Model-Driven Objective Functions in MPC using Economic Engineering Systems Theory

With an Application to Supply-Chain Scheduling at Shell

A.J.J. Meegdes

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Model-Driven Objective Functions in MPC using Economic Engineering Systems Theory

With an Application to Supply-Chain Scheduling at Shell

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

A.J.J. Meegdes

January 18, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Royal Dutch Shell PLC. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Model-Driven Objective Functions in MPC using Economic Engineering Systems Theory

by

A.J.J. Meegdes

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: January 18, 2021

Supervisor(s):

dr.ir. M.B.Mendel

Reader(s):

prof. dr. ir. B.H.K. De Schutter

dr. A. Jamshidnejad

ir. C. Hutters

ir. B.J.G. Overdevest

Abstract

This thesis introduces a theory for model-driven objective functions in Model Predictive Control (MPC) algorithms. For scheduling supply chains, such model-driven objective functions allow the MPC algorithm to make optimal scheduling decisions by anticipating future changes in product flow and transfer price dynamics. Including such dynamics introduces new insights in the decision-making process for supply chains, as current supply-chain management relies on professional expertise and modelling techniques with static product flows and transfer prices.

In this thesis, a dynamic model for product flows and transfer prices at a storage depot in the supply chain is developed with Economic Engineering Systems Theory. We develop a model-driven objective function for profit-maximization in an MPC scheduling algorithm using the Economic Engineering storage depot model. The advantage of the model-driven objective function is the ability to assess the product flow and transfer price dynamics that affect the revenues and costs for various decisions. The MPC algorithm for scheduling shipments towards storage depots includes the constraints in the supply chain and offers the potential to control processes in the supply chain in a dynamic and automated way.

This thesis applies the modelling technique and scheduling algorithm to the refined oil product supply chain of Shell for Germany, Austria and Switzerland (DACH). The algorithm automates processes that form the bridge between the yearly tactical planning and the day-to-day scheduling operations. Supply-chain companies like Shell benefit from the scheduling algorithm by optimal decision-making, additional time for strategic activities and less room for human error.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank

dr. ir. Max Mendel for his academic supervision and the enthusiasm and drive he conveys to the students.

ir. Bart Overdevest for giving me the opportunity to do an internship at Shell and his industry supervision.

ir. Coen Hutters and my peers in the Economic Engineering group for their feedback and contributions during our meetings and discussions.

Shefin Punamadathu for sharing his supply-chain knowledge at Shell and guidance.

Delft, University of Technology | A.J.J. Meegdes
January 18, 2021

Chapter 1

Introduction

1-1 Control Engineer's Perspective on Supply-Chain Scheduling

Schedulers in the supply chain experience conditions that are changing constantly. The resulting product flow and transfer price dynamics affect the revenues and costs for supply-chain companies. They lose value due to supply-chain problems ranging from inefficient transportation of finished goods, to excessive inventories and shortages [18]. By modelling the product flow and transfer price dynamics at a storage depot in the supply chain, it becomes possible to anticipate the changing conditions during scheduling operations to avoid such problems.

Closer cooperation between control engineers and supply-chain experts introduces realism to the modelling and decision-making for supply chains [21]. Modelling storage depots in the supply chains as dynamic systems provides insight into potential revenues and costs for various scheduling decisions. Using control techniques allows us to make optimal scheduling decisions with the objective to maximize profits.

In this thesis, we develop a scheduling algorithm for shipments towards and from storage depots in the supply chain in a dynamic and model-driven way. This control engineering approach for supply-chain scheduling deals with the continuously changing conditions and disruptions, and puts automation within reach. We visualize this scheduling process with a semi-automated setting in Figure 1-1. In the figure, the scheduling algorithm is part of the Model Predictive Control (MPC) block, which replaces spreadsheets and professional expertise in current supply-chain scheduling.

We develop the MPC block consisting of the model for a storage depot in the supply chain and the scheduling algorithm with:

1. Modelling the dynamics of product flows and transfer prices in the supply chain using Economic Engineering Systems Theory (Chapter 3).
2. MPC with a model-driven objective function based on the Economic Engineering modelling technique (Chapter 4).

To the best of the authors' knowledge, no study has been conducted on model-driven objective functions in MPC with Economic Engineering Systems Theory and the application to supply-chain scheduling. Section 1-2 and Section 1-3 discuss the Economig Engineering (EconE) modelling technique and model-driven objective functions in MPC for the scheduling algorithm with the development choices in more detail.

