INF
Informatik

Hochschule Reutlingen
Reutlingen University

Hochschule Reutlingen

Reutlingen University

– Digital Business Engineering –

Master–Thesis

Template for a Master Thesis at Reutlingen University

Vorname Nachname
Meinestr. 42
71088 Holzgerlingen

Matrikelnummer : 690420

Betreuer:	Prof. Dr. ING. Vorname Nachname
Zweitbetreuer:	Prof. Dr. Vorname Nachname
Abgabedatum:	30.09.2022



Sperrvermerk

Diese Thesis basiert auf internen und vertraulichen Daten des Unternehmens ¡vollständiger Unternehmensname¿. Diese Thesis darf unbefugten Dritten ohne ausdrückliche Zustimmung des Unternehmens und des Verfassers nicht zugänglich gemacht werden. Eine Vervielfältigung und Veröffentlichung der Thesis ohne ausdrückliche Genehmigung - auch in Auszügen - ist nicht erlaubt. Die Sperrfrist endet fünf Jahre nach Einreichung der Thesis am dd.mm.yyyy.

Abstract

Thesis abstract (same as dissertation abstract) answers the main questions of an entire paper. This short summary is a single page of text. It needs to show key methods used in a work, problems analyzed and gathered outcomes of a complex research work. This kind of academic assignment requires profound knowledge. People, who read abstracts, prefer those summaries (not executive summaries) that remain short, but very informative, with presented implications and clear study results. [Original source:
https://prothesiswriter.com/blog/how-to-write-a-perfect-thesis-abstract]
If you write your thesis in english check out https://grammarly.go2cloud.org/SH3Yg, to proof read your writing.

Kurzfassung

Der Abstract ist eine Kurzzusammenfassung einer wissenschaftlichen Arbeit, er wird dieser vorangestellt. Ein Abstract enthält alle wichtigen Informationen in Kürze zu Forschungshintergrund, Aufbau, Methodik, Datenauswertung und auch die Forschungsergebnisse. Bei einem Abstract handelt es sich aber nicht um einen Buchklappentext der Spannung aufbauen soll – im Gegenteil, das Darstellen der Ergebnisse hilft dem Leser, die Relevanz der Arbeit zu beurteilen. Das Abstract schreiben bedeutet also, all diese wichtigen Dinge zu berücksichtigen.
Der Abstract sollte nicht länger als ein bis zwei Seiten sein und darf nie mit dem Schlussteil verwechselt werden, da er unabhängig vom Text zu verstehen ist. Oft wird ein Abstract in englischer Sprache gefordert, beispielsweise zur Bewerbung auf internationalen Konferenzen.
[Original source: https://www.bachelorprint.de/wissenschaftliches-schreiben/abstract-schreiben-beispiele/#:~:text=Der%20Abstract%20ist%20eine%20Kurzzusammenfassung,Datenauswertung%20und%20auch%20die%20Forschungsergebnisse.]

Acknowledgement

First and foremost, I have to thank my research supervisors, [Ms Name Surname], [Mr Name Surname] and [Ms Name Surname]. Without their assistance and dedicated involvement in every step throughout the process, this paper would have never been accomplished. I would like to thank you very much for your support and understanding over these past four years.
I would also like to show gratitude to my committee, including [Ms Name Surname], [Ms Name Surname], [Ms Name Surname], [Mr. Name Surname] and [Ms Name Surname]. [Ms Name Surname]was my first-year [subject] professor at [Name]University. Her teaching style and enthusiasm for the topic made a strong impression on me and I have always carried positive memories of her classes with me. I discussed early versions of the [Name of the subject] with [Mr. Name Surname]. She raised many precious points in our discussion and I hope that I have managed to address several of them here. Even though I have not had the opportunity to work with [Mr. Name Surname], [Mr. Name Surname], or [Ms Name Surname], previously, the impact of their work on my own study is obvious throughout this dissertation.
In March 2012, I went to [Name] University for several weeks to study with [Ms Name Surname]. My time at [Name] University has been highly productive and working with [Ms Surname] was an extraordinary experience. Much of the analysis presented in Section III is owed to my time at [Name of the University]. [Mr Name Surname] at the [Name] University kindly assisted me with the statistical analysis in this dissertation and was very patient with my knowledge gaps in the area. I must also thank two colleagues at the Department of [Name], [Name Surname] and [Name Surname], for giving me the retreat to have this thesis rushed to the printer.
Getting through my dissertation required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past three years. I cannot begin to express my gratitude and appreciation for their friendship. [Name Surname], Name Surname], [Name Surname] and [Name Surname] have been unwavering in their personal and professional support during the time I spent at the University. For many memorable evenings out and in, I must thank everyone above as well as [Name Surname], [Name Surname] and [Name Surname]. I would also like to thank [Name Surname] who opened both her home and heart to me when I first arrived in the city.
Most importantly, none of this could have happened without my family. My grandmother, who offered her encouragement through phone calls and letters every week – despite my own limited devotion to correspondence. With her own brand of humor, [Name Surname] has been kind and supportive to me over the last several years. To my parents and my sister – it would be an understatement to say that, as a family, we have experienced some ups and downs in the past three years. Every time I was ready to quit, you did not let me and I am forever grateful. This dissertation stands as a testament to your unconditional love and encouragement. [Original source: https://prothesiswriter.com/blog/how-to-write-an-acknowledgement-for-a-thesis]

Contents

Sperrvermerk **I**

1 Introduction **1**

2 Foundations **3**
2.1 Markov Decision Process . . . 3
2.2 Reinforcement Learning . . . 3

3 Methodology **5**
3.1 The Two Training Scenarios . . . 5
3.2 Framework for Evaluating the Performance of a Reward Function . . . 6

4 Main Content **11**
4.1 Here is a Table . . . 11
4.2 Equations . . . 11
4.2.1 Here is a Single Line Equation . . . 11
4.2.2 Here is a Multiline Equation . . . 12
4.2.3 Inline Math . . . 12
4.3 Figures . . . 12
4.3.1 Here is a Figure . . . 12
4.3.2 Here are Two Figures in One . . . 12
4.4 The SI Package . . . 12
4.5 Fancy Boxes . . . 13
4.6 Code Snippet . . . 13
4.7 Here is a Section . . . 14
4.7.1 This is a Subsection . . . 14

5 Literature Review **15**

6 Findings and Observations **17**

7 Conclusion and Future Work **19**
7.1 Conclusion . . . 19
7.2 Future Work . . . 19

Acronyms **23**

Glossary **25**

A Beispiel für ein Kapitel im Anhang **27**
A.1 Beispiel für einen Abschnitt im Anhang . . . 27

Selbstständigkeitserklärung **29**

List of Figures

2.1 The agent–environment interaction in a Markov decision process [1, p. 48] 3

3.1 KUKA KR 10 R1100 sixx, as Screenshot from the ISG-virtuos Visualization 6
3.2 Performance metrics for second order feedback control system (Haldia Institute of Technology, [2]) 7
3.3 The three main aspects for evaluating an Reinforcement Learning (RL) algorithm, defined by an example experiment, (rolling average over 200 steps) 8

4.1 Two experiments with sparse reward. The robot reaches the goal only once indicated by a non-zero reward at step 49 500 12
4.2 The agent purposefully makes itself blind to obstacles. 13

Chapter 1

Introduction

Die Einleitung dient dazu, den Leser auf die Arbeit einzustimmen, die Aufgabenstellung in einer allgemein verständlichen Form kurz und knapp zu beschreiben und anschließend die Gliederung der Arbeit kurz zu motivieren und zu beschreiben, so dass der Leser weiß, was er in welchen Kapiteln lesen wird, bzw. lesen kann.
Die Einleitung sollte einen Umfang von anderthalb Seiten nach Möglichkeit nicht überschreiten. Hintergrund dafür ist, dass Einleitung und Zusammenfassung zusammen den Rahmen der Arbeit bilden. Dieser Rahmen muss derart verfasst sein, dass eine völlig unbeteiligte Person (z.B. ein Abteilungsleiter in einem Unternehmen) innerhalb von höchstens 10 Minuten erfassen kann,

- in welchem Themengebiet sich die Arbeit bewegt,
- was die prinzipielle Aufgabenstellung und
- wie die prinzipielle Vorgehensweise war,
- welche Ergebnisse erzielt wurden,
- welche weitere Entwicklung noch möglich ist.

Chapter 2

Foundations

In this chapter, the foundations for Reinforcement Learning and reward functions most relevant to this thesis are presented. All terms and concepts, necessary to understand the explanations in later chapters, shall be explained in the following. The experiments depicted in this thesis are trained with the DQN algorithm implementation in Stable-Baselines3 [3]. Therefore Q-Learning will be explained in greater detail.

2.1 Markov Decision Process

The Markov decision process (MDP) is a mathematical concept, that provides a framework for simulating decision-making processes. Based on these symbols, a MDP can be defined as a tuple $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$ where S is a finite set of states, A is a finite set of actions, $\gamma \in (0, 1]$ is the discount factor, $T : S \times A \times S \to [0, 1]$ is the transition probability distribution, and $R : S \times A \times S \to \mathbb{R}$ is the reward function. [4] As explained by Puterman ET al. [5], "The decision maker's objective is to choose a sequence of actions called a policy, that will optimize the performance [cumulative reward] of the system over the decision making horizon. Since the action selected at present affects the future evolution of the system, the decision maker cannot choose his action without taking into account future consequences "

2.2 Reinforcement Learning

Reinforcement Learning is a type of Machine Learning, that is based on MDPs. Sutton ET al. [1, p. 53ff] defines it as follows: "The learner and decision-maker is called the agent. The thing it interacts with, comprising everything outside the agent, is called the environment. These interact continually, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment also gives rise to rewards, special numerical values that the agent tries to maximize over time." These relationships are presented in figure 2.1

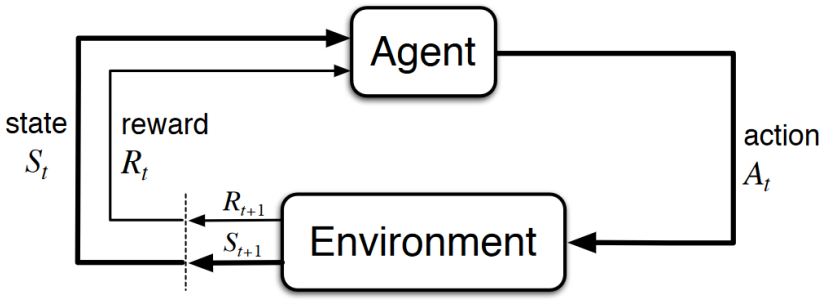


Figure 2.1: The agent–environment interaction in a Markov decision process [1, p. 48]

The environment can be a simulation or it can be a system in the real world with a digital interface. Usually, a simulation is used, as this speeds up the training process. The agent performs actions in the environment and gets signals back from the environment, as described in the MDP.
Reinforcement Learning works with discrete steps. At each step, the agent gets some information from the state of the environment, also called an observation.

Chapter 3

Methodology

In this chapter, the methodology and the setup for the experiments are described. First, the virtual commissioning software, and its limitations are presented. Then it is explained, how this software is utilized as the environment and how the RL code interacts with it. Then the two scenarios, used for the experiments in chapter 4 and chapter **??** are presented. Finally, a framework for evaluating the performance of a reward function is introduced.
All actions, including the reset, are implemented utilizing an NC file. An NC file is a standardized way to write G-Code for industrial machines. The reset method is implemented as a

```
f"G90 G01 X1={j0} Y1={j1} Z1={j2} A1={j3} B1={j4} C1={j5} F10000\n"
```

command. G90 G01 defines an absolute movement of the robot to the position defined by the six joint angles j0 to j5. The step method is implemented as a

```
f"G91 G01 X1={j0} Y1={j1} Z1={j2} A1={j3} B1={j4} C1={j5} F1000\n"
```

command. G91 G01 defines a relative movement of the robot to the position defined by the six joint angles j0 to j5. At every step in the RL training, the chosen action is translated into a line of NC-Code, which is written in a file and then executed by the NC-Kernel in virtuos.
To train the RL agent, the OPEN AI Stable Baselines 3 (SB3) [3] Python module is leveraged. Amongst the algorithms provided by SB3 the Deep Q-Networks (DQN) algorithm, explained in section **??**, is used. DQN is a sample efficient RL algorithm with a discrete action space.
Most of this setup came as a prerequisite to the thesis. As part of this thesis, the setup was modified, so that actions are executed via an NC-file. This required changes in the Python script and the virtuos model.

3.1 The Two Training Scenarios

There are two scenarios. One scenario is utilized in chapter 4, where the robot is trained to move to a specified position. The second is utilized in chapter **??**, where the robot is trained to move to a specified position whilst avoiding obstacles. The second scenario is an extension of the first. It has added obstacles, distance sensors, collision detection, and more inputs to the Q-function.
Both scenarios have only one robot. The " KUKA KR 10 R1100 sixx", as presented in figure 3.1 has the technical data listed in table 3.1. It is a common type of industrial robot for small to medium-sized handling and assembling tasks and is therefore a great robot for the proof of concept in this project.
In the first scenario, the robot starts the episode in the position illustrated in figure 3.1. The goal of the agent is to reach the red point, which is a fixed point within the reach of the robot. The starting position is the position of the robot arm, where all joints have an angle of 0 . In this state, the Tool Center Point (TCP) has the coordinates $[0.7233, 0.0, 0.9951]$. The TCP is an arbitrary point at the center of the tool (in our case it is a gripper), defined by the manufacturer. The goal has the coordinates $[0.6, 0.6, 0.6]$. The euclidean distance between the starting position and the

Table 3.1: Technical data for "KUKA KR 10 R1100 sixx" [6]

Maximum reach	1101 mm
Maximum payload	10 kg
Pose repeatability (ISO 9283)	± 0.03 mm
Number of axes	6
Mounting position	Floor
Footprint	320 mm x 320 mm
Weight	approx. 55 kg

goal is 0.73 m. To achieve the goal, the agent can move only one of the six axes by 2° incrementally each step. The goal is considered achieved if the TCP of the robot is within 5 cm of the target position. With this setup, the minimum amount of steps, to reach the goal, is about 30 steps.

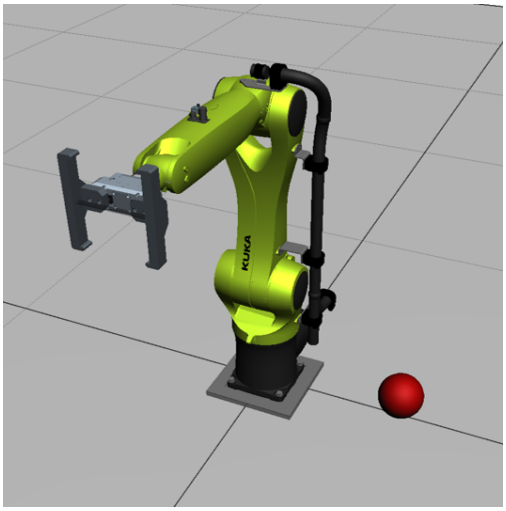


Figure 3.1: KUKA KR 10 R1100 sixx, as Screenshot from the ISG-virtuos Visualization

The parameters used for the Stable Baselines 3 [3] DQN algorithm are as listed in the following table 3.2:
All experiments are performed using a single PC with an Intel I7 4770K and a GTX 1060 3GB.
The second scenario is an extension of the first. It has added obstacles, distance sensors, collision detection, and more inputs to the Q-function. Further details on how the second scenario differs from the first are explained in section **??**.

3.2 Framework for Evaluating the Performance of a Reward Function

In this thesis, the performance of reward functions is evaluated under three main aspects:

- time to convergence
- final performance of policy
- stability

The goal is, to maximize all three aspects. Some improvements in the system can improve all three aspects at once, whilst others may increase one, but slightly decrease the other two. The objective of this thesis is to find and understand how changes in the systems affect the three performance aspects.

Table 3.2: Parameters for the Stable Baselines 3 [3] DQN algorithm

neural network architecture	[input(6), 64, 64, 64, output(13)]
learning rate	0.001
buffer size	1 000 000
learning starts	1000
batch size	128
tau	1
gamma	0.99
train freq	every 4 steps
gradient steps	-1
target update interval	100
max grad norm	10
exploration fraction	0.1
exploration initial eps	1
exploration final eps	0.05

The three aspects are inspired by performance metrics for feedback control systems. Feedback control systems are evaluated against a step response, where the system starts with an initial offset from the target value, and aims to minimize the difference between actual value and target value as fast as possible. In figure 3.2 such a step response is shown. The system starts at an initial value of 0 and aims to reach a value of 1 as fast and precise as possible.

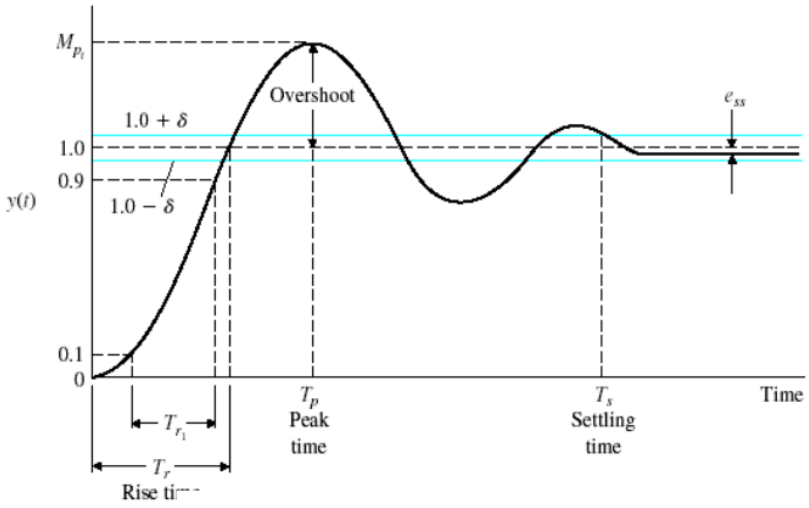


Figure 3.2: Performance metrics for second order feedback control system (Haldia Institute of Technology, [2])

The settling time T_s is synonymous with the *time to convergence* aspect. The steady-state error e_{ss} is synonymous with the *final performance of policy* aspect. The *stability* aspect is not as easy to compare with the equivalent in feedback control systems, as RL algorithms have different mechanisms and different reasons for unstable behavior. During the training, the agent may suddenly diverge and "unlearn" the correct policy.
To further clarify, what the three main aspects for evaluating an RL algorithm mean in practice, and how they can be read from a graph of the training progress, an example scenario is analyzed. A RL training run is a sequence of steps, where the agent collects experiences and learns from them. At the start of a training run, the weights and biases in the neural network are randomly initialized based on a random seed. The training process is divided into episodes, limited to n steps.

In the experiments in this thesis the limit is 600 steps. After each episode, the environment is deterministically reset to an initial state. To track the performance during the training, the cumulated episode return is logged after each episode. The intuition is that the higher the reward, the better the policy of the agent.
Comparing the return of the episodes is a common practice in RL research. Yet this practice is not sufficient for the objective of this thesis, as here the reward functions are experimented with. When using different reward functions between experiments, the intuition that a higher reward implies a better policy no longer holds true. A better way to compare two episodes is to evaluate how well the agent has achieved the goal according to another metric. For this thesis, we can compare if and in how many steps the agent has moved the TCP to the target position. This is indicated by the *total steps in the episode*. If the total steps in an episode is less than the maximum steps (600 steps in our case), then we know, that the agent has reached a terminal state, causing the episode to be terminated before the timestep limit of 600 steps. In the experiments in chapter 4 no negative terminal, like *"robot crashed "* , exists, so terminating the episode early means that the agent has reached the target position.
To illustrate, how the three performance metrics can be deduced based on the graph of the total steps for each episode, the example experiment in figure 3.3 is analyzed.

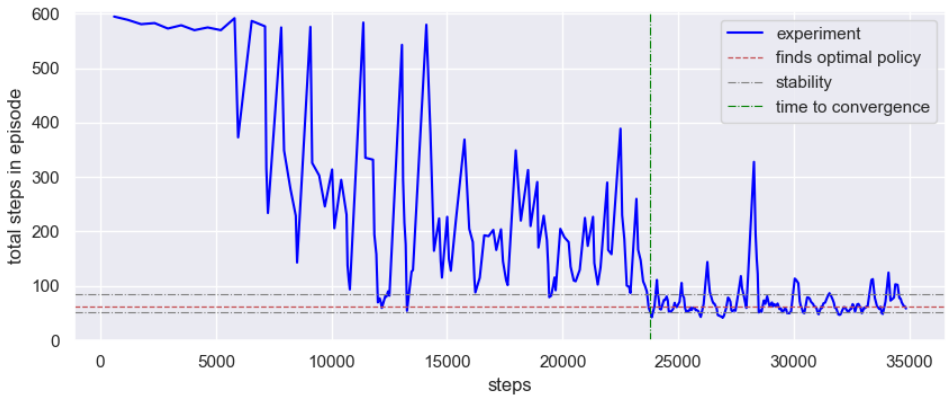


Figure 3.3: The three main aspects for evaluating an RL algorithm, defined by an example experiment, (rolling average over 200 steps)

The *time to convergence* aspect is defined by the elbow method, as this method is commonly used in machine learning. Convergence is achieved if the performance of the policy does not improve significantly anymore. To better examine this aspect, the experiment performance should be averaged with a rolling-window average. This makes it easier to visually pinpoint if the average performance of the policy is not improving anymore. If a rolling average is used for a figure, the window size is noted in the figure description.
The *final performance of policy* aspect is defined by how good the policy of the agent is after the learning process has converged. For the use case in this thesis, the performance can be evaluated based on how fast the agent reaches the target position. In particular, the performance is based on how many steps the agent takes to reach the target position in each episode. The fewer steps a policy takes to reach the target position, the better the final performance of the policy. To evaluate the final performance, only the performance at the end of the training process is considered. Determining, when the training process is finished is quite arbitrary. However, the performance should asymptotically converge to its final performance during the training process.
The *stability* can not be judged easily by a single value. Different aspects are contributing to the stability of a RL training. One aspect is the standard deviation σ of the performance after convergence has been achieved. Another aspect of the stability is the resilience to divergence. RL algorithms sometimes diverge away from a sensible policy due to the natural instability and

sensibility towards the random seed.



A more rigorous numerical definition performance metrics would be preferable. Nevertheless, for this thesis, the performance aspects are mostly determined by visually interpreting the training plots as illustrated with figure 3.3. This methodology is used because RL algorithms are very sensitive towards the random seed, and multiple training runs need to be performed to achieve a representative image of the training performance. Measurements should not be presented as more precise than the system it measures. A numerical one-dimensional metric as a performance measure would not represent the unprecise and random nature of RL. Therefore visual comparisons for the performance aspects are deemed sufficient for this thesis.

The reward function developed in this thesis should be generalizable for similar scenarios. A reward function, that is designed for a scenario with one obstacle should also perform well in a scenario with three other obstacles. Parts of the reward function, that can not be generalized should be abstracted so that they can be easily parameterized externally.

Chapter 4

Main Content

Somewhere in your Thesis you need to show the work that you have done should be described. Earlier in the Thesis you can show and summarize work others have done. Later in your Thesis, you should focus on the work you have done.
I will use this section to show some examples for figures and table and stuff

4.1 Here is a Table

In table 4.1, the experiments performed in this chapter are listed alongside an explanation for why they are performed.

Table 4.1: **Experiments in chapter 4**

Experiment	Figures	Explanation
Sparse Reward	4.1	serves as a comparison to show that dense reward is superior
Negative Dense Reward	**??**; **??**	shows that a dense reward improves the performance, but this one has stability issues
Negative Dense Reward (0 Positive Terminal Reward)	**??**	tests if the stability can be improved by changing the terminal reward
Negative Dense Reward (0 Positive Terminal Reward; Lower Learning Rate)	**??**	tests if the stability can be improved by lowering the learning rate
Dense PositiveRreward with Reward Shaping	**??**; **??**	improves stability and introduces reward shaping
Convergence Test	**??**	tests the convergence and stability by training longer
Dynamically Reduced Reward vs. Baseline	**??**; **??**	tests if a dynamically reduced reward can improve performance

4.2 Equations

And math stuff.

4.2.1 Here is a Single Line Equation

$$Q^*(s,a) = \sum_{i=0}^{\infty} \gamma^i r_i \leq \sum_{i=0}^{\infty} \gamma^i = \frac{1}{1-\gamma} = \frac{1}{1-0.99} = 100 \tag{4.1}$$

4.2.2 Here is a Multiline Equation

$$\text{distance} = \sqrt{(x_{\text{goal}} - x_{\text{TCP}})^2 + (y_{\text{goal}} - y_{\text{TCP}})^2 + (z_{\text{goal}} - z_{\text{TCP}})^2} \tag{4.2}$$

$$\text{reward} = -\left(\frac{\text{distance}}{\text{maxdistance}}\right) \tag{4.3}$$

4.2.3 Inline Math

You can also use an inline math environment like so $\frac{1}{10}$.

4.3 Figures

for graphs and images

4.3.1 Here is a Figure

Figure 4.1 shows two experiments with a sparse reward function, set to run for 162000 steps (one day of computing each).

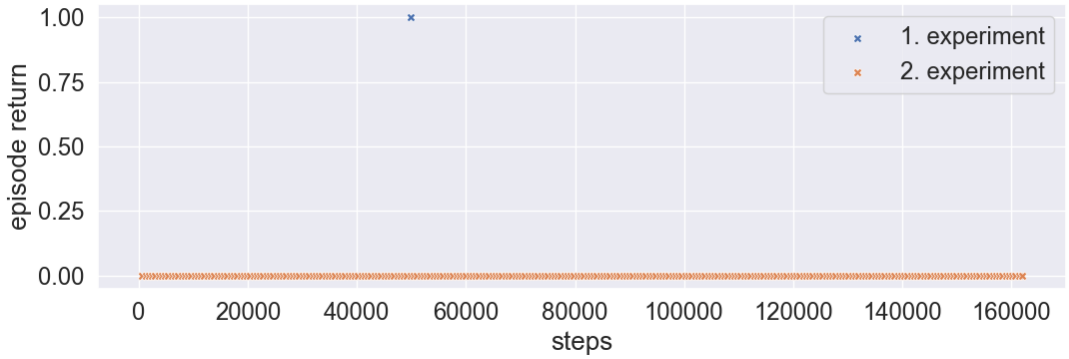


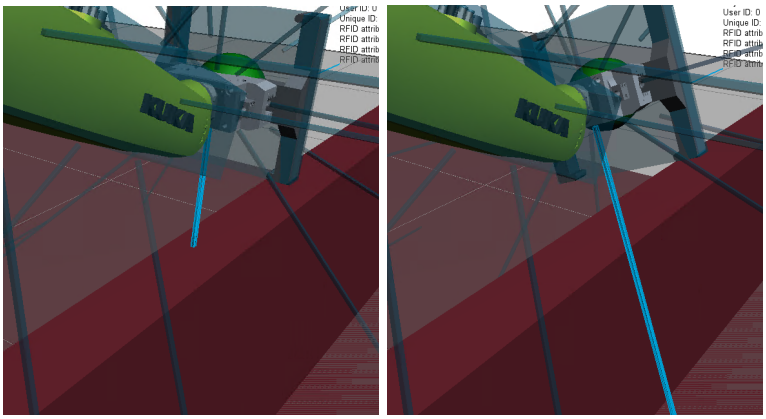
Figure 4.1: Two experiments with sparse reward. The robot reaches the goal only once indicated by a non-zero reward at step 49 500

4.3.2 Here are Two Figures in One

Always mention the figure 4.2 with a `figure~\ref{key}`. The makes sure, that there is no newLine in between the two things.

4.4 The SI Package

Here are examples for when you need to **use numbers and units in your text**. When you use numbers you need to make sure, that the format is consistent. Use , or . for german or international notation. Have a consistent spacing between number and unit.
The **SI package** does that for you. Just define all your values with a `\SI{13000}{MPa}` to get the output 13 000 MPa.
To quickly open an environment like an itemize for example just type `\nameOfEnvironment` then VSCode will suggest a snippet for you, that you can select with enter. try it out and make an itemize thing below.



(a) agent sees wall (b) agent moved axis 6

Figure 4.2: The agent purposefully makes itself blind to obstacles.

4.5 Fancy Boxes

Make **fancy boxes** if you are feeling creative. the package awesomebox https://ctan.org/pkg/awesomebox?lang=de is already included in the template.



To ensure that the agent does not learn to wait in front of the target, the following condition is required to hold:

$$\text{terminal reward} > \text{max intermediate reward} + \text{discount factor} \cdot \text{terminal reward} \quad (4.4)$$



use them for important stuff



or warn your reader



show where there might be some problems with your research. Like the number of participants is to low. This shows that you are a good researcher and know the limitations of your findings.



put emphasise on something

4.6 Code Snippet

You can insert code snippets using the listings package:
The complete reward function developed for the first scenario without obstacles is presented as Python code below. This reward function combines all findings from chapter 4.

Listing 4.1: reward function for: move to target position

```
def reward_Function(TCP_coordinates, steps_count_this_episode):
    x_TCP, y_TCP, z_TCP = TCP_coordinates
    # calculate the Euclidean distance to the target coordinates
    distance = np.sqrt((X_TARGET - x_TCP) ** 2 + (Y_TARGET  - y_TCP) ** 2 + (
    Z_TARGET  - z_TCP) ** 2 )
    # if the step limit is reached, the episode is done
    if steps_count_this_episode >= STEP_LIMIT:
        done = True
    # calculate the reward
    reward = 1 - (distance / POSITION_REWARD_MAX_DISTANCE) ** 0.4
    # if the TCP is in close proximity to the target, the agent gets the positive
    terminal reward and the episode is done
    if distance <= GOAL_ACHIEVED_AT_DIST_IN_M:
        reward = POSITIVE_TERMINAL_REWARD
        done = True
    return reward, done

```

You can also do inline formatted code like this `reset()`.

4.7 Here is a Section

Always create a label for your section, so that you can reference it in your text.

4.7.1 This is a Subsection

This is a subsection.

This is a Subsubsection

This is a subsubsection.

Chapter 5

Literature Review

In your thesis you can integrate a literature review. Check out the pdf at https://www.unil.ch/files/live/sites/hec/files/doc/master/mscis/how-to-do-a-literature-review-for-your-master-thesis.pdf to learn more

Chapter 6

Findings and Observations

[Original source: https://www.oxbridgeessays.com/blog/dissertation-findings-discussion-sections/] Depending on your preference for writing, the findings and discussion sections can be the most rewarding sections of your total dissertation. By this point, you actually get to write about what you have done, rather than what others have said about your subject area. Granted that at some point in the discussion you are going to have to link back to this previous research. But you still have the opportunity to demonstrate how you have met that coveted gap in the research and generally made a useful contribution to knowledge.
There are many ways to write up both your findings and discussion. In shorter dissertations, it might make sense to have both of these comprise one section. In longer pieces of work, these chapters are usually separate.

Chapter 7

Conclusion and Future Work

In this chapter, the thesis is concluded and opportunities for future work are presented.

7.1 Conclusion

The conclusion is the very last part of your thesis or dissertation. It should be concise and engaging, leaving your reader with a clear understanding of your main findings, as well as the answer to your research question.
In it, you should:

- Clearly state the answer to your main research question
- Summarize and reflect on your research process
- Make recommendations for future work on your topic
- Show what new knowledge you have contributed to your field
- Wrap up your thesis or dissertation

[Original source: https://www.scribbr.com/dissertation/write-conclusion/]

7.2 Future Work

There should some text describing what research questions have been discovered.

Bibliography

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. Adaptive computation and machine learning series, Cambridge, Massachusetts: The MIT Press, second edition ed., 2018.

[2] A. BHATTACHARYYA, "Lecture Note, The Performance of Feedback Control Systems."

[3] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[4] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Teaching Multiple Tasks to an RL Agent using LTL," p. 10, 2018.

[5] M. L. Puterman, *Markov decision processes*, vol. 2 of *Stochastic Models*. Elsevier, Jan. 1990.

[6] "KUKA KR 10 R1100 sixx," Feb. 2021.

Acronyms

DQN Deep Q-Networks.

MDP Markov decision process.

RL Reinforcement Learning.

SB3 Stable Baselines 3.

TCP Tool Center Point.

Glossary

euclidean distance In mathematics, the Euclidean distance between two points in Euclidean space is the length of a line segment between the two points. It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem, therefore occasionally being called the Pythagorean distance..

virtual commissioning virtual commissioning is a technique, where the physical production environment is modeled in a 3D simulation, so that new hardware and new software can be tested without halting the production line.

Appendix A

Beispiel für ein Kapitel im Anhang

Anhänge werde in LaTeX durch den Schalter `\appendix` eingeleitet. Anschließend wird die Dokumentstruktur mit Kapiteln und Abschnitten wie bisher fortgesetzt. Die Nummerierung wird automatisch angepasst.

A.1 Beispiel für einen Abschnitt im Anhang

Selbstständigkeitserklärung

Ich versichere, dass ich diese Thesis/Arbeit ohne fremde Hilfe selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlichen oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe. Die Arbeit wurde noch keiner Kommission zur Prüfung vorgelegt und verletzt in keiner Weise Rechte Dritter.

....

Ort, Datum Vorname Nachname