Reinforcement Learning for Inventory Control in Supply Chains

An Approach towards Robust and Decentral Models

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Institut für Technologie (KIT)
genehmigte

Dissertation

von

M.Sc. Wi.-Ing. Lena Bergmann

aus Stuttgart

Tag der mündlichen Prüfung: 27.10.2023
Hauptreferent: Prof. Dr.-Ing. Kai Furmans
Korreferent: Prof. Dr. Stefan Helber

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorandin in der Zentralabteilung Logistik bei der Robert Bosch GmbH. Ich möchte mich an dieser Stelle bei allen Personen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Herrn Prof. Kai Furmans, Leiter des Instituts für Fördertechnik und Logistiksysteme (IFL), gilt mein besonderer Dank für die Übernahme des Hauptreferats sowie für viele weitere Dinge; insbesondere für die Inspiration mich überhaupt den wissenschaftlichen Herausforderungen zu stellen, Ihrer unglaublichen Vielzahl an Ideen und Ihre Zeit in den gemeinsamen Diskussionen.

Prof. Stefan Helber und Prof. Jan Gerrit Korvink danke ich für die Übernahme des Korreferats und des Prüfungsvorsitzes. Ich schätze Ihre werschätzende Art der Prüfungsdurchführung und die geteilten Ratschläge.

Den Doktorand*innen vom IFL und der Robert Bosch GmbH verdanke ich, dass ich während der Promotionszeit immer von Gleichgesinnten umgeben war, die an verwandten Themen gearbeitet und mit ähnlichen Widrigkeiten gekämpft haben. Danke für Euren Beistand, gemeinsame Mittagessen, Ausflüge, Telefonate, Diskussionen und Erfahrungen. Auch meinen Kolleg*innen und Vorgesetzt*innen bei der Robert Bosch GmbH und den zahlreichen Studierenden in dieser Zeit gilt ein großer Dank für die Möglichkeit der Unternehmenspromotion, Eure Arbeit, Unterstützung und Diskussionen sowie die geteilte Zeit. Zusätzlich habe ich aus der Arbeit und den Diskussionen mit den Partnern des Forschungsprojektes *Dynamic Production Network Broker* viel wertvollen Input erhalten und tolle Erfahrungen gesammelt.

Ein großer Dank gilt meinem Partner, meinen Freund*innen und meiner Familie für Eure bedingungslose Unterstützung, die mich durch viele Hochs und Tiefs gebracht hat. Danke für aufbauende Gespräche, Karten und Briefe, Motivationsreden und natürlich auch Ablenkung.

Stuttgart, im Oktober 2023 Lena Bergmann

Abstract

In supply chain operations, inventory management represents a well-established challenge that involves the trade-off between product availability and stock holding costs. Research has tackled this topic extensively. The emergence of reinforcement learning through, for example, influential contributions such as *AlphaGo* and *AlphaZero*, has spurred further investigation into the application of inventory management for other problem domains.

The aim of this study was to merge the well-established issue of inventory management with the state-of-the-art solution approach of reinforcement learning. Therefore, existing research on this integration was evaluated and scrutinized to identify any research gaps. A key issue that arose pertained to the generalization of these models. As machine learning models heavily rely on the data on which they are trained, the primary concern is how to create a reinforcement learning model that can withstand shifting environments, including demand, replenishment time, and cost parameter traits. Consequently, the fundamental model was enhanced through an adaptable state and action space, and additional data points supplemented the initial basic state space. An analysis indicated that this extension facilitates the improved adaptation of the model to fluctuating environmental parameters.

When this study examined beyond a single supply chain location to encompass a linear supply chain with multiple locations, an additional question arose. A comparison between decentralized and centralized inventory decisions revealed that the decentralized model performed similarly well when compared with the central model. Additional findings indicated that demand shocks did not appear to significantly impact either model when

the bullwhip effects caused with and without a demand shock were compared.

Finally, the developed model was applied to real-world data from a chosen range of products. Promising results were obtained, but they require validation with further testing.

Kurzfassung

Das Bestandsmangement ist ein seit langem bekanntes Problem des Supply Chain Managments. Dabei geht es darum, das Gleichgewicht zwischen Produktverfügbarkeit und Bestandskosten zu finden. Es gibt eine breite Palette von Forschungsarbeiten, die sich bereits mit diesem Thema befassen. Mit dem Bekanntwerden des Reinforcement Learnings durch prominente Arbeiten wie *AlphaGo* und *AlphaZero* wurden auf der anderen Seite mehr und mehr Anwendungen des Reinforcement Learnings auf andere Problemdomänen untersucht.

Das Ziel der vorliegenden Arbeit war es, das bekannte Problem des Bestandsmanagements mit der innovativen Lösungsmethode des Reinforcement Learning zu verbinden. Daher wurden bestehende Arbeiten zu dieser Kombination gesichtet und auf ihre Forschungslücke hin analysiert. Eine Frage, die sich dabei stellte, ist diejenige nach der Generalisierbarkeit dieser Modelle. Da maschinelle Lernmodelle in hohem Maße von den Daten abhängen, mit welchen sie trainiert werden, ist die erste zu beantwortende Frage, wie ein Reinforcement Learning Modell designt werden kann, so dass es gegenüber veränderlichen Umgebungsparametern wie Nachfrage, Wiederbeschaffungszeit und Kostenparametern robust agiert. Dazu wurde das entwickelte Basismodell um einen adaptiven Zustands- und Aktionsraum erweitert, sowie der erste einfache Zustandsraum um weitere Datenpunkte ergänzt. Die Analyse und Auswertung anhand verschiedener Datenreihen zeigte, dass diese Erweiterung zu einer besseren Anpassung des Modells an veränderliche Umweltparameter führte.

Als weitere Fragestellung wurde eine aus mehreren Standorten bestehende lineare Lieferkette untersucht. Der Vergleich von dezentral und

zentral getroffenen Bestandsentscheidungen für diese Supply Chain führte zu dem Ergebnis, dass das dezentrale Modell im Vergleich zum zentralen erstaunlich gut agierte. Ein weiteres Ergebnisse war, dass beide Modelle relativ robust gegenüber Nachfrageschocks sind. Dafür wurde der Bullwhip Effekt in der Supply Chain mit und ohne Nachfrageschock verglichen.

In einem letzten Schritt wurde das entwickelte Modell auf reale Daten einer Auswahl von Produkten angewendet. Die Ergebnisse sind vielversprechend, müssen aber durch weitere Tests validiert werden.

Contents

Abstract iii

Kurzfassung v

1 Introduction 1
1.1 Motivation and Objective 1
1.2 Structure of the Thesis 3

2 Basics 5
2.1 Supply Chains 5
2.2 Inventory Management 8
2.3 Reinforcement Learning 12

3 Existing Approaches towards Reinforcement Learning for Inventory Control 17
3.1 Modeling of the Supply Chain 18
3.2 Modeling of a Reinforcement Learning System 23
3.3 Data 28
3.4 Chapter Conclusion 31

4 Model Structure of Reinforcement Learning for Inventory Control 33
4.1 The Environment 33
4.1.1 The Supply Chain Model 34
4.1.2 Synthetic Data Creation 38
4.2 The Agent and its Actions 40
4.3 Training the RL Model 40
4.4 Reward Function 42
4.5 Implementation 43

4.6 General Setup for Analysis 44
4.6.1 Performance Indicators 45
4.6.2 Presumptions Regarding Model Behavior 46
4.6.3 Statistical Validation of Testing 47
4.6.4 Parameter Setting 48
4.6.5 Benchmark Algorithm 50
4.7 Chapter Conclusion 53

5 A Robust Reinforcement Learning Model for Inventory Control 55
5.1 Robustness 55
5.2 Extension of the Basic Model Through Adaptive State and Action Space Scaling 56
5.3 Extension of the Basic Model by Cost Parameters 65
5.4 Analysis of System Behavior Compared with the (r, s, q) Approach 69
5.5 Chapter Conclusion 83

6 Reinforcement Learning for Inventory Control in Linear Multi-Stage Suppl Chains 85
6.1 Model Implementation 85
6.2 Analysis of System Behavior of the Decentral and Central Approaches 86
6.3 Analysis of the Bullwhip Effect Caused by a Demand Shock . . 94
6.4 Chapter Conclusion 98

7 Application to Real-World Data 99
7.1 Selection of the Data and Assumptions 99
7.2 Results 103
7.3 Chapter Conclusion 105

8 Conclusion 107
8.1 Summary 107
8.2 Outlook 109

Nomenclature 113

Bibliography 117

Publication List 133

1 Introduction

The question is too good to spoil with an answer.

– Robert Koch

This chapter provides a summary of the motivation for this study and the corresponding research questions. These factors drove the structure of the work, which is also briefly outlined.

1.1 Motivation and Objective

The problem of inventory management is a well-known and often treated problem in the field of logistics. Each supply chain (SC) , whether it has a small decision scope like the grocery store around the corner or a globally steered SC, must answer the basic question of how to stay at the service of customers without accumulating too much stock. In particular, during the current SC disruptions caused by the COVID-19 pandemic and the shortages in semiconductors and raw materials, the topic of inventory management and its corresponding solutions has generated interest. This interest is evident from the statistics for the search term “inventory management” on Google Trends, as depicted in Figure 1.1, which exhibits consistent and stable popularity with occasional fluctuations. The extensive research conducted on the replenishment problem for all types of SCs, ranging from simple systems to supply networks that accommodate single or multiple products, with or without due dates, reflects the significant interest in the topic.

By contrast , the level of interest in the concept of deep reinforcement learning (RL) had not been measured for a long time, when it finally began to emerge in Google searches during the late 2010s. The development of RL agents continues to create a stir in the media. While academic researchers had already been working with and advancing RL for a considerable period, public attention was mainly directed toward the accomplishments of DeepMind. For instance, one of the initial models, *AlphaGo*, was introduced in October 2015 and gained its first victory against a well-regarded Go player in early 2016, which was reported in the media. Later, this development was followed by Master and *AlphaGo Zero*, which are both agents that learned to play the traditional game of Go. Furthermore, Alpha Zero also acquired skills to play chess and shogi, and the latest development of DeepMind can master visually intricate Atari games. (DeepMind 2021) The advancement of deep RL, which was previously an unknown field, served as a catalyst for further research in various application areas. This, in turn, led to the creation of the work presented in this thesis. The aim of this study was to address a prevalent logistical issue, namely the replenishment problem, using a modern technological solution from the machine learning toolkit—namely RL.

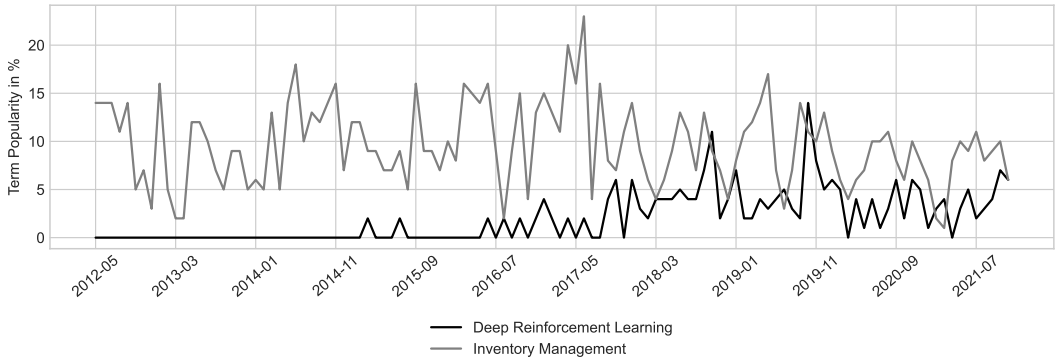

Figure 1.1: *GoogleTrends* Report of Term Popularity for the Terms *Inventory Management* and *Deep Reinforcement Learning* Over the Last Years

1.2 Structure of the Thesis

Providing a framework for the present work, the following research questions served as the foundation for further studies and the structure of the thesis, as depicted in Figure 1.2:

How might one develop an RL system for SCs that contend with stochastic demand and stochastic replenishment times and are inclusive of forecast errors? Which SC characteristics are crucial to incorporate? After conducting a literature review, which is presented in Chapter 3, it became evident that various approaches exist regarding the implementation of RL in inventory management. Notably, the amount of information included in the state space, varying actions taken, and assumed SC models range widely. Consequently, a basic RL system was developed, which is presented in Chapter 4. The chapter outlines the model's design decisions and abstraction levels, as well as the implementation and setup for future investigations.

How reliable and robust are the decisions made by the basic model? How is the dependability affected by the structure of the state and action space, and how competently does the developed model perform compared with a traditional policy? The conduct of the model was evaluated and contrasted with a conventional inventory policy. The primary model was further developed and assessed, as presented in Chapter 5. By adapting the scaling of the state and action space, the RL model performed effectively in unfamiliar environmental settings.

How should an RL system be designed for linear SCs that contain several stages? How does a central approach perform versus decentrally trained models and how do they react to demand shocks? The formerly developed robust model was applied to linear SC structures that contained several locations. By contrast, the model was expanded to address the replenishment issue in linear SCs with only one RL agent. A comparison of the models and their reactions to demand shocks is illustrated in Chapter 6.

How would such a previously defined and theoretically tested model perform on real data? Theoretical research has its advantages, but it is more interesting to a wider audience when it is successfully applied to real-world use cases. Therefore, the developed model was applied to several products and their corresponding demand series. Decisions and resulting key performance indicators (KPIs) were compared with the results of the theoretical work as well as with reality in Chapter 7.

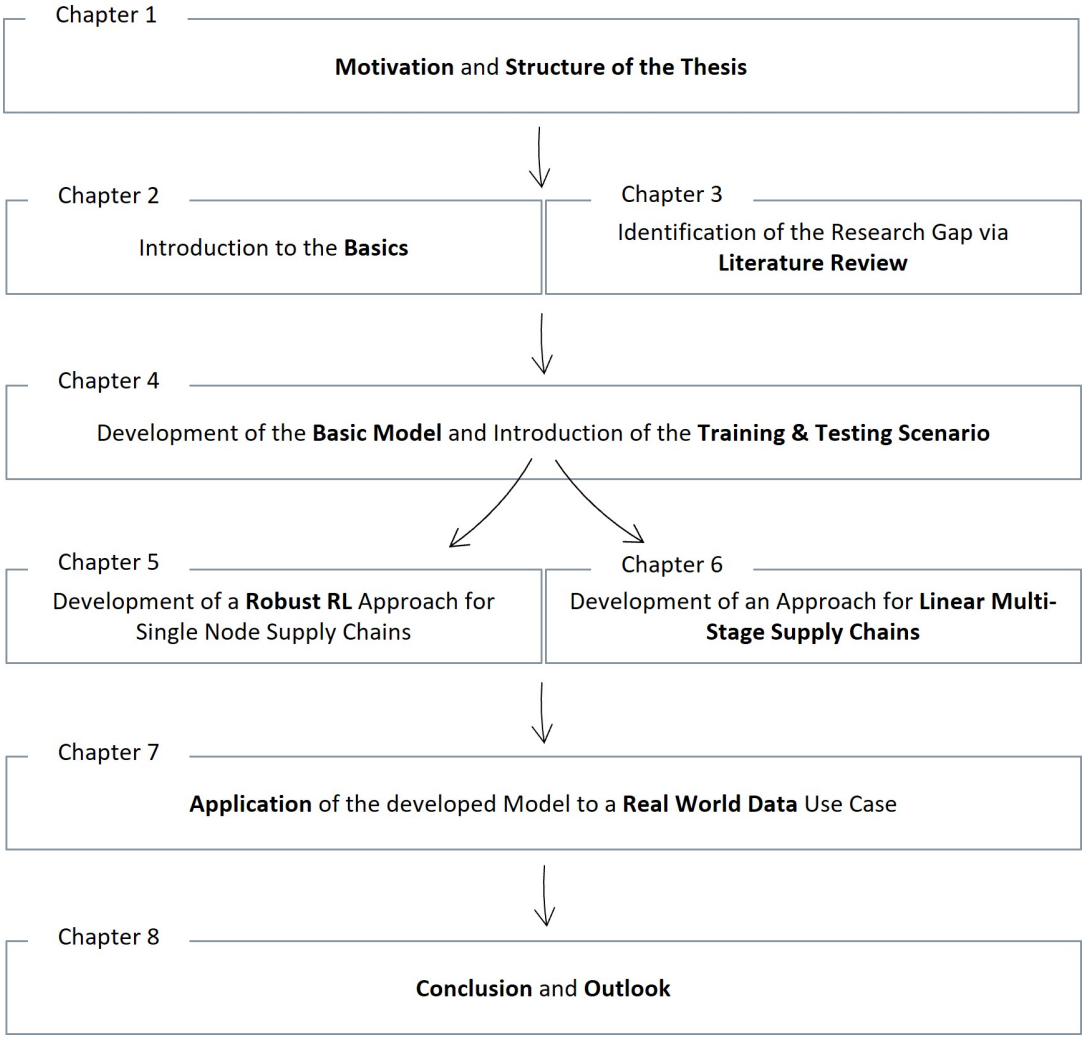

Figure 1.2: Structure of the Thesis

2 Basics

Most of us forget the basics and wonder why the specifics don't work.
– Garrison Wynn

In this chapter, the three main topics—SCs, inventory management, and RL—are addressed. A brief introduction to each topic is provided in the following sections.

2.1 Supply Chains

The Council of Supply Chain Management Professionals (CSCMP) defines the term *supply chain* (SC) as the flow of materials and information from the acquisition of raw materials to the transformation and delivery of finished products to the end user. In most cases, there are several actors from different companies working together in an SC. (CSCMP 2013)

By representing SCs in graphs, the different actors and their relationships can be abstracted by the nodes and the directed links between them. Sources in the context of graphs refer to resources, while sinks characterize customers or demand in general. (Arnold et al. 2008, p.6), (Tempelmeier 2015) Looking at different SCs and their locations and material flows, Arnold et al. (2008, p.161, p934 ff) distinguish three main SC structures, which are depicted in Figure 2.1. The first is represented by a directed graph that is convergent so that there are two or more sources, several intermediate nodes, and a sink. Arnold et al. (2008, p.935) refer to such a network as a many-to-one network, which fits the structure of procurement and produc-

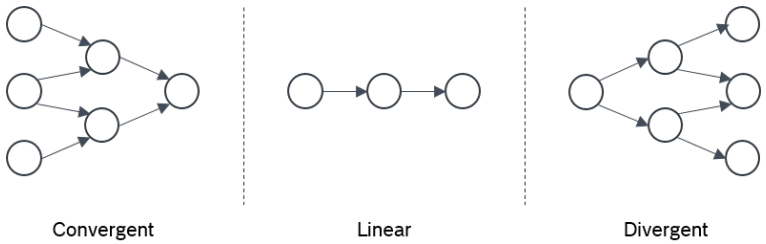

Figure 2.1: Schematic Supply Chain Structures, Based on Arnold et al. (2008, p.161, 934 ff.)

tion supply networks. By contrast, one-to-many networks are characterized by one source, several intermediate nodes, and two or more sinks. In the context of SCs, they are called distribution networks. Real SCs have some form of the abovementioned networks, or a mixture of them, but they are often represented in terms of a linear chain clustered around the respective tasks of making, sourcing, and delivering. (Arnold et al. 2008, p.934 ff.)

In SCs, all kinds of flows exist, including material, information, product, and financial flows. Therefore, the actors who are closer to the customer are referred to as downstream actors, while those who are closer to the source of materials are referred to as upstream actors in an SC (p.3 ff) (Ivanov et al. 2019) (Stadtler and Kilger 2008)

Such an SC must be managed, which is where the term *supply chain management* (SCM) comes in. SCM refers to the coordination and collaboration of actors inside and outside a company in the SC, which includes all parts from sourcing, procurement, and all logistical matters (CSCMP 2013).

Schuh (2007) describes the concept of a central planning and coordinating body for managing internal SCs. He highlights that the advantage of this appears to be the alignment of local decisions with SC-wide objectives and therefore better overall planning results. However, a central planning instance leads to a higher coordination effort, which becomes more complicated as the number of instances to be planned increases, and it would be even higher for SCs that involve multiple companies.

The concept of decentralization replaces the central planning instance with several autonomous agents. Becker et al. (2015) and McFarlane et

al. (2003) define a system as decentralized if there is more than one decision-making instance that is not provided with complete information. The decision-making units are often linked to physical elements. Decentralized decision making has the advantage of reducing the complexity of local decisions, such as by allowing simpler methods to be used at one stage of the SC. (Arnold et al. 2008, p.125)

Zijm et al. (2019), Arnold et al. (2008, p.7 f.) and Schuh (2007) name several key aspects of SCM to describe their goals:

- **Availability of products, materials and information:** This is one of the main objectives of SCM. As the term SC already implies, the primary objective of SCs is to ensure the right amount of materials, semi-finished products, or finished goods at the right time and place. At each step in the process, material and information are supplied for the planning instances of the next steps. The availability of information becomes even more important when one considers SCs with actors from different companies.
- **Cost efficiency:** This has long been a determining factor in production and logistics planning, where the aim is to achieve maximum output with minimum input. However, the frame of reference for cost efficiency must be kept in mind. For example, cost efficiency in one production unit may be achieved through large batch sizes, which in turn may lead to high levels of work-in-process inventory between SC units—and therefore to lower cost efficiency in the system as a whole.
- **Customer focus:** This has become increasingly important in recent years as customers have demanded product variation and differentiation and the market has moved from a push market to a pull market.

Regarding the main tasks of SCM, Arnold et al. (2008, p.194) identify three levels of planning: the first and strategic level is the general SC design, namely the selection of locations and partners. The second, tactical, level is SC planning at a more detailed level, which includes the timely consideration of demand forecasts, orders, inventories, transport resources, and more. Lastly, at the operational level of SC execution, the aim is to imple-

ment the abovementioned plans, which is facilitated by the information and communication available. During execution, it is also necessary to constantly react to the real situation that the SC is confronted with.

2.2 Inventory Management

As mentioned in the previous section, inventory management is a tactical level of planning. The term *inventory management* is often used in a general context to describe the management processes that ensure the availability of products (CSCMP 2013); thus, it includes one of the main objectives of SCM. Axsäter (2015) uses the term *inventory control* to refer to the inventory management decisions that directly affect inventory levels. However, *inventory management* is often used synonymously. Other terms are *replenishment* (Axsäter 2015) and *ordering policies* (Arnold et al. 2008, p.153), which often refer to the same thing. Interestingly, because replenishment policies are planned in advance, they belong to the tactical level of SC planning. However, with the day-to-day replenishment decisions due to changing situations, it is of course a large part of operational "doing".

Inventories exist because of the process characteristics of the different SC processes, and therefore, they cannot be decoupled from them. Procurement, for example, may want to order in large batch sizes to obtain certain volume discounts. Transportation, on the other hand, is tied to the capacity of certain means of transportation, and therefore, it tends to order in different batch sizes. Sales, on the other hand, may want to keep a high level of stock to be able to respond to any customer request. Finally, the finance department does not want high levels of inventory because it ties up capital that is not available in the form of cash. Inventory management balances all of these different objectives, which can be summarized as ensuring the availability of materials and products while keeping stock levels sufficiently low. (Axsäter 2015) (Tempelmeier 2015)

Inventory is also a buffer against all kinds of uncertainties in SCs. Uncertainty comes from all directions—starting with customer demand, which in

most cases is not deterministic but varies over time. For planning purposes, demand forecasts are made, which contain the uncertainty of forecast errors, as most forecasting models are not perfect. Another uncertainty arises from stochastic process times during the replenishment period. These are caused by machine or transport breakdowns, organizational decisions during production (e.g., prioritization), and limited capacities (e.g., material, machinery, and personnel). Inventory is also held against these uncertainties. (Tempelmeier 2015)

The performance of inventory management is measured against the abovementioned objectives: the ability to deliver is expressed in terms of service level. In general, whether an incoming order can be fulfilled from stock should be measured. As orders consist of one or more units, there are different methods for calculating the corresponding service levels depending on the focus—that is, whether it is on the ability for an order to be fulfilled at all or on the quantity of the order that can be fulfilled. (Tempelmeier 2015)

The need for sufficiently low stock levels compared with the achieved service levels is often expressed in terms of the associated costs. According to Axsäter (2015), Berling (2005), Silver et al. (2017) and Zipkin (2000), various types of costs are associated with inventory management:

- The **holding cost** per unit and time should be related to the value of capital tied up in inventory. It is often determined as a percentage of the unit value. In addition, costs such as material handling, storage, organizational, and damage costs could be considered.
- **Ordering costs** are fixed costs for each replenishment, regardless of the batch size. They may include handling costs such as inspections and invoices, order forms, or authorizations.
- **Shortage costs** occur when an ordered product cannot be delivered to the customer on time. Either the order is backordered and the customer waits for the order, or sales are lost because the customer switches to another supplier or product. In the first case, the cost could be extra administration, extra transport costs, or discounts for

late deliveries. In the case of lost sales, the originally planned revenue contribution is also lost. In most cases, the cost of shortages is difficult to determine.

The formalized problem of inventory management is the so-called *replenishment* or *inventory control* problem. Prestwich et al. (2012) defines this problem for a set of planning periods T with demand d_t arising in each planning period t. The objective is to find an appropriate replenishment plan given the planning objectives and constraints. The plan consists of corresponding orders o_t for each period t.

A well-known phenomenon in the field of SCM is the bullwhip effect (BWE), which Forrester (1961) describes as a small change in demand at the customer side that leads to more severe changes as it moves up the SC from retailer to distributor to manufacturer. Lee et al. (1997) define the symptoms of the BWE as excess inventory, poor demand forecasting, either insufficient or excessive capacity, poor customer service, uncertain production planning, and high costs of correction.

The main causes of BWE lie in four main reasons identified by Lee et al. (1997):

- **Demand forecast updates** occur at each stage of the SC and information about changes is **not shared**. This results in individual updates depending on the demand signal received from the downstream stage. These individual updates include safety stocks and other factors, such as those below, which simply result in higher demand for the next stage; however, they are not explained and therefore cannot be interpreted by the supplier stage.
- **Order batching** occurs for cost efficiency in transport and production or for organizational reasons. This can result in orders being piled up until a certain quantity or date is reached. This naturally distorts the perception of demand at the upstream node.
- **Price fluctuations** lead to irregular buying behavior, as low prices may encourage forward buying and therefore high demand when prices are

low. When prices rise again, buying stops as the accumulated stock is first used up. Such fluctuations make planning difficult.

- **Rationing and shortage games** occur when a supplier is unable to meet all demand but rather only part of it. Due to rationing, each customer may receive only a fraction of what was requested. This leads to exaggerated and thus distorted demand on the part of the customers, because they know they will only receive part of the order.

Lee et al. (1997) also suggest solutions to the abovementioned problems, such as avoiding multiple demand forecasts, reducing order lot sizes, keeping prices stable, and avoiding shortage games by rationing through past demand figures.

The BWE can be quantified, as Fransoo and Wouters (2000) demonstrate, by comparing the coefficients of variance of demand in two stages of the SC. Another possibility is to follow the approach of Chen et al. (2000) by comparing the variances of customer demand and the resulting demand at the supplier. Whichever approach is chosen, since the BWE affects the demand variance, a measure such as one of the two mentioned would be appropriate.

The BWE can be experienced in a simulation game called the *Beer Game*, which was developed at the MIT Sloan Management School in the 1960s. Goodwin and Franklin (1994) describe it for the use of a computer simulation to teach managers what happens with their decisions. The structure resembles a simple linear SC of four stages, namely retailer, wholesaler, distributor, and factory, facing customer demand. The demand starts out steady but suddenly changes to another level, which the BWE illustrates very impressively with all of the aforementioned problems.

There is a broad and impressive research field of mathematical models for solving the problem of inventory and thus order coordination in SCs. An attempt to summarize them here is not made; rather, the interested reader may wish to consult the work of Axsäter (2015) and Tempelmeier (2015).

In addition to the mathematical models mentioned above, there are approaches such as vendor managed inventory (VMI). This attempts to facil-

itate the coordination of inventories by shifting the responsibility from the customer to the supplier. The idea is to reduce information asymmetries and unite individual interests between SC stages, which in most cases are independent entities with their own agendas. (Arnold et al. 2008, p.272)

2.3 Reinforcement Learning

Sutton and Barto (2018) define RL very broadly, with the basic idea explained by the schema in Figure 2.2: An agent interacts with an environment by performing different actions. The environment processes these actions and transitions to a new state, which also leads to a corresponding reward for the action taken. Based on the reward and the new state, the agent decides on its next action.