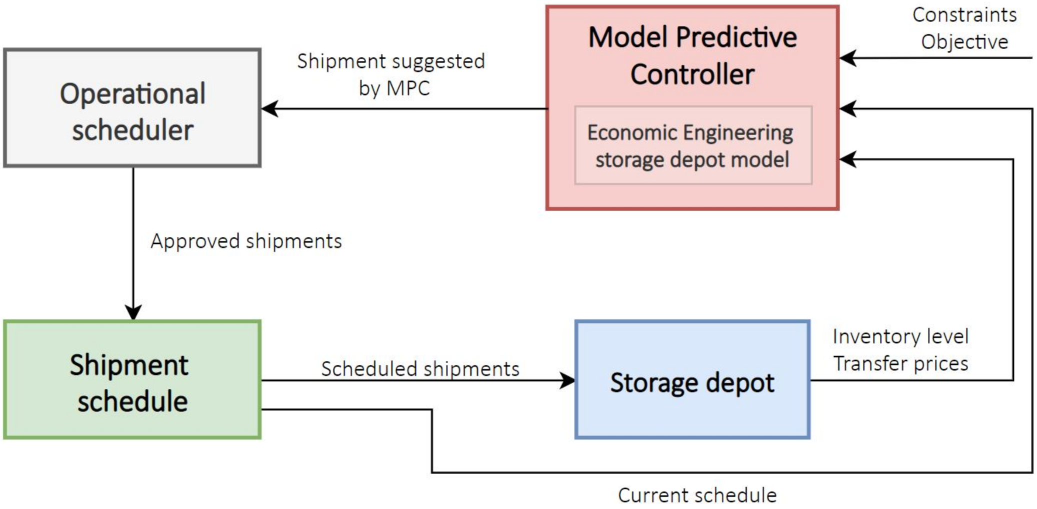


Figure 1-1: Model-driven and semi-automated decision-making process for scheduling shipments towards and from the storage depot in a supply chain. This thesis develops the MPC block that collaborates with the operational scheduler in a decentralized setting.

In Figure 1-1, the MPC algorithm and operational scheduler together have the task of scheduling shipments towards and from the storage depot in the supply chain. The shipments are collected in the shipment schedule. We develop the scheduling algorithm for one storage depot and this is called a decentralized approach. For large supply chains, it means that every storage depot has its own unique storage depot model and MPC algorithm. Ultimately, all production sites, storage depots and sales channels (as visualized in Figure 1-2) have their own unique model and MPC algorithm that collaborate to schedule all shipments in the supply chain in a completely automated way.

Since this thesis serves as proof of concept for the integration of Economic Engineering Systems Theory in MPC with the focus on scheduling shipments at one storage depot, the choice for a decentralized setting is made. Eventually for larger networks with more storage depots, centralized or distributed settings can be used for the scheduling process. In the centralized setting, one MPC algorithm schedules the shipments for multiple storage depots at once. The distributed setting schedules the shipments for one storage depot and uses information of other elements in the supply chain in the decision-making. The centralized and distributed settings are out of the scope of this thesis, but we recommend these as future research.

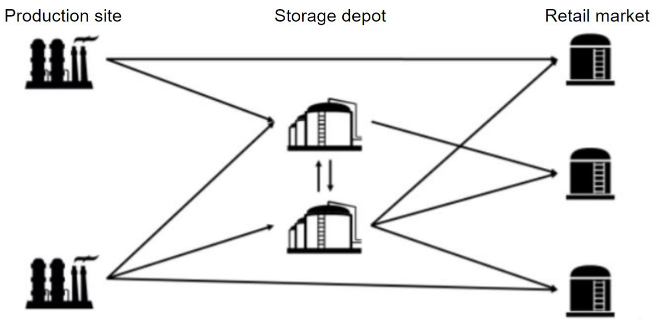


Figure 1-2: Illustrative example of locations and connections in a supply chain [55]. In this thesis, we develop a MPC algorithm for the scheduling of shipments towards storage depots.

1-2 Dynamic Supply-Chain Modelling with Economic Engineering Systems Theory

With Economic Engineering Systems Theory [33], we model the dynamic behaviour of transfer prices, product flows and resulting inventory levels at storage depots in the supply chain. Economic Engineering Systems Theory is a technique of economic modelling that uses analogies between principles in economics and physical laws. EconE models use causal and dynamic relationships to describe economic systems. The development of an EconE model for a storage depot resulted from the observations of supply-chain concepts at Shell, which are discussed in more detail in Chapter 2.

In Chapter 3, we develop the modelling technique to include product flow and transfer price dynamics in the decision-making for supply chains. The dynamical behaviour in the supply chain is caused by rapid changing conditions and disruptions, the resulting dynamics are difficult to interpret with only professional expertise. We consider the storage depot as the center in the modelling, where supply and demand come together. The modelling of supply chains in this work is based on analogies with Newtonian mechanics to describe the dynamics of product flows and transfer prices around the storage depot. For supply-chain scheduling, such product flow and price dynamics provide insights into the potential profits.

The modelling technique is applicable to supply chains in general. With the building-block approach discussed in Section 3-3, the storage depot models can be unique and as detailed as is required for the application. The modelling technique is focused on storage depots, but production sites, retail stations and other sales channels can also be modelled. The building blocks would be similar for models for refineries and retail stations. A conceptual model for retail stations is shown in Section 7-2.

For the oil industry, research by Orie [39] is a pioneering effort into the use of Economic Engineering Systems Theory for oil-economic systems on which this thesis is built.

