obscuring the camera, thus assessing whether the model's actions are primarily predicated on the visual information or if they are a result of rote replication of manually provided commands. The ability to forge meaningful correlations between visual inputs and corresponding actions is indispensable for efficient learning and generalization. Therefore it is a critical validation of the model's capabilities.

Data

Figure 4.8 shows the trajectory followed by the robot arm in the "No Camera" trial.

In stark contrast to the previous trials, the robot arm displays a significant deviation from the expected trajectory. There is no observed movement along the Z-axis, indicating that the arm does not perform the necessary vertical displacement to lift or lower the object. Instead, the arm exhibits some movement along the Y-axis, a behaviour not seen in the previous trials. Most of the movement, however, is along the X-axis.

The absence of a "Wait" option for the robot arm could potentially explain this unusual behaviour. Without this option, the robot arm could not wait for clear visual input, resulting in it moving primarily along the X and Y-axes.

It's clear from this trial that the robot arm relies heavily on visual inputs to make its decision, as indicated by the dramatic change in the movement trajectory when deprived of these inputs. This shows that the model's ability to map visual information to corresponding actions is a vital part of its operational paradigm.

The results from this trial emphasize the importance of integrating mechanisms that allow the robot arm to adapt and respond appropriately in the absence of clear visual input, a common scenario in real-world applications.

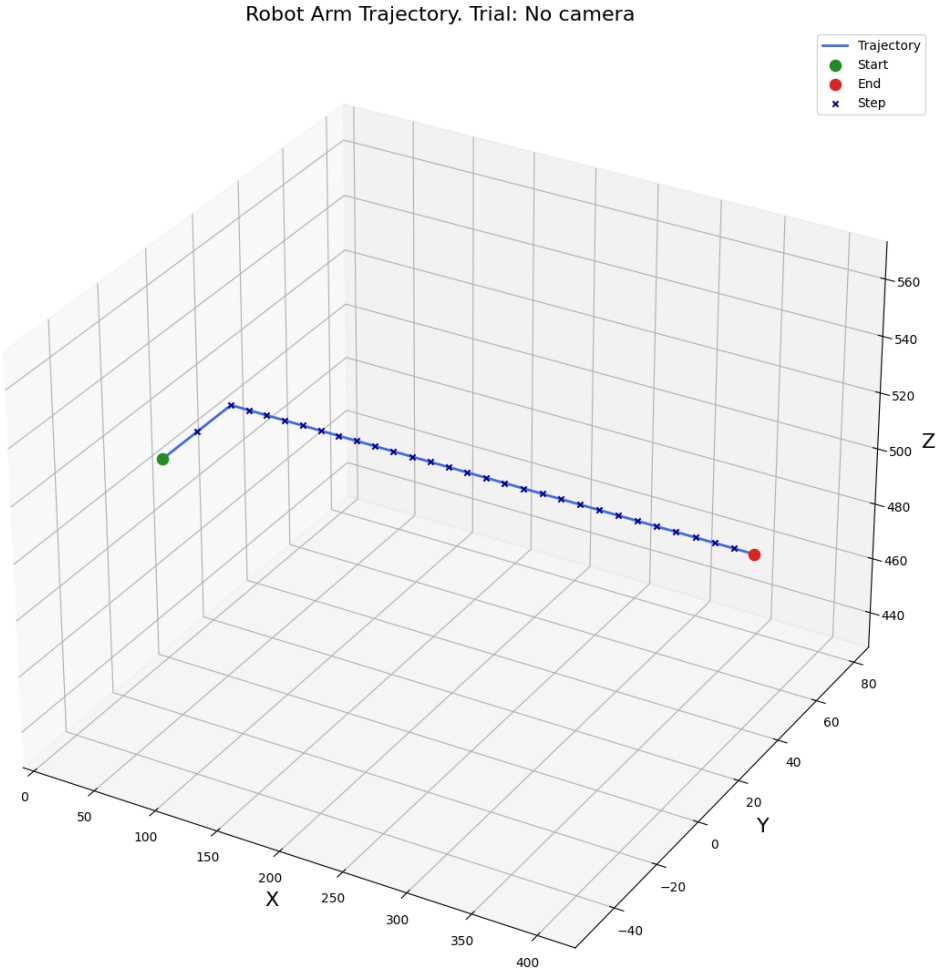


Figure 4.8: Robot Arm Trajectory. Trial: No Camera

4.2.4 Trial Run 4: Short Rope

Conditions

The difference between the baseline and this trial is the reduced length of the rope that is tied to the hook.

Relevance

This trial scrutinizes the model's capacity to accommodate alterations in the physical configuration of the robotic arm. By introducing a shortened rope, the model's ability to recalibrate its actions in response to this change serves as an indicator of its adaptability and robustness. The significance of this experiment is underscored in real-world applications where the crane might be tasked with managing disparate tools or components, necessitating appropriate adjustments in its operational behaviour.

Data

Figure 4.9 depicts the trajectory followed by the robot arm in the "Short Rope" trial.

In this experiment, the rope tied to the hook had a significantly reduced length. In response to this change, the trajectory of the robot arm illustrates an interesting adaptation. The arm appears to move largely in alignment with the baseline trial along the X and Y-axes. However, a crucial adjustment is seen in the Z-axis movement. Recognizing the short rope's limitation, the robot arm compensates by moving lower along the Z-axis than observed in the baseline trial. The final position of the object can be seen in Figure 4.10.

This adaptation indicates that the model is capable of adjusting to changes in the physical configuration of the robot arm. It recognizes the change in rope length and responds by modifying the vertical displacement of the arm to appropriately position the object.

These results highlight the model's adaptability and its ability to recalibrate its operations in response to alterations in the robot arm's configuration. The ability to adjust to such changes is essential for real-world applications where the physical parameters of the task can often vary. This trial demonstrates that the model is capable of such adaptation, which bodes well for its robustness in varied operational scenarios.

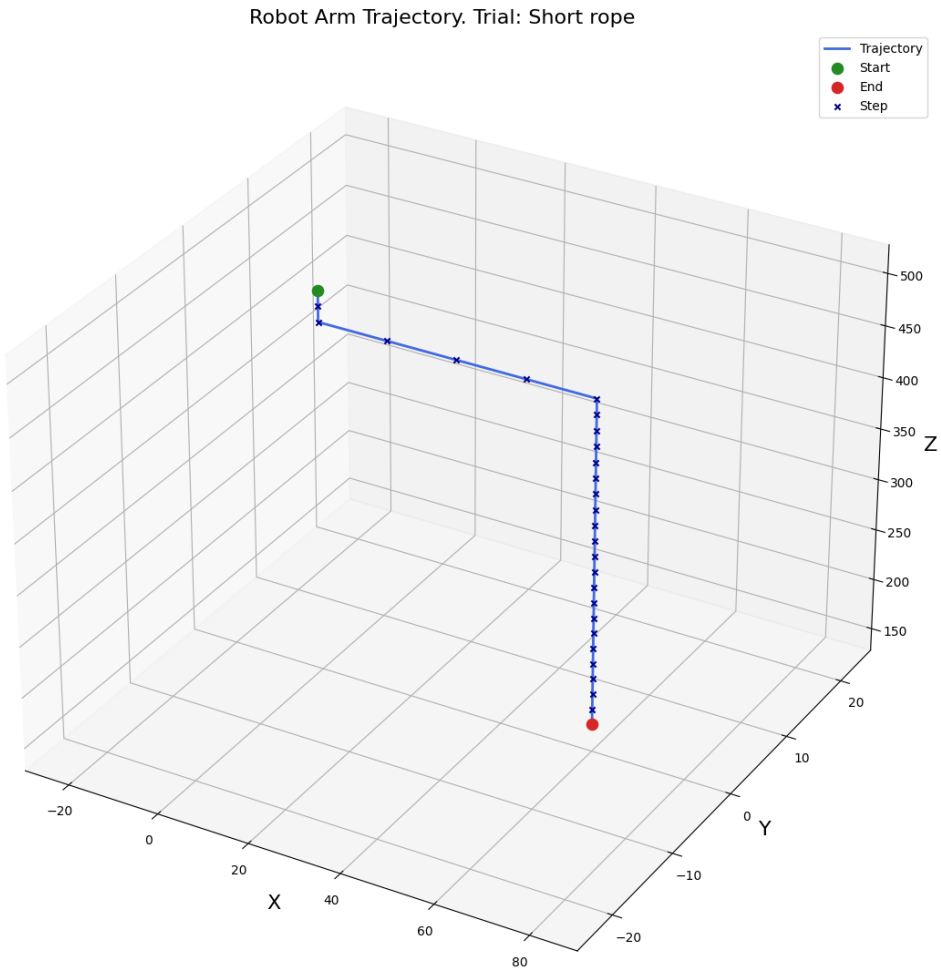


Figure 4.9: Robot Arm Trajectory. Trial: Short Rope

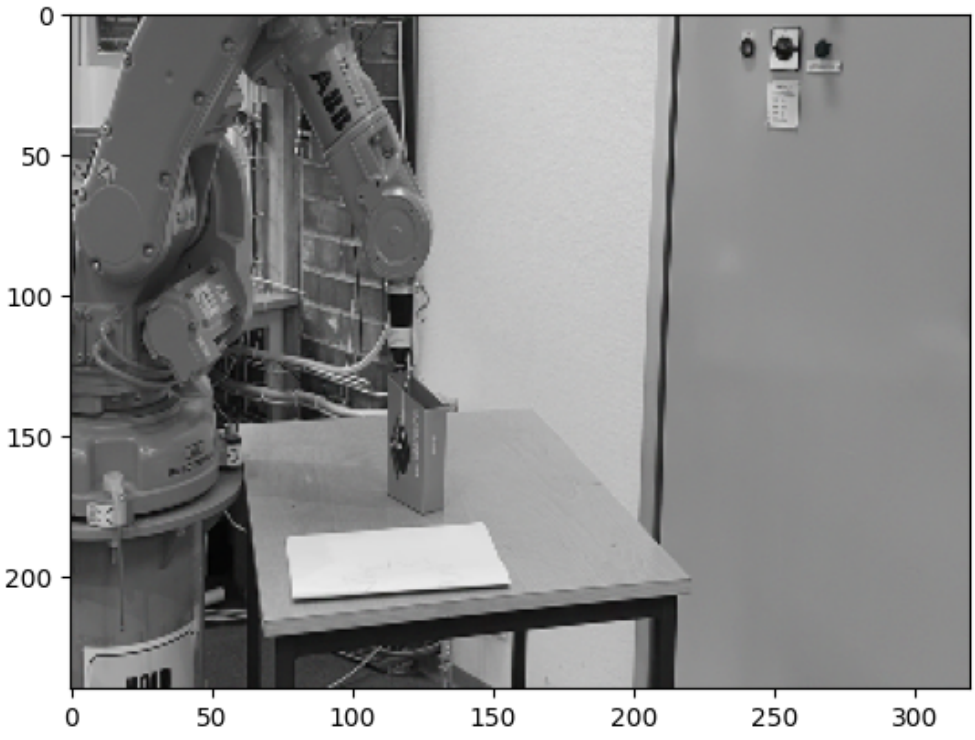


Figure 4.10: Endpoint position. Trial: Moved Sheet

4.2.5 Trial Run 5: Blue Object

Conditions

In this trial, the usual green object was replaced by a blue one with a different shape.

Relevance

This trial challenges the model's ability to manage diverse objects, a critical facet for its applicability across an extensive range of tasks within lifting operations. By introducing an object that differs in both colour and shape and which is previously unseen by the model, this trial evaluates the model's proficiency in successfully executing the task. This assessment is integral to demonstrating the model's capacity to generalize across distinct situations, an aspect that underscores the model's effectiveness and adaptability in a multitude of real-world scenarios.

Data

Figure 4.11 shows the trajectory followed by the robot arm in the "Blue Object" trial.

In response to the change of conditions, the robot arm's trajectory was nearly identical to that observed in the baseline trial along the X and Y-axes. This indicates that the change in the object's colour and shape did not affect the arm's horizontal movement. However, a distinct difference is noticed in the Z-axis movement. Recognizing the need to accommodate for the different shape of the object, the robot arm adjusts by moving lower along the Z-axis to appropriately position the object on the table. The final position of the object can be seen in Figure 4.12.

This demonstrates the model's capacity to adapt to changes in the task conditions. It suggests that the model can recognize differences in object characteristics and adjust its movements accordingly. This is crucial for real-world applications where tasks often involve handling objects of different shapes and sizes.

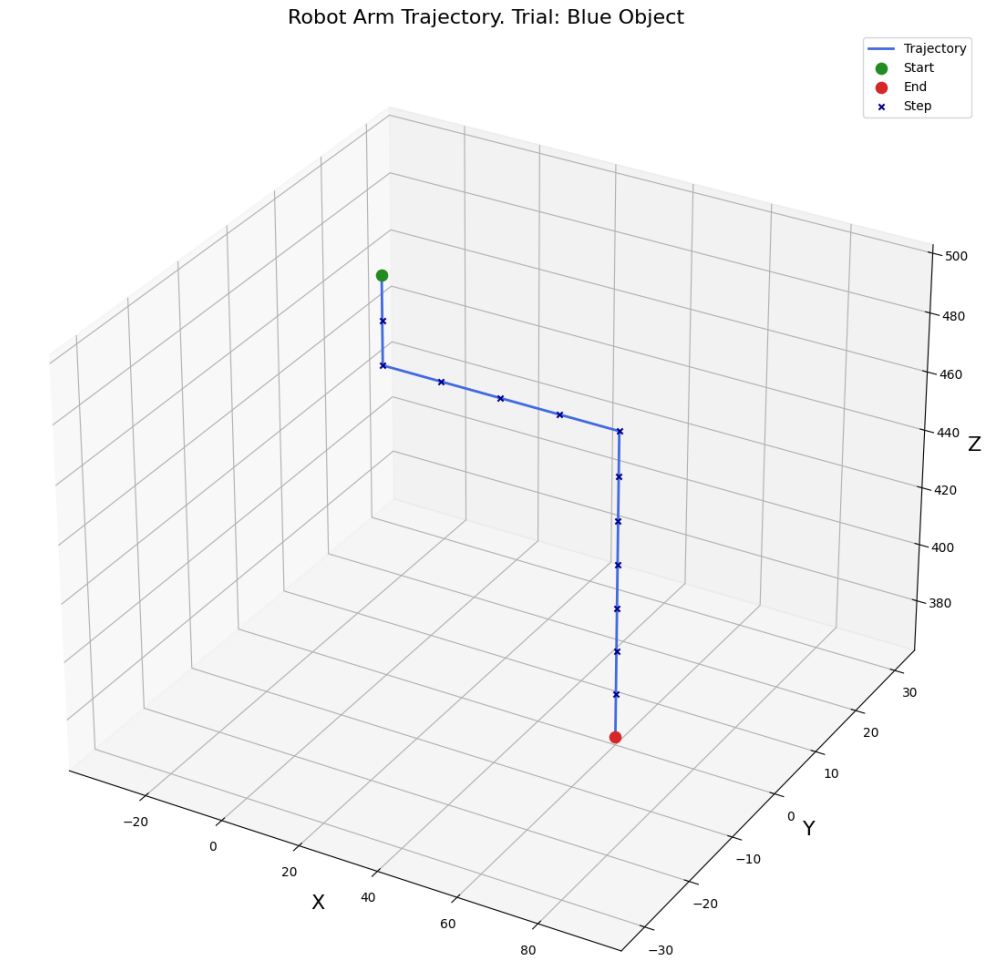


Figure 4.11: Robot Arm Trajectory. Trial: Blue Object

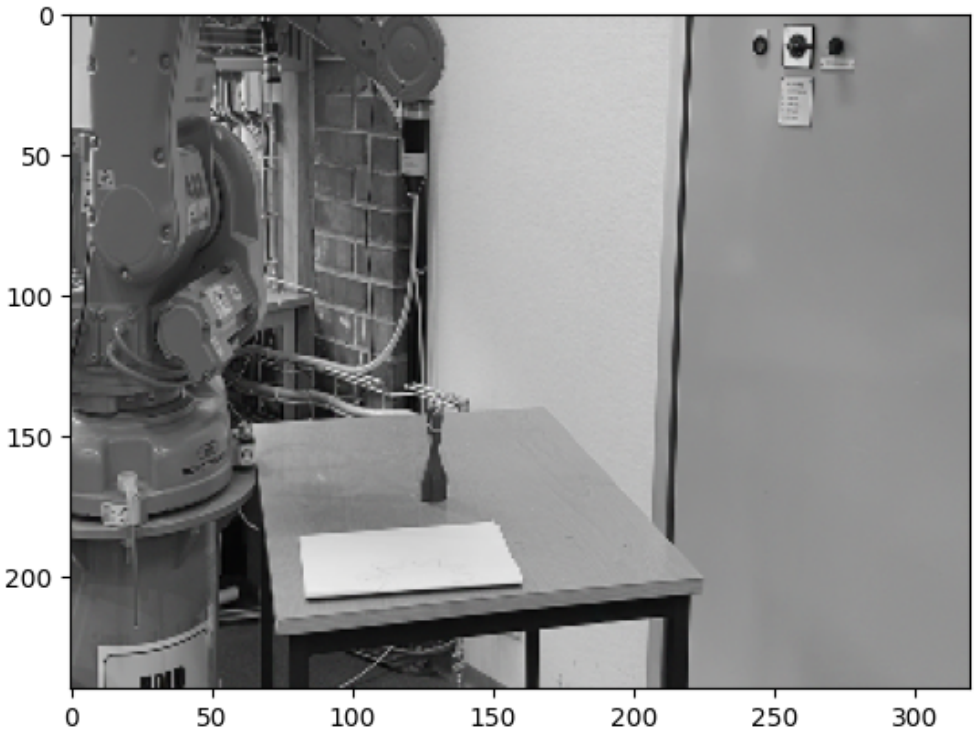


Figure 4.12: Endpoint position. Trial: Moved Sheet

4.3 Neural Network Design

Table 4.1: Behaviour Cloning on full dataset of 18 examples after 10 epochs

Number of Neurons in CfC	CNN type	CNN Size	Training Results
32	CNN with Batch Normalization	Small	Accuracy: 0.9825 Loss: 0.0399
		Large	Accuracy: 0.98 Loss: 0.0363
	Impala-CNN	Small	Accuracy: 0.9850 Loss: 0.0394
		Large	Accuracy: 0.9805 Loss: 0.1406
64	CNN with Batch Normalization	Small	Accuracy: 0.982 Loss: 0.0365
		Large	Accuracy: 0.9815 Loss: 0.0389
	Impala-CNN	Small	Accuracy: 0.9820 Loss: 0.5828
		Large	Accuracy: 0.9815 Loss: 0.6676

The CNN blocks use three layers of tf.keras.layers.Conv2D with kernel size 8, 4, 3 and stride 4, 2, 2. The large blocks has 32, 64, 128 filters, while the small blocks have 16, 32, 32 filters. More information can be found in ConvCfC.py in the GitHub repository.

The large batch normalized CNN has 39 435 392 trainable parameters, while the small has 9 851 456 trainable parameters. The CfC with 64 neurons has 47 028 trainable parameters, while CfC with 32 neurons has 18 960 trainable parameters. In terms of memory a 64 neurons CfC network with a large normalized CNN uses 150.61 MB, while a 32 neuron CfC with a small normalized CNN uses 37.65 MB.

The large Impala-CNN has 9 954 720 trainable parameters and uses 38.15 MB of memory when paired with the large CfC network. The small Impala-CNN has 2 480 272 trainable parameters and uses 9.53 MB of memory when paired with a small CfC network.

Chapter 5

Discussions

This chapter focuses on the analysis and discussion of the results obtained from the experiments.

5.1 Five Trials

Five experiments were conducted, with conditions modified for each trial. The results of each experiment are subsequently discussed, with an emphasis on the properties of the proposed approach that each experiment revealed.

5.1.1 Trial Run 1: Original (Baseline)

In the first trial, the same setup employed during the Imitation Learning (IL) phase was used. This approach aimed to gauge the model's performance against the demonstrations. The worst outcomes considered were randomized actions, suggesting an untrained model, or inaction, indicating the model's inability to generate an output. The model performed similarly to the demonstration, albeit with a slight depth offset in the basket placement. Smooth movement and no unexpected behaviour were observed, indicating the model's understanding of the fundamental task requirements. The model's difficulty in understanding depth, demonstrated by the consistent failure to place the basket on the white paper, supports findings in existing Sim-to-real gap research [17].

5.1.2 Trial Run 2: Moved Sheet

In the second experiment, the white sheet of paper was relocated to a different part of the table. The model did not acknowledge this as a new objective and replicated the behaviour observed in Trial Run 1, placing the basket with a depth offset relative to the white sheet.

5.1.3 Trial Run 3: No Camera

The third trial was characterized by sensory deprivation, where the camera was covered, rendering the image stream black. The model responded by moving horizontally continuously without attempting to lower the basket. It was noted that the model lacked an output for waiting, suggesting the necessity of such an option in an industrial setting.

5.1.4 Trial Run 4: Short Rope

The fourth experiment involved shortening the rope from which the basket was hanging to simulate varying scenarios. Despite this change, the model mirrored the behaviour seen in the baseline experiment and placed the basket on the table by increasing the number of downward actions.

5.1.5 Trial Run 5: Blue Object

The final experiment tested the model's generalization capabilities in a Zero-shot learning environment. The green basket was replaced with a blue tool possessing a tuning fork shape. The model performed comparably to the baseline experiment, indicating its ability to reason that the task was to place the object on the table. Despite its disregard for the white sheet of paper as the placement zone, the model showed an inclination to focus on the object on the hook, regardless of its nature.

5.2 Trends and Patterns

5.2.1 Model Adaptability

One of the most important trends that were observed during the trial was the model's adaptability to changes in the physical configuration of the environment. As shown in the "Short Rope" and "Blue Object" trials, when significant alterations were introduced to the setup, the model managed to demonstrate its ability to adjust the movements to accommodate said alterations. Specifically, it managed to recognise the difference along the Z-axis, which indicates a good grasp of vertical displacement requirements of the task conditions based on the visual input.

5.2.2 Dependence on Visual Inputs

Another noteworthy pattern that emerged was the model's heavy reliance on visual inputs for decision-making, as demonstrated in the "No Camera" trial. When the camera input was covered, the robot arm's movements significantly deviated from the expected trajectory. This showcases the model's dependence on visual cues for performing its tasks efficiently. While this can be seen as an asset when visual information is rich and reliable, it may pose challenges in scenarios where visual information is compromised.

5.2.3 Insensitivity to Subtle Environmental Changes

While the model displayed admirable adaptability to major alterations in the setup, it seemed less sensitive to more subtle elements in the environment. In the "Baseline" trial it became apparent that the model did not manage to identify the white sheet as a designated placement zone. In the "Moved Sheet" trial, the model didn't adjust its trajectory to account for the new placement of the white sheet.

This could be tied to the limitations of the 2D camera sensor used for gathering visual data. While they capture the horizontal and vertical aspects of the scene, they do not provide depth information. This makes it challenging for the model to accurately gauge the relative distances between objects in the environment, which is a crucial factor when determining the correct placement of objects.

Without depth perception, the white sheet and other objects in the environment may simply appear as two-dimensional shapes of different colours. As such, the model might not have been able to differentiate between the white sheet and similar-coloured objects in the cluttered background, hence its failure to recognize the sheet as the designated placement zone.

5.2.4 Consistency in Performance

Across all trials, the model demonstrated consistency in attempting to achieve the given task, even when faced with unfamiliar or altered circumstances. This was evident in the fact that all trials resulted in the robot arm successfully placing the object down, albeit not always in the desired location.

These trends and patterns speak to the crux of our research question, which aimed to investigate the model's adaptability, reliance on visual cues, sensitivity to environmental changes, and consistency in performance. Understanding these trends will enable us to refine the model further, improving its utility and reliability in real-world lifting operations.

5.3 Relevant Findings

This study has brought forth several findings that are significant to our understanding of autonomous lifting operations using Behaviour Cloning (BC) and Phasic Policy Gradient (PPG).

5.3.1 Performance of Behaviour Cloning

Initial findings indicate that BC can effectively learn and replicate the control policy demonstrated by the human operator. The robotic arm successfully imitated the broad task of moving the object from the starting point to the end. However, the model's inability to identify the white sheet as the target placement zone in the baseline and "Moved Sheet" trials demonstrated limitations in its capacity to discern crucial environmental cues. This observation suggests that additional depth information or improved training might be required to enhance the model's understanding of task-relevant objects in its environment.

While the model did not learn where the placement zone was in regards to depth, the saliency map shows that the model did get an intuition about the general objective during BC. In Figure 4.3 it is clear that the model focuses on the table, the object and the robot arm. There are also areas in the bottom, especially on the right frame that get attention from the model without any specific reason or this hurting performance.

5.3.2 Robustness to Physical Alterations

The model displayed a degree of robustness to physical changes in the environment, as evidenced in the "Short Rope" and "Blue Object" trials. Despite the alterations to the hook's length and the object's shape and colour, the model made appropriate adjustments in the Z-axis, suggesting it can generalize to alterations in the physical task set-up. This is the first indication of the algorithm being able to perform Zero-shot learning.

5.3.3 Reliance on Visual Inputs

The "No Camera" trial highlighted the model's heavy reliance on visual inputs for decision-making. When faced with a lack of visual input (i.e., images with only black pixels), the robot arm failed to maintain a proper trajectory and exhibited random movement, confirming the critical role of visual data in the model's operation.

These findings collectively underscore the potential of behaviour cloning and PPG approach for autonomous robotic arm control. They also identify areas for future improvement, particularly in enabling the model to discern subtler environmental changes and function effectively with minimal visual inputs. This knowledge is crucial for refining the model and advancing its applicability in real-world lifting operations.

5.4 Key Takeaways

In the scope of this research, the Closed-form Continuous-time (CfC) Artificial Neural Network (ANN) was employed in a lifting operation using the proposed LOKI-G algorithm. High accuracy was exhibited by the CfC ANN in the Imitation Learning (IL) phase across all configurations

tested. In the Reinforcement Learning (RL) phase, robust and reliable behaviour was displayed, with commendable performance observed during the Zero-Shot Learning task. Although the basket's placement was proximate to the designated target, a consistent depth offset was noted, likely attributable to the utilization of a single camera sensor devoid of depth information.

As delineated in Table 4.1 in 4.3 Neural Network Design, minimal variation was observed in model performance with different hyperparameters. A notable exception was the increased loss recorded when employing Impala-CNN across all configurations, except the small one. This observation suggests that the performance was not constrained by the neuron count in the CfC ANN, indicating the feasibility of utilizing a smaller CfC ANN. Considering the additional factors present in an industrial setting compared to a lab, a larger 64 neuron CfC ANN was selected for the trial runs, paired with the small CNN with Batch Normalization. No discernible latency was detected during the lifting operation, hinting at the potential viability of a more compact model.

During the Behaviour Cloning (BC) phase, the model appeared to form a general understanding of the key areas of interest, as visualized in Figure 4.3 of the saliency map. Attention given to seemingly irrelevant bottom corner areas did not adversely impact model performance. While benchmark outperformance was not the central goal of this experiment, it was proposed that saliency maps from BC could be compared with those from longer periods of RL. This comparison, conducted with the same input image, could illuminate any improvements in the model's reasoning and pinpoint its primary focus areas.

The camera placement in the experiment was acknowledged as unrepresentative of an industrial crane, where it would typically be affixed to the base or tip of the boom, rendering the crane invisible to the model. This difference in setup could potentially lead to better performance by directing model attention towards the object and the environment, rather than the crane. For instance, in our experiment, the saliency map in Figure 4.3 indicated that action decisions were partly based on the robot arm's position. Given the research's aim to develop an adaptable and robust solution for real-world lifting operations, the model's explainability is of paramount importance. As such, visualizing saliency maps on the trained BC model serves as a vital tool in fostering trust among crane users and manufacturers.

5.5 Implications of the results

The findings of this study suggest that a sparse Artificial Neural Network (ANN), once robustly trained and easily explainable, can be effectively trained on the edge using minimal examples. This discovery has the potential to expedite the development of autonomous cranes without the need for simulators, thereby circumventing the Sim-to-real gap. Avoiding this gap could allow the initial performance of the crane to be both safe and efficient from the onset of the lifting operation. Prior studies have indicated a Sim-to-real gap of 41.1-52.6% depending on the sensors used, after 50 hours of training on a 64 core CPU and NVIDIA A100 80GB GPU [17].

ANN results are frequently measured against benchmarks for performance. This approach provides easily understandable quantitative measurements but has led to progressively larger models due to their exceptional results when trained on cloud compute [28]. This increase in computational power usage negatively impacts carbon emissions. It is hypothesized that the use of sparse ANN on the edge could reduce the carbon footprint in comparison to larger models trained using simulation [8].

The use of explainable models such as CfC ANN and tools like saliency maps can potentially enhance user trust. Industrial applications often prohibit the use of black box models; therefore, having a trustworthy, easily implementable algorithm that provides visualizable attention could eliminate some barriers to the widespread adoption of autonomous lifting operations. The repository for this research is publicly available and open-source[1].

[1]Repository for the project: https://github.com/R-Liebert/LOKI-G

5.6 Limitations of the study

This study has been conducted on a robot arm that was simulating a crane. As this was conducted in a lab, the light conditions were reasonably consistent, the environment was static and natural disturbances were absent. The only sensor used was a single camera. There have still not been experiments performed on multimodal CfC ANN [13]. To only rely on one image stream poses an increased risk towards sensor failure and would limit the applications to those where all information is within the camera's field of view.

5.7 Recommendations for Future Research

This study served to provide experimental proof of concept. A logical next step might be to investigate the Zero-shot learning capabilities further by teaching a robot a specific task and then altering the environment and the task itself. Although the investigation aimed to explore if a larger model would negatively impact performance due to increased computational time, this was found not to be the case. Therefore, future research could explore the development of more compact CfC ANNs through optimization of the design with hyperparameter tuners such as SigOpt-Lite [25].

Considering that most cranes are equipped with a range of sensors, measuring parameters from hook load to boom position, the development of a multimodal CfC ANN could potentially enhance safety in autonomous crane operations.

To further ensure safety during lifting operations, an intriguing approach may lie in the utilization of BarrierNet [30]. Introducing a safety layer between the CfC ANN and the outputs/controller could allow for quantitative research to measure the number of interventions required by the BarrierNet on a model trained using LOKI-G versus one trained in simulation. This would provide an opportunity to document the difference in carbon footprint between the methods.

Since the commencement of this thesis, several advancements have been made in the field. One notable development is "A Walk in the Park" [26]. In contrast to the current experiment, which used PPG in the RL phase, this work is based on a Soft Actor Critic (SAC). Another promising technique for use in the IL phase is the application of Inverse soft-Q learning in IQ-learn [10]. The current experiment employed basic Behaviour Cloning in the IL phase. As LOKI-G presents a general approach, there is potential for improvement through the utilization of novel IL and RL methods with faster convergence to train the CfC ANN.

Chapter 6

Conclusions

In order to reach the primary objective, the following secondary objectives were identified in the Introduction chapter.

- **Analyzing the open-source frameworks for performing IL and RL in an industrial setting, with a focus on the eligibility of the LOKI algorithm and CfC ANN.**

  The Open-Source rllib library was the most promising candidate as a framework for performing IL and RL in an industrial setting, with a focus on the eligibility of the LOKI algorithm and CfC ANN. Rllib is "the industry-standard reinforcement learning Python framework" and promises "a fast path to production" [2]. However, it became apparent that the documentation was outdated, and the framework does not support RNN for IL. Consequently, it became necessary to build everything from scratch regarding data and model management, which was successfully completed. As such this objective can be considered to be met.

- **Adapting the LOKI algorithm to offline IL and state-of-the-art (SOTA) RL.**

  The LOKI-G algorithm was developed with the purpose of meeting this goal and can be found on the GitHub provided in this thesis. This objective was also met.

- **Creation of a custom environment for performing lifting operations. Create logging tools for State-Action pairs used in offline IL and reward function for RL.**

  A custom environment was created in order to facilitate communication between the Robot Arm and the algorithm. It was capable of logging State-Action pairs in both online and offline parts of the trials. This objective has been met.

- **Evaluate the performance of the model when performing the lifting operation, focusing on risk minimization, efficiency and explainability.**

  The performance of the model was evaluated and described in the Discussion chapter of this thesis. We have managed to achieve explainability by adding a saliency map. This objective has also been met.

Considering all of the secondary objectives have been mapped, we can conclude that there is indeed potential for integrating a generalized version of the 'LOKI' fast policy learning method with Closed-form Continuous-time Neural Networks. This thesis shows that the described method is suited for developing an efficient, robust and transparent solution for real-world lifting operations using minimal training examples.

6.1 Summary of main findings

During this research, it has been shown that a sparse neural network trained on a sub-optimal expert manages to perform a lifting operation in a safe and efficient with minimal examples. While

the precision of the basket placement was offset by the target, this was a consistent behaviour and likely linked to sensory input. It has been proven that small networks suitable for training and operation on edge devices perform adequate lifting operations and further research in this direction of environmentally friendly development of machine learning is justifiable as a countermeasure to increasingly large neural networks in the cloud.

Our proposed algorithm LOKI-G, Algorithm 2, performed as expected in a controlled environment.

6.2 Reflection

During the course of this research, we discovered that the act of waiting, a common aspect of human performance, plays a crucial role. In the process of designing a model for controlling cranes and other industrial equipment, attention is easily directed towards actions where an actuator is moved. However, it was found that a human operator frequently engages in periods of inactivity during a lifting operation. If the model is not provided with the option to wait, it invariably seeks an action that involves moving an actuator. This tendency can lead to undesirable behaviour, as in many scenarios, the most appropriate action is no action.

While the importance of quality data is heavily emphasized in machine learning curricula, reflections were made on how this principle affected performance in a real-world setting. The demonstrations on which the model was trained were collected by two individuals in a lab, each observing the arm from different directions. This situation could be equated to having two stereo cameras, two motion sensors, and two microphones positioned at 0°and 90°relative to the robot arm. In contrast, the model had only one stationary camera at its disposal. The stark discrepancy in the volume of data available to the model, as compared to the humans, appears disproportionate when comparing the model's performance with human performance. To ensure a fair comparison with human performance, the data collected for the model should ideally match the volume of data a human operator would use to perform the same task.

During the preliminary project, there were anticipations of performing the experiment using industrial-grade hardware and open-source software. Regrettably, this did not materialize. Proprietary hardware solutions and inaccurate documentation significantly constrained the outcomes and put pressure on the time limits. While this issue is not unique to this experiment, it has a greater impact when the research has limited previously related experiments to draw upon. The academic representation of the experiments would have potentially benefited substantially from a more focused approach, concentrating on the basic performance of the proposed algorithm rather than adopting an industrial perspective.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 1. ISBN: 1581138385. DOI: 10.1145/1015330.1015430. URL: https://doi.org/10.1145/1015330.1015430.

[2] Inc. Anyscale. *Scalable, state of the art reinforcement learning*. 2023. URL: https://www.ray.io/rllib.

[3] Erik Brynjolfsson and Andrew McAfee. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. 1st. W. W. Norton & Company, 2014. ISBN: 0393239357.

[4] Carlos Celemin et al. *Interactive Imitation Learning in Robotics: A Survey*. 2022. arXiv: 2211.00600 [cs.RO].

[5] Ching-An Cheng et al. *Fast Policy Learning through Imitation and Reinforcement*. 2018. DOI: 10.48550/ARXIV.1805.10413. URL: https://arxiv.org/abs/1805.10413.

[6] Karl Cobbe et al. "Phasic Policy Gradient". In: *CoRR* abs/2009.04416 (2020). arXiv: 2009.04416. URL: https://arxiv.org/abs/2009.04416.

[7] Felipe Codevilla et al. *Exploring the Limitations of Behavior Cloning for Autonomous Driving*. 2019. arXiv: 1904.08980 [cs.CV].

[8] Jesse Dodge et al. *Measuring the Carbon Intensity of AI in Cloud Instances*. 2022. arXiv: 2206.05229 [cs.LG].

[9] Florian Fuchs et al. "Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning". In: *CoRR* abs/2008.07971 (2020). arXiv: 2008.07971. URL: https://arxiv.org/abs/2008.07971.

[10] Divyansh Garg et al. *IQ-Learn: Inverse soft-Q Learning for Imitation*. 2022. arXiv: 2106.12142 [cs.LG].

[11] Nathan Gavenski et al. "How Resilient Are Imitation Learning Methods To Sub-Optimal Experts?" In: *Intelligent Systems: 11th Brazilian Conference, BRACIS 2022, Campinas, Brazil, November 28 – December 1, 2022, Proceedings, Part II*. Campinas, Brazil: Springer-Verlag, 2022, pp. 449–463. ISBN: 978-3-031-21688-6. DOI: 10.1007/978-3-031-21689-3_32. URL: https://doi.org/10.1007/978-3-031-21689-3_32.

[12] Ramin Hasani et al. *Closed-form Continuous-time Neural Models*. 2021. DOI: 10.48550/ARXIV.2106.13898. URL: https://arxiv.org/abs/2106.13898.

[13] Ramin Hasani et al. *Inventing liquid neural networks*. Youtube. URL: https://youtu.be/iRXZ5vQ6mGE?t=526.

[14] Ramin Hasani et al. *Liquid Structural State-Space Models*. 2022. DOI: 10.48550/ARXIV.2209.12951. URL: https://arxiv.org/abs/2209.12951.

[15] C. Huyen. *Designing Machine Learning Systems: An Iterative Process for Production-ready Applications*. O'Reilly Media, Incorporated, 2022. ISBN: 9781098107963. URL: https://books.google.no/books?id=BAy%5C_zgEACAAJ.

[16] Eric Jang et al. "BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning". In: *5th Annual Conference on Robot Learning*. 2021. URL: https://openreview.net/forum?id=8kbp23tSGYv.

[17] Ludvik Kasbo. *Reducing the Sim-To-Real Gap in Reinforcement Learning for Robotic Grasping with Depth Observations*. 2022. URL: https://hdl.handle.net/11250/3037793.

[18] J. Kober, J. Andrew (Drew) Bagnell, and J. Peters. "Reinforcement Learning in Robotics: A Survey". In: *International Journal of Robotics Research* 32.11 (Sept. 2013), pp. 1238–1274.

[19] Alexandre Lacoste et al. "Quantifying the Carbon Emissions of Machine Learning". In: *CoRR* abs/1910.09700 (2019). arXiv: 1910.09700. URL: http://arxiv.org/abs/1910.09700.

[20] Mathias Lechner, Ramin M. Hasani, and Radu Grosu. *Neuronal Circuit Policies*. 2018. arXiv: 1803.08554 [q-bio.NC].

[21] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. "Model-based Reinforcement Learning: A Survey". In: *CoRR* abs/2006.16712 (2020). arXiv: 2006.16712. URL: https://arxiv.org/abs/2006.16712.

[22] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. "No-Regret Reductions for Imitation Learning and Structured Prediction". In: *CoRR* abs/1011.0686 (2010). arXiv: 1011.0686. URL: http://arxiv.org/abs/1011.0686.

[23] Stuart Russell, Daniel Dewey, and Max Tegmark. *Research Priorities for Robust and Beneficial Artificial Intelligence*. 2016. arXiv: 1602.03506 [cs.AI].

[24] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG].

[25] SigOpt. *SigOpt-Lite*. 2023. URL: https://github.com/sigopt/sigoptlite#readme.

[26] Laura Smith, Ilya Kostrikov, and Sergey Levine. *A Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning*. 2022. arXiv: 2208.07860 [cs.RO].

[27] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[28] Pablo Villalobos et al. *Machine Learning Model Sizes and the Parameter Gap*. 2022. arXiv: 2207.02852 [cs.LG].

[29] Yuhui Wang et al. *Truly Proximal Policy Optimization*. 2020. arXiv: 1903.07940 [cs.LG].

[30] Wei Xiao et al. "BarrierNet: Differentiable Control Barrier Functions for Learning of Safe Robot Control". In: *IEEE Transactions on Robotics* (2023), pp. 1–19. DOI: 10.1109/TRO.2023.3249564.

[31] Xinyan Yan, Byron Boots, and Ching-An Cheng. *Explaining Fast Improvement in Online Imitation Learning*. 2021. arXiv: 2007.02520 [cs.LG].