The agent's learning is controlled by the reward signal as it defines the goal of the RL process. A reward signal is sent from the environment to the agent after an action has been performed. It depends on the action performed and the current state of the environment. The goal of the agent is to maximize the rewards received over time. The reward signal therefore defines the learning process by rewarding good and punishing bad actions. As the reward signal is immediate feedback regarding what is good, the value function instead indicates what is good in the long run. It can be said to accumulate rewards starting from a particular state and considering what states are likely to follow. For example, a state may have a low immediate reward but a high value because the states that follow it are highly rewarded. A value function can be a simple function or look-up table, or it can involve complex calculations and searches. The value function itself is updated during the training process by a special RL algorithm that combines the old policy with updates of the recently received state–r eward pair. (Rebala et al. 2019) (Sutton and Barto 2018)

Sutton and Barto (2018) then distinguish between two main methods for solving RL problems that are illustrated in Figure 2.3 - namely tabular and approximate methods. Tabular methods enumerate all possible states and

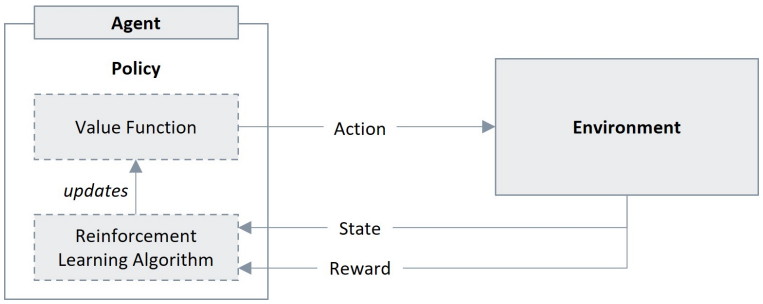

Figure 2.2: Schematic Representation of RL, Based on Sutton and Barto (2018)

stores the corresponding value for each possible action. This limits the state space to known and visited states during training. However, it may not be efficient for large state spaces, which is why the second family of methods exists. With approximate methods, one attempts to find a function that maps states and actions to values. As this requires fewer parameters than a full enumeration of the state space, this is a major advantage of approximate methods. A typical function approximator would be an artificial neural network (ANN). Its properties allow generalization to unknown states and actions, which would overcome one of the limitations of tabular methods. (Rebala et al. 2019) Solutions for the first family of methods (i.e., tabular) are often optimal, as policies can be determined in an exact manner. Examples include dynamic programming, Monte Carlo methods, and temporal difference learning. For the second family of methods, as the name suggests, solutions are often only approximate. Examples include on and off policy methods and policy gradient methods. (Sutton and Barto 2018)

The attentive reader may have already recognized the basic idea of RL from another context—namely that the mathematical model behind RL is the so-called *Markov decision process* (MDP). MDPs are time-discrete stochastic control processes that arise when a decision not only has immediate effects but also influences future decisions. The theory of MDPs can be found in Waldmann and Stocker (2013). MDPs can be solved using various methods, such as linear programming as well as value and policy iteration if the full model of the MDP is known. If, for example, the reward

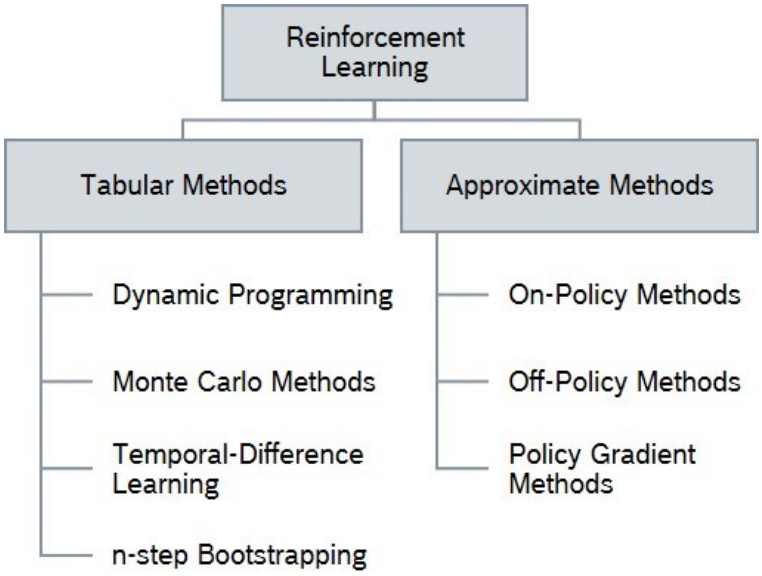

Figure 2.3: Methods of RL, Based on Sutton and Barto (2018)

function or transition matrix is not known, then this can be overcome by an RL agent. (Li et al. 2006) (Taylor and Tuyls 2010)

In most of the aforementioned RL methods, the main idea is to learn the action-value relations to derive an optimal policy. In the context of policy gradient methods, this is different, as instead of learning the action–value relations, the policy is learned directly. This means that while the value function can be used to learn the policy, it is not needed for the action selection itself when applied. Within the group of policy gradient methods, there is a subgroup called actor–critic methods, where the actor learns the policy while the critic learns the value function. (Sutton and Barto 2018) (Rebala et al. 2019) An advantage of policy-based methods is that they can generate stochastic policies, which is useful for some types of problems where the policy is stochastic rather than deterministic. A popular example of this is poker, where the decision to bluff or not to bluff is best made stochastically. Another advantage of policy-based methods is their ability to deal with incomplete state information. When state information is incomplete, as in the case of poker where one does not know one's opponent's hand, the best deterministic action is stochastic. (Rebala et al. 2019)

For all kinds of machine learning algorithms, and therefore also for policy gradient methods in RL, the subject of hyperparameter optimization is crucial. Learning algorithms are mainly controlled by a number of parameters that determine the learning rate, step sizes to be taken, discount factor gamma, clipping factors, and many more, and which are referred to as hyperparameters. (Rebala et al. 2019) (Schulman et al. 2017) (Zhang et al. 2021) These parameter settings are known to have a huge impact on the performance of the model, and therefore, they must be set according to the specific domain. (Zhang et al. 2021) The choice of the right combination of hyperparameters can be made by experts through experience as well as trial and error. Alternatively, there is increasing development in automating the hyperparameter search through a structured search of the possible settings, resulting in plug-and-play solutions for implementation.

For more information on RL, policy gradient methods, and machine learning methods in general, including the principle of ANNs, Rebala et al. (2019) provide a good overview. For an in-depth introduction to the working principle of policy gradient methods in particular and RL in general, the work of Sutton and Barto (2018) is a good reference.

3 Existing Approaches towards Reinforcement Learning for Inventory Control

Learn from yesterday, live for today, hope for tomorrow.
The important thing is not to stop questioning.
– Albert Einstein

As mentioned in Section 2.2, a wide variety of approaches exist for solving the replenishment problem. Therefore, this chapter presents a review of the literature considering the combination of RL and inventory control. The focus is on replenishment planning, which considers one product to be planned, as opposed to the joint replenishment problem, which involves a variety of products. Nevertheless, there are examples of this case, as it contains interesting approaches on other levels and should not be omitted.

A few approaches to solving the replenishment problem using RL already exist, with promising results. Most of them claim that their models work well, and in some cases even more effectively than some benchmark policies. Therefore, the following sections compare the existing approaches in terms of their modeling of the SC, which represents the environment for the RL setting; the modeling of the RL setting itself; and the data used for training and testing.

3.1 Modeling of the Supply Chain

All authors in the defined literature focus on SCs. The differences lie in the modeled characteristics of the SCs on the one hand and the centrality of the planning instance on the other. This seems to be of particular interest when SCs do not only consist of a single SC location but rather represent whole supply networks that consist of several stages and a varying number of nodes at each stage with their own and independent interests. The centrality of planning can therefore be determined by the level at which planning occurs. Centralized planning refers to a policy that receives input to make decisions for an entire supply network and makes decisions for all of the SC nodes that it contains. By contrast, decentralized approaches seek to incorporate different policies, such as one for each node, that find a good local decision for each node and work well together in a supply network. The training of the policies in the decentralized case could be performed together or separately. The simple assumption for centrally planned SCs is a shared reward function, whereas for decentralized SCs it is a non-shared reward function. This favors the advantages of the decentralized models as they could be trained separately and then assembled into a planning network at runtime.

As Table 3.1 indicates, a variety of approaches exist. Earlier approaches seem to favor centralized planning models, while more recent publications seem to favor decentralized planning more often. Nevertheless, earlier approaches, such as those of Chang Ouk. Kim et al. (2005), Chang Ouk Kim et al. (2008), Yang and Zhang (2015) and Xu et al. (2009)), have already compared a centralized with a decentralized SC.

'Classical' centralized approaches, such as those of Giannoccaro and Pontrandolfo (2002), Chaharsooghi et al. (2008), Sun and Zhao (2012), Prestwich et al. (2012), Hubbs et al. (2020) and Gokhale et al. (2021) model some kind of master planner who makes decisions for all nodes equally and interdependently. This explains the design of holistic reward functions for the whole SC, as there is only one decision-making instance.

The central approach in the studies of Chang Ouk. Kim et al. (2005), Chang Ouk Kim et al. (2008), Kwon et al. (2008), Sui et al. (2010) and Zarandi et al. (2013) is similar to VMI in that the second stage decides whether to ship to the various nodes in the first, more downstream stage. This concept leads to the abovementioned classical approach, as the authors have only considered two-stage SCs, in which case a VMI is conceptually a central planning instance. For larger SCs, the concept would need to be adapted.

In the decentralized case of the other publications, there is a decision maker for each SC node. Regarding the reward function, there are two possibilities during training: each agent attempts to optimize its own reward locally and could therefore be trained separately from the other agents, or different agents share a reward function similar to the one in the central case and are therefore trained together. Most of the mentioned studies have modeled the problem with separate reward functions. The exception is Meisheri et al. (2019), who model a more complex supply network that consists of a local distribution center and 1,000 retailers, each selling 220 products. To reduce complexity, a decentralized approach seems appropriate in their case. As far as hints could be found in their paper, Meisheri et al. (2019) train a policy that is implemented for each retailer that orders from the local distribution center and one of its products. The reward function for this policy includes a shared parameter that somehow connects the individual policies, in addition to product-specific rewards such as cost. The authors call the common parameter the fairness parameter to ensure similar inventory levels across all products, rather than some with low and others with high levels.

As Kara and Dogan (2018) consider the simplest SC consisting of one node at one stage, the distinction between a centralized and a decentralized approach cannot be made.

Regarding the SC models considered, the variety ranges from simple linear one-stage and one-node models in the study of Kara and Dogan (2018) to huge divergent models like the one of Meisheri et al. (2019). The SC structure of the considered models is mostly a linear or divergent one; according to current knowledge , no one has yet investigated a convergent one. A con-

vergent SC structure would correspond to an assembly system or similar. This may be due to the assumption that the development of a linear model into a convergent one is simply a matter of using the bill of materials and the product structure itself to calculate the appropriate replenishment quantities once the general quantity required has been determined.

Speaking of supply networks or even linear chains, the BWE is an interesting topic to investigate. Notably, only Yang and Zhang (2015) have analyzed the resulting BWE when applying their adaptive inventory control.

Most of the authors listed in Table 3.1 have considered a single product case that corresponds to the replenishment problem. Sui et al. (2010), Meisheri et al. (2019) and Gokhale et al. (2021) consider multiple products and therefore attempt to find a solution to the so-called joint replenishment problem.

Thus, differences in SC modeling lie in the consideration of replenishment times between different stages, how to deal with unfulfilled demand, and whether capacity constraints exist. For replenishment lead times, there is again a wide range, from zero lead times to stochastic lead times.

In reality, lead times are usually greater than zero due to organizational processes, such that if an order is placed on day t, in the best case with order acceptance by telephone, supplier availability, and immediate delivery processing, then the order will be delivered early the next day. (Tempelmeier 2015) In addition, the assumption of stochastic lead times is useful for events such as stock-outs at the supplier, stochastic process times of logistics processes, or upstream production processes. (Tempelmeier 2015)

Unmet demand can be dealt with in two ways: In the case of backordering, it is assumed that the customer is willing to wait for their product when a stock-out occurs. Backorders are therefore filled as quickly as possible. For sales organizations, this may not always be the case. This is why the concept of lost sales exists, which occur when the customer does not find the desired product available and therefore chooses an alternative. This is what happens in the study of Dogan and Güner (2015). It can be said that

it is not always easy to detect lost sales if the customer does not inform the supplier. (Zijm et al. 2019)

Capacity constraint is considered in some cases and for different objects: Giannoccaro and Pontrandolfo (2002), Hubbs et al. (2020) and Gokhale et al. (2021) have considered a warehouse with a limited number of storage locations, while Kwon et al. (2008) and Zarandi et al. (2013) have considered a limited production capacity, as do Hubbs et al. (2020). This is also the case for each node in the beer game setting considered in Mortazavi et al. (2015). Sui et al. (2010) and Meisheri et al. (2019) have modeled limited transport capacity (e.g., by the number of trucks), which works like batch sizes in the former and is expressed by volume and weight limits in the latter publication.

Regarding the different focus of the approaches, Jiang and Sheng (2009) and Dogan and Güner (2015) have integrated price sensitivity and pricing algorithms into their SCs, while Kara and Dogan (2018) integrate the age information of products and their expiry date. In both areas, namely price and product perishability, the literature could be expanded; however, it should be limited to these three to keep the focus on the present problem.

Reference	SC Structure	# Nodes/ Stage	Plan- ning	Shared Reward Function	# Prod- ucts	Replenishment Times	Unmet Demand	Capacitated
Giannoccaro and Pontrandolfo (2002)	L	(1, 1, 1)	C	Yes	1	L∼U(1, 3)	BO	Warehouse
Chang Ouk. Kim et al. (2005)	D	(1, 4)	Both	Both	1	Determ.	LS	-
Chaharsooghi et al. (2008)	L	(1, 1, 1, 1)	C	Yes	1	L∼U(0, 4)	BO	-
Chang Ouk Kim et al. (2008)	L	(1, 1)	Both	Both	1	Determ.	BO	-
Kwon et al. (2008)	D	(1, 4)	C	Yes	1	Determ.	LS	Manufacturer
Jiang and Sheng (2009)	D	(10, 80)	DC	No	1	Determ.	LS	-
Chang Ouk Kim et al. (2010)	L	(1, 1, 1) / (1, 1, 1, 1)	DC	No	1	Determ.	-	-
Sui et al. (2010)	D	(1, 10)	C	Yes	2	Determ.	LS	Transport
Sun and Zhao (2012)	L	(1, 1, 1, 1)	C	Yes	1	-	BO	-
Prestwich et al. (2012)	L / D	(1, 1, 1) / (1, 2)	C	Yes	1	L = 0	-	-
Zarandi et al. (2013)	D	(1, 4)	C	Yes	1	Determ.	BO	Supplier
Dogan and Güner (2015)	D	(1, 2)	DC	No	1	L = 0	LS / BO	-
Mortazavi et al. (2015)	L	(1, 1, 1, 1)	DC	No	1	L∼U(a, b)	BO	Each Node
Yang and Zhang (2015) (and Xu et al. (2009))	L	(1, 1)	Both	Both	1	Determ.	-	-
Kara and Dogan (2018)	L	(1)	-	-	1	Determ.	LS	-
Meisheri et al. (2019)	D	(1, 1000)	DC	Partially	220	-	LS	Transport
Hubbs et al. (2020)	L	(1, 1, 1, 1)	C	Yes	1	Determ.	LS / BO	Warehouses
Gokhale et al. (2021)	D	(1, 4)	C	Yes	100	Determ.	BO	Warehouses, Production

Table 3.1: Comparison of Literature Concerning Supply Chain Modeling

3.2 Modeling of a Reinforcement Learning System

The general RL setting described in Section 2.3 can be used to distinguish the different publications in terms of the training algorithm used, the function approximator for the value function, the design of the reward function, the action to be taken, and the input state representation.

Some of the approaches focus on the development of a specific training algorithm that should be adapted to the inventory management setting. This is the case for Chang Ouk Kim et al. (2008)), who develop an action–reward learning algorithm into an asynchronous one. The asynchronous part considers the fact that rewards for the same action can change over time. Furthermore, Kwon et al. (2008) extend a myopic RL algorithm to a case-based one, which they call case-based myopic RL. The idea behind it is to discretize the state space to more effectively deal with the otherwise huge state space. Developing this idea further,Jiang and Sheng (2009) seem to develop a case-based RL approach, which they call case-based RL. Moreover, Prestwich et al. (2012) develop a training algorithm based on an evolutionary algorithm to set the weights for the ANN that represent the value function.

The other group of publications focus on the RL setting and the modeling of the different parts; therefore, they make use of already known algorithms. This is the case for Giannoccaro and Pontrandolfo (2002), who use the SMART Algorithm of Das et al. (1999) and compare it with a benchmark policy. Furthermore, Chang Ouk. Kim et al. (2005), Chang Ouk Kim et al. (2010) and Yang and Zhang (2015)/Xu et al. (2009) have used an action–reward method, but they have not specified it in detail. There is a large fraction of Q-learning approaches by Chaharsooghi et al. (2008), Sui et al. (2010), Sun and Zhao (2012), Dogan and Güner (2015), Mortazavi et al. (2015) and Kara and Dogan (2018), with the latter pair also implementing the SARSA algorithm. Lastly, Zarandi et al. (2013) implement a temporal difference learning approach.

The more recent approaches of Meisheri et al. (2019), Hubbs et al. (2020) and Gokhale et al. (2021) have also incorporated the idea of actor–critic (A2C) algorithms. Learning algorithms are said to be actor–critic ones if they are able to learn approximations to both the policy and the value function. The actor corresponds to the learned policy, while the critic corresponds to the value function. (Rebala et al. 2019) Meisheri et al. (2019) adapt the A2C to its parallel computation setting, while Gokhale et al. (2021) compare the A2C algorithm with two other policy gradient methods, namely trust region policy optimization (TRPO) and proximal policy optimization (PPO). Hubbs et al. (2020) use a standard PPO training algorithm.

As explained in Section 2.3, the value function can be calculated exactly and stored in a table or something similar, or approximated by some kind of function approximator (e.g., an ANN). For publications where the function approximator is explicitly specified, the results can again be found in Table 3.2. The approach of Zarandi et al. (2013) stands out as they use a fuzzy rule-based function to approximate the value function.

The approaches can then be distinguished by the action the authors decide to take and the reward function that judges the goodness of the action taken. In the current literature, there are two main groups of actions taken. The different groups can be seen in Table 3.2. The first large group decides directly on the quantity to be ordered from the supplier at the level of the SC node. Sui et al. (2010) can be assigned to this group, as their model formally decides on the quantity sent from the distribution center to the retailers. However, as they assume a VMI approach, this corresponds to the quantities ordered by the retailers when adapted to a non-VMI approach. The second group decides on the safety lead time of the SC node, the idea behind which is to decide on a certain safety lead time given a certain state and to adjust the reorder points accordingly. The decision itself can be expressed in terms of the safety factor that the SC node needs to hold or the number of stocking units that need to be added to the reorder point.

Those decisions are judged by different measures, where again a distinction between two main groups can be drawn: the first group judges based on the cost provoked by the taken action. These cost functions differ in the

positions they consider. Most of the time, holding costs for stocked items and penalty costs for late deliveries are considered. Fixed ordering costs have only been integrated by Giannoccaro and Pontrandolfo (2002), Prestwich et al. (2012) and Mortazavi et al. (2015). Giannoccaro and Pontrandolfo (2002) and Mortazavi et al. (2015) additionally integrate transportation costs, whereas Sui et al. (2010) only add transportation cost. Furthermore, Hubbs et al. (2020) integrate ordering costs but not on a fixed basis and at the unit level. Moreover, Dogan and Güner (2015) and Gokhale et al. (2021)integrate manufacturing costs alongside additional cost parts, which when combined with the revenue made sum to the total profit function they consider as a reward. As Kara and Dogan (2018) and Meisheri et al. (2019) include the perishability of products, they also reward actions by the cost for items that have expired. Meisheri et al. (2019) additionally include a cost factor for a leveled order behavior.

Another way to assess the quality of the action taken is to examine the resulting service level. Here, the idea is to meet a predefined target service level, as opposed to generally rating higher service levels higher. This helps to limit inventory levels, which would otherwise become extremely high. Chang Ouk. Kim et al. (2005), Kwon et al. (2008), Jiang and Sheng (2009), Chang Ouk Kim et al. (2010) and Yang and Zhang (2015)/Xu et al. (2009) have chosen this option. Chang Ouk Kim et al. (2010) use the deviation from the target service level for the retailer node and the customer waiting time for order fulfillment for all upstream nodes as a reward signal.

Most of the aforementioned publications have used the stock level at the different SC nodes as a state description. This has also been done by Giannoccaro and Pontrandolfo (2002), Chaharsooghi et al. (2008), Kwon et al. (2008), Mortazavi et al. (2015) and Hubbs et al. (2020). The approaches distinguish between stocks in storage and stocks in transit. Some include the sum of both, while others include only on-hand stocks.

By contrast, Jiang and Sheng (2009) do not include any stock position at all, but rather only the actual reorder point and the expected demand. This is consistent with the approach of not deciding on order quantities, as many other authors have done, but on the reorder point. Sun and Zhao

(2012) and Zarandi et al. (2013)) include the backorder position in addition to the inventory position in the input state representation, while Prestwich et al. (2012) include time information in the form of the actual time period. Chang Ouk. Kim et al. (2005), Jiang and Sheng (2009), Chang Ouk Kim et al. (2010), Sui et al. (2010) and Yang and Zhang (2015)/Xu et al. (2009) have included historical demand data in their state description. They mostly use it to make some kind of time series forecast to derive future demand.

The input state representation advances when authors consider more specialized problems, such as in the following examples: Dogan and Güner (2015) consider a mixed ordering and pricing problem, and therefore, they include the previous term price. Kara and Dogan (2018) consider the perishability of products in their problem setting, and therefore, they include the age information in the state description. They also compare whether including age information in addition to stock levels in the input state representation improves the results.

Noteworthily, Meisheri et al. (2019) model the states in a much more elaborate manner than their predecessors. In addition to the inventory levels of each product, since they consider a multi-product problem, and the aggregated forecast of future demand as product-related inputs, they add a list of product metadata, such as the standard deviation of the forecast error, unit volume and weight, and an indicator of product deterioration due to not being called off by orders. As the last two inputs, the total volume and total weight of all planned products are modeled. This can be explained by the constraint of transport volume and weight that the authors add to their problem.

Reference	Training Algorithm	Function Approximator	Reward Function	Action	Input		
					IP	D_{hist}	Other
Giannoccaro and Pontrandolfo (2002)	Standard	-	Cost (F, T, H, P)	Order Amount	✓		
Chang Ouk. Kim et al. (2005)	Standard	-	Service Level	Safety Lead Time	✓	✓	
Chaharsooghi et al. (2008)	Standard	Q-Table	Cost (H, P)	Order Amount	✓		
Chang Ouk Kim et al. (2008)	Self-development	-	Cost (H, P)	Safety Lead Time			
Kwon et al. (2008)	Self-development	-	Service Level	Safety Factor	✓		
Jiang and Sheng (2009)	Self-development	-	Service Level	Order-Up-To-Level		✓	✓
Chang Ouk Kim et al. (2010)	Standard	-	Target Service Level, Waiting Time	Safety Lead Time	✓	✓	
Sui et al. (2010)	Standard	ANN	Profit (H, P, T)	Order Amount	✓	✓	
Sun and Zhao (2012)	Standard	-	Cost (H, P)	Order Amount	✓		
Prestwich et al. (2012)	Self-development	ANN	Cost (F, H, P)	Order Amount	✓		✓
Zarandi et al. (2013)	Standard	Fuzzy Rule-Based Function	Cost (H, P)	Order Amount	✓		
Dogan and Güner (2015)	Standard	-	Profit (H, P, M)	Order Amount, Price	✓		✓
Mortazavi et al. (2015)	Standard	Q-Table	Cost (F, T, H, P, M)	Order Amount	✓		
Yang and Zhang (2015) (and Xu et al. (2009))	Standard	-	Service Level	Safety Factor	✓	✓	
Kara and Dogan (2018)	Standard	-	Cost (P, E)	Order Amount	✓		✓
Meisheri et al. (2019)	Self-development	ANN	Cost (P, E, N)	✓	✓	✓	
Hubbs et al. (2020)	Standard	ANN	Profit (F, H, P)	Order Amount	✓		
Gokhale et al. (2021)	Standard	ANN	Revenue - Cost (H, P, M)	Order Amount	✓		

Table 3.2: Comparison of Literature Concerning RL in Inventory Control

3.3 Data

In most publications, it is not particularly clear which data are used for training and which are used for evaluation. Therefore, this section refers to data in general and assumes a reasonable split between training and evaluation data sets, unless otherwise specified. In all of the publications mentioned thus far, the demand data are the main varying component and therefore serve as training data, comparable to training data in supervised or unsupervised learning. The chosen demand scenarios can be distinguished by the underlying distributions, such as simple stationary, elaborated stationary, and non-stationary demand. Simple ones use fixed data sets; for example, Sun and Zhao (2012). Chaharsooghi et al. (2008) and Sui et al. (2010) use uniformly distributed demand, the latter with varying intervals, whereas Prestwich et al. (2012) add a stochastic summand to the deterministic demand, thus creating a discrete distribution of demand.

In addition, Dogan and Güner (2015), Hubbs et al. (2020), Kara and Dogan (2018) and Giannoccaro and Pontrandolfo (2002) create their data sets with two Poisson, one Gamma and one Erlang distributed random variables, respectively; thus, they fall into the stationary demand group. Giannoccaro and Pontrandolfo (2002) and Kara and Dogan (2018) work with a fixed mean while varying the shaping and scaling parameters to generate different demand distributions. Giannoccaro and Pontrandolfo (2002) also test agents trained on one demand pattern on another for robustness. Furthermore, Gokhale et al. (2021) use all three distributions to model demand while comparing the effect of the distributions on the convergence and speed of different training algorithms.

Moreover, Chang Ouk. Kim et al. (2005), Chang Ouk Kim et al. (2008), Kwon et al. (2008), Jiang and Sheng (2009), Chang Ouk Kim et al. (2010), Zarandi et al. (2013), Mortazavi et al. (2015), Yang and Zhang (2015) and Xu et al. (2009) have used non-stationary demand data for training. They have modeled demand as a random variable following a normal or Poisson distribution with randomly varying means and variance coefficients. This

variation is realized by a factor X~U(a,b)), which is added to the respective means and applied every T~U(c,d) intervals with a, b, c, d $\in \mathbb{N}$.

Real-world demand data have only been used by Meisheri et al. (2019)), who use a public data set from the bricks and mortar industry.

Despite the earlier assumption of a reasonable split between training and test data sets, it remains interesting to see whether the models were trained on one demand pattern and tested on another for robustness reasons. The short results are provided in the last column of Table 3.3. The vast majority of studies have used different demand scenarios in the form of non-stationary demand and/or changing variances or other parameters, such as lead times and costs. However, in the papers of Chang Ouk Kim et al. (2008), Kwon et al. (2008), Jiang and Sheng (2009), Chang Ouk Kim et al. (2010), Sui et al. (2010), Sun and Zhao (2012), Prestwich et al. (2012), Yang and Zhang (2015) and Kara and Dogan (2018)), it is not clear whether, for each scenario, a separate model was trained on that scenario, or whether a model was trained on one scenario and tested on others. The uncertainty may stem from the word *simulation*, which may be used in some publications as equivalent to the term *training*; however, there is still no clear distinction between the training and testing phases.

Chaharsooghi et al. (2008) describe how the same model was used for two scenarios, one with deterministic and the other with uniformly distributed demand, as well as for Mortazavi et al. (2015) with non-stationary demand. Some authors have focused on different research questions, such as Dogan and Güner (2015)), who investigate different retailer sales strategies; Meisheri et al. (2019), who use real data; and Gokhale et al. (2021), who focus on comparing training algorithms in combination with different demand distributions.

To the best of the present author's knowledge, only Giannoccaro and Pontrandolfo (2002) have explicitly stated that they trained a model on one demand pattern and then tested it on larger variances to test the robustness of the model.

Reference	Demand Distribution	Stationary	Testing of Robustness
Giannoccaro and Pontrandolfo (2002)	D~Erl($\frac{n}{\mu}$, $\frac{1}{CV^2}$)	Yes	Yes
Chang Ouk. Kim et al. (2005)	D~N(50, 10) and D~N(50+X, σ)	No	-
Chaharsooghi et al. (2008)	D~U(0, 15)	Yes	No
Chang Ouk Kim et al. (2008)	D~N(100+X, σ)	No	-
Kwon et al. 2008	D~N(50+X, σ)	No	-
Jiang and Sheng (2009)	D~N(20+X, σ)	No	-
Chang Ouk Kim et al. (2010)	D~N(200+X, σ	No	-
Sui et al. (2010)	D~Poi_{λ}	Yes	-
Sun and Zhao (2012)	Predefined Demand Time Series	Yes	-
Prestwich et al. (2012)	Discrete Demand	Yes	-
Zarandi et al. (2013)	D~N(50+X, σ)	No	No
Dogan and Güner (2015)	D~$Poi_{\lambda=5}$	Yes	No
Mortazavi et al. (2015)	D~$Poi_{\lambda+X}$	No	No
Yang and Zhang (2015) (and Xu et al. (2009))	D~N(50, 10) and D~N(50+X, σ)	No	-
Kara and Dogan (2018)	D~Gamma($\frac{1}{CV^2}$, $\frac{p}{\mu=20}$)	Yes	-
Meisheri et al. (2019)	Demand Data from Stores	-	No
Hubbs et al. (2020)	D~$Poi_{\lambda=20}$	Yes	No
Gokhale et al. (2021)	D~Poi_{λ}, N(μ, σ) and Gamma(p, b)	Yes	No

Table 3.3: Literature Comparison Data

3.4 Chapter Conclusion

This chapter has provided a detailed analysis of the current RL approaches to inventory management. The focus has been less on the algorithms used than on the modeling of the whole RL setting, such as the environment in the form of SCs, the input state representation, the design of the training, and the data. The following research gaps have been determined to exist in the current state of knowledge:

As far as the representation of the input state is concerned, a wide variety exists between different approaches, as demonstrated in Section 3.2. However, no investigations or comparisons have been conducted into how it should be designed for the application of RL in inventory management to achieve good results. Furthermore, in relation to the design of training and testing data, which data the models have been trained and then tested on is not particularly clear in the existing literature. An interesting question is whether a model trained on one demand scenario learns to abstract from it to another when tested.

Moving on to SCs and supply networks, some approaches have distinguished between centralized and decentralized training approaches. Again, a comparison of the two systems would be useful, as the possibility of decentralized, separate training would allow for easier adaptation to changing SCs. As the BWE is a well-known problem in SCs, the study of its evolution under the different approaches is an underrepresented topic in the literature.

As a final conclusion regarding the data used to test the developed models, it can be stated with certainty that the literary canon lacks the use of real data instead of systematically generated data. Created data seem to be useful because they can be controlled; therefore, certain conclusions can be drawn on a more artificial and thus solid basis. Nevertheless, models such as this one should also work for real-world applications. This problem is addressed later in Chapter 7.

4 Model Structure of Reinforcement Learning for Inventory Control

The secret of modeling is not being perfect.
– Karl Lagerfeld

Following the schematic structure of RL depicted in Figure 2.2 and explained in Section 2.3, the aim of this chapter is to describe the general setting, assumptions, and modeling choices for using RC for inventory management. It starts with a short description of the environment, namely the SC model and the possibility of creating synthetic data to obtain a structured training and testing scenario later. This is followed by a description of the agent, the actions it could take, and the corresponding learning algorithm with parameter settings and a reward function. The chapter concludes with a description of the implementation of the whole model and the analysis set up for the following chapters.

4.1 The Environment

The environment is an SC model, the characteristics of which are described in Section 4.1.1. Then, the method of generating synthetic training and test data and the justification of the range of values of the model parameters are discussed in Section 4.1.2.

4.1.1 The Supply Chain Model

The underlying theory of SC modeling is based on the understanding of models of the organization Verein Deutscher Ingenieure (2014), which states that a model is a simplified representation of an existing or planned system. Simplifications are made according to the characteristics relevant to the study. The present model is characterised according to the characteristics of SC models in Section 3.1. A schematic representation can be found in Figure 4.1.

In an abstract way, the actors in an SC can be described as individual SC nodes n, all of which act according to the following properties: each node faces a demand $d_{n,t}$ from the downstream node in period t and orders material, semi-finished or finished goods, hereinafter referred to as material in general, of size $o_{n,t}$ from the upstream node. From a node's point of view, upstream nodes are referred to as suppliers, while downstream nodes are referred to as (internal or external) customers. The stocks at each node $I_{n,t}$ are affected by these material movements such that demands from the customer node have a negative effect on stocks while realized orders placed at the supplier node have a positive effect on stocks.

In the following, the notation for the different types of stocks from Tempelmeier (2015) is used: I_t^p is the stock that physically exists in the warehouse. When demand arrives and is confronted with an on-hand inventory of zero, this results in backorders. Backlogs are accumulated in the backlog inventory I_t^f. Orders that have already been placed but have not yet arrived are referred to as ordered stock I_t^o. The calculated available stock I_t^d then results from $I_t^p - I_t^f + I_t^o$.

External customer demand arrives on a discrete time axis (e.g., daily or weekly) and follows a discretised gamma distribution. For example, Axsäter (2015) and Tempelmeier (2015) propose using a gamma distribution for demand modeling. Compared with normally distributed demand, realisations of gamma-distributed random numbers are non-negative. In addition, the gamma distribution is highly versatile with its two shape and scale parame-

ters for modeling other distributions, such as the Erlang distribution, as its special cases.

Similar to the concept of scheduling agreements in the automotive sector, the external end customer provides a forecast of its demand for a given forecast interval. Once a demand amount is forecast, it is not fixed, but it can still change until the call-off day, which is modeled using the forecast error. The forecast error follows a discretized normal distribution and is modeled as a function of average demand.

The demand cannot be influenced by marketing promotions or other similar activities. Demand is fulfilled from stock; therefore, a make-to-stock model is considered.

Transport times from one node to another, production times, and possible downtimes are aggregated in the replenishment time. Replenishment times are stochastic and follow a discretized gamma distribution. As Tempelmeier (2015) mentions, replenishment times are often modeled as a deterministic parameter. This is not the case in reality, where lead times are mostly stochastic and unplanned interruptions occur.

Demand and replenishment lead times are treated as independent random variables. Unsatisfied demand does not result in lost sales, but it is backordered and satisfied as soon as stock levels permit. Partial backorders are allowed. Warehouse capacity is not limited, allowing for unlimited inventory. The first and most upstream node in the observed SC is faced with an unlimited supply of material; thus, material availability is guaranteed and only delayed by replenishment lead times.

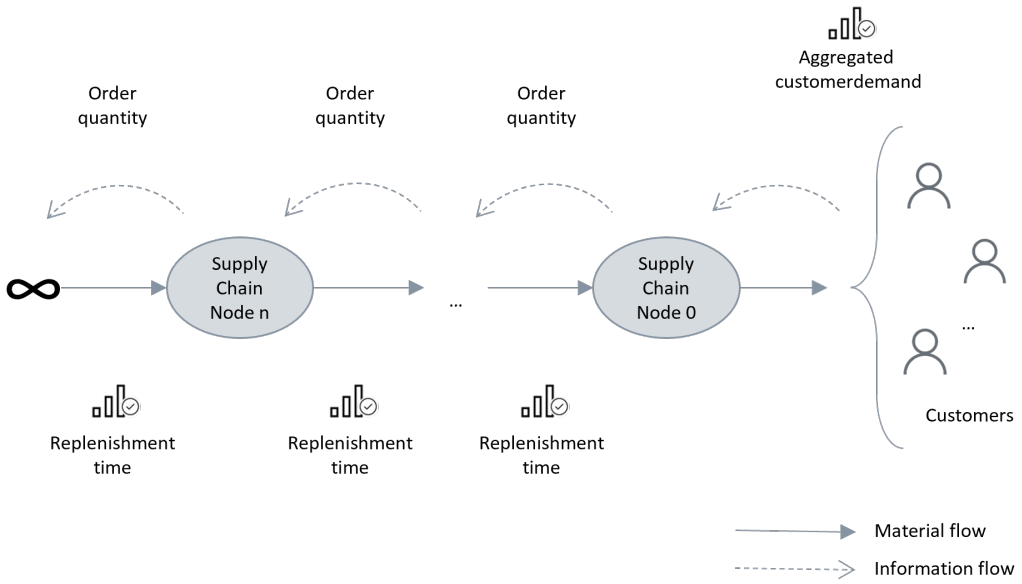

Figure 4.1: Schematic Linear SC Model

By including the possibility of forecast errors as well as stochastic lead times, the present approach differs from most of the approaches in Section 3.1; however, it makes the modeling a little more realistic and the resulting model a little more applicable to reality. According to Davis (1993) and Verwijmeren et al. (1996), the main sources of uncertainty in SCs are those that arise from supplier performance, manufacturing, transport processes, and customer demand. Integrating stochastic lead times covers the uncertainties in manufacturing and transportation processes as well as those that arise from supplier performance. Uncertainty in customer demand is modeled by not only stochastic demand but also forecasting errors, which are empirical in many SCs.

According to the nature of RL, the simulation of the SC is conducted in discrete time steps t, starting with the decision taken by the agent and ending with the execution of the same. The aforementioned logic in updating the different quantities, such as demand, ordered and received quantities, as well as stock levels, can be observed in the schematic representation of the simulation process in Figure 4.1.1. Today's orders $o_{n,t}$ are stored as future stock, also known as order stock.

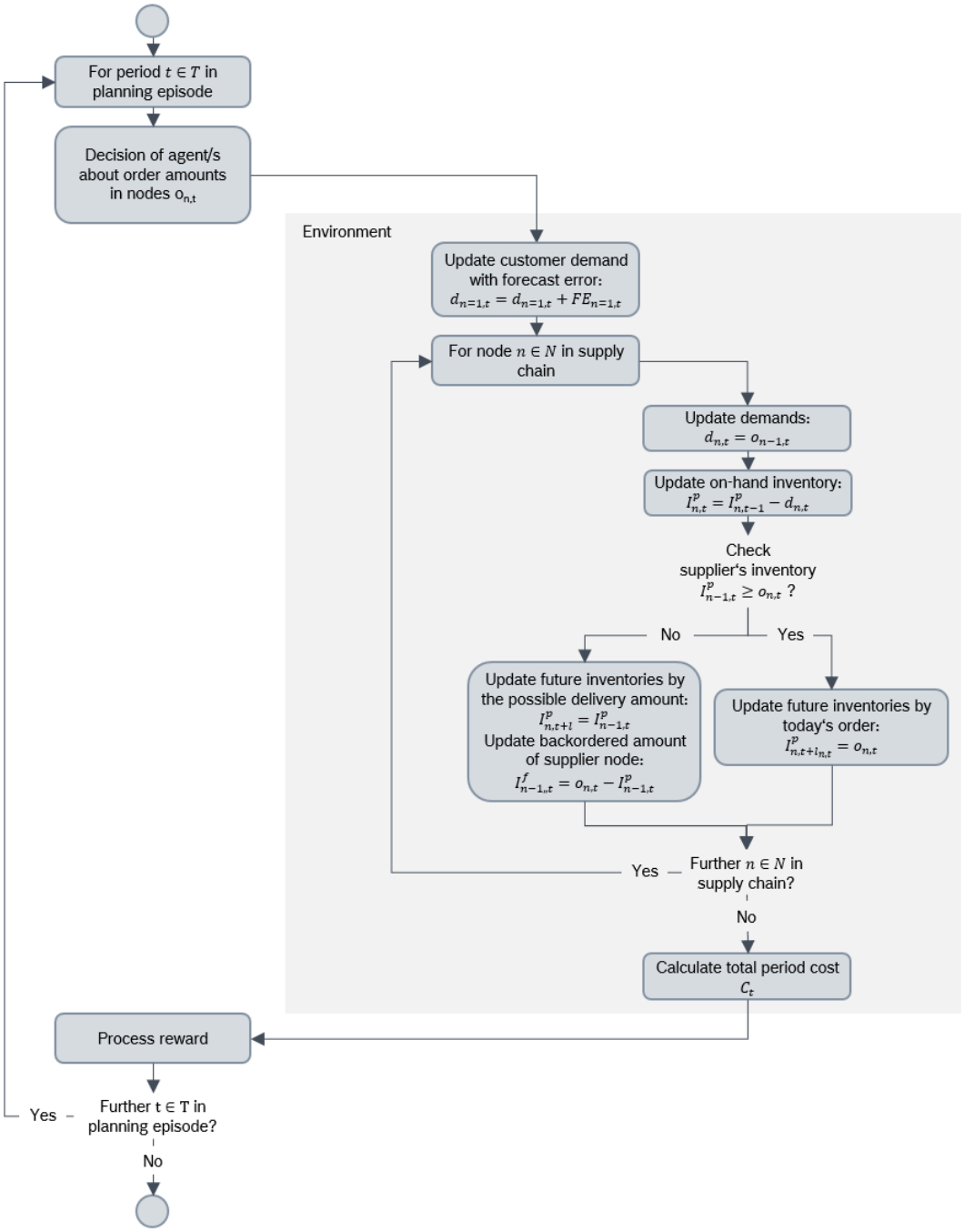

Figure 4.2: Time-Discrete Simulation of a SC

4.1.2 Synthetic Data Creation

Throughout most of the present work, the data are generated synthetically to control the different characteristics to be analysed during the parameter study. Nevertheless, the model is applied to a real-world example in Chapter 7. As explained earlier, a gamma distribution is used to model demand quantities and replenishment times. To obtain a variety of data scenarios, time series of customer demand and replenishment lead times are created by inputting the desired mean and coefficient of variation (CV) of the distribution. The input is mapped to the specific shape and scale parameters of the gamma distribution using Equation 4.3 and Equation 4.4.

With the CV of a distribution of the random variable X being

$$CV(X) = \frac{\sqrt{Var(X)}}{E(X)} \tag{4.1}$$

and the mean and variance of a gamma distribution in dependence of the gamma-specific shape and scale parameter p and b being

$$E(X) = \frac{p}{b}, Var(X) = \frac{p}{b^2} \; for \; X \sim Gamma(p, b) \tag{4.2}$$

the shape parameter p and the scale parameter b result in the following:

$$CV(X) = \frac{\sqrt{(Var(X)}}{E(X)} = \frac{\sqrt{\frac{p}{b^2}}}{\frac{p}{b}} = \frac{1}{\sqrt{p}}$$

$$\Leftrightarrow p = \frac{1}{CV(X)^2} \tag{4.3}$$

$$E(X) = \frac{p}{b} \Leftrightarrow b = \frac{p}{E(X)} \tag{4.4}$$

The influence of the mean and CV can be seen in Figure 4.3, which presents the probability density functions (PDFs) of gamma-distributed variables. The mean remains the same for all curves, while the CV is varied according to the legend provided. For a constant mean $E(X)$, a lower

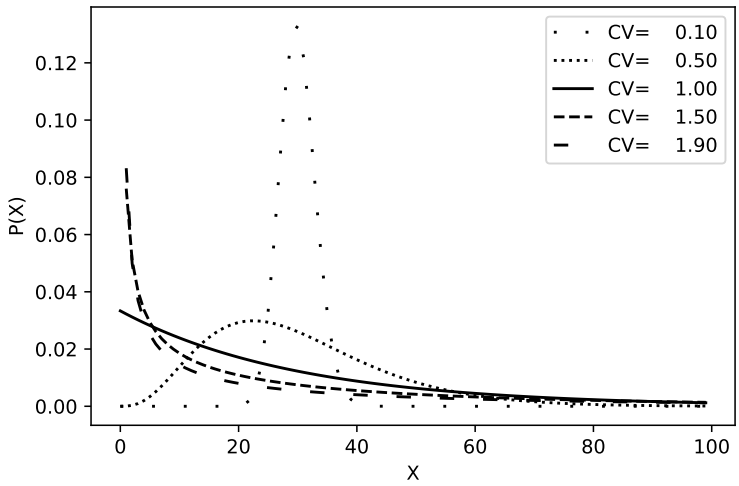

Figure 4.3: PDF of Gamma Distribution with $E(X) = 30.0$ and Varying CVs

$CV(X)$ implies a lower variance $Var(X)$ compared with the mean. Conversely, a higher $CV(X)$ implies a higher variance compared with the mean.

The discretization of the distributions occurs to account for the discrete nature of the products demanded or the full time units, such as hours, days, or weeks, required for replenishment. Discretization is performed by adding the probabilities to the integers until a certain probability mass $1-\varepsilon$ is reached. The remaining ε is distributed equally among the m entries of the probability vector:

$$P(X = x) = F(x + 0.5) - F(x - 0.5) + \frac{\varepsilon}{m} \tag{4.5}$$

In the case of the gamma-distributed replenishment times, a replenishment time of zero is not considered. The probability is therefore removed and added to the remaining entries according to their probability ratios:

$$P(X = x) = P(X = x) + P(X = 0)\frac{P(X = x)}{\sum_{i=1}^{m} P(X = i)} \tag{4.6}$$

The forecast error is modeled as a normally distributed random number with $FE \sim N(\mu = 0, \sigma)$. The limits of the range of values or maximum devi-

ations that the forecast error takes $\mp FE_{max}$ are ratios of the mean demand and determine σ, which satisfies the following relation:

$$FE_{max} \stackrel{!}{=} F^{-1}(1 - \frac{\varepsilon}{2}) \tag{4.7}$$

4.2 The Agent and its Actions

Using the replenishment problem in Section 2.2 as a theoretical basis, the agent can be given different decision scopes for different problem instances. Problem instances differ in the number and connectivity of SC nodes. Various approaches to the scope of an agent exist. When considering SCs with more than one SC node, there could be one agent per SC node, which would result in a scenario with multiple decentralized decisions. Such a scenario was described along with its advantages and disadvantages in Section 3.1. The alternative, also described in Section 3.2, is a centralized approach where one agent orders for several SC nodes.

Regardless of the central planning setting, the actions or decisions taken by the agent remain the same - they concern the replenishment quantity $o_{n,t}$ for node n in period t. In decentralized approaches, the decision consists of a single number, whereas the centralized approach outputs a vector of numbers. Thus, the agent is introduced to the actual state of the SC and decides according to its developed policy. The action space is limited by a maximum order amount $o_{n,t}^{max}$.

4.3 Training the RL Model

Following the advantages already explained in Section 2.3, the study of Gokhale et al. (2021) and several previous studies, the agent's policy in the present model is trained by the so-called PPO. In terms of the aforementioned categories, it could be classified as a policy gradient method with an

actor-critical approach. Gokhale et al. (2021) state that PPO is relatively robust to different demand distributions when applied in an inventory control setting. More generally, Schulman et al. (2017) confirm its applicability to more general problem settings while still performing well. This is achieved by the stability of a TRPO combined with much simpler implementation possibilities, which is introduced by Schulman et al. (2017) and called PPO.

In general, the difference between standard supervised learning and RL is that supervised learning uses a static training data set. RL instead uses training data that are constantly generated by the agent as it interacts with the environment. Thus, the training data is constantly changing, which can lead to instability in the training process. This generally makes training stability a problem in RL. With instability in the training data, huge changes in the policy are made, which could destroy the previous training effort. TRPO is a group of learning algorithms that address this problem. The general idea is to not overdo the policy updates and thus promote a more stable training process. For this reason, TRPO implements a so-called KL constraint along with the objective function, which limits the updating of the policy to a narrow, so-called trust region. Instead of a separate constraint, PPO includes this constraint in the objective function, which is much easier to implement. (Schulman et al. 2017)

Compared with offline learning, online learning algorithms are not as sample-efficient. Where offline learning benefits from an experience buffer that is used multiple times for policy updates, online learning algorithms use a sample once and then move on to the next sample. (AurelianTactics 2018) PPO additionally approaches this issue through two parallel working threads and is thus able to act in a more sample-efficient manner: in the first thread, several agents generate samples of experience episodes. These sampled episodes are used from occasionally in a second thread for a common policy update. (Schulman et al. 2017)

These facts have made PPO a state-of-the-art training algorithm in recent years; therefore, it is the preferred alternative for the present work.

4.4 Reward Function

Rewards are vital in developing a sound policy that aligns with the problem-solving objectives. The present model defines rewards on a per-action and per-step basis in the environment. As in many other approaches described in Section 3.2, the accumulated SC cost C_t of a single period is the reward signal. Using a cost function offers the advantage of unifying the inventory and service level dimensions. Tests have indicated, and the logical explanation suggests, that using only one of the objectives as a reward signal, either low inventory levels or high service levels, leads the model to satisfy only that specific objective while ignoring the other. Furthermore, preliminary studies have demonstrated that, for the current scenario, a detailed reward system, such as costs, is more effective than simpler rewards, such as ± 1 or comparable options. Modeling rewards in this manner could be advantageous in situations where specific requirements exist for a target level of service, or for a minimum or maximum stock level, for example.

In accordance with experts in the field of inventory control and the overall goals of inventory management, the following cost components are incorporated: holding costs of h_n accrue for every item stocked during planning period t and each node n, as well as backorder costs of b_n for every item not delivered on time by node n and during period t, resulting in a shortage of inventory $I^f_{n,t}$. It is important to acknowledge that placing orders incurs fixed ordering costs of K_n per order at node n during period t. This cost is independent of the order quantity and is applied once per order. The cost for one period is then calculated as the sum of costs for all nodes in the SC:

$$C_t = \sum_{n=0}^{N} h_n(I^p_{n,t}) + b_n(I^f_{n,t}) + K_n * \mathbb{1}_{\{o_{n,t}>0\}}(o_{n,t}) \tag{4.8}$$

4.5 Implementation

The aforementioned aspects of the RL setting were implemented within the framework of OpenAI, an organization that conducts research in the field of artificial intelligence. This non-profit organization is controlled by investors such as Microsoft. (OpenAI 2021)

OpenAI has created a framework that simplifies the implementation of various RL applications. This is facilitated by an environment interface, pre-existing and accelerated training algorithms, and a hyperparameter optimization solution. The details of the environment interface are outlined in Brockman et al. (2016a), and corresponding code examples are available in Brockman et al. (2016b). A version of the aforementioned PPO by Schulman et al. (2017) has already been implemented and tested in *Stable Baselines*, a fork of OpenAI's baseline algorithms. The related code is available in the work of Hill et al. (2018). Adopting established running algorithms and frameworks is considered beneficial for reducing errors and the time spent on implementing already validated discoveries. The present study specifically focuses on the use of training algorithms for particular environments, instead of their further development.

As the policy network, the pre-existing MlpPolicy of *Stable Baselines* is used. The activation function used in this network is the tangent hyperbolicus (tanh). This network comprises a standard multilayer perceptron that includes two hidden layers, each of which contains 64 neurons. (Hill et al. 2018) The goal was to commence with a relatively uncomplicated and standard network. While larger and more complex networks were also tested, they did not yield better results.

Hyperparameter optimization for the algorithms of *Stable Baselines* can be performed using the *Optuna* package. The package provides an efficient framework for hyperparameter optimisation studies by defining the parameters to be optimized and their value ranges. (Akiba et al. 2019) *Stable Baselines RL Zoo*, OpenAI's *Spinning Up* documentation, and AurelianTactics (2018) served as expert input for the chosen hyperparameters and value ranges documented in Table 4.1.

Parameter	Distribution	Value Range
n_steps	Categorical	{32, 64, 128, 256}
batch_size	Categorical	{16, 32, 64, 128, 256, 512, 1024, 2048}
nminibatches	-	$max(1, \lfloor \frac{n\_steps}{batch\_size} \rfloor)$
noptepochs	Categorical	{1, 5, 10, 20, 30, 50}
cliprange	Log-Uniform	[0.1, 0.4]
gamma	Log-Uniform	[0.9, 0.9999]
lambda	Uniform	[0.8, 1.0]
ent_coef	Log-Uniform	[1e-8, 0.1]
learning_rate	Log-Uniform	[1e-5, 1.0]

Table 4.1: Hyperparameters to Optimize Using *Optuna* and the Corresponding Value Ranges

When updating a policy, it is crucial to determine how many experience episodes should be sampled before the policy is updated. This behavior is guided by the parameters *minibatches*, *n_steps*, and *noptepochs*. Experience is collected during *n_steps*, and then a stochastic gradient descent is performed of *minibatches* size for *noptepochs* epochs. (AurelianTactics 2018) (Schulman et al. 2017)

The following hyperparameters govern the progression from the old to the new policy: *cliprange* specifies the allowable margin between the new and old policies, and *lambda* and *gamma* are applied in the *generalized advantage estimation* to shape rewards further. (Schulman et al. 2015)

The *ent_coef* parameter determines the entropy term, which fosters exploration and prevents early convergence. The final optimization is for the *learning_rate*, which measures the strength of the weights' update for the underlying ANN. (Hill et al. 2018) (Kingma and Ba 2014)

4.6 General Setup for Analysis

This section presents the framework for assessing the model's performance. In the following subsections, relevant performance metrics, the validated statistical structure, and the parameter configurations are delineated.

4.6.1 Performance Indicators

As introduced in Section 2.1 and Section 2.2, the primary goals for an SC are to meet customer demand and to maximise efficiency. In subsequent chapters, the analysis mainly focuses on the achieved service level and the mean cost per period, while average stock levels, order frequencies, and order sizes are also considered when deemed relevant.

The event-oriented α service level, as defined by Tempelmeier (2000), is used to measure the service level of each node. This measures the probability that a customer demand D_t during period t can be fulfilled using the physical stock I_t^p during the same period t.

$$\alpha_t = P(D_t \leq I_t^p). \tag{4.9}$$

Mean costs are calculated as the average cost per planning period, denoted by t:

$$\overline{C} = \frac{\sum_{t \in T} \sum_{n \in N} C_{n,t}}{T}. \tag{4.10}$$

In certain instances, the mean demand during the corresponding planning episode is used to scale the average costs $\overline{C}$ to obtain the cost efficiency per requested unit:

$$\overline{C_{Eff}} = \frac{\overline{C}}{E(D)}. \tag{4.11}$$

A comparable efficiency may be achieved by dividing the average stock levels, denoted by $\overline{I^p}$, by the mean replenishment time, resulting in a cost KPI that is independent of replenishment times:

$$\overline{I^p_{Eff}} = \frac{\overline{I^p}}{E(L)}. \tag{4.12}$$

Both equations (Equation 4.11 and Equation 4.12) allow for the reduction of the KPI by the influencing factor, resulting in comparable average values.

4.6.2 Presumptions Regarding Model Behavior

For the purpose of system analysis, the developed models are expected to exhibit specific behavior. Therefore, the following assumptions regarding the model behavior were defined, taking the various parameters of the SC into consideration:

(P0): Increasing demand results in a rise in the total average cost as there are more units within the system that require handling and stocking. Conversely, decreasing demand results in a decrease in the total average cost. However, this should not suggest a change in the average cost per unit. It is expected that the model would induce a stable or reduced average cost per unit $\frac{\overline{C}}{E(D)}$ due to the inclusion of scaling effects.

(P1): The SC faces a potential risk in the growing variance of demand. A model would be expected to deal with this risk, such as by stocking extra units. However, this solution is expected not to result in a reduction of the service level. On the other hand, in the case of falling demand variance, a stable service level should be maintained.

(P2): Following Little's law, the number of items in the system changes with the time in the system. (Arnold and Furmans 2009) Hence, a longer or shorter replenishment time would lead to an increase or decrease in the number of items in the system, respectively. Presently, the inventory system's units equate to stock levels. However, the system's average inventory level per replenishment time unit ($\frac{\overline{IP}}{E(L)}$) is expected to remain stable, ensuring a proportionate response of the system.

(P3): A fluctuation in replenishment times presents the next risk to the SC. The system's behavior is expected to lead to consistent service levels.

(P4): The cost parameters' relationship determines the respective significance of stock levels, order frequency, and service level. As a result, the connection of these cost parameters sets targets for how to reduce expenses:

(a) High holding costs compared with the other cost parameters lead to lower inventory levels;

(b) High fixed ordering costs lead to fewer orders, resulting in higher ordering volumes and inventory levels;

(c) High backorder costs lead to a high level of service.

4.6.3 Statistical Validation of Testing

Here, simulation runs and parameter combinations are considered. A single simulation run entails the simulation of one SC system over a predefined episode that comprises T time periods. Parameter values remain constant during a given run, with only random number realizations differing due to the use of several different seed values for the random number generator. A scenario denotes a distinct parameter combination and is scrutinized over 385 simulation runs. The size of the dataset was collected with a 95% confidence level and a 5% margin of error. The mean performance indicators for one scenario were derived from 385 runs.

To demonstrate that the sample size of 385 is sufficient for obtaining reliable results, the relative standard error of mean is calculated and analysed for all scenarios and resulting performance indicators in the respective chapters. The definition of relative standard errors is defined using

$$SE_x = \frac{s}{\overline{x}\sqrt{N}} \tag{4.13}$$

where s is the sample standard deviation, $\overline{x}$ is the sample mean, and N equals 385 for each scenario. The considered performance indicators are the mean cost per period and the service level.

	Letter	Parameter Value
Number of SC Locations	N	{1, 3}
Time Horizon of One Episode	T	170
Forecast Interval	f	30
For Distribution Discretization	ε	0.01

Table 4.2: General SC Settings for the Experiments

4.6.4 Parameter Setting

The SC parameters presented in Table 4.2 determine diverse training and testing scenarios. These parameters are derived from the SC model specified in Section 4.1:

The number of nodes included in the linear SC is determined by N. This value varies depending on the SC under analysis. For the first and basic scenario when examining the model's robustness, N is set to 1. However, when considering a linear SC, N is set to 3. A planning episode, which has a planning horizon of T periods, comprises customer demand forecast obtained by the agent who has the ability to anticipate f periods ahead. The time horizon of one episode is set at 170 periods. Assuming that days are the unit of these periods, this equates to approximately half a year, which is considered a reasonable planning horizon based on past experience. This planning horizon coincides with a forecasting horizon of 30 periods or one month. The chosen horizon is justified by the fact that fixed horizons in today's Enterprise Ressource Planning (ERP) systems for certain products typically vary between two and four weeks.

The threshold for truncating the distributions is $1 - \varepsilon$ when determining the distributions of the unique probabilistic values and discretizing them.

Customer demand is characterized by its mean $E(D)$ and coefficient of variation $CV(D)$, while the mean $E(L)$ and coefficient of variation $CV(L)$ describe the replenishment time distribution. A number between 0 and 1 characterizes the ratio at which the forecast error deviates most from the mean customer demand. Lastly, parameters such as holding cost h, fixed order cost K, and backorder cost b describe the cost structure; h and b rep-

resent the monetary amounts (MUs) to be paid per item when in stock or backordered, respectively; and K denotes the MUs per order.

Two distinct testing scenarios are chosen. The first involves testing on the parameter combination that the model was trained on, which is referred to as the *original* data set. The second scenario involves systematically varying parameter values in different ranges to assess the model's response to changing environmental factors. Certain values for each data set and parameter are listed in Table 4.3. The purpose of utilizing these two distinct data sets is to scrutinize the model's resilience, which is outlined in the following chapter. Research question 2, which investigates the model's efficacy when the parameter values deviate significantly from the known values, provided the impetus for this inquiry. Hence, the values selected for parameter variation mainly differ in terms of the order of magnitude.

The model was trained using the original data set, where the parameter settings were inspired by numerical examples in various publications but were chosen randomly. Nevertheless, the specific values were selected in a manner that accommodated variations in lower and higher dimensions. The average demand of 30 appears to be a suitable starting point, as demands in the range of 10 or in the hundreds may be lower or higher, respectively. Similarly, average replenishment times were determined: As an average replenishment time of 7 days might be realistic for scenarios where suppliers and production are based in Europe and products are readily available, lower replenishment times could be achieved by partnering with local SC providers. By contrast, over-sea transport and extended production times contribute to longer replenishment periods. An appropriate starting point for the CVs would be 1.0, which allows for variability in a lower and higher value range. The cost parameters were largely influenced by the works of Axsäter (2015), Tempelmeier (2015), and Goodwin and Franklin (1994). These were then simplified into the current cost ratios.

The study involved a single- and a two-factorial experiment to investigate parameter variation. The first experiment focused on single factors to draw initial conclusions. The latter experiment examined the interdependency

	Assumed Distribution	Parameter	Parameter Value	
			Original	Parameter Variations
Demand	Gamma	$E(D)$	{30}	{1, 5, 10, 50, 100, 500, 1000}
		$CV(D)$	{1.0}	{0, 0.5, 1.0, 1.5, 2.0, 2.5}
Replenishment Times	Gamma	$E(L)$	{7}	{1, 5, 10, 15, 20, 50}
		$CV(L)$	{1.0}	{0, 0.5, 1.0, 1.5, 2.0, 2.5}
Holding Cost	Deterministic	h	{1}	{0, 1, 10, 50, 100, 500, 1000}
Fixed Order Cost	Deterministic	K	{10}	{0, 1, 10, 50, 100, 500, 1000}
Backorder Cost	Deterministic	b	{100}	{0, 10, 50, 100, 500, 1000, 5000, 10000}

Table 4.3: SC Parameter Value Ranges for Creating the Original Data Set and Further Parameter Variations

between certain factors to verify the previous adaptations of the model under more varied circumstances.

4.6.5 Benchmark Algorithm

A variety of different approaches exist for finding a solution to the replenishment problem. Most of them are to be found in the area of operations research that seeks elaborated mathematical solutions or provides certain heuristics. Several inventory policies could be considered standard policies for how much and at what time to restock. These are the fundamental policies that every student of logistics learns initially, which many ERP systems continue to rely upon. These policies serve as a basic comparison as they are well-known, straightforward, and essential.

The policies are defined by four parameters, namely r, s, S, and q. Here, r and s are responsible for triggering a single order, while S and q specify the quantity to be ordered. The control of stock levels is based on r time intervals, and s can be construed as the minimal stock level that activates order placement once it has fallen below that level. The number of items to be ordered varies depending on whether it is required to top up the stocks to reach S or if it follows a standard order amount of q. In various permutations, the parameters create specific replenishment strategies, such as the (r, s, q) policy considered here. (Tempelmeier 2015)

Review periods later are set to daily intervals of discrete time, such that they are equivalent to the conditions of the RL model. An iterative proce-

dure for fixing the order quantity q and then the reorder point s is recommended by Tempelmeier (2015). Initially, q^* is determined by the economic order quantity (EOQ) formula, which depends on the fixed order cost K, inventory holding cost h, and demand D within the considered time unit. Axsäter (2015) takes a more lenient approach by assuming that the demand for the planning interval cannot be predicted. Instead, the expected average demand in the given time unit is used. Another extension of the classical EOQ formula integrates backordering costs b per unit and time, which is introduced in Axsäter (2015). This finally results in q^*:

$$q^* = \sqrt{\frac{2KE(D)(h+b)}{hb}} \tag{4.14}$$

The reorder point, denoted by s, serves as a safeguard against demand that may arise during the risk interval of the replenishment time frame. This occurs when a new order has been triggered due to the falling below of s while the order has not yet been fulfilled. When the demand and replenishment time are present as two separate distributions, the demand distribution during replenishment time D' can be obtained by convolving the demand distribution for every possible replenishment time and then weighting the result with the probability of each replenishment time (Tempelmeier 2015):

$$P\{D' \leq d'\} = \sum_{l=l_{min}}^{l_{max}} P\{D \leq d | L = l\} P\{L = l\} \tag{4.15}$$

A secondary factor must also be considered; that is, when the stock level falls below s, the available stock does not always match it precisely, and there may already be a difference. Tempelmeier (2015) refers to this shortfall as undershoot and suggests adding the probability of undershoot U to the demand probability during the replenishment time D' to determine the final value of s. He calculates the probability of undershoot U as follows:

$$P\{U = u\} = \frac{1 - P\{D \leq u\}}{E(D)} \tag{4.16}$$

The ultimate probability distribution, denoting the demand during replenishment time and accounting for the likelihood of undershoot, can be represented by the final random variable $D'' = D' + U$, which is constructed by convolving both random variables D' and U. (Tempelmeier 2015)

Finally, the service level α^* to be achieved must be considered when determining the value of s. As a result, s is determined based on the accumulated probability of the vector entry of D'' that achieves the desired service level. In contrast to Tempelmeier (2015), the α service level is used in the present study instead of the β service level. This might result in slightly worse service levels, as the α service level is known as the stricter service level measurement.

$$P\{D'' \leq s\} \overset{!}{\geq} \alpha^* \tag{4.17}$$

Following Tempelmeier (2015), the (r, s, q) policy has advantages due to its combination of the (s, q) and (r, S) policies. The use of a fixed ordering amount q stabilises order sizes, while limiting reordering and review points to discrete values allows coordination with other processes.

However, it should be noted that this inventory policy operates under different framework conditions than the proposed RL approach. The conventional approach uses a target service level as input, which must be met every time. By contrast, the RL model can prioritize other optimization objectives over the service level. This is particularly relevant when considering changing cost parameters during later analysis phases. When backorder costs are comparatively low in relation to the other two parameters, the (r, s, q) strategy would still endeavour to achieve the target service level at a higher resulting cost. By contrast, the RL approach permits service levels to decrease to reduce costs. However, this deviation primarily applies in scenarios that involve changes to cost parameters.

4.7 Chapter Conclusion

This chapter has outlined the fundamental RL model for inventory control. Subsequent research and developments build upon these core principles. The creation of synthetic data facilitates the design of future training and testing scenarios. However, an intriguing aspect related to the action and state space remains unexplored, which is addressed in Chapter 5.

5 A Robust Reinforcement Learning Model for Inventory Control

Jeder Mensch hat ein Brett vor dem Kopf – es kommt nur auf die Entfernung an.
– Marie von Ebner-Eschenbach

Current machine learning models are usually designed for specific use cases and data sets, resulting in the need for retraining when they are applied to diverse scenarios or data. Given that adaptation training is time-intensive, this chapter's motivation is to develop a model that can adjust to various SC environments without requiring retraining and could thus be designated as *robust*.

First, a brief definition of the term *robustness* is presented, followed by various expansions of the fundamental model outlined in Chapter 4.

5.1 Robustness

Following the definition of robust production systems by Stricker and Lanza (2014), robustness necessitates that a production system can operate at high performance levels despite disturbances. Disturbances are characterized as changes in customer demand or general variations in process times due to unforeseen events, which are similar to the aforementioned variations in demand and replenishment times (Stricker and Lanza 2014).

Furthermore, this study considers varying cost structures to encompass a wide range of materials. It is crucial to note that different materials may have diverse underlying cost structures, such as in cases where the purchase price and holding cost significantly exceed the transportation cost.

5.2 Extension of the Basic Model Through Adaptive State and Action Space Scaling

The function of ANNs necessitates the scaling of input and output values for optimal performance. This is due to the deployment of activation functions, which are usually assigned within individual neurons, and typically include the sigmoid or tanh function. The function range of both is constrained to a relatively small interval, as illustrated by Figure 5.1. The sigmoid function is mapped to the interval between $(0, 1)$, while the tanh function is mapped to $(-1, 1)$. As can be seen from the graphs, the interval for the x-axis value is severely restricted, which makes it difficult to obtain y-values with proper differentiation. Thus, for either of the two activation functions employed, preprocessing the input data and the output of the ANNs through scaling, normalization, or standardization is recommended.

Furthermore, the transformation of inputs enables equal weighting of originally disparate parameters. By standardizing the input values, the influence of larger parameters is reduced to prevent a disproportionate influence on the outcome.

The analysis of this approach is initially conducted using a model whose state space is solely determined by the inventory level - the computational available stock in period t. Its composition was explained in Section 4.1. The aim is to begin with a state space as low-dimensional as possible. The literature research in Section 3.2 demonstrated that in certain cases, solely information regarding the inventory level is sufficient for making sound decisions. Taylor and Tuyls (2010) identify the omission of domain information in the state space as *domain reduction* and *domain hiding*. The avail-

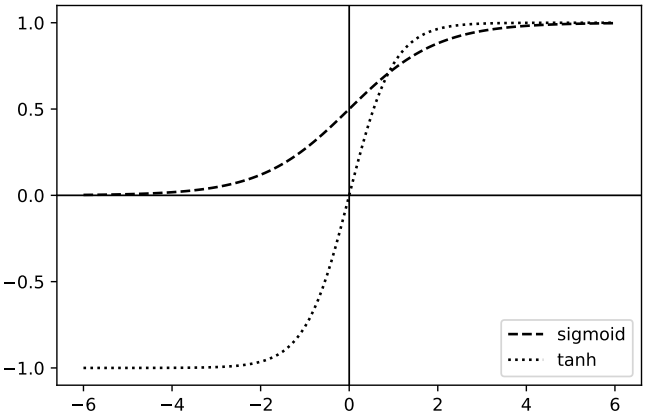

Figure 5.1: Value and Definition Range of the Sigmoid and Tanh Functions

able computational stock combines information on both on-hand and in-transit inventory, leading to reduced information. However, environment information such as demands, replenishment times, and cost parameters are hidden from the RL model. According to Taylor and Tuyls (2010), these two methods fall under the category of *abstraction*, which is seen as a way to increase the feasibility of RL models in complex environments.

This study proposes the concept of adaptive scaling of state and action space. It is crucial to transform input parameters for balanced and steady training procedures. The main aim of the study is to enable the model to adjust its actions, even when environmental attributes (e.g. demand) and the replenishment of resources are changing.

There are two common scaling methods, namely standardization and normalization. Standardization transforms the data into a standard normal distribution (refer to Equation 5.1); conversely, normalization, also known as min-max scaling, shifts the data into the desired value range (refer to Equation 5.2) (Bhandari 2020):

$$X' = \frac{X - \mu(X)}{\sigma(X)} \tag{5.1}$$

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.2}$$

Both standardization and normalization require prior knowledge of the value range of the specified data. Since knowledge regarding inventory levels is unavailable but is rather dependent on process characteristics and selected actions, the objective is to determine a suitable constant to scale down inventory levels to a significant interval, wherein the RL model can differentiate the input more accurately. A similar approach is investigated for the action space, which is determined by the output of the activation function. Said output is mapped to either $(0, 1)$ or $(-1, 1)$. A constant limit is also required for scaling. Defining it as a variable size would help to set maximal replenishment quantities to values that correspond to changing demand and replenishment characteristics.

For an adaptive scaling factor, the concept that emerged in Section 4.6.5 is used. Demand during replenishment time for preventing undershoot D" as a scaling factor would require knowledge of demand and replenishment time distributions. By using Equation 4.6.5 as the basis, an appropriate value of α can be chosen such that $s(\alpha)$ is high enough to normalize the inventory level and action around $(-1, 1)$.

As the computationally available inventory levels can be both positive and negative, they are scaled by simply dividing the original value by the positive constant.

$$I^{p'} = \frac{I^p}{s(\alpha)},\ with\ s(\alpha)\ defined\ by\ P\{D'' \leq s\} \overset{!}{\geq} \alpha \tag{5.3}$$

The constant stated earlier is also used for the action space. As the activation function in *Stable Baselines* is modeled using a tanh function, the output corresponds to a range of $(-1, 1)$. (Hill et al. 2018) Therefore, the constant is used to adjust the output to a positive value range, which can be interpreted as a corresponding quantity of replenishment.

$$a = \frac{(a'+1)}{2} * s(\alpha),\ with\ s(\alpha)\ defined\ by\ P\{D'' \leq s\} \overset{!}{\geq} \alpha \tag{5.4}$$

To evaluate this scaling method, the model is trained on one specific parameter setting, previously described as the original data set. The environment's unpredictability causes variations in the actual demand, forecast errors, and replenishment times based on the assumed distributions.

To address the randomness of the training process, 10 distinct models are trained with differing seeds for $3x10^6$ training steps. Before training each model, hyperparameter optimization is conducted using *Optuna*, with 50 possible parameter combinations over $3x10^5$ training steps.

The selected model exhibits the lowest mean cost per episode over the 385 tested episodes, with minimal variation; therefore, it meets the criterion of being the *best*. The average mean cost per period stands at 308.55 [MU] with a standard deviation of 105.89, and the corresponding service level is 98% on average with a deviation of 2%.

To assess the impact of the normalization factor, the analysis begins by testing the previously described parameter variations. Test data are generated by drawing realizations of the parameters defined by the range of values. The training was performed on realizations from the original parameter combination, but now the parameters vary significantly beyond those encountered during training. If the results for this expanded data set are consistent with those of the original parameter combination, then the ability to adapt to changing environments can be demonstrated.

Therefore, one model named $RL_{Baseline}$ that uses a fixed normalization factor is compared with another model, $RL_{Adaptive}$, which uses an adaptive normalization factor. The raw data for these results can be found in Bergmann (2023b).

First, the sample sizes of the test data sets are confirmed by considering the relative standard errors of the mean. In all scenarios, the following simulation runs exhibit low relative standard errors of the mean (refer to Equation 4.6.3), as demonstrated in Figure 5.2. The KPI of mean cost per

period has a value range of $[0, 0.0652]$, whilst the service level is in the range of $[0, 0.0151]$, with the majority of 95% being within these intervals.

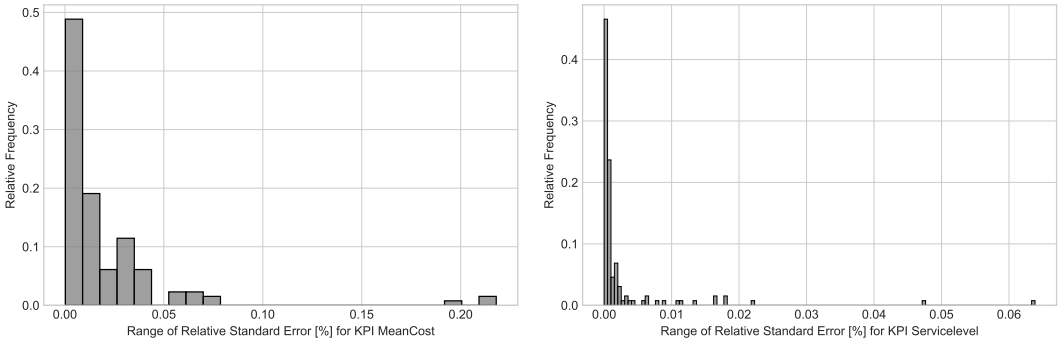

Figure 5.2: Relative Standard Errors of the Mean of the KPIs Mean Cost and Service Level Over All Data Sets Used in the Following

The scaling factor incorporates details on demand and replenishment attributes. Initially, the demand parameters are assessed. To quantify the cost's dependence on each factor, the previously defined KPI of cost efficiency (refer to Equation 4.11) is considered along with the absolute mean cost per period and service level. The outcomes are depicted in Figure 5.3.

Mean costs per period increase significantly with the rise in mean demand for the $RL_{Baseline}$ model compared with the $RL_{Adaptive}$ model. The first upper graphic illustrates the KPI of cost efficiency, which remains fairly constant for the $RL_{Adaptive}$ model but varies for the $RL_{Baseline}$ model. The cost efficiency of both models is similar for the expected demands of 5, 10, and 50. As the models were trained on an expected demand of 30, the KPIs unsurprisingly differ only marginally for similar values to the original. Instead, the $RL_{Adaptive}$ model outperforms for demands of 1 and 100 or greater. The rise in mean cost for the $RL_{Baseline}$ model in the next plot of the upper part of Figure 5.3 is explained by a drop in service level for larger expected demands. This can be explained by the mean order size depicted in the following graph, which does not adjust to increasing demands for larger values. The $RL_{Adaptive}$ model follows the bisecting angle and orders on average what is requested, while the $RL_{Baseline}$ model deviates from it for larger anticipated demands.

When the CV of the demand is varied, a slightly cheaper performance of the $RL_{Baseline}$ model for lower CVs is observed. This can be attributed to lower stock levels initially for the $RL_{Baseline}$ model, which do not lead to lower service levels and hence appear more efficient. Nonetheless, this advantage is rapidly depleted as the stock levels decrease excessively, leading to lower service levels, which in turn results in higher backorder costs as the CV of demand surpasses $CV(D) > 1.0$.

Referring to the assumption made in Section 4.6, one can infer that the $RL_{Adaptive}$ model confirms (P0) as the escalating expected demand does not lead to an increase in the average cost per unit, which remains quite stable at a mean of 12.26 [MU] with a standard deviation of 3.32 per period. (P1) is supported to some extent by the results, where a minor decline in service level is observed for higher CVs, but only from >99% to >97% when CVs of 2.5 are considered. By contrast, the $RL_{Baseline}$ model does not comply with assumptions (P0) and (P1).

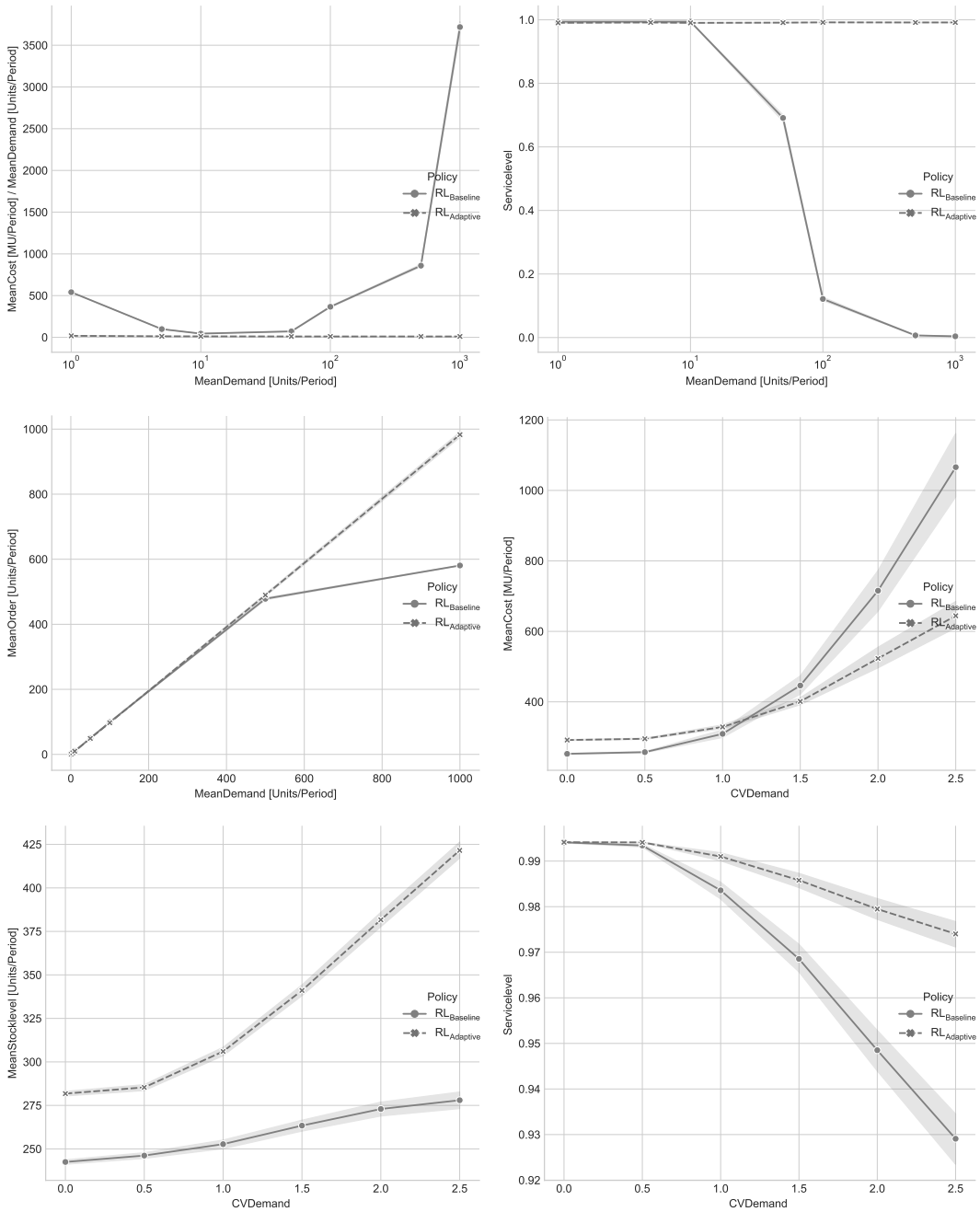

Figure 5.3: Comparison of the $RL_{Baseline}$ and $RL_{Adaptive}$ Models in Terms of Mean Cost, Service Level, and Mean Order Size when Demand Parameters are Varied

The conclusion drawn after examining the replenishment parameters of the mean expected replenishment time and its CV are depicted in Figure 5.4.

According to (P2), an increase in the number of stocks in the system with higher average replenishment times is anticipated. However, considering stock efficiency (i.e. stock levels per replenishment time unit), one would expect the values to be relatively consistent. These findings are illustrated in the first graph for the $RL_{Adaptive}$ model. The stock efficiency exhibits a mild positive gradient but remains stable for mean replenishment periods up to 50. The interpretation of the stock efficiency of the $RL_{Baseline}$ model must be coupled with the absolute stock levels and the subsequent service level graphs. The efficiency declines with higher replenishment times, which may initially be interpreted as beneficial due to scale effects. However, it is important to note that when examining the overall stock levels, negative levels are observed after the 20 replenishment periods. These negative levels account for the drop in service levels, which occur after only 10 replenishment periods.

Values below the trained seven replenishment periods - in this case, the replenishment times of one period - are not handled fully adequately by either model. The $RL_{Baseline}$ model results in excessive overstock, as evidenced by the stock efficiency graph, while the $RL_{Adaptive}$ model lowers its stock to a minimum. Unfortunately, this minimum level does not seem sufficient, and service levels remain relatively low despite its high overall levels.

The thesis formulated in (P3) states that service levels do not decrease as the CVs increase. The final row of Figure 5.4 reveals a noteworthy pattern, namely that the $RL_{Baseline}$ model attains superior service levels compared with the $RL_{Adaptive}$ model for lower CVs. The inflection point is around a CV of 1.0, where both models were trained; thereafter, the $RL_{Adaptive}$ model exhibits stability when faced with CVs up to 2.5. Upon examining the following graph, it becomes apparent that the $RL_{Adaptive}$ model's response to increasing CVs is due to an increase in inventory levels. This is suitable for all CVs ≥ 1.0 but results in an excessively large decrease in the low variance segment where CV < 1.0. For that case, the $RL_{Baseline}$ model performs better with no adaptation at all, as it leads to stable service levels and relatively smaller holding instead of backorder cost.

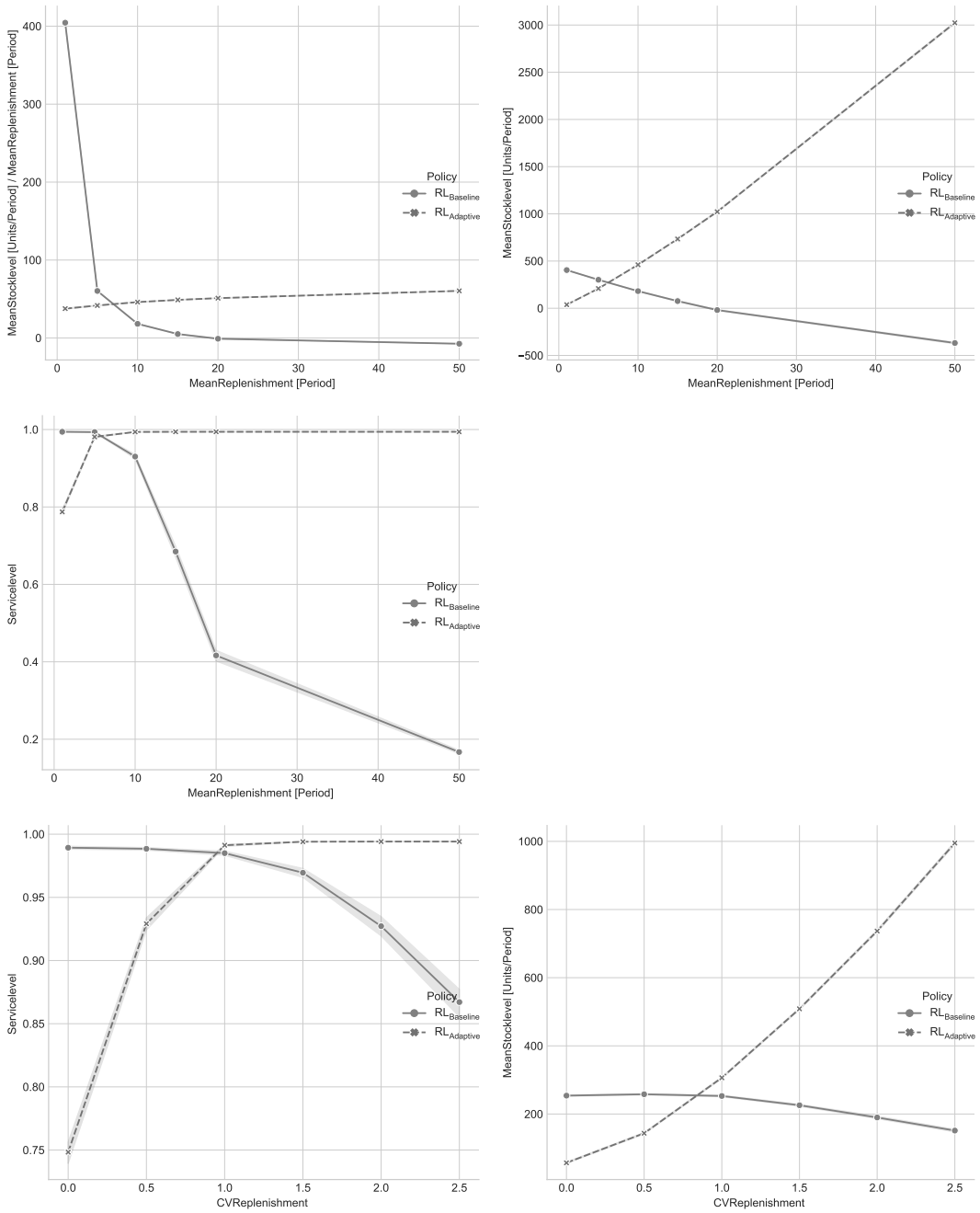

Figure 5.4: Comparison of the $RL_{Baseline}$ and $RL_{Adaptive}$ Models in Terms of Stock and Service Levels when Replenishment Time is Varied in its Mean and CV

Based on the presented analysis, this study concludes that implementing an adaptive scaling factor, as opposed to a fixed factor, is conducive to de-

veloping a robust RL strategy, particularly for the scenario of increasing and more variable demand and replenishment parameters.

However, upon analyzing the behavior of the $RL_{Adaptive}$ model towards changes in cost parameters that also impact their mutual ratio, one must reject the expectations outlined in (P4). As illustrated in Figure 5.5, no identifiable pattern emerges that would support the notion of model adaptation. Mean stock levels remain constant despite the significant increase in holding costs. Fixed order costs also increase without affecting order frequencies. Additionally, varying backordering costs do not result in any changes in service levels compared with the other two cost parameters.

Therefore, how improvements in this field could be achieved using an extended state space that incorporates cost information is explored in the following section.

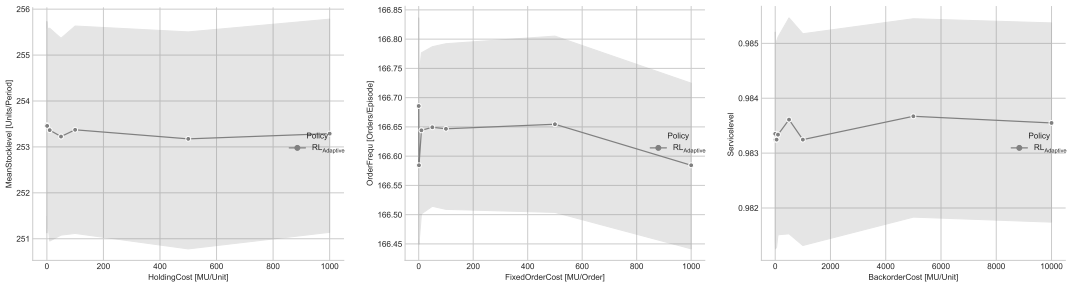

Figure 5.5: Adaptations of $RL_{Adaptive}$ to Changing Cost Parameters

5.3 Extension of the Basic Model by Cost Parameters

The previous section concluded by highlighting the model's inability to adjust to varying cost parameters. As the $RL_{Adaptive}$ model lacks insight into cost conditions present within the state space, information pertaining to cost parameters must be incorporated in the next stage. Herein, the model is referred to as $RL_{Adaptive+Cost}$.

The cost parameters are scaled by the ratio of each parameter in relation to the sum of all cost parameters. Rather than scaling with an arbitrarily high constant, the dependencies of the different parameters are considered. To scale each cost parameter p, Equation 5.5 is applied:

$$p' = \frac{p}{\sum_{p \in P} p}, \; with \; P = \{h, K, b\} \; and \; \forall p \in \{h, K, b\}. \tag{5.5}$$

The resulting model with a distinct input state is acquired through the aforementioned process of hyper-optimizing and training 10 distinct models before selecting the best one. The chosen model achieves an average cost per period of 318.95 [MU] with a standard deviation of 61.49 when it is applied to the original data set. Additionally, it attains an average service level of 99% with a 1% deviation.

The impact of the supplementary data in the state space can be seen in the top row of Figure 5.6: The reaction of the $RL_{Adaptive+Cost}$ model to modifications in cost parameters significantly exceeds that of the $RL_{Adaptive}$ model for each cost parameter. As the proportion of holding costs increases, the assumption expressed in (P4a) can be confirmed. In general, inventory levels decrease as holding costs rise. This would result in a decrease in service levels as demand cannot be reliably met. However, the increasing significance of holding costs compared with backorder costs leads to less of a rise in the mean cost overall than with the $RL_{Adaptive}$ model, resulting in greater stability. The tipping point at which the performance of the $RL_{Adaptive+Cost}$ model outstrips that of the $RL_{Adaptive}$ model is noticeable when the holding cost parameter tops the fixed order cost parameter of 10 and the backorder cost parameter at a level of 100 [MU].

A comparable scenario is noted for fluctuating backorder costs featured in the lower row of Figure 5.6: The $RL_{Adaptive+Cost}$ model can adjust its actions to declining backorder cost parameters that are akin to the ones of a fixed order cost of 10. Its response is a decrease in service level in case of backorder costs lower than 50. In this context, the service level is no longer of significance to the overall cost. By contrast, when the cost of backorders exceeds the combined costs of holding and fixed order, service levels tend

to be consistently high, even surpassing those of the $RL_{Adaptive}$ model. This trend is also evident in the mean cost curve, which is initially similar in both models, including the trained backorder cost of 100. However, beyond this point, the mean cost of the $RL_{Adaptive}$ model clearly stands above that of the other. As a result, P4c can be affirmed.

The image alters when the fixed order cost parameter is modified: The model's response to increasing order cost appears to be favourable as the frequency of orders decreases. However, an examination of the average overall cost reveals that the $RL_{Adaptive}$ model still incurs lower costs per period. The point at which the models' curves diverge can be located at the training value of 10 for fixed order costs. The divergence is even greater at the cost of 100, which relates to backordering costs. As a result, service levels are no longer crucial from this stage onwards. The model can be inferred to be unable to strike a suitable balance between the still substantial backorders and the increasing fixed order costs. While (P4b) can be affirmed, it must be noted that the treatment of fixed order costs does not align with the expectations outlined in the introduction of (P4), since the average cost per period increases.

Overall, further tests demonstrate that the suggested modification to the state space yields comparable behavior to that of the $RL_{Adaptive}$ model when confronted with fluctuating mean demands and replenishment times. Moreover, the overall favourable changes within the policy aimed at changing cost parameters confirm the usefulness of incorporating such parameters into the model.

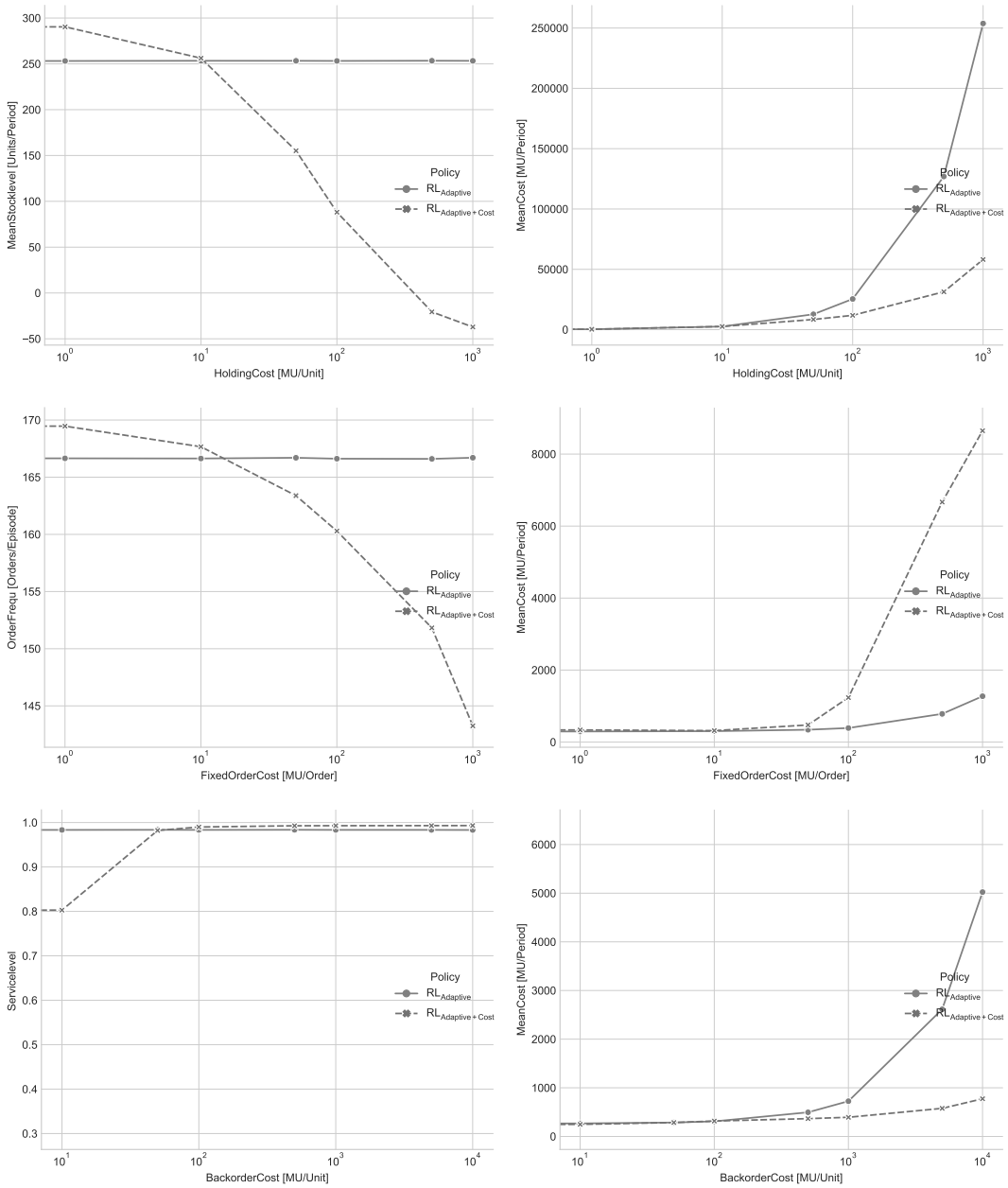

Figure 5.6: Adaptations of $RL_{Adaptive+Cost}$ to Changing Cost Parameters

5.4 Analysis of System Behavior Compared with the (r, s, q) Approach

Finally, the performance of the original and amplified data sets is compared in test episodes using the (r, s, q) inventory policy described earlier. In this case, the review period is set to discrete points of time with $r = 1$ to ensure similar conditions for the RL and conventional approach. For the rest of the policy, it is referred to as the (s, q) policy; s and q are calculated in accordance with the formulas expressed earlier. The expected demand amount over a time horizon of T periods determines $E(D)$ in Equation 4.6.5. This is equivalent to multiplying the expected demand per time period by T. The (s, q) inventory policy is set to a target service level of $\alpha = 0.95$.

Figure 5.7 presents a comparison of the three models - $RL_{Adaptive}$, $RL_{Adaptive+Cost}$ and the (s, q) model - for the testing scenarios with varying demand and replenishment parameters. Overall, the RL models seem to have lower mean costs for all possible input parameters. However, as the average demand and CV of the demand increase, making them highly unpredictable for the RL approaches, the advantage over the (s, q) model remains but is diminished. Notably, for the lowest mean replenishment of one period and the lowest CV of 0.0, almost no replenishment time is required, and the (s, q) model outperforms both of the RL models. It can be concluded that state-of-the-art policies are proficient at managing these replenishment times. Additionally, it is worth noting that there is minimal variation between the identified models regarding changes in mean replenishment times, while the CV remains fixed.

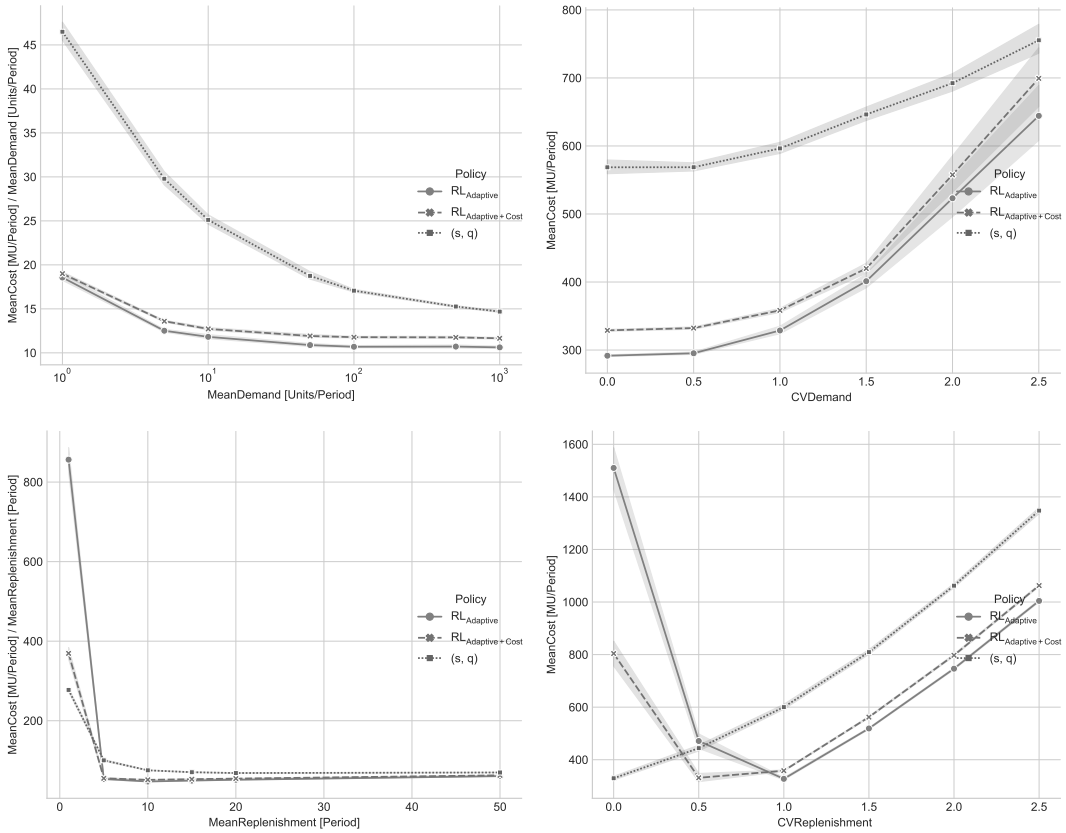

Figure 5.7: Comparison of the $RL_{Adaptive}$, $RL_{Adaptive+Cost}$, and (s,q) Models when Demand and Replenishment Parameters are Varied

When the cost parameters are altered, as illustrated in Figure 5.8, both the $RL_{Adaptive}$ and (s,q) models exhibit similar trends in behavior. However, the $RL_{Adaptive}$ model provides lower mean costs than the (s,q) model. Notably, the $RL_{Adaptive+Cost}$ model yields even lower mean costs than both of the aforementioned ones, specifically when variations in holding and backorder costs are considered. This trend cannot be confirmed in the case of the $RL_{Adaptive+Cost}$ model's response to increasing fixed order costs since they result in costs many times greater than those of the other models. This behavior has already been explored.

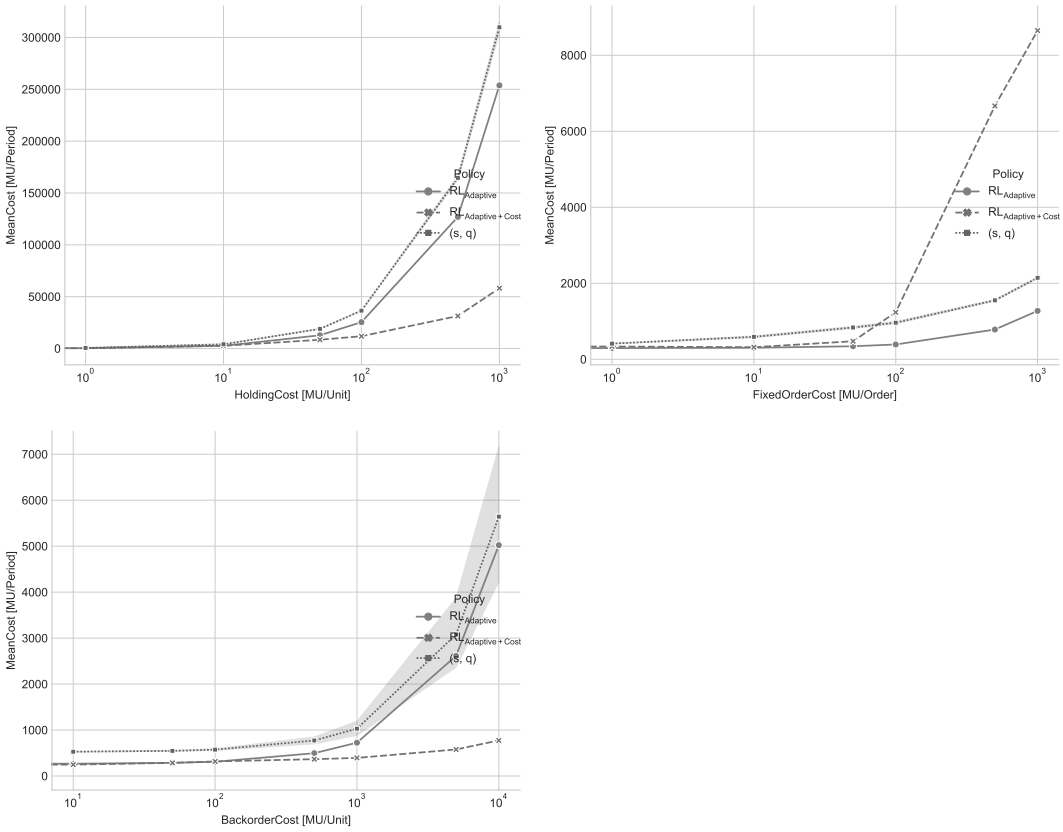

Figure 5.8: Comparison of the $RL_{Adaptive}$, $RL_{Adaptive+Cost}$, and (s, q) Models when Cost Parameters are Varied

Several two-factorial and one three-factorial test scenarios must be conducted to analyze the system's behavior further. The aim is to evaluate the interaction of different factors. The test focuses on demand parameters such as mean demand and the CV of demand, replenishment parameters such as mean replenishment time and the CV of replenishment time, and three cost parameters. These parameters are varied using a full-factorial approach. The analysis also considers the full-factorial variation of the two variances, namely the CV of demand and the CV of replenishment time. Intermediate values are omitted based on single-factorial studies, where a linear behavior is anticipated. Higher limits are also established for the CVs. The ensuing comparison is narrowed to the $RL_{Adaptive+Cost}$ and (s, q) approaches for the sake of clarity.

Figure 5.9 presents the outcomes for the factors of mean demand and the CV of demand. One can infer that the RL policy in the left diagram results in a relatively stable mean cost per period and demanded unit. Additionally, one can observe that the mean cost is higher for lower mean demands, with this difference being more significant for lower CVs. Moreover, it should be noted that a higher CV leads to an increase in mean cost. This can be attributed to the adaptation of the model with more securities, such as higher stock levels, to account for the increased volatility in demand. Compared with the (s, q) policy, the RL policy aims for a stable mean cost per period and unit demanded, while the (s, q) policy results in a declining mean cost per unit. When the two curves for each CV are compared, the mean costs are usually lower for the RL approach in all cases except for the scenario of high variability in demand with $CV(D) = 3.0$ and a mean demand of $E(D) = 100$.

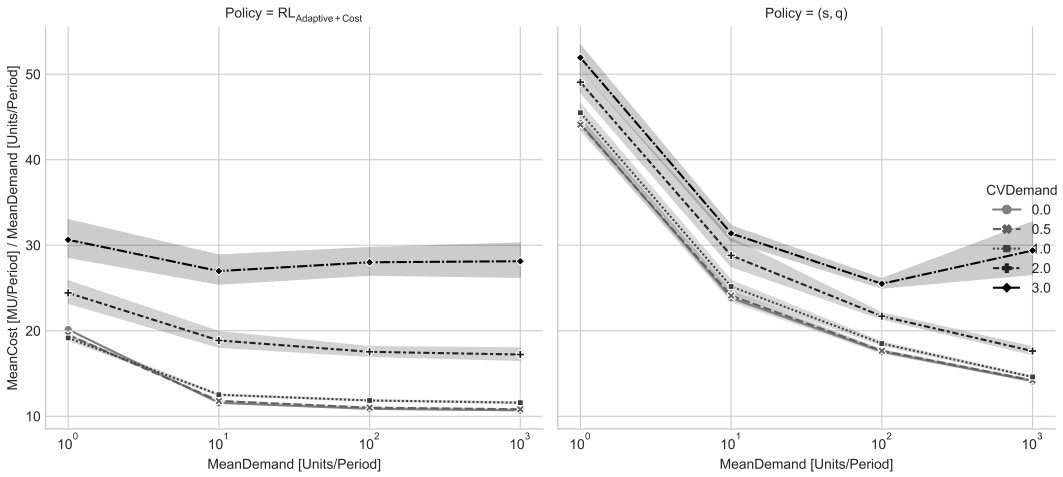

Figure 5.9: Comparison of Mean Cost per Demanded Unit for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Demand Parameters are Varied

Considering the service level as the next KPI of interest, Figure 5.10 portrays its variation across different demand parameters. At first glance, the chart on the left suggests a significant drop in service level for the RL policy with increasing CV. However, it is important to note that the lower limit of the y-axis indicates a service level of 97%, which remains satisfactorily high. It can be stated that increased demand variances result in decreased

service levels for the RL policy. However, this decline remains within an acceptable range. On the other hand, the service level of the (s, q) policy remains relatively stable for higher variances, except when the demand is high at $E(D) = 1000$. This decrease in service level could explain the higher average cost depicted in the preceding figure. Comparing the service level in relation to the mean and CV of demand reveals that both models display a trend where scenarios with a lower mean demand are more prone to being impacted by an increased variability of demand than scenarios with a higher mean demand.

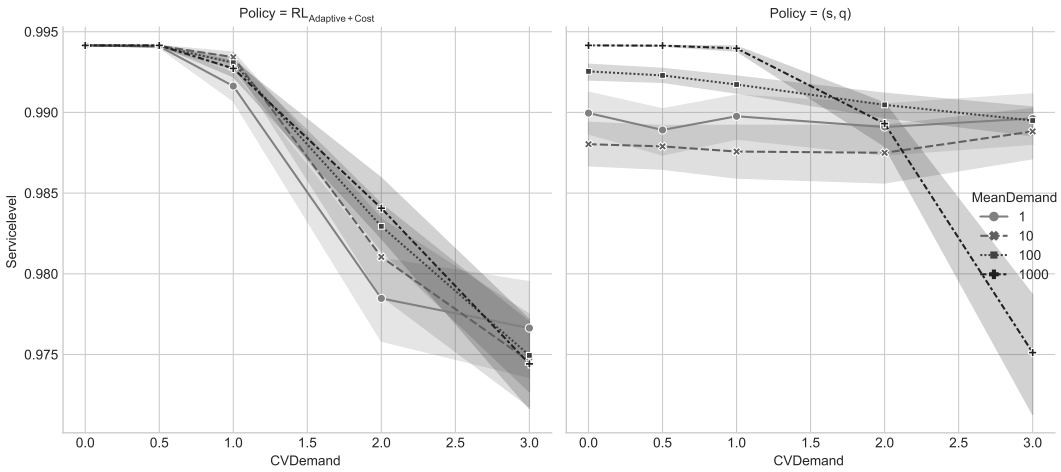

Figure 5.10: Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Demand Parameters are Varied

The parameters of mean and CV of replenishment times require consideration. The aggregated results are presented in Figure 5.11. The KPI for this analysis is the mean stock level per period of replenishment time, which demonstrates that a change in replenishment time does not result in an over-proportional change in mean stock levels. The initial observation indicates that for both policies, the mean stock levels increase as the CV of replenishment time increases. This reaction appears suitable for both models as additional stock offsets the security against replenishment variance. The RL policy's average stock levels remain stable with minimal fluctuations when transitioning from a mean replenishment time of $E(L) = 1$ to $E(L) = 7$. For lower CVs (i.e., $CV(L) \leq 2.0$), mean stock levels are some-

what higher, whereas for the highest tested CV (i.e., $CV(L) = 3.0$), it is slightly lower than those for mean replenishment times above $E(L) = 1$. When transitioning to higher unknown mean replenishment times, such as $E(L) = 40$, the RL policy maintains a relatively stable mean stock level with a slight increase, while the (s, q) policy exhibits a slight decrease. The difference in mean stock levels between the first two replenishment times, where $E(L) = 1$ and $E(L) = 7$, is considerably more marked and less consistent for the (s, q) policy compared with the RL policy. Upon examining the specific curves with the same CVs, the mean stock levels for the RL policy are reported to be generally lower.

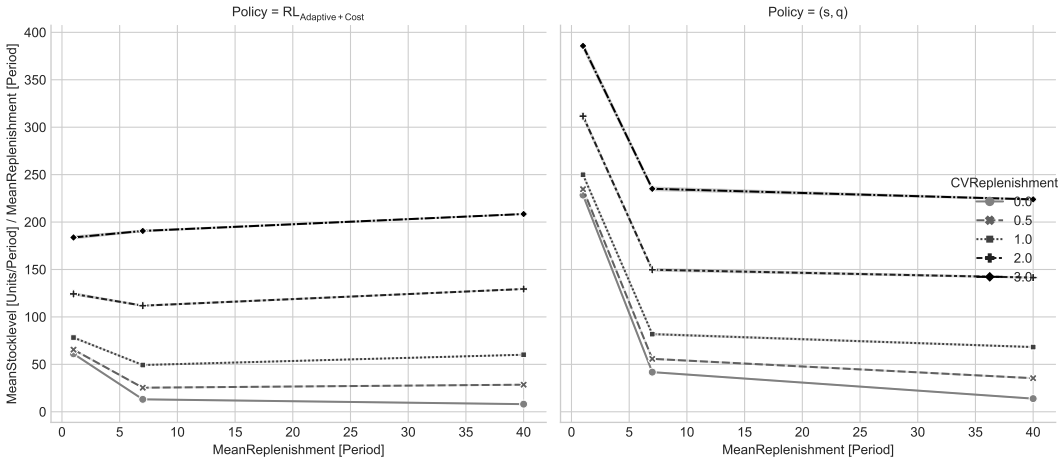

Figure 5.11: Comparison of Mean Stock Level of the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Replenishment Time Parameters are Varied

Figure 5.12 provides an overview of the development of service levels when replenishment time parameters are varied. Compared with the RL policy, the (s, q) policy exhibits consistently high service levels regardless of $E(L)$ and $CV(L)$. By contrast, CVs lower than $CV(L) = 1.0$ lead to a drop in service level for the RL policy, while $CV(L) \geq 1.0$ combined with mean replenishment times greater than the known value of $E(L) = 7$ result in stable service levels. On the contrary, the RL model struggles to handle a significantly shorter mean replenishment time with $E(L) = 1$, as service levels only exceed 95% for $CV \geq 2.0$. To conclude, the RL model seems to encounter issues with low replenishment times and variances.

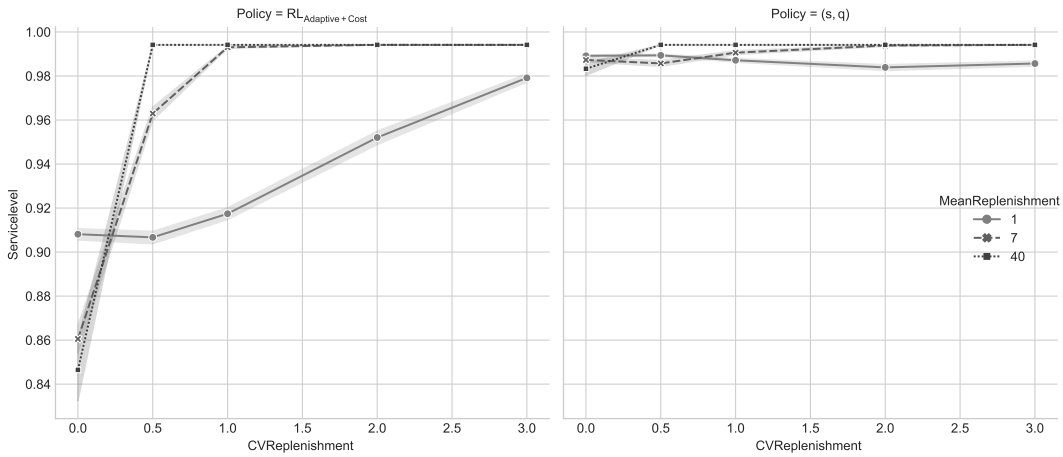

Figure 5.12: Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Replenishment Time Parameters are Varied

Upon examination of the mean cost KPI in Figure 5.13, it becomes apparent that both models display similar behavior. This indicates that the RL policy offsets higher backorder costs caused by lower service levels with lower stock holding costs resulting from lower mean stock levels. Only the curve of $CV = 0.0$ demonstrates a significantly greater cost for the RL policy compared with the (s, q) policy. This reflects how the (s, q) policy is a better fit for "easier" replenishment time settings.

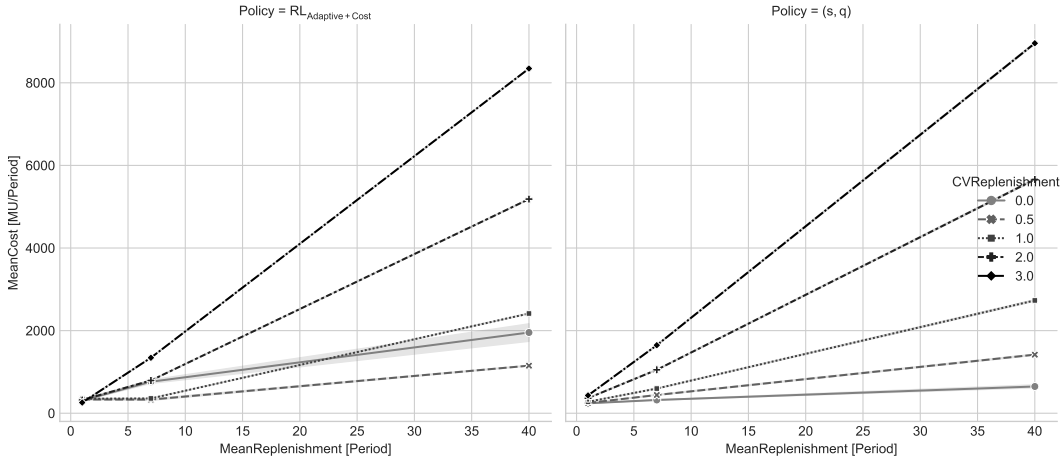

Figure 5.13: Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Replenishment Time Parameters are Varied

The combined analysis of holding, fixed order, and backorder costs is considered in terms of cost parameters. Different stock levels, order frequencies, and service levels are implicated by varying the cost parameter ratios.

The findings of the single-factor experiments are generally supported by the three-factor analysis of the cost parameters: As holding costs increase, mean stock levels tend to decrease when the RL model is applied. This can be observed in the upper row of Figure 5.14. Additionally, mean stock levels are generally higher with increased backorder costs, which can be attributed to the growing significance of high service levels as backorder costs escalate. Surprisingly, the average stock levels decrease from left to right as the fixed order costs increase. Usually, one would expect the average stock levels to increase with rising fixed order costs due to fewer and larger orders. However, this behavior is due to the anomaly of the RL model described earlier.

For the (s, q) model, a similar trend can generally be seen in the lower row of Figure 5.14, namely that an increase in holding costs leads to a decrease in the mean stock level. However, reactions to rising holding and backorder costs are not as significant as for the RL model. Conversely, the increase in stock levels with rising fixed order costs confirms the previously stated expectation.

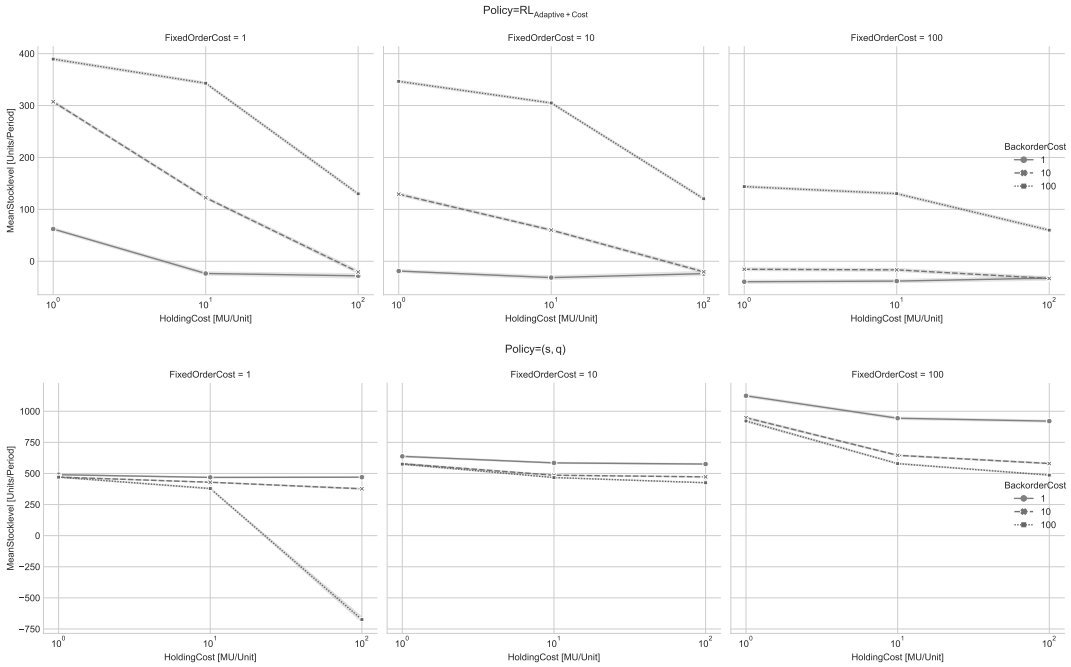

Figure 5.14: Comparison of Mean Stock Level for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Cost Parameters are Varied

When order frequency is considered as a performance measure that is expected to respond to increasing fixed order costs, the general trend of both models is confirmed, as seen in Figure 5.15. That is, as fixed order costs increase, the order frequency of both models decreases.

It can be stated that the RL model typically has a higher order frequency than the (s, q) approach. Moreover, it exhibits less adaptation in order frequency than the (s, q) approach when various slopes are examined. With a higher backorder cost, the frequency is generally greater. The intriguing aspect is that an anomaly in the RL model becomes more evident when holding costs increase. Specifically, as holding costs rise from left to right in the upper row of Figure 5.15, there is a more pronounced contrast between the backorder cost curves and lower order frequencies. Thus, the model prefers lower mean stock levels through manipulating the ordering frequency.

For the (s, q) model, it is evident that elevated fixed order costs reduce order frequencies, while increased holding costs result in an increase in fre-

quency, as the goal is to maintain fewer products in stock and order more frequently. Similarly, for this model, a high backorder cost leads to an uptick in the order frequency as smaller fixed order amounts $q*$ are necessary.

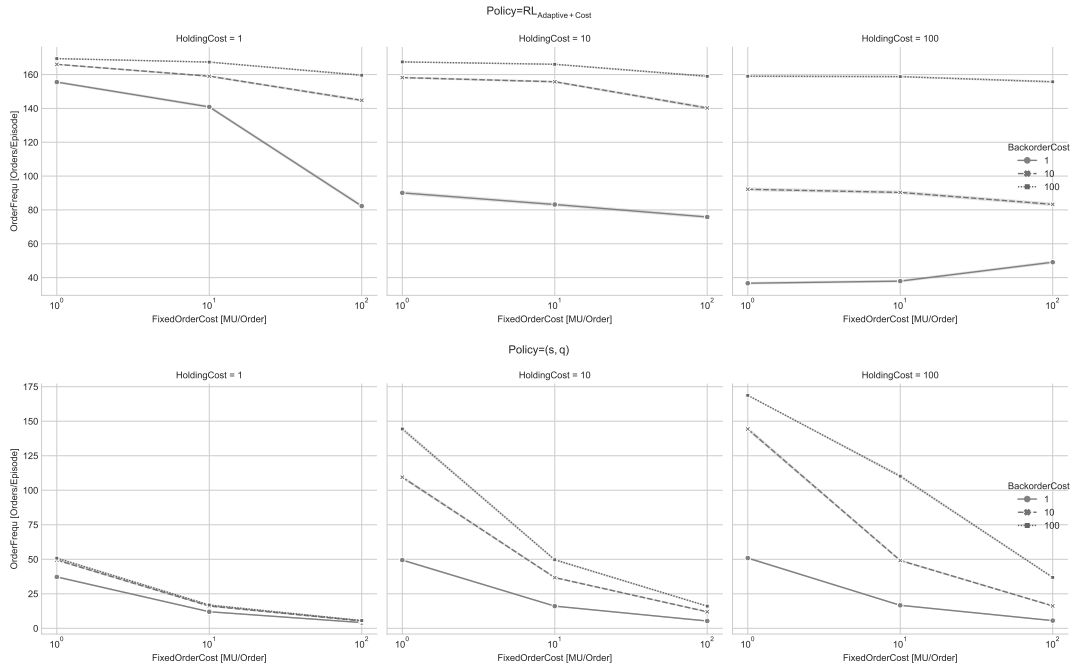

Figure 5.15: Comparison of Order Frequency for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied

Figure 5.16 illustrates the contrast in service levels for both models when the cost parameters vary. The lower row of the chart displays the (s,q) model, which notably maintains a consistent service level. This stems from the model's approach, where the reorder point s^* is fixed based on a singular target service level. An intriguing observation arises when fixed order costs are set to 1 whilst holding and backorder costs are set to 100. Examining the cost parameters in Equation 4.6.5, it is evident that low ordering costs result in an order amount influenced by the fraction $\frac{K(h+b)}{h*b} = \frac{2}{100}$. This leads to an order size that is incapable of meeting the demand in a single time period, which is illustrated by the order frequency chart in Figure 5.15. The model places orders during each time period to meet the demand.

Furthermore, service levels for the RL method differ greatly between very low and very high ones. In general, higher backorders generate an increase

in service level when no other cost factor dominates the cost ratios with a very high value. This can be observed, for example, when backorder costs are under 100 [MU] while holding costs are at 100 [MU], which is represented by the dotted line and the first two points in each graph of the upper row in Figure 5.16. Also noteworthy is the prevalence of fixed order costs at 100 [MU] and the lower service levels when fixed order costs are lower.

For cases where all cost parameters hold equal value, a service level of approximately 67% is achieved.

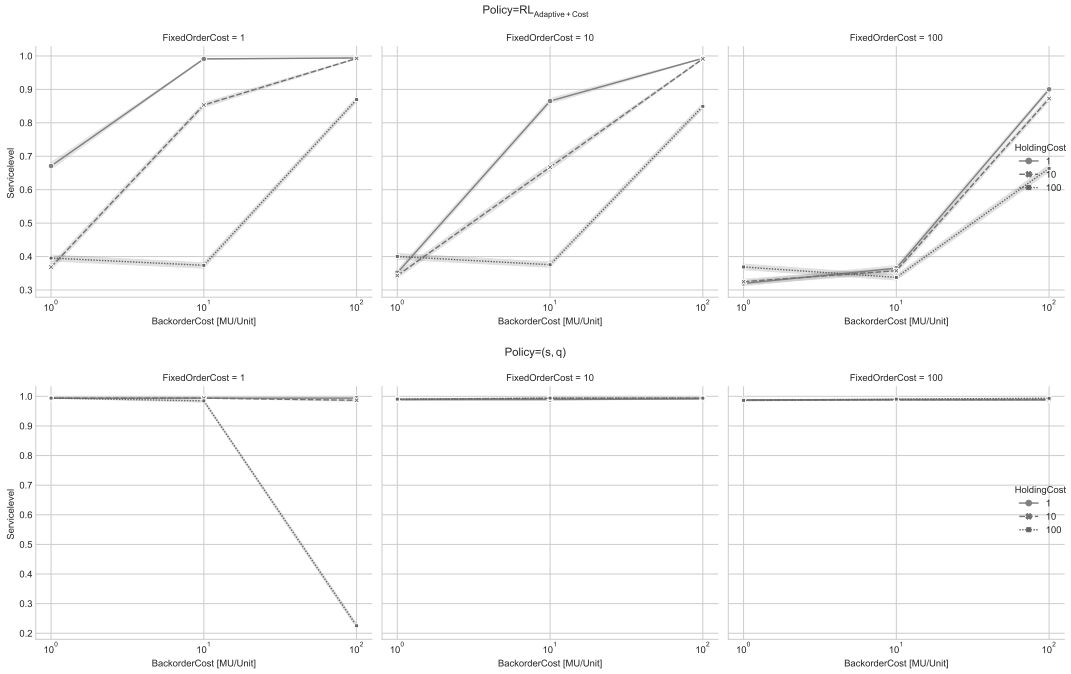

Figure 5.16: Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Cost Parameters are Varied

Leaving aside the findings that relate to the RL model's capability to adjust to changing cost parameter ratios, an issue remains, namely whether this feature results in lower costs than conventional methods. The varying y-axis ranges in Figure 5.17 could provide some insights in this regard: While the limits for the RL approach extend to $14,000$ [MU], the (s, q) model's scale reaches $80,000$ [MU]. Upon comparing the average values and standard deviations of mean cost per period, displayed in Table 5.1, the visual impression and detailed differences are confirmed to be the same: Whatever

the cost parameter combination, the average costs are lower for the RL approach.

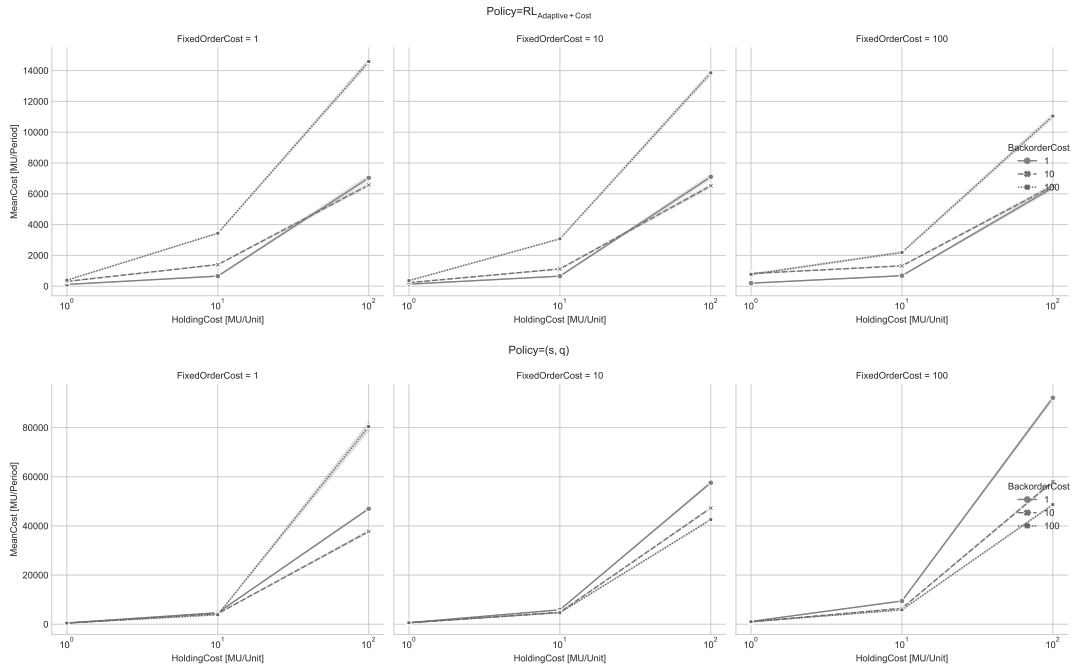

Figure 5.17: Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Cost Parameters are Varied

Holding-Cost	FixedOrder-Cost	Backorder-Cost	$RL_{Adaptive+Cost}$		(s, q)	
			E(MeanCost)	Std(MeanCost)	E(MeanCost)	Std(MeanCost)
1	1	1	111.73	13.37	490.23	35.39
1	1	10	309.49	22.31	470.97	29.03
1	1	100	391.18	25.10	471.02	29.73
1	10	1	131.76	20.96	639.09	59.72
1	10	10	228.39	63.74	583.44	49.25
1	10	100	361.17	33.01	595.26	88.98
1	100	1	193.54	24.92	1128.11	100.39
1	100	10	822.63	203.14	959.73	80.00
1	100	100	769.44	492.24	1,011.14	288.65
10	1	1	652.65	103.52	4,682.26	304.69
10	1	10	1,402.50	155.80	4,293.25	248.20
10	1	100	3,438.79	225.94	3,841.35	451.37
10	10	1	650.54	109.32	5,853.97	473.71
10	10	10	1,119.00	130.27	4,861.23	331.20
10	10	100	3,072.70	221.03	4,674.55	285.69
10	100	1	680.04	107.68	9,450.90	929.66
10	100	10	1,319.62	202.42	6,472.38	539.06
10	100	100	2,191.90	552.21	5,842.92	457.79
100	1	1	7,035.75	1,679.81	46,984.76	2,983.24
100	1	10	6,567.64	1,083.78	37,756.39	5,431.27
100	1	100	14,586.10	1,745.21	80,422.50	17,988.10
100	10	1	7,101.34	1,605.05	57,577.35	4,582.13
100	10	10	6,518.98	1,141.82	47,262.41	2,971.11
100	10	100	13,851.17	1,649.16	42,552.08	2,748.09
100	100	1	6,394.73	1,395.23	92,145.41	8,682.85
100	100	10	6,510.73	1,048.21	58,042.43	4,767.57
100	100	100	11,046.83	1,337.31	48,675.30	3,548.64

Table 5.1: Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when Cost Parameters are Varied

As part of the latest factorial testing, the interaction between the two CVs for demand and replenishment time is analyzed. The findings are presented in Figure 5.18 and Figure 5.19. Overall, the results in Figure 5.18 reveal a pattern in the curve for the (s, q) policy: An increase in demand CV or replenishment time CV results in higher costs. Nonetheless, in both models, the impact of rising CVs of replenishment time appears to outweigh that of increasing CVs of demand, particularly for CVs ≥ 2.0, where the various curves converge. The graph on the left reveals anomalies in the RL policy. It demonstrates that the RL model cannot handle uncomplicated SC settings, which is evidenced by the exorbitant costs for the CV combinations of $CV(D) = 0.0$ and $CV(L) = 0.0$, as well as for $CV(D) = 0.5$ and $CV(L) = 0.0$, which are comparable to the higher CVs. The impact of zero variance in replenishment times compared with varying CVs of demand can be clearly seen in the attained service levels in Figure 5.19. The poorest result, leading to an inferior service level, was brought about by the aforementioned combination of $CV(D) = 0.0$ and $CV(L) = 0.0$. Raising the CV of demand alongside a constant CV of replenishment time $CV(L) = 0$ leads to an increase in service levels.

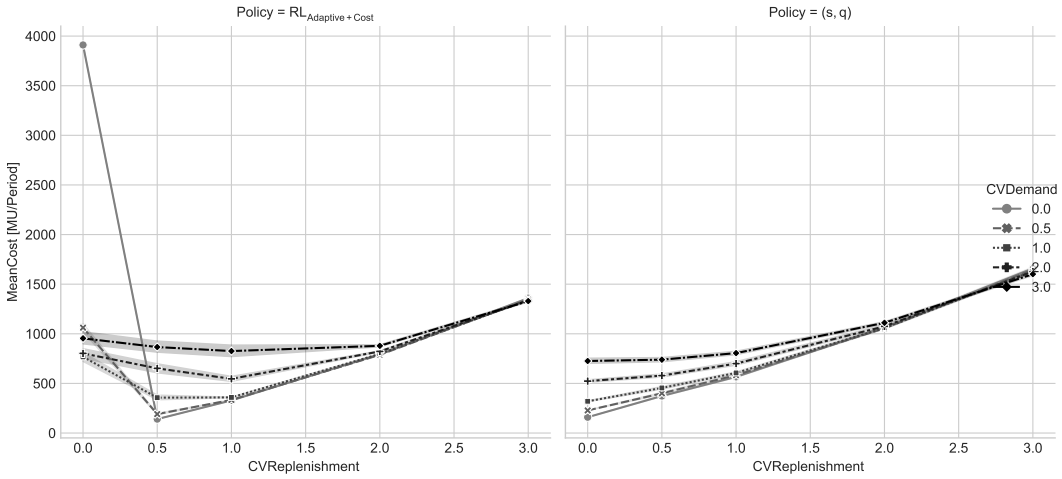

Figure 5.18: Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when CVs of Demand and Replenishment Time are Varied

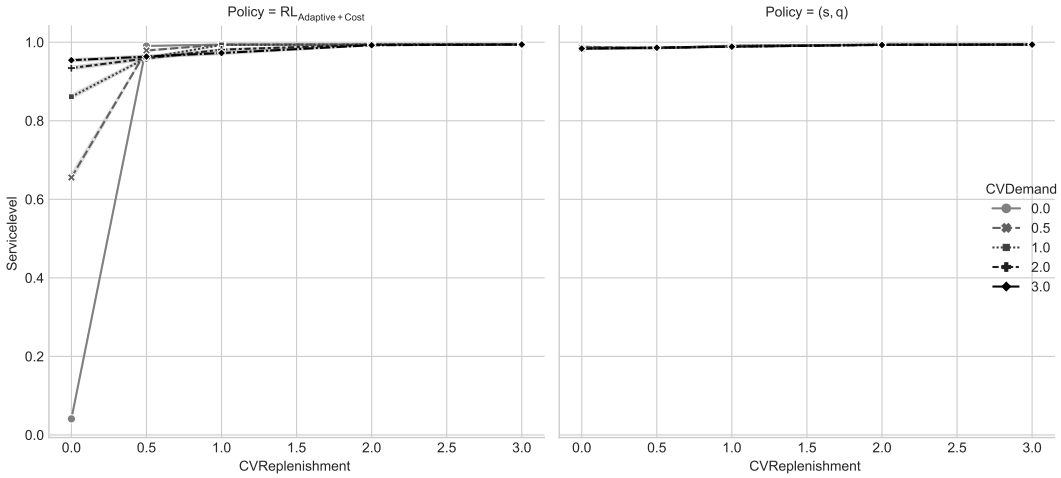

Figure 5.19: Comparison in Service Level for the $RL_{Adaptive+Cost}$ and (s, q) Approaches when CVs of Demand and Replenishment Time are Varied

5.5 Chapter Conclusion

Overall, this chapter has presented a one-stage model that demonstrates stable ordering performance in response to changes in environmental parameters, including demand, replenishment variables, and cost parameters. The models developed within this study surpassed the effectiveness of a simple conventional inventory policy when tested on an unfamiliar data set. This is a noteworthy and optimistic outcome, as the RL model did not receive training on these value ranges. However, it can still adapt to the changing surroundings with its adaptable state and scale spaces, which were introduced in this chapter. The presumptions made in Section 4.6 could be confirmed for almost all cases except the response to fixed order costs. Additionally, for simple environments, a need exists to restrict the use of the RL model, as it appears to be more effective for more complex environments with higher average demands, replenishment times, and variances. This was particularly noticeable in the final analysis, where the absence of demand and replenishment variability resulted in poorer outcomes than a basic, state-of-the-art inventory policy.

6 Reinforcement Learning for Inventory Control in Linear Multi-Stage Suppl Chains

A ship in port is safe, but that's not what ships are built for.
– Grace Hopper

As the literature suggests, there is growing interest in managing not only one SC node but also several nodes within the entire SC. Two main planning paradigms can be distinguished - namely a central planning approach and a decentralized planning approach. The basic characteristics and advantages of each approach have already been described in Chapter 2. The previous chapter's results and the time-intensive training process of machine learning models raise the following question: Can several decentral models find similarly good solutions to a central planning approach, thus saving time and energy? This chapter presents the two models to be compared, followed by a comparison of the already known KPIs and the BWE they cause.

6.1 Model Implementation

Two approaches can be distinguished, namely central and decentral planning. Centrality refers to the location where decisions about order dates and amounts are made. In a decentralized approach, decisions are made by several RL agents, one for each SC node and location. In a centralized

approach, one main RL agent makes decisions for all SC nodes simultaneously.

Therefore, the central agent's action space is $A_t = o_{0,t} \times o_{1,t} \times ... \times o_{N-1,t} \times o_{N,t}$ including the amount of ordered material $o_{n,t}$ for each node n in period t, while the state space includes the inventory levels $I^d_{n,t}$ for each SC node n in t. It is therefore equal to $I^d_t = I^d_{0,t} \times I^d_{1,t} \times ... \times I^d_{N-1,t} \times I^d_{N,t}$ and additionally includes the corresponding cost parameters. This follows from the findings for the state space of Chapter 5 and accordingly results in more information for the centralized agent.

6.2 Analysis of System Behavior of the Decentral and Central Approaches

To proceed, a linear SC consisting of three actively planned nodes is considered.

For the decentralized and independent agents, the best performing model from Chapter 5 is selected and implemented for each of the three SC nodes.

For the centralized approach, a superior agent is trained that decides for three SC nodes as a whole; therefore, it is provided with inventory levels and cost parameters of all SC nodes at once. Following the procedure in Chapter 5 for leveling the effect of stochastic training processes, 10 agents are trained on different seeds and the best performing one is selected. Instead of $3x10^5$ periods for hyperparameter optimization and $3x10^6$ for training, experiments have demonstrated that the central model must be optimized and trained with double the number of periods. Since both, the state space and the action space, grow with the number of considered SC nodes, the model requires more experience periods to find its policy. The resulting best agent is compared to N independent RL agents interacting in a linear SC of N nodes. Tests are performed on the original data set.

The resulting model yields a mean cost per period of 839.72 [MU] with an average standard deviation of 142.79. Service levels of 99% are achieved with a standard deviation of 1%. All raw data for this and the following results can be found in the data collection in Bergmann (2023b).

Sample sizes are confirmed by considering the relative standard errors of the means over the simulation runs of all original and further data sets of the selected models. Standard errors can be observed in Figure 6.1 and are generally low. The majority of 95% are in the range of $[0, 0.0883]$ for the mean cost per period KPI and of $[0, 0.0212]$ for the service level KPI.

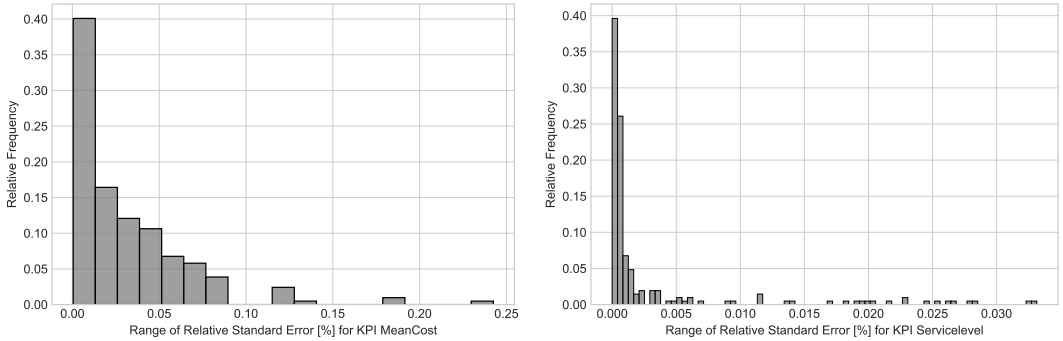

Figure 6.1: Relative Standard Errors of Mean of the KPIs Mean Cost and Service Level Over All Data Sets Used in the Following

After comparing the decentral and central models on the original data set, one can state the following: In general, a decentralized model has been found that is as successful as the centralized one. Considering the information advantage of the central model, this is a rather surprising result. Next, the results of the mean KPIs and the general tendency of both models are statistically validated by the results of a Mann-Whitney U test, which indicates a difference in the mean costs but no significance in the level of service obtained. The exact results can be found in Table 6.1 and the visualization in Figure 6.2.

Nevertheless, the behavior of the two models is examined below on the basis of the assumptions made in Section 4.6. The aim is to study the effects of changing environmental parameters. Therefore, a systematic vari-

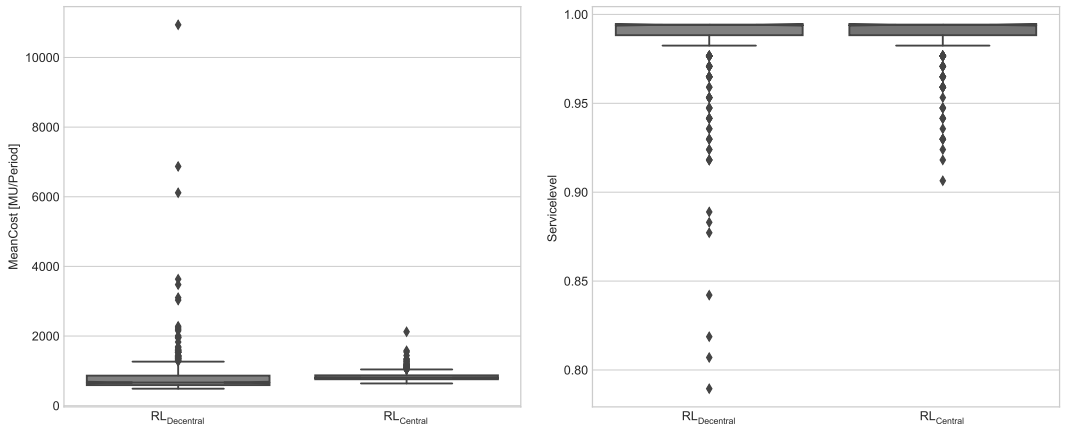

Figure 6.2: Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models in Terms of Mean Cost and Service Level

		μ	σ	U	p-value
Mean Cost	Decentral Model	869.12	774.13	43,538.0	0.000
	Central Model	839.72	142.79		
Service Level	Decentral Model	0.99	0.02	73,576.0	0.412
	Central Model	0.99	0.01		

Table 6.1: Comparison of Statistical Measures of the Two Models

ation of certain parameters is performed according to the value ranges in Section 4.6.

When a change in demand is considered, the graph on the left in Figure 6.3 depicts the aforementioned cost efficiency. An initial drop in unit cost is followed by a relatively stable level as demand increases. As the models have been trained on a mean demand of 30, a first conclusion is that both models are able to handle mean demands of a much higher magnitude. The expected observations of (P0) can thus be confirmed. Compared with the higher magnitudes, the range below the level of $E(D) = 5$ becomes more difficult for both models with a significant increase in the mean cost per unit. In general, the mean costs per period and per unit are at a similar level for both models, with them being lower for the decentralized model for higher demands $E(D) \geq 5$ and vice versa for $E(D) = 1$. Service levels during an increase in mean demand are generally at a high level, with only the edges of the confidence interval reaching values below a 99% service level.

Varying the CV of demand reveals a different picture in the second row of Figure 6.3: For CVs $CV(D) \geq 1.5$, the centralized model seems to perform better in terms of lower average costs with higher service levels. However, it should be noted that the service level for the decentralized model is still above 94% for CVs up to 2.5. The results for (P1) are therefore reported in such a way that service levels do not remain stable with higher variance in demand, but they still remain at a high level for both models.

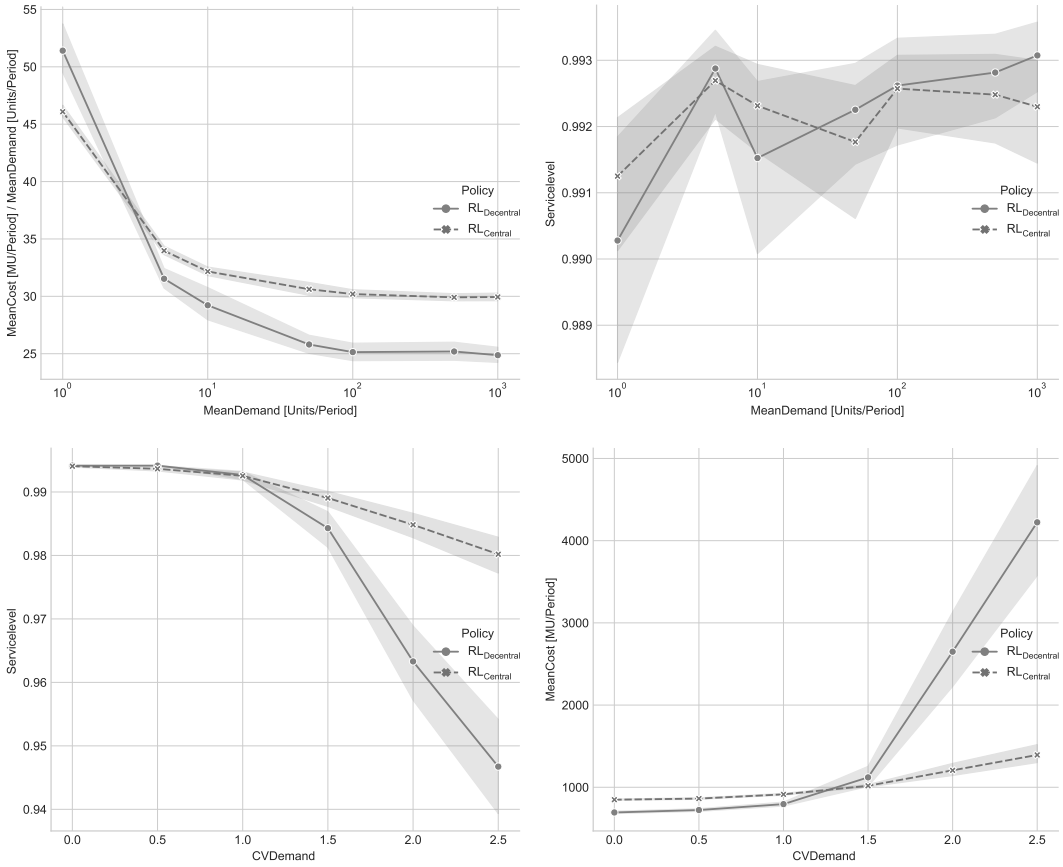

Figure 6.3: Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Mean Demand and the CV of Demand are Varied

The next test scenario changes the replenishment characteristics. On the one hand, the expected replenishment times are varied, while on the other hand, the corresponding CVs are varied. The assumption (P2) of stable mean stock levels per replenishment time period is largely confirmed by the upper-left graph in Figure 6.4. The top-right plot of service level

against changing mean replenishment lead times indicates a relatively stable service level at higher mean replenishment lead times. Sticky points are again found when the mean replenishment time falls below the seven periods known from training. In this case, the decentralized model seems to stock too few units, while the centralized policy seems to stock too many units at the right time. This is reflected in a drop in service level for both models, although the drop is greater for the decentralized model than that for the centralized model.

When the variance of replenishment times is considered, the expected behavior analysed in the direction of (P3) can be observed in the two lower graphs of Figure 6.4. An increasing variance in replenishment times leads to higher mean stock levels, with those of the central model generally being higher. For all CVs greater than the known $CV(L) \geq 1.0$, service levels are stable at a high level. It is only at lower CVs that both models do not seem to cope so well. This is reflected in the low service levels for the decentralized model, which are again lower than those for the centralized model.

When the resulting costs of changing the replenishment characteristics are examined, a similar relationship between the models can be observed as in the case of the demand parameters: In general, for magnitudes higher than the known values, the decentralized model seems to lead to lower costs when the mean values of demand or replenishment time are changed. However, this is not the case as the variance increases, where the central model dominates. This is also partially true for the ranges below the known values, as described above.

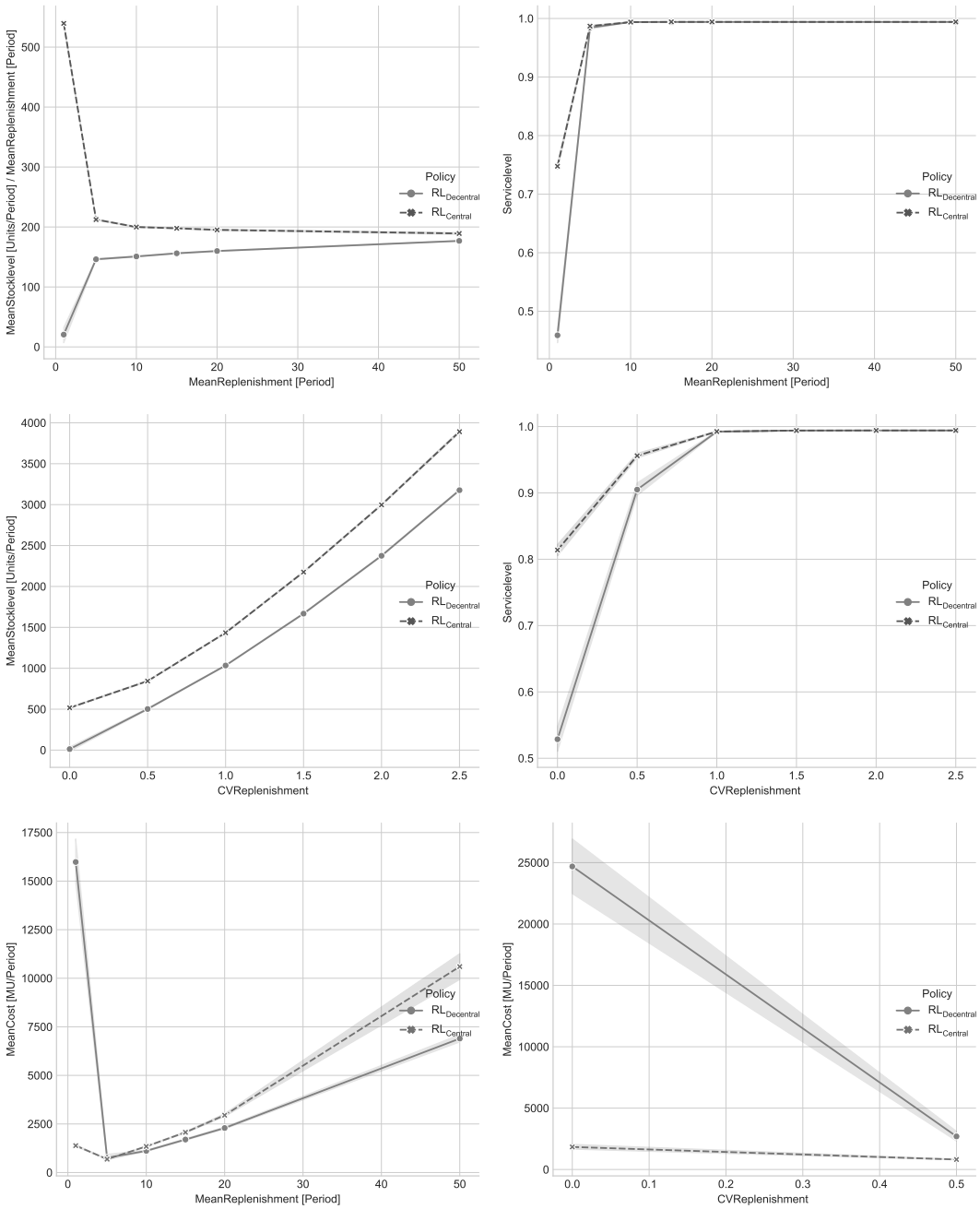

Figure 6.4: Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Mean Replenishment Time and the CV of Replenishment Time are Varied

A final test scenario is provided by varying the three different cost parameters by a single factor. The results are presented in Figure 6.5. Starting with

the holding costs, one can observe the expected decrease in the average inventory levels, as anticipated in (P4a). In addition, one sees a drop in service level when holding costs reach the amount of fixed order costs at a value of 10. Holding costs that exceed the values of 10 and 100 become similar and greater than the fixed order and backorder costs, respectively. As a result, the service level is less important than holding costs. This becomes even more obvious when one examines the last data points that reach holding costs of > 100 [MU].

With varying fixed order costs, one would expect fewer orders as these drive up costs. Interestingly, the centralized model seems to have a policy of ordering less frequently in general than the decentralized model. Nevertheless, for both models, the number of orders during a planning episode decreases as fixed order costs increase. Again, as fixed order costs approach the backorder cost range of 100, service levels become less important and therefore decrease drastically.

A final observation can be made by varying the backorder cost. Backorder costs have a direct impact on service levels: As soon as they are lower or similar to the other cost parameters, the service level is also low. On the other hand, all tested values that are higher than the other cost parameters lead to a stable service level in both models at the corresponding and previously known service level. From this point on, increasing backorder costs only lead to an increase in total costs because a similar number of backordered items become more expensive. The first peak in average cost is caused by the lower service level.

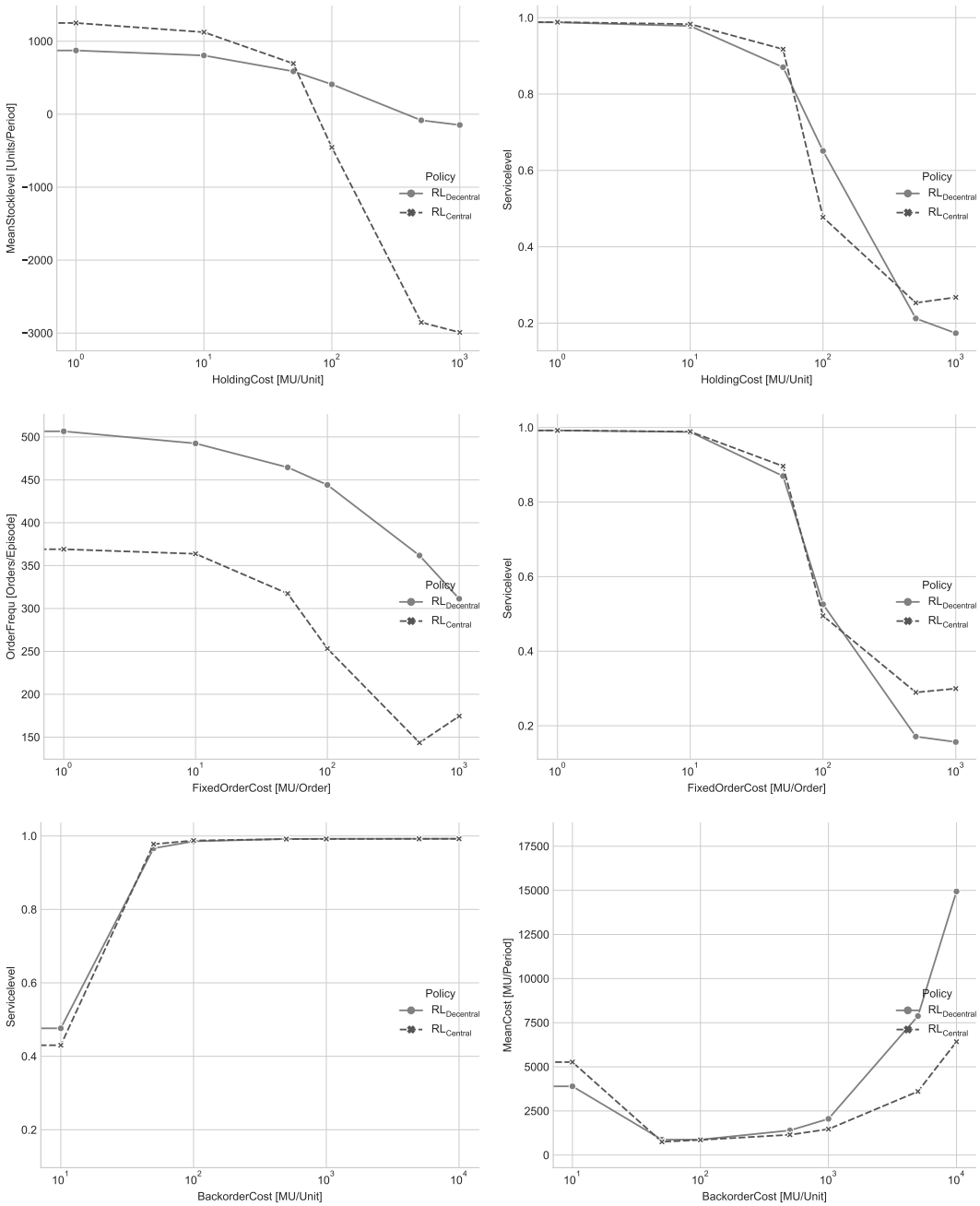

Figure 6.5: Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Cost Parameters are Varied

To summarize the analysis, both models exhibit very similar responses to changing environmental parameters. The centralized model seems to han-

dle more variable demand better than the decentralized model. The same can be said for more variable replenishment times. The cost parameters are treated very similarly, such that both models adapt to changing cost ratios and attempt to optimize the costs by considering the dominant cost parameter. This would lead to low average inventory levels, low order frequencies, and high service levels in the face of high holding costs, fixed order costs, and backorder costs.

6.3 Analysis of the Bullwhip Effect Caused by a Demand Shock

As described in Chapter 2, several reasons exist for why the BWE occurs. In the present scenario, the causes of price fluctuations can be ignored, as prices are assumed to be fixed. Order bundling occurs naturally, as the models consider fixed order costs and therefore bundle orders. As a result, the measure of BWE is expected to be positive regardless of a disruption in demand. Therefore, the same demand evolution with and without a disruption is compared to obtain a sense of the inherent BWE due to order batching. The two main causes investigated are the remaining causes: By comparing the decentralized and centralized approaches, the effect of information centrality across the SC is investigated.

Two types of data set are used: Chapter 4's original data set is used as the baseline data set, while the data set that models the demand shock is constructed in a very similar way, with the exception of an unexpectedly high demand in the middle of an episode. This unexpectedly high demand is equal to $10xE(D)$, as this is a value that will not be reached by the underlying demand distribution, which is gamma distributed with E(D)=30 and CV(D)=1.0.

By comparing several performance indicators of these two data sets, the BWE and the resulting changes in the KPIs caused by the demand shocks should be quantified. The first KPI measured is that of the approach pre-

sented by Chen et al. (2000), which relates the variance in order values to the variance in demand:

$$BWE_n = \frac{Var(O_{n,t})}{Var(D_{n,t})}, \forall n \in N, t \in T \tag{6.1}$$

To this end, a high BWE would imply a higher variance of ordered than demanded quantities. This is exactly the effect that Forrester (1961) observed when describing the BWE.

Examining the results in Figure 6.6 and Table 6.2, one can observe different responses to the demand shock. The decentralized model in the first node, the one closest to the customer, exhibits an increase in variance when faced with the demand shock. By contrast, the two subsequent nodes even exhibit a decrease in the mean of the measured BWE. The situation for the central model is different for the observation at the first SC node: There, the unexpected increase in demand leads to a decrease in the measured BWE. The following two nodes also remain relatively stable. For the second node, a significant change in the BWE measurement must be rejected, while for the third node a significant increase can be reported, but at a low level. This seems to indicate that the first node in the SC is already able to absorb the demand shock without propagating the demand variance further down the SC.

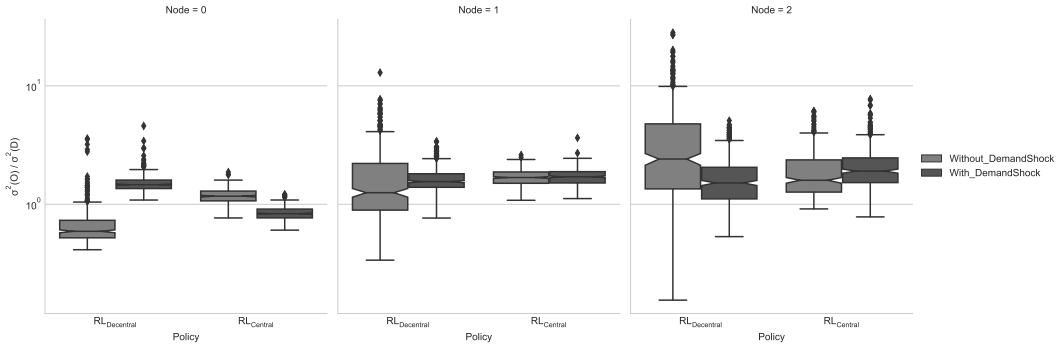

Figure 6.6: BWE Measurement with Equation 6.1 per Node and Model

A further analysis is conducted by examining the KPIs of mean cost per period and the resulting service level of the SC. Figure 6.7 presents a com-

				μ	σ	U	p-value
$\frac{\sigma^2(O)}{\sigma^2(D)}$	Decentral Model	Node 0	Without Demand Shock	0.690	0.364	4470.0	0.000
			With Demand Shock	1.516	0.300		
		Node 1	Without Demand Shock	1.860	1.535	55,430.0	0.000
			With Demand Shock	1.684	0.450		
		Node 2	Without Demand Shock	3.911	4.175	47,656.0	0.000
			With Demand Shock	1.758	0.877		
$\frac{\sigma^2(O)}{\sigma^2(D)}$	Central Model	Node 0	Without Demand Shock	1.188	0.166	4671.0	0.000
			With Demand Shock	0.844	0.103		
		Node 1	Without Demand Shock	1.711	0.272	73,655.0	0.441
			With Demand Shock	1.712	0.280		
		Node 2	Without Demand Shock	1.987	1.013	59,354.0	0.000
			With Demand Shock	2.175	1.006		

Table 6.2: Comparison of BWE Measures of the Two Models when Exposed to a Demand Shock

				μ	σ	U	p-value
Mean Cost	Decentral Model	Node 0	Without Demand Shock	367.37	380.39	8,017.0	0.000
			With Demand Shock	3,835.97	5457.24		
		Node 1	Without Demand Shock	222.67	170.96	4,394.0	0.000
			With Demand Shock	1,765.40	1,988.18		
		Node 2	Without Demand Shock	234.92	191.01	4,270.0	0.000
			With Demand Shock	1,246.78	872.62		
Mean Cost	Central Model	Node 0	Without Demand Shock	338.49	90.87	47,080.0	0.000
			With Demand Shock	493.59	312.07		
		Node 1	Without Demand Shock	325.02	13.91	52,213.0	0.000
			With Demand Shock	317.78	16.74		
		Node 2	Without Demand Shock	179.64	122.27	58,617.0	0.000
			With Demand Shock	254.34	272.30		

Table 6.3: Comparison of Mean Cost Measures of the Two Models when Exposed to a Demand Shock

parison of mean cost per period and service level before and after the demand shock for each model and node. Comparing the mean costs without and with the demand shock reveals an increase in the mean costs per period for both models when a demand shock occurs. Table 6.3 provides means and standard deviations as well as the results of a Mann-Whitney U test are given. Nevertheless, the increase in average costs is significantly higher for the decentralized model than for the centralized one. Looking at the lower service level graphs, it is clear that the increase in mean cost is caused by a decrease in service level.

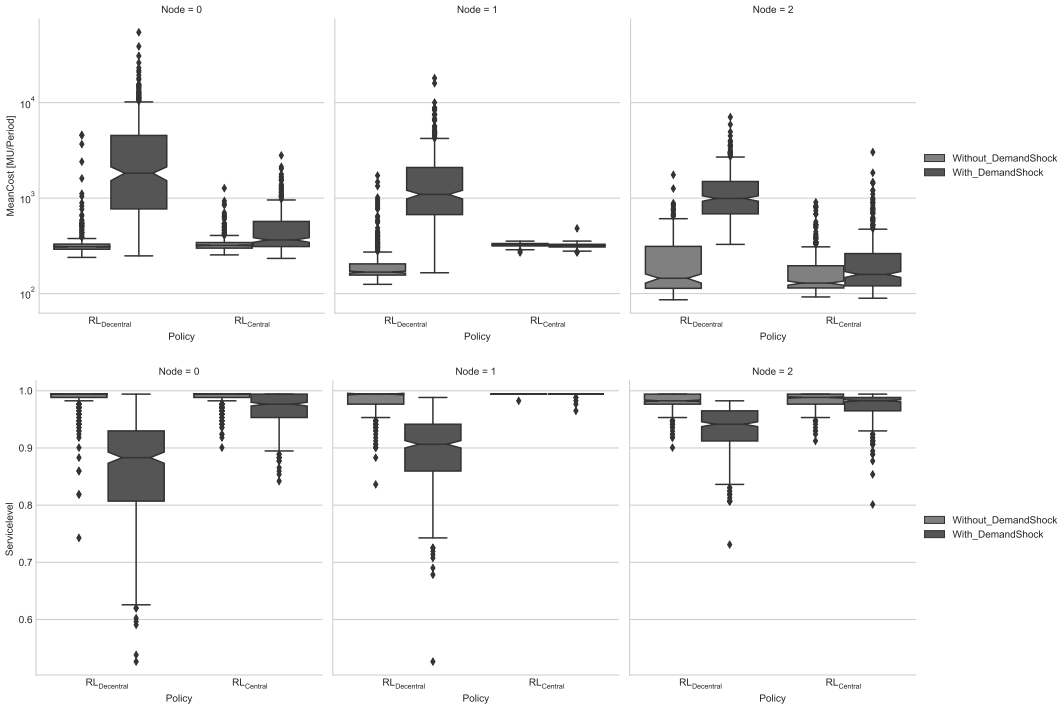

Figure 6.7: Change in Mean Cost and Service Level per Node and Model when Confronted by a Demand Shock

The expected increase in demand due to the introduced demand shock is not unambiguously measurable by Equation 6.1. However, the effect can be seen in the KPIs used in the system analysis above.

6.4 Chapter Conclusion

The results of this chapter can be summarized as follows:

It was possible to find a decentralized policy that works similarly well when plugged into an SC several times compared with a centralized model. This is promising because the decentralized models were not adapted and not newly trained, but rather taken as they were trained for a single SC location and switched in series for a linear SC. This has the advantage of not requiring retraining for new SC configurations.

When tested on unknown data, both the centralized and decentralized models performed quite well, especially when average demand and replenishment times and CVs became higher than the known and trained values. By contrast, lower values seemed to cause problems in the form of higher costs. A possible countermeasure would be to attempt to train on generally lower values or to use conventional models when confronted with less complex environments.

Demand shocks, as modeled in the present work, seemed to mainly affect service levels and the resulting costs, but they did not involve a high amplification of demand throughout the SC. It is here that the centralized model seemed to demonstrate its advantages over the decentralized model, as the losses in service level and the resulting backorder costs were lower for the centralized model.

7 Application to Real-World Data

Everyone you meet is fighting a battle you know nothing about. Be kind. Always.
– Ian MacLaren

Given the theoretical basis of the previous three chapters, the aim of this chapter is to assess the applicability of the model to real-world data. Therefore, the selection of data and the underlying assumptions are presented in the first section followed by the presentation of the results when the RL model is applied to real-world data. Then, the chapter concludes with a discussion of the objectives achieved and the weaknesses of the approach.

7.1 Selection of the Data and Assumptions

The theoretical examples use data points such as demand data in the form of realized and forecast demand as well as replenishment data and cost parameters, such as holding, fixed order, and penalty costs. For the real-world use case, an equivalent must be found. Today, most companies use ERP systems to plan their operations. These range from systems for high-level network planning in the SCM area to manufacturing execution system (MESs) for detailed planning. (Schuh 2007) One of the most commonly used ERP systems is that provided bySAP™
texttrademarkṪhe SAP™ ERP system is as well used by the company whose data are used for this real data use case.

As the present work deals with the problem of replenishment and the question of how much and when to reorder material or intermediate products, the field of application falls within SAP™ materials planning, which takes place towards the supplier. The task is often performed by an inventory and production planner, supported by the *MD04* transaction, which provides an overview of future requirements and orders already placed. The requirements in this case come in the form of a secondary requirement triggered by a planned or production order. In addition, the SAP™ ERP system is able to propose a replenishment plan based on the settings that correspond to the SC of the particular material. On this basis, the inventory planner can accept or adjust the proposed replenishment plan.

The present work picks up at this point: Data from the *MD04* transaction are used as input for the RL policy to generate a replenishment plan. Therefore, several specific products are considered, and the evolution of the corresponding stock levels, demand, and orders placed by the dispatcher are observed over a certain time interval. Replenishment times play a minor role in this approach, as the dispatcher usually determines the specific date on which the material is to be delivered. It is then the responsibility of the supplier to deliver on time on that particular date. Of course, replenishment time plays a role in the short-term nature of orders, as the supplier requires sufficient lead time to be able to deliver on time. Under this assumption, the results of the RL model's decisions in this particular case are to be seen as pre-planning rather than live planning on a daily basis. Therefore, the lead times are set to the deterministic value of one day.

To select specific materials and their part numbers in specific locations, the procedure in Figure 7.1 is followed to retrieve a random and heterogeneous collection of different products. The database for all the data is a data lake that contains the raw SAP™ tables. These are listed in the appropriate sections below.

First, all material-location combinations in the time interval between 01-Jul-21 and 01-Nov-21 that appear in the demand tables (i.e., the secondary demand [RESB]) are considered. Secondary demand is generated by production orders, which in turn are generated by customer demand; 500 of

these combinations are chosen arbitrarily. For material-location combinations for which there are at least 10 demand periods and for which price information is available, the above-mentioned data points of demand, corresponding stocks (LQUA), orders manually executed by the MRP controller (EKET/EKPO), and price information (MBEW) are retrieved. Price information is used to calculate holding costs and is therefore mandatory. The requirement of at least ten demand points is based on the need to derive a reasonably adequate distribution of demand, while also not excluding too many materials. Another requirement is the availability of stock data for this specific time interval. Inventory data are important for two reasons: First, it is necessary to define initial inventory levels for the time interval under consideration; and second, a comparison of average stock levels would at least be interesting for evaluating the performance of the RL policy. Stock levels at the specific start date are rarely available. It is therefore assumed that the initial stock levels correspond to the average stock levels obtained from the data. This results in a total of 111 unique material-location combinations.

After this preprocessing of the data, the RL agent from Chapter 5 is applied. It calculates the scaling factors for each material introduced in Chapter 5 and predicts orders for each period and material based on the updated stock levels and corresponding cost parameters. The agent is not trained on the described data; rather, it is used as is.

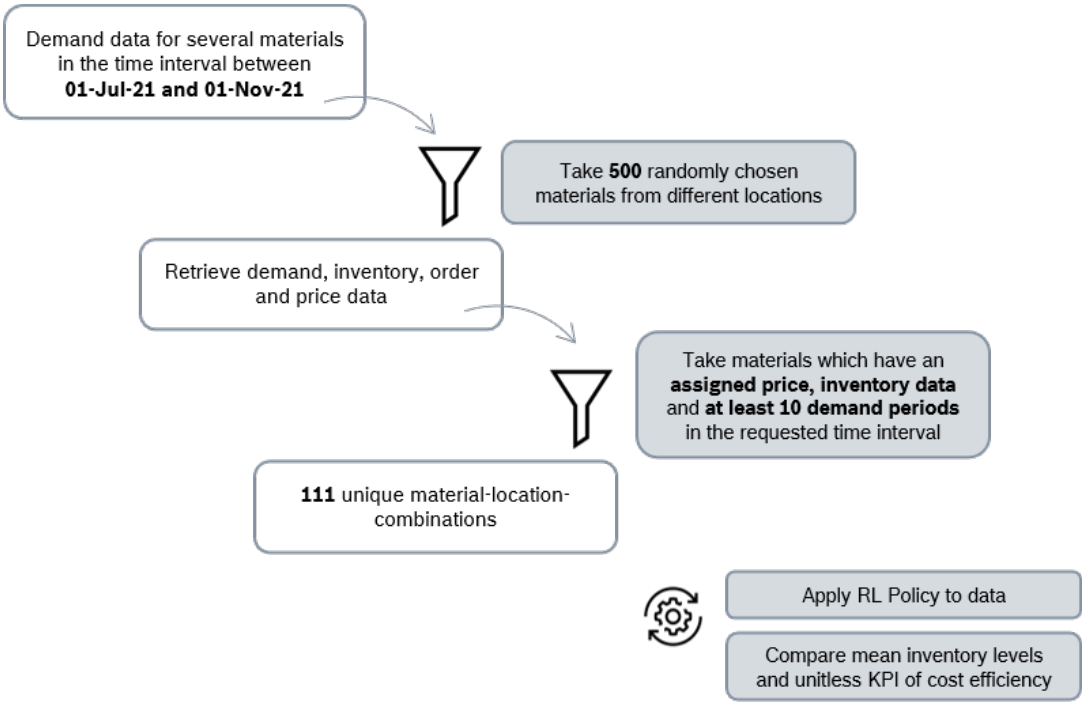

Figure 7.1: Procedure for Selecting Material-Location Combinations

The cost parameters are known to be difficult to determine and are therefore derived from several sources: Holding costs comprise the administrative costs of warehousing plus handling and capital costs. Reality has demonstrated that administrative costs are negligible, while capital costs are known to be the major component. Capital costs are usually determined as a percentage of the product price, with a percentage higher than the interest rate charged by a bank. (Axsäter 2015) Arnold et al. (2008, p.232) suggest a capital cost rate between 4% and 20%; a more tailored rate can be found with Equation 7.1 from Pulic (2005-2015):

$$\text{Stock interest rate} = \text{interest rate} * \frac{\varnothing \text{storage time}}{360} \tag{7.1}$$

A working capital study by Deloitte (2019) reports average holding times for certain industries, such as 53 days for automotive and 79 days for en-

gineering. Considering the current low interest rates close to zero[1] and assuming at least 1% for ease of calculation, the capital rate would actually be between 0.15% and 0.22%. Nevertheless, a comparison is made with a more traditional capital rate of approximately 10%. Holding costs are therefore calculated as a fraction of 10% of the material price.

Fixed order costs are highly dependent on the routes and means of transport used. In addition, each company optimises its transport routes through a specific system of full truck load (FTL), less than truck load (LTL), and milk run organization. For the present example, an average value of 400€per order is assumed, which results from the evaluation of internal historical and publicly available data on intra-European routes by truck and different types of charges.

According to Axsäter (2015), backorder costs are difficult to estimate because they usually consist of a number of different parameters. They could include the cost of re-organising after a delivery is backordered, the consequences of a missing part on the assembly line, or the cost of buying the part elsewhere at a higher price. Since backorders should be avoided at all costs, this is reflected in an arbitrarily high backorder cost of 1000€per unit.

7.2 Results

The data sample obtained by using the aforementioned procedures consists of 111 unique material-location combinations from 12 different sites, and it can be found in Bergmann (2023a). Demand and price data are distributed as illustrated in Figure 7.2, where the left side of the graph shows the demand characteristics, while the right side shows the distribution of mean demand and corresponding prices. One can see that the mean demand per day for a share of approximately 92.8% of the materials lies in the interval up to 1000 units per day, while the CVs seem to be comparatively high with

1. European Central Bank (ECB) Interest Rates https://www.ecb.europa.eu/stats/policy_and_exchange_rates/key_ecb_interest_rates/html/index.en.html

the 90% quantile at 3.05. Prices for 90% of the materials are in the interval between (0, 228.25).

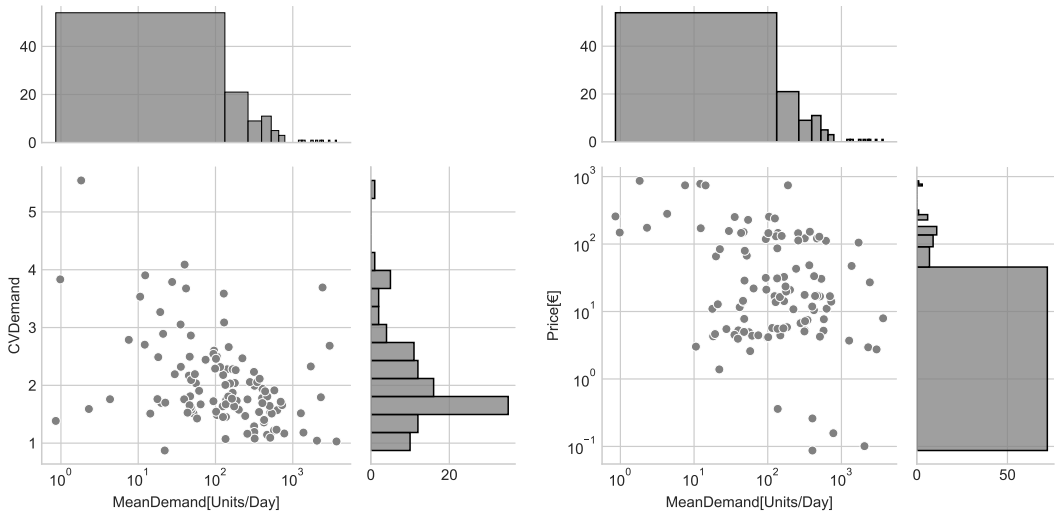

Figure 7.2: Distribution of Input Data

After applying the RL policy to the demand series, it can be generally stated that the service levels for all demand series are 100%, as was achieved in reality. The focus of interest is now on the comparison with reality for the resulting KPIs.

First, the average stock levels for the 111 selected materials are examined: In 90 cases of the selected 111, the mean stock levels can be reported lower and could be reduced by an average of 1919.42 units when the RL policy is switched to. The graph to the left of Figure 7.3 presents a histogram of all absolute reduced stock levels. Putting this absolute number of reduced stocked units in relation to the corresponding average demand in the right graph, the reduction is a multiple of up to 40 times the average demand and on average 8.8 times the average demand.

The following unitless KPI is used as another way to assess the average cost per unit demanded and price. The general level of demand and price-dependent holding costs have been identified as the main drivers of absolute costs. Consequently, these should be neutralized by the following formula as an adaptation of Equation 4.11 in Section 4.6:

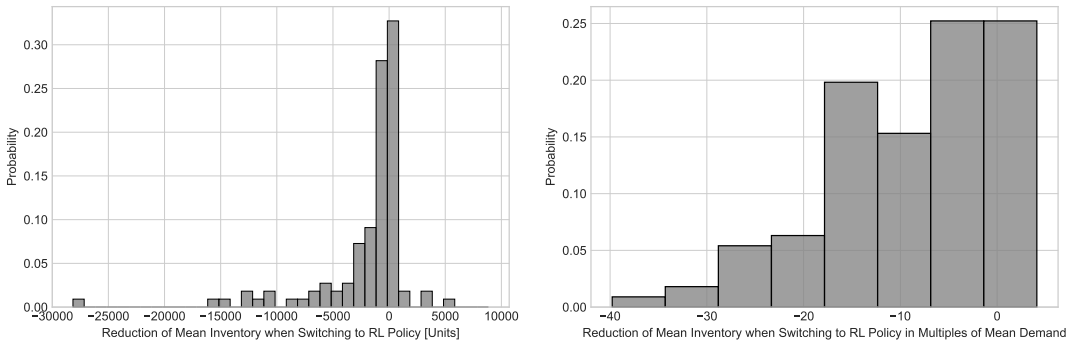

Figure 7.3: Histogram of the Yielded Inventory Reduction when the RL Policy is Applied to the Data

$$\overline{C_{Eff}} = \frac{\overline{C_t}}{E(D)h} \tag{7.2}$$

The KPI distribution for the theoretical case when the robust model $RL_{Adaptive+Cost}$ and the KPI distribution are tested for the current real data use case can now be compared. The mean KPIs are 27.86 and 14.66, respectively, which promise satisfactory behavior of the RL policy when it is applied to the real data. The KPIs from the theoretical test scenarios start at a lower level, with a small peak in the single digits; however, they have a longer right tail towards worse KPI ranges. This is confirmed by the standard deviations of 56.54 and 14.92 for theory and reality, respectively.

7.3 Chapter Conclusion

Overall, this small example of applying the RL policy to arbitrary real-world part numbers demonstrates several things: Under the assumptions made, inventory could be reduced with similar to better cost efficiency. Due to the data situation, no one-to-one comparisons could be made. Nevertheless, the auxiliary KPIs are suitable for determining the general trend. This opens up the possibility of applying the model to real use cases; yet, this does not release one from the obligation to conduct further studies. It is recommended that the database should be strengthened in terms of cost pa-

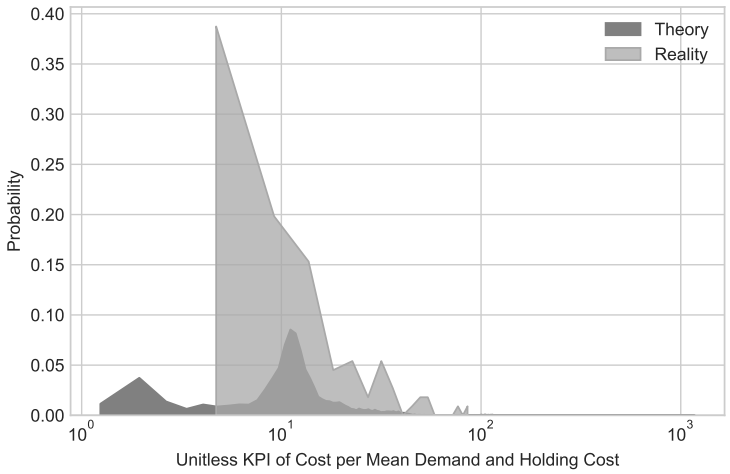

Figure 7.4: Histogram of KPI in Theoretical Experiments and when Applied to Real-World Examples

rameters and, in particular, time series that are complete in terms of stock and order information. This would allow a highly accurate comparison of the model's behavior with reality. In addition, it might be helpful to add other data features, such as information on the grouping of the material into an A, B, or C classification. Such data points could provide an indication of why the RL policy might not work so well for some of the materials.

If the results are still promising, a next step could be to test the model not only on historical data but also on a live system. Past data have the disadvantage of having states where all of the changes of the day have already happened and are aggregated into a clear-looking state description of that particular day. By contrast, data normally change throughout the day, forecast demand changes as it gets closer to the call-off day, and both are subject to fluctuations that a real-time policy would also face.

8 Conclusion

I regret nothing in life but the things I have not done.

– Coco Chanel

This chapter summarizes the main findings of this thesis and outlines further research and applications using extended models.

8.1 Summary

The present research was conducted under the assumptions of mainly three research questions, which are restated and responded to as follows:

How might one develop an RL system for SCs that contend with stochastic demand and stochastic replenishment times and are inclusive of forecast errors? Which SC characteristics are crucial to incorporate?

The literature review in Chapter 3 revealed how previous approaches have addressed certain model decisions concerning the underlying SC model as well as those for the RL part. By describing the SC environment in Chapter 4, the requirements for the setting became clear: An agent should decide on the action replenishment quantities and dates. Unlike most previous approaches in the literature, the SC environment was modeled with stochastic replenishment times and the probability of forecast errors. The training algorithm and the policy and value networks chosen are standard ones from the work of *OpenAI*, as the focus should be on the design decisions rather than on the development of yet another efficient training algorithm and policy approximator. After the description of the environ-

ment and the assumptions about the synthetic training data, the creation of the test scenarios was clarified in the later sections of Chapter 4.

How reliable and robust are the decisions made by the basic model? How is the dependability affected by the structure of the state and action space, and how competently does the developed model perform compared with a traditional policy?

Systematically generated synthetic training and test data helped to answer the second research question in Chapter 5. The aim was to develop a RL model that could be adapted to different SC environments without further training. The major advantage of such a model would be its consistent performance when applied to different products with lower or higher demand, with more or less variance, different replenishment times, and different cost structures. Thinking of a company, it is easy to imagine a large number of products that could be planned in this way, and where not having to train for each product would save much time. To achieve this, adaptive scaling of the state and action space was used. In addition, the state space consisted of the information points of stock level and cost parameters. Compared with no adaptive scaling, a significant cost reduction was achieved. The inclusion of the cost parameters also helped the model to adapt to changing ratios of the cost parameters; for example, when holding costs increased relative to the other cost parameters, the model attempted to reduce the average stock levels to reduce costs.

How should an RL system be designed for linear SCs that contain several stages? How does a central approach perform versus decentrally trained models and how do they react to demand shocks?

In Chapter 6, a comparison was made between a decentralized and a centrally controlled linear SC. The best model as determined by the robustness analysis in Chapter 5 was taken as an agent for each SC node in an SC that consisted of three nodes. In addition, a central agent was trained to make decisions for all three nodes simultaneously. The analysis demonstrated that it is possible to find a decentralized policy that performs at the same level as the central one when switched in series. In addition, an analysis of

responses to demand shocks was investigated. Overall, the sudden increase in demand mainly affected the decentralized model, while the centralized model seemed to cope well.

How would such a previously defined and theoretically tested model perform on real data?

Chapter 7 explained how the real-world database was created, which specific part numbers were chosen, and how the model worked on these time series. The well-known problem of the lack of sufficient and complete data necessitated assumptions, which should be reviewed in future work. For the present work, the results of the analysis point in a promising direction, as a reduction in average stock levels can be achieved without a reduction in service levels.

8.2 Outlook

The research does not stop here: As far as SC models are concerned, it is easy to imagine that they will become more complicated with increasing globalization, and that they are not at all represented by the present model. The number of relationships, suppliers, and customers increases, but the number of uncertainties could also be modeled more precisely. A useful extension could be to disaggregate replenishment times as is and instead model the availability of transport and production resources and failure probabilities.

In these more complex environments, there may be other possible extensions of the state and action space: Of course, research could go in the direction of giving the model more information up to every available data point of the SC. Actions could consist not only of the quantity to be ordered on a given day but perhaps also of a complete replenishment plan for a given period in the future. Interesting actions could be inspired by the decision space of today's schedulers, such as speeding up the delivery of orders by organizing a special tour, reprioritizing production, or negotiating with different suppliers.

The example of work such as Hubbs et al. (2020) indicates that the topic has great potential for future work: The authors developed something like standard environments for several well-known operations research problems, including the case of inventory management. Standardizing problems in the form of implemented code could make these problem instances more accessible and therefore more widely used in the future.

Finally, there seems to be almost no limit to the extension of the current model, especially if one considers one of the many studies that have predicted the possible future of SCs. For example, the KPMG study *Road to Everywhere - Future of SCs* [1] and several articles on the future of manufacturing [2] have identified the following three main trends defined by often-quoted buzzwords:

The first trend is that of SC transparency. Driven by the B2C market, the need for greater SC transparency is also emerging in the B2B market. Customers want to know about the status of their delivery, possible disruptions, and the resulting consequences. Technologies such as sensor data, RFID, and GPS tracking are bringing an increasing number of devices into the Internet of Things (IoT) age. As a result, there will be many more data points to process, enabling a more realistic digital twin of supply chains and factories.

This leads to the second trend in cyber-physical systems, which is that existing digital copies of reality can be assembled using the collected data. These could then be combined to form increasingly larger networks of cyber-physical systems, all of which interact across company boundaries. In addition to traditional SC players, another emerging group of future players will be SC platforms that offer SC, manufacturing, or logistics as a service. The consequence of this is short-term and constantly changing SC partners.

1. https://home.kpmg/xx/en/home/services/advisory/management-consulting/optimize-your-sector-operations/future-supply-chain.html
2. https://vdma-verlag.com/home/future_manufacturing_DE.html#modal-cookiewarning

Moreover, the sheer volume of data and the sheer number of partners are leading to increases in complexity and data, while speed and responsiveness remain crucial factors. This gives rise to the third trend—namely decision automation. To process the huge amount of data, pre-sort it, handle standard decisions, and free up human decision makers for more impactful questions, an increasing number of decisions will be made by systems. These systems, embedded in the aforementioned cyber-physical systems, will consist of multiple agents, both human and digital, interacting with each other.

The present work is intended to lay the foundations for dealing with the aforementioned future trends, and will certainly need to be developed further in order to achieve this vision.

Nomenclature

General

b	Scale parameter of the gamma distribution
p	Shape parameter of the gamma distribution
A2C	Actor-Critic Algorithm
ANN	Artificial Neural Network
BWE	Bullwhip Effect
CSCMP	Council of Supply Chain Management Professionals
CV	Coefficient of variation
E(X)	Expected value of the randomly distributed variable X
EOQ	Economic Order Quantity
ERP	Enterprise Resource Planning
IoT	Internet of Things
MDP	Markov Decision Processes
MES	Manufacturing Execution System
MU	Monetary units
PDF	Probability Density Function
PPO	Proximal Policy Optimization

RL Reinforcement Learning

SC Supply Chain

SCM Supply Chain Management

tanh Tangent hyperbolicus

TRPO Trust Region Policy Optimization

Var(X) Variance of the randomly distributed variable X

VMI Vendor Managed Inventory

Literature Review

BO Backorders

C Central planning

D Divergent supply chain structure

DC Decentral planning

E Expired Items cost

F Fixed ordering cost

H Holding cost

IP Inventory position

L Linear supply chain structure

LS Lost sales

M Manufacturing cost

N Leveling cost

P Penalty cost

T Transportation cost

Model

$\overline{C}$	Mean cost per period
$\overline{I^p}$	Mean stock level
b_n	Backorder cost per unit at node n
C_t	Total cost in t
D	Random variable of customer demand
d_t	Realized amount of demand in period t
$d_{n,t}$	Realized amount of demand at node n in period t
FE	Random variable of forecast error
$fe_{n,t}$	Realized forecast error in period t for node n
h_n	Holding cost per unit at node n
$I^p_{n,t}$	Physical available inventory at node n in period t
K_n	Fixed order cost per order at node n
L	Random variable of replenishment time
$l_{n,t}$	Realized replenishment time in period t for node n
N	Total number of nodes in the supply chain
n	Index for a node in the supply chain
o_t	Order amount in period t
$o_{n,t}$	Ordered amount at node n in period t
T	Number of periods in a planning episode
t	Index for the planning period

Bibliography

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. *Optuna: A Next-generation Hyperparameter Optimization Framework*. Accessed March 29, 2024. http://arxiv.org/pdf/1907.10902v1.

Arnold, Dieter, and Kai Furmans. 2009. *Materialfluss in Logistiksystemen*. 6. Aufl. 2010. VDI-Buch. Berlin, Heidelberg: Springer Berlin Heidelberg. http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1544165.

Arnold, Dieter, Heinz Isermann, Axel Kuhn, Horst Tempelmeier, and Kai Furmans. 2008. *Handbuch Logistik*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72929-7.

AurelianTactics. 2018. *PPO Hyperparameters and Ranges*. https://medium.com. Accessed May 11, 2021. https://medium.com/aureliantactics/ppo-hyperparameters-and-ranges-6fc2d29bccbe.

Axsäter, Sven. 2015. *Inventory Control*. Vol. 225. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-15729-0.

Bauer, Christian, Klaus Bauer, Sebastian Becker, Lena Bergmann, Manuel Beuttler, Hans-Peter Bock, Eike Broda, et al. 2021. *Abschlussbericht des Forschungsprojekts "Broker für Dynamische Produktionsnetzwerke"*. https://doi.org/10.5445/IR/1000141238.

Becker, Till, Daniel Weimer, and Jürgen Pannek. 2015. "Network structures and decentralized control in logistics: topology, interfaces, and dynamics." *International Journal of Advanced Logistics* 4 (1): 1–8. https://doi.org/10.1080/2287108X.2015.1012329.

Bergmann, Lena. 2023a. *Parameters and KPIs from the Real Data Use Case.* https://doi.org/10.35097/1683.

———. 2023b. *Results for the Different Parameter Tests of the Developed RL Agents and the Benchmark Algorithm.* https://doi.org/10.35097/1682.

Bergmann, Lena, Johannes Dümmel, and Yinglai Tang. 2021. "Deriving Machining Processes from Technical Drawings: An Approach Motivated by Cloud Manufacturing Platforms Using Convolutional Neural Networks." *Industrie 4.0 Management,* no. 37 (5), 21–25.

Berling, Peter. 2005. "On Determination of Inventory Cost Parameters." PhD Dissertation, Lund University.

Bhandari, Aniruddha. 2020. *Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization.* https://www.analyticsvidhya.com/blog/. Accessed April 7, 2024. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016a. *OpenAI Gym.* Edited by GitHub. Accessed April 7, 2024. https://github.com/openai/gym.

———. 2016b. *OpenAI Gym.* Accessed April 7, 2024. http://arxiv.org/pdf/1606.01540v1.

Chaharsooghi, S. Kamal, Jafar Heydari, and S. Hessameddin Zegordi. 2008. "A reinforcement learning model for supply chain ordering management: An application to the beer game." *Decision Support Systems* 45 (4): 949–959. https://doi.org/10.1016/j.dss.2008.03.007.

Chen, Frank, Zvi Drezner, Jennifer K. Ryan, and David Simchi-Levi. 2000. "Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information." *Management Science* 46 (3): 436–443. https://doi.org/10.1287/mnsc.46.3.436.12069.

CSCMP. 2013. *Supply Chain Management Terms and Glossary*. Accessed January 8, 2021. https://cscmp.org/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx.

Das, Tapas K., Abhijit Gosavi, Sridhar Mahadevan, and Nicholas Marchalleck. 1999. "Solving Semi-Markov Decision Problems Using Average Reward Reinforcement Learning." *Management Science* 45 (4): 560–574. https://doi.org/10.1287/mnsc.45.4.560.

Davis, Tom. 1993. "Effective Supply Chain Management." *Sloan Management Review,* no. 34, 35.

Deloitte, ed. 2019. *Working Capital Studie: Flüssige Mittel und gebundenes Kapital*. Accessed September 10, 2021. https://www2.deloitte.com/content/dam/Deloitte/de/Documents/finance/working-capital-studie-2019.pdf.

Dogan, Ibrahim, and Ali R. Güner. 2015. "A reinforcement learning approach to competitive ordering and pricing problem." *Expert Systems* 32 (1): 39–48. https://doi.org/10.1111/exsy.12054.

Forrester, J. W. 1961. "Industrial Dynamics." *MIT Press, Cambridge.*

Fransoo, Jan C., and Marc J.F. Wouters. 2000. "Measuring the bullwhip effect in the supply chain." *Supply Chain Management: An International Journal* 5 (2): 78–89. https://doi.org/10.1108/13598540010319993.

Giannoccaro, Ilaria, and Pierpaolo Pontrandolfo. 2002. "Inventory management in supply chains: a reinforcement learning approach." *International Journal of Production Economics* 78 (2): 153–161. https://doi.org/10.1016/S0925-5273(00)00156-0.

Gokhale, Apoorva, Chirag Trasikar, Ankit Shah, Arpita Hegde, and Sowmiya Raksha Naik. 2021. “A Reinforcement Learning Approach to Inventory Management.” In *Advances in Artificial Intelligence and Data Engineering,* edited by Niranjan N. Chiplunkar and Takanori Fukao, 1133:281–297. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-3514-7{\textunderscore}23.

Goodwin, Jack S., and Stephen G. Franklin. 1994. “The Beer Distribution Game: Using Simulation to Teach Systems Thinking.” *Journal of Management Development* 13 (8): 7–15. https://doi.org/10.1108/02621719410071937.

Hill, Ashley, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, et al. 2018. *Stable Baselines.* Edited by GitHub. Accessed April 7, 2024. https://github.com/hill-a/stable-baselines.

Hubbs, Christian D., Hector D. Perez, Owais Sarwar, Nikolaos V. Sahinidis, Ignacio E. Grossmann, and John M. Wassick. 2020. *OR-Gym: A Reinforcement Learning Library for Operations Research Problems.* Accessed April 7, 2024. http://arxiv.org/pdf/2008.06319v2.

Ivanov, Dmitry, Alexander Tsipoulanidis, and Jörn Schönberger. 2019. *Global Supply Chain and Operations Management.* Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-94313-8.

Jiang, Chengzhi, and Zhaohan Sheng. 2009. “Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system.” *Expert Systems with Applications* 36 (3): 6520–6526. https://doi.org/10.1016/j.eswa.2008.07.036.

Kara, Ahmet, and Ibrahim Dogan. 2018. “Reinforcement learning approaches for specifying ordering policies of perishable inventory systems.” *Expert Systems with Applications* 91:150–158. https://doi.org/10.1016/j.eswa.2017.08.046.

Kim, Chang Ouk, Ick-Hyun Kwon, and Jun-Geol Baek. 2008. "Asynchronous action-reward learning for nonstationary serial supply chain inventory control." *Applied Intelligence* 28 (1): 1–16. https://doi.org/10.1007/s10489-007-0038-2.

Kim, Chang Ouk, Ick-Hyun Kwon, and Choonjong Kwak. 2010. "Multi-agent based distributed inventory control model." *Expert Systems with Applications* 37 (7): 5186–5191. https://doi.org/10.1016/j.eswa.2009.12.073.

Kim, Chang Ouk., J. Jun, J. K. Baek, R. L. Smith, and Y. D. Kim. 2005. "Adaptive inventory control models for supply chain management." *The International Journal of Advanced Manufacturing Technology* 26 (9-10): 1184–1192. https://doi.org/10.1007/s00170-004-2069-8.

Kingma, Diederik P., and Jimmy Ba. 2014. *Adam: A Method for Stochastic Optimization.* Accessed April 7, 2024. http://arxiv.org/pdf/1412.6980v9.

Kwon, Ick-Hyun, Chang Ouk Kim, Jin Jun, and Jung Hoon Lee. 2008. "Case-based myopic reinforcement learning for satisfying target service level in supply chain." *Expert Systems with Applications* 35 (1-2): 389–397. https://doi.org/10.1016/j.eswa.2007.07.002.

Lee, Hau L., Venkata Padmanabhan, and Seungjin Whang. 1997. "The bullwhip effect in supply chains." *Sloan Management Review,* no. 38, 93–102. Accessed April 7, 2024. https://sloanreview.mit.edu/article/the-bullwhip-effect-in-supply-chains/.

Li, Lihong, Thomas J. Walsh, and LittmanMichael L. 2006. "Towards a Unified Theory of State Abstraction for MDPs." *International Symposium on Artificial Intelligence and Mathematics (ISAIM)* 4:5–15.

McFarlane, Duncan, Sanjay Sarma, Jin Lung Chirn, C.Y Wong, and Kevin Ashton. 2003. "Auto ID systems and intelligent manufacturing control." *Engineering Applications of Artificial Intelligence* 16 (4): 365–376. https://doi.org/10.1016/S0952-1976(03)00077-0.

Meisheri, Hardik, Vinita Baniwal, Nazneen N. Sultana, Balaraman Ravindran, and Harshad Khadilkar. 2019. *Reinforcement Learning for Multi-Objective Optimization of Online Decisions in High-Dimensional Systems.* arXiv preprint arXiv:1910.00211. Accessed April 7, 2024. http://arxiv.org/pdf/1910.00211v1.

Mortazavi, Ahmad, Alireza Arshadi Khamseh, and Parham Azimi. 2015. "Designing of an intelligent self-adaptive model for supply chain ordering management system." *Engineering Applications of Artificial Intelligence* 37:207–220. https://doi.org/10.1016/j.engappai.2014.09.004.

OpenAI. 2021. *OpenAI - About.* Accessed February 25, 2021. https://openai.com/about/.

Prestwich, S. D., S. A. Tarim, R. Rossi, and B. Hnich. 2012. "A neuroevolutionary approach to stochastic inventory control in multi-echelon systems." *International Journal of Production Research* 50 (8): 2150–2160. https://doi.org/10.1080/00207543.2011.574503.

Pulic, Armin, ed. 2005-2015. *Lagerkennzahlen.de: Kennzahlen für Lagerwirtschaft und Bestandsmanagement.* Accessed September 10, 2021. http://www.lagerkennzahlen.de/.

Rebala, Gopinath, Ajay Ravi, and Sanjay Churiwala. 2019. *An Introduction to Machine Learning.* Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-15729-6.

Schuh, Günther. 2007. *Produktionsplanung und -steuerung: Grundlagen, Gestaltung Und Konzepte.* VDI-Buch. Dordrecht: Springer. http://gbv.eblib.com/patron/FullRecord.aspx?p=416991.

Schulman, John, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. *High-Dimensional Continuous Control Using Generalized Advantage Estimation.* Accessed April 7, 2024. http://arxiv.org/pdf/1506.02438v6.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. *Proximal Policy Optimization Algorithms*. Accessed April 7, 2024. http://arxiv.org/pdf/1707.06347v2.

Silver, Edward A., David F. Pyke, and Douglas J. Thomas. 2017. *Inventory and production management in supply chains*. Fourth edition. Boca Raton, London, and New York: CRC Press Taylor & Francis Group. http://lib.myilibrary.com/detail.asp?ID=979632.

Stadtler, Hartmut, and Christoph Kilger. 2008. *Supply Chain Management and Advanced Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74512-9.

Stricker, N., and G. Lanza. 2014. "The Concept of Robustness in Production Systems and its Correlation to Disturbances." *Procedia CIRP* 19 (4): 87–92. https://doi.org/10.1016/j.procir.2014.04.078.

Sui, Zheng, Abhijit Gosavi, and Li Lin. 2010. "A Reinforcement Learning Approach for Inventory Replenishment in Vendor-Managed Inventory Systems With Consignment Inventory." *Engineering Management Journal* 22 (4): 44–53. https://doi.org/10.1080/10429247.2010.11431878.

Sun, Ruoying, and Gang Zhao. 2012. "Analyses about efficiency of reinforcement learning to supply chain ordering management." In *IEEE 10th International Conference on Industrial Informatics,* 124–127. IEEE. https://doi.org/10.1109/INDIN.2012.6301163.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement learning: An introduction*. Second edition. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press.

Taylor, Matthew E., and Karl Tuyls. 2010. *Adaptive and learning agents: Second workshop, ALA 2009, held as part of the AAMAS 2009 conference in Budapest, Hungary, May 12, 2009 ; revised selected papers.* Vol. 5924. Lecture notes in computer science Lecture notes in artificial intelligence. Berlin: Springer. https://doi.org/10.1007/978-3-642-11814-2. http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10369925.

Tempelmeier, Horst. 2000. "Inventory service-levels in the customer supply chain: Lagerbezogene Servicegrade in Logistikketten." *OR-Spektrum : quantitative approaches in management* 22 (3): 361–380.

———. 2015. *Bestandsmanagement in Supply Chains.* 5., erweiterte und verbesserte Auflage. Norderstedt: BoD - Books on Demand.

Verein Deutscher Ingenieure. 2014. *Simulation of systems in materials handling, logistics and production,* 3633.

Verwijmeren, Martin, Piet van der Vlist, and Karel van Donselaar. 1996. "Networked inventory management information systems: materializing supply chain management." *International Journal of Physical Distribution & Logistics Management* 26 (6): 16–31. https://doi.org/10.1108/09600039610125170.

Waldmann, Karl-Heinz, and Ulrike M. Stocker. 2013. *Stochastische Modelle: Eine anwendungsorientierte Einführung.* 2., überarb. und erw. Aufl. EMIL@A-stat Medienreihe zur angewandten Statistik. Berlin: Springer.

Xu, Junqin, Jihui Zhang, and Yushuang Liu. 2009. "An adaptive inventory control for a supply chain." In *2009 Chinese Control and Decision,* 5714–5719. https://doi.org/10.1109/CCDC.2009.5195218.

Yang, Songpo, and Jihui Zhang. 2015. "Adaptive inventory control and bullwhip effect analysis for supply chains with non-stationary demand." In *The 27th Chinese Control,* 3903–3908. https://doi.org/10.1109/CCDC.2015.7162605.

Zarandi, Mohammad Hossein Fazel, Seyed Vahid Moosavi, and Marzieh Zarinbal. 2013. “A fuzzy reinforcement learning algorithm for inventory control in supply chains.” *The International Journal of Advanced Manufacturing Technology* 65 (1-4): 557–569. https://doi.org/10.1007/s00170-012-4195-z.

Zhang, Baohe, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. 2021. “On the Importance of Hyperparameter Optimization for Model-based Reinforcement Learning.” *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics,* no. 130, 4015–4023. Accessed April 7, 2024. http://proceedings.mlr.press/v130/zhang21n.html.

Zijm, Henk, Matthias Klumpp, Alberto Regattieri, and Sunderesh Heragu, eds. 2019. *Operations, Logistics and Supply Chain Management.* Lecture Notes in Logistics. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-92447-2.

Zipkin, Paul Herbert. 2000. *Foundations of inventory management.* Boston: McGraw-Hill.

List of Figures

1.1 *GoogleTrends* Report of Term Popularity for the Terms *Inventory Management* and *Deep Reinforcement Learning* Over the Last Years 2
1.2 Structure of the Thesis 4

2.1 Schematic Supply Chain Structures, Based on Arnold et al. (2008, p.161, 934 ff.) 6
2.2 Schematic Representation of RL, Based on Sutton and Barto (2018) 13
2.3 Methods of RL, Based on Sutton and Barto (2018) 14

4.1 Schematic Linear SC Model 36
4.2 Time-Discrete Simulation of a SC 37
4.3 PDF of Gamma Distribution with $E(X) = 30.0$ and Varying CVs 39

5.1 Value and Definition Range of the Sigmoid and Tanh Functions 57
5.2 Relative Standard Errors of the Mean of the KPIs Mean Cost and Service Level Over All Data Sets Used in the Following . . 60
5.3 Comparison of the $RL_{Baseline}$ and $RL_{Adaptive}$ Models in Terms of Mean Cost, Service Level, and Mean Order Size when Demand Parameters are Varied 62
5.4 Comparison of the $RL_{Baseline}$ and $RL_{Adaptive}$ Models in Terms of Stock and Service Levels when Replenishment Time is Varied in its Mean and CV 64
5.5 Adaptations of $RL_{Adaptive}$ to Changing Cost Parameters 65
5.6 Adaptations of $RL_{Adaptive+Cost}$ to Changing Cost Parameters . . 68

5.7 Comparison of the $RL_{Adaptive}$, $RL_{Adaptive+Cost}$, and (s,q) Models when Demand and Replenishment Parameters are Varied . 70
5.8 Comparison of the $RL_{Adaptive}$, $RL_{Adaptive+Cost}$, and (s,q) Models when Cost Parameters are Varied 71
5.9 Comparison of Mean Cost per Demanded Unit for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Demand Parameters are Varied . . 72
5.10 Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Demand Parameters are Varied 73
5.11 Comparison of Mean Stock Level of the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Replenishment Time Parameters are Varied 74
5.12 Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Replenishment Time Parameters are Varied 75
5.13 Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Replenishment Time Parameters are Varied . . 75
5.14 Comparison of Mean Stock Level for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied 77
5.15 Comparison of Order Frequency for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied 78
5.16 Comparison of Service Level for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied 79
5.17 Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied 80
5.18 Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when CVs of Demand and Replenishment Time are Varied 82
5.19 Comparison in Service Level for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when CVs of Demand and Replenishment Time are Varied 83

6.1 Relative Standard Errors of Mean of the KPIs Mean Cost and Service Level Over All Data Sets Used in the Following 87
6.2 Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models in Terms of Mean Cost and Service Level 88

6.3 Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Mean Demand and the CV of Demand are Varied 89
6.4 Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Mean Replenishment Time and the CV of Replenishment Time are Varied 91
6.5 Comparison of the $RL_{Decentral}$ and $RL_{Central}$ Models when Cost Parameters are Varied 93
6.6 BWE Measurement with Equation 6.1 per Node and Model . . 95
6.7 Change in Mean Cost and Service Level per Node and Model when Confronted by a Demand Shock 97

7.1 Procedure for Selecting Material-Location Combinations . . . 102
7.2 Distribution of Input Data 104
7.3 Histogram of the Yielded Inventory Reduction when the RL Policy is Applied to the Data 105
7.4 Histogram of KPI in Theoretical Experiments and when Applied to Real-World Examples 106

List of Tables

3.1 Comparison of Literature Concerning Supply Chain Modeling 22
3.2 Comparison of Literature Concerning RL in Inventory Control 27
3.3 Literature Comparison Data 30

4.1 Hyperparameters to Optimize Using *Optuna* and the Corresponding Value Ranges 44
4.2 General SC Settings for the Experiments 48
4.3 SC Parameter Value Ranges for Creating the Original Data Set and Further Parameter Variations 50

5.1 Comparison of Mean Cost for the $RL_{Adaptive+Cost}$ and (s,q) Approaches when Cost Parameters are Varied 81

6.1 Comparison of Statistical Measures of the Two Models 88
6.2 Comparison of BWE Measures of the Two Models when Exposed to a Demand Shock 96
6.3 Comparison of Mean Cost Measures of the Two Models when Exposed to a Demand Shock 96

Publication List

Christian Bauer et al. 2021. *Abschlussbericht des Forschungsprojekts "Broker für Dynamische Produktionsnetzwerke"*.
https://doi.org/10.5445/IR/1000141238

Lena Bergmann et al. 2021. “Deriving Machining Processes from Technical Drawings: An Approach Motivated by Cloud Manufacturing Platforms Using Convolutional Neural Networks.” *Industrie 4.0 Management,* no. 37 (5), 21–25