1-3 Supply-Chain Dynamics with Model-Driven Objective Functions in MPC

We develop the scheduling algorithm for shipments at a depot with MPC as control strategy for the decision-making. MPC is an advanced technique for controlling processes while satisfying constraints that uses an internal model for predicting system behaviour over a predefined prediction horizon [9, 30]. The ability to specify a objective function on which is optimized repeatedly for finite-time horizons while accounting for future behaviour is a main advantage of MPC [31]. The features of MPC to take into account future dynamics and operational constraints makes it suitable for the control of supply chains [21]. In Section 2-3, we continue the discussion on the suitability of MPC in supply chains compared to other control strategies.

In Chapter 4, we develop the theory for model-driven objective functions in MPC with Economic Engineering Systems Theory. We use a profit-maximizing objective function with variables and parameters derived from the modelling technique for supply-chain dynamics at a storage depot. As a result, the MPC algorithm (scheduling algorithm) outputs optimal scheduling decisions anticipating future changes in product flow and price dynamics. We develop the scheduling algorithm in a semi-automated setting such that the professional expertise remains part of the decision-making. Ultimately, the goal remains to completely automate supply-chain processes such that experts can focus on strategic and innovating activities.

In Chapter 5, we apply the modelling technique and scheduling algorithm to storage depots in the refined oil product supply chain of Shell in Germany, Austria and Switzerland (DACH). We show the potential by assessing the predictive performance for supply-chain dynamics of the EconE modelling technique. Furthermore, we demonstrate the decision-making with the EconE model-driven objective function in MPC for scheduling shipments at a storage depot in the refined oil product supply chain. The developed Graphical User Interface (GUI) shows how the scheduling algorithm works in a semi-automated setting.

Finally, we discuss identified directions for future research and additional findings in Chapter 7.

Chapter 2

Supply-Chain Management and Modelling Techniques

2-1 Introduction

Current supply-chain management relies on relatively static modelling approaches and professional expertise in order to keep businesses running. The continuously changing conditions and often occurring disruptions increase the demand for a dynamic approach to operations in supply-chain management. This chapter aims to further expose these shortcomings in current supply-chain management with a focus on the oil industry.

The current state of operations for supply chains in the oil industry is discussed in more detail in Section 2-2. We narrow the discussion on operations in the oil industry down to the supply chain of refined oil products. We discuss general supply-chain management and related modelling techniques with a focus on Model Predictive Control (MPC) in Section 2-3.

2-2 Supply-Chain Processes in the Oil Industry

Oil is one of the most important materials in the world and billions of barrels of oil are consumed annually [49]. The oil industry is critical in driving the global economy and the derived products of the oil industry are essential in other vital industries [22]. The oil comes out of the ground as crude oil and refineries transform the crude oil into refined oil products ready for customer use.

2-2-1 The Oil Industry Value Chain

In all industries, activities take place to transform the input of raw materials, knowledge, labor and capital in end products. A value chain visualizes these activities and helps to identify the activities that create value throughout the chain. These created values constitute to the prices for products throughout the chain, which are transfer prices. Transfer prices are the prices that selling department of a company charge for product to the buying department of possibly the same company. Transfer pricing is heavily used in the oil industry, refineries pay for crude oil to producers, depots to suppliers and so on.

A simplified value chain for the oil industry is shown in Figure 2-1. The upstream and downstream are the main segments of the value chain and Integrated Oil Companies (IOCs) like Shell perform all of these activities.

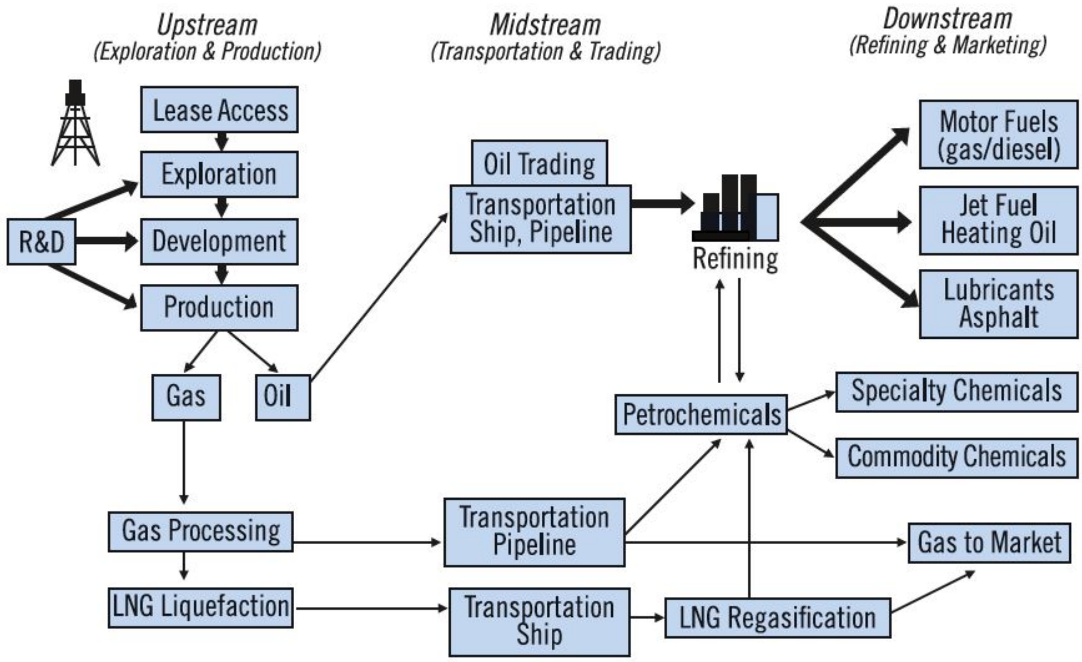


Figure 2-1: Overview of activities in the oil industry value chain [20].

In Figure 2-1, the midstream segment represents the transportation and trading in the end-to-end value chain. In the oil industry, the midstream segment is usually considered to be a part of the upstream or downstream operations. Figure 2-2 depicts the structure of transportation of oil in the value chain. Since there is a difference between the transportation in the upstream and in the downstream segment, we refer to crude oil transportation as part of the upstream segment and to transportation of refined oil products as part of the downstream segment. The upstream transportation has the task to move crude oil from wells to the refinery and this is mainly done by shipping and pipeline. After refining, four transport options move the refined oil products to the customer side. These four options have all unique advantages and drawbacks.

Pipelines are a convenient way of moving oil. Crude oil pipelines are usually larger than product pipelines. Despite the fact that a pipeline infrastructure is expensive to build, it is an inexpensive way of transporting the product. Ship transport used for moving oil is also a cheap option. Tank barges can carry huge amounts of product and they can be used over water. Rail transport is also used in many regions, but good rail infrastructure is necessary. Although more rail tanks cars in trains are needed to transport huge amounts of product, it is a cost-effective method [20]. Finally, trucks are mainly used for secondary transportation, which means the transportation of products from storage depots to end-users. Tanker trucks can carry smaller amounts and are more expensive than the other transportation modes. The transportation of oil is challenging and it requires accurate coordination across the entire supply chain.

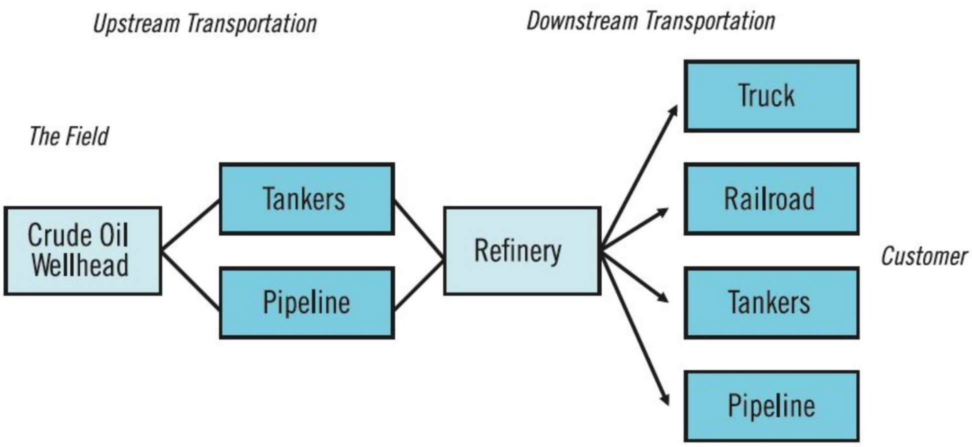


Figure 2-2: Modes of transport in the up- and downstream segment of the oil industry [20].

The Upstream Segment

The upstream segment of the oil industry is all about wells and collecting the crude oil. These activities include locating wells, drilling them and all other operations needed. The supply chain of crude oil towards refineries is part of the upstream segment and is challenging with a lot of companies involved in the process [1]. Orie [39] developed an model with Economic Engineering Systems Theory for the crude oil supply chain to forecast the crude oil prices.

Oil is a fossil fuel, it owes this name since it was formed from remains of living millions of years ago. The pressure and heat turned the organic material into simpler substances, nowadays known as hydrocarbons, which are the components that are refined from crude oil in the refinery. Crude oil itself is actually a useless product and its value is simply the value of its derived products. The proportions of derived products is dependent on the quality of oil. Hence, the value of the crude oil in the upstream segment is roughly determined by two factors: the costs followed from the researching and production, and the oil quality.

Crude oils are produced at different places around the world and the crude oil is unique in all these places. Therefore, each specific crude oil is usually named by the name of the country, the region or the field. The uniqueness is achieved through the different make-ups of particular crude oils. Crude oil is a complex mixture of numerous different hydrocarbons with all their own properties, in addition crude oils also contain contaminants. The value of crude oil is determined by its quality. The quality of crude oil is mainly determined by its density and sulfur content [45].

The Refining Process

Crude oil itself cannot be used in its raw form and the value of it is based on the products that can be refined from the crude oil. Refining is the process how oil is turned into petroleum products that can be used as end product or as intermediate. In all refineries this process takes places, but nowhere does this happen in exactly the same way. Refineries differ in design, processing sequences, capacities and many other ways.

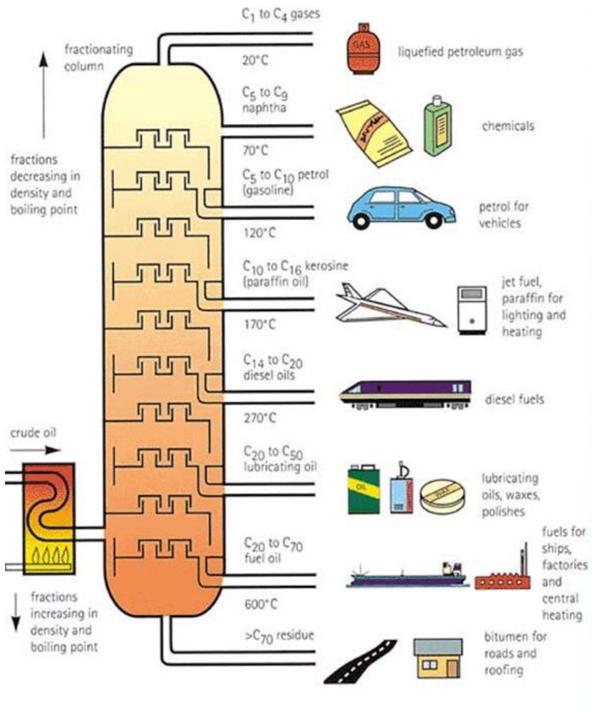


Figure 2-3: Conversion of crude oil to different refined oil products with distillation [46].

Every refinery is designed for specific (range of) grade(s) of crude oil and each grade gives different yields. Refining yields are the different proportions of products refined from the crude oil and they are mainly dependent on the quality of the crude oil. Figure 2-3 gives an overview of the refined oil products that result from the distillation of crude oil. There are four main groups for the refinery output: gases, gasolines, middle distillates and residuals [17]. Gasoline is the lightest liquid refined product group in the refinery. The products in this group have high value and the most familiar product is motor gasoline. Middle distillates are the refined products in the range between the lighter (gases and gasolines) and heavier products. The middle distillate group include amongst others diesel fuel, jet fuel and light heating oil.

All these main product groups have different grades with different specifications. All the different products have their own usage and from the refinery they must all reach their destination, which is done by the refined oil product supply chain.

The Downstream Segment

The processing, transporting and selling of refined products made from crude oil are the downstream operations of the oil industry. Oil refining, supply and trading, product marketing, wholesale and retail are all activities that take place in the downstream segment. The oil industry provides thousands of products to end-user customers around the world. Familiar products are gasoline, diesel, jet fuel, heating oil and asphalt, and also less familiar products (petrochemicals) like lubricants, synthetic rubber, plastic.

All the different businesses in the downstream segment have their own tasks and interests. However, they have to collaborate well in order to be successful. Section 2-2-2 elaborates on the convergence of all the downstream operations in the refined oil products supply chain.

2-2-2 The Refined Oil Products Supply Chain

The starting point of the product supply chain is at the refinery when the crude oil is processed and the refined products are ready for distribution to the end-users. Refineries convert crude oil to all the different oil products. The oil products are distributed from the refineries to the different depots that distribute them further. The derived products are stored in storage depots located along the supply line. In Figure 2-4, the storage depot is the central point where all product flows go in and out. There is transfer pricing throughout the entire product supply chain, because each buying party pays for the product to the selling party.

Depots have different sources for their incoming batches of oil product. It is likely to have a depot that functions both for throughput to another depot and as last depot in the supply chain before the oil product is sold. Hence, depots can have refineries, other depots, hydrocarbon dealmaking and imports as their source of incoming product as shown in Figure 2-4. The source hydrocarbon dealmaking means that refined products are entering the supply chain through smart deals with other companies. For example, a company has low stocks of

Automotive Gasoil (AGO) in the south of Germany and high stocks of AGO in Rotterdam, another oil company with high stocks in the south of Germany needs an amount of AGO in Rotterdam, in such a situation it is lucrative for the two companies to agree on an exchange since they both eliminate their transportation costs.

From the depots, there are also multiple destinations for the product. These destinations all fall under different channels. Two important sales channels are retail and commercial fuels. Retail sites are supplied with oil products from the depots, these retail stations can be company-owned or third parties that buy and pick up product. The commercial fuels channels are for example airports or trucking companies that need large amount of product with which is agreed in term deals for a certain number of years. Spot deals are commonly occurring at the commercial fuels channel as well.

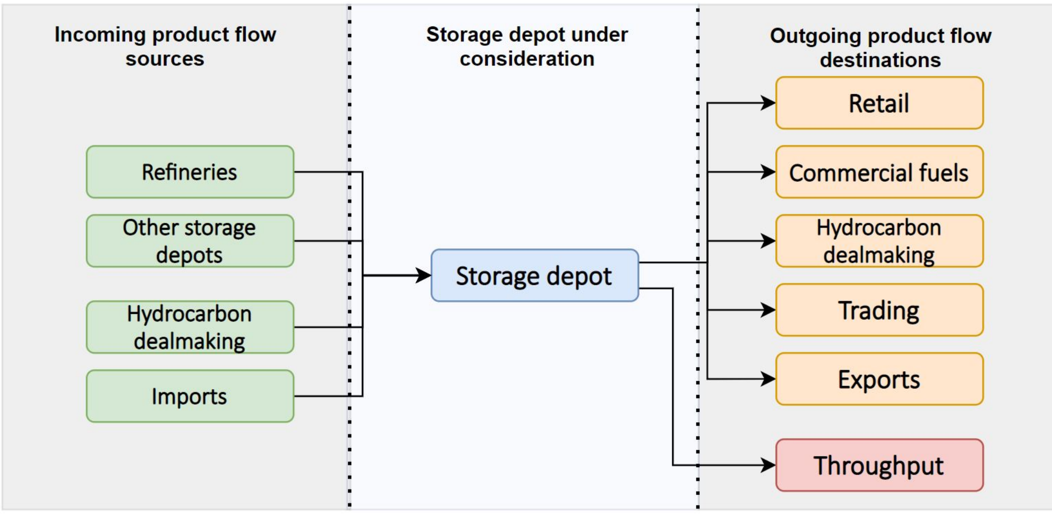


Figure 2-4: Supply and demand channels connected to a storage depot in the refined oil product supply chain.

Figure 2-4 indirectly visualizes the role of a depot in balancing the supply and demand of the refined products in the supply chain. As the middle man where flows of product go in and out, the inventory management for depots is crucial in meeting customers demand. The task of balancing the supply and demand throughout the supply chain while creating maximum value for the company is known as channel optimization. This task is an integrated process where all the parties with dependencies in the downstream value chain are collaborating. However, the balancing becomes complicated when one of the parties on the supply or demand side experiences critical changes due to for example disruptions on the supply chain.

The products reach the destinations by transportation. The different destinations for the products also require different ways of transportation with different costs, sizes and duration. The end-users come in various forms as well. Scheduling the supply chain is a complex process in which all possibilities must be carefully considered in order to efficiently get the product to the end-users. There are numerous effects like downtime of refineries or environmental issues that have major impact on the supply chain. All parties involved have to deal with these effects. As a result, making the right decision in those circumstances is challenging. The focus of this research is on the supply chain of refined oil products along the Rhine river, which is further elaborated on in Section 2-2-3.

2-2-3 The Rhine River Supply Chain

Oil products in the area along the Rhine river are mainly supplied by refineries in the Rotterdam area and the German Ruhrgebiet. The transportation of the oil products coming from the refineries to major storage depots along the Rhine river is mainly done by shipping and pipeline. The oil products are stored in the storage depots until they are picked up at the depot for sales channels or a depot-to-depot transfer.

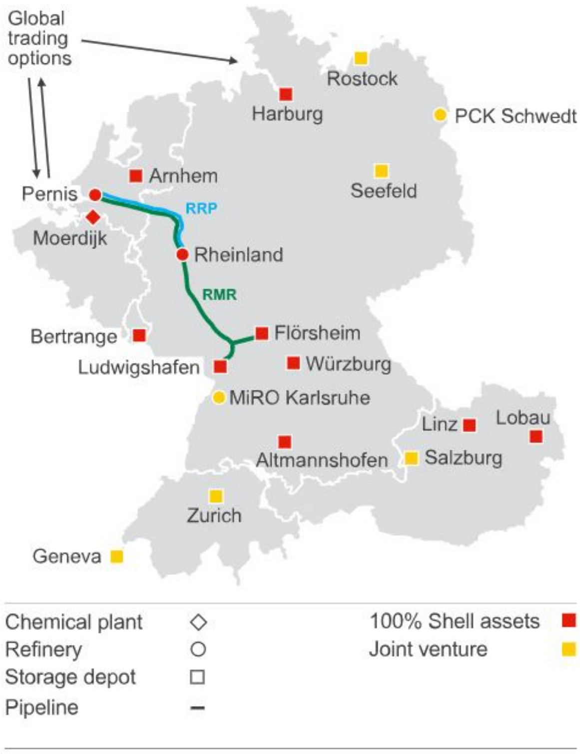


Figure 2-5: The Rhine envelope of Shell [42].

The Rhine envelope of Shell covers the largest market for oil products in Europe and many of the other important clusters of the petrochemicals industry [42]. The location of the supply chain along the Rhine river makes it an important artery in the large consumer markets of Western and Central Europe.

In the Rhine river envelope there are over 50 depot locations that provide customers with product. There are four modes of transportation to supply the oil products in the downstream segment. However, every depot has its own characteristics. Some depots are only accessible by truck or train as not all depots in the supply chain are located directly along the Rhine river. Other depots might have high flexibility in modes of transport and are accessible by pipeline and ship as well. Figure 2-5 visualizes important elements of the supply chain in the Germany, Austria and Switzerland (DACH) region.

Parts of the Rhine river are unnavigable when the water levels are low. The reason for the low water levels are dry summers with a lack of rainfall. Cargo vessels cannot be fully loaded on segments of the Rhine and hence the transportation is highly affected and becomes way more expensive in the supply chain. The shipping in some segments of the Rhine river might be even eliminated. It is complicated to predict the behavior of the supply chain when parts fall away. Figure 2-6 shows the evolution of the Rhine water levels together with the barge freight rates. Stakeholders have to take more strategic measures to prepare themselves structurally for the consequences of climate change on Rhine transportation [52].

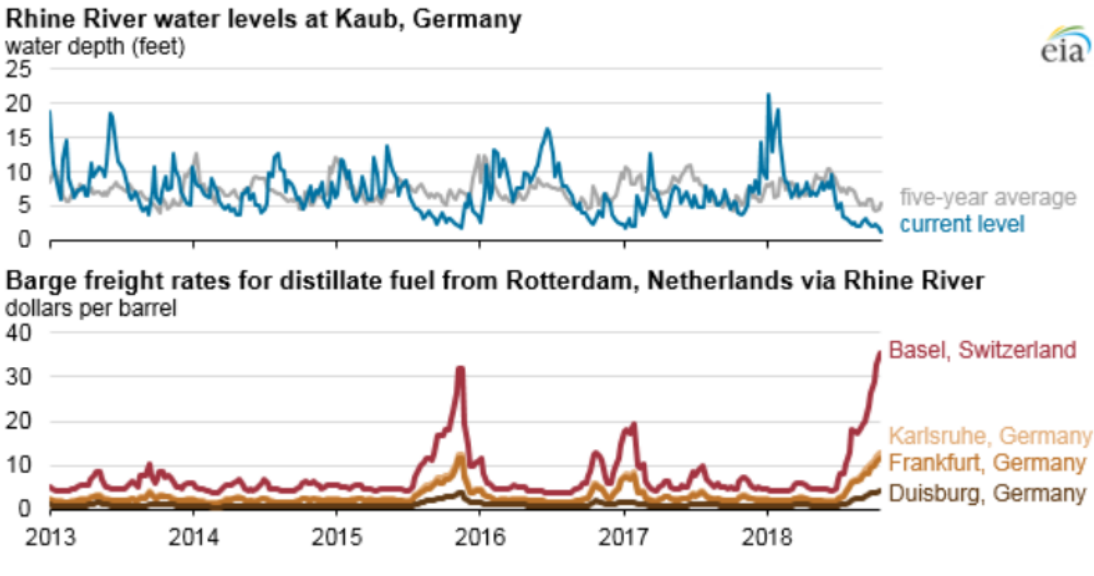


Figure 2-6: Evolution of Rhine water levels at Kaub and its effect on barge freight rates [48].

2-2-4 Transfer Pricing in the Supply Chain

Supply chains are increasingly subjected to transfer pricing [14], as is the refined oil product supply chain. In a complex and comprehensive supply chain as the Rhine envelope of Shell, transfer pricing is used for the prices at which different parties in the supply chain transact with each other. The product that flows through the supply chain has a value that increases with costs made by activities. Transportation, storage and other costs are all adding value to the product. The transfer prices are based on the total value in the product up to that point and the margin the seller of the product adds.

Considering Figure 2-5, Flörsheim pays a transfer price for product to the refinery in Pernis.

Then, the product is transported to Flörsheim from Pernis and stored. The Würzburg depot picks up product at Flörsheim and pays then a (higher) transfer price to Flörsheim. Retail stations that are supplied from Würzburg pay again another transfer price. In that way, all entities (refineries, depots, retail stations) run their own intercompany businesses in which they want to make as much profits as possible.

In reality, the prices and costs vary over time where in practice more static pricing is used to account for the transfer pricing. Improving the modelling by making these prices and costs in the supply chain dynamically evolving over time could result in large competitive advantages.

2-3 Current Supply-Chain Management and Modelling

Supply-chain management

Supply-chain modelling is usually focused on the planning of the supply chain. The models complement in the decision-making for the planning and this is categorized for different cycles of planning operations. Figure 2-7 gives a general overview of the planning operations for different cycles and related matters.

The first one is the business planning with a yearly cycle. The purpose of the business planning is determining the long term strategy and what the budgets and targets are. The business planning is high level and has to do with the structure of the supply chain or long-term plans of the company for certain goals.

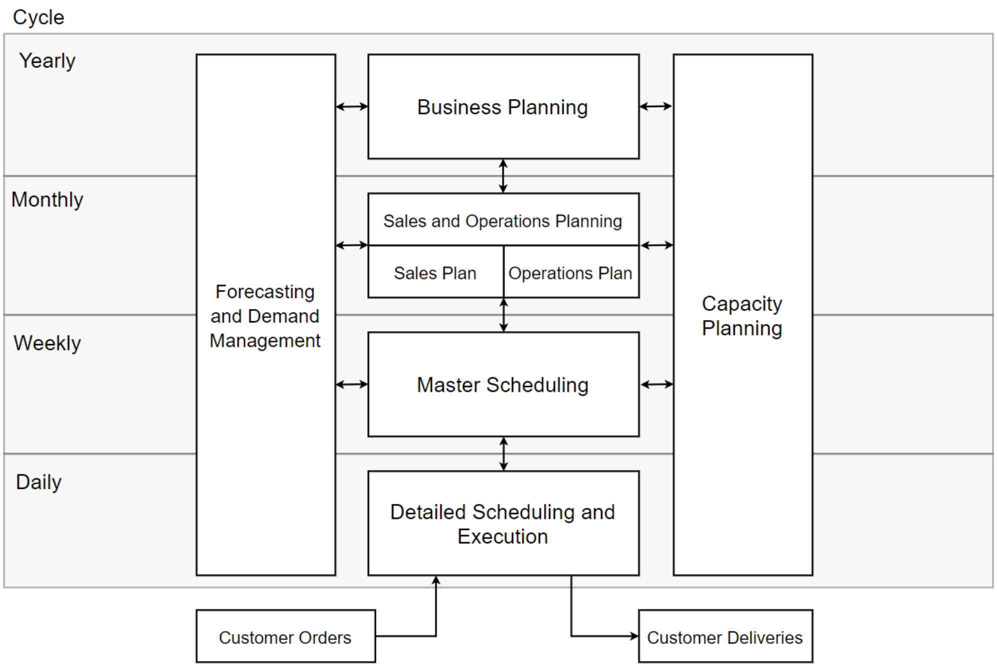


Figure 2-7: General supply-chain management activities in a company based on Wallace [54].

Secondly, sales and operations planning is for the mid-term and mediates between the long-term and short-term decision-making. The goal of sales and operations planning is more tactical to efficiently create value within short-term operations to support the strategic direction of the company [37]. For the refined oil product supply chain, the allocations of refined products to storage depots or the transportation modes are part of the tactical aspect.

The master scheduling is necessary for more short-term decisions. It includes the more detailed and specific tasks like resource allocations or customer order taking. The goal for the master scheduling is meeting customer needs and to balance the demand and supply at all times [26]. When demand exceeds supply, the company cannot provide enough product to meet total customer need. As a results, costs increase due to delays and additional transport premiums. The service levels may suffer from these effect [27, 23]. If supply exceeds demand, inventories increase and profit margins are possibly squeezed due to price cuts and discounting. Collaboration and information-sharing between the supply and demand side can provide the competitive edge that enables more profitability [15, 43].

Furthermore, the detailed scheduling and execution follows the master scheduling by ensuring that things are set in motion and executed. In Figure 2-7, the forecasting and demand management and capacity planning represent respectively the demand and supply aspects in the supply chain. All types of planning and scheduling are integrated with each other. Tactical planning for sales and operations is the most integrated as it deals with the strategic goals and the essential operational tasks of the company.

Variability in orders and inventories is a main driver of the dynamical behavior of supply chains. Feedback, interaction and time delays are causes of variability and complex behaviour in the supply chain [19]. The decision matrix in Figure 2-8 helps to understand the challenges that variability brings to supply-chain management. Make-to-stock (MTS) is the process that supply is matched to anticipated consumer demand, so the product is stocked for future

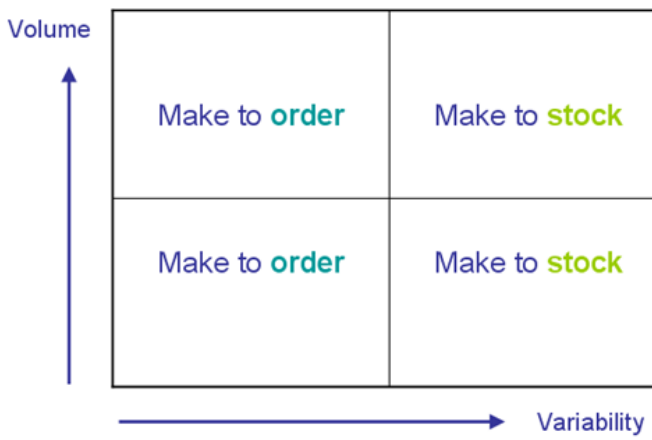


Figure 2-8: Make-to-order and make-to-stock approaches to different levels of volume and variability in orders [35].

sales that have not taken place at the time of production [5]. Make-to-order (MTO) is the process where the production is commenced after orders from consumers are coming in. The processes in the oil product supply chain are clearly more like MTS than MTO with all the depots along the supply chain where product is stored. High variability with high volume in orders require the inventories in the supply chain to have sufficient stocks.

Mathematical programming and control

Mathematical Programming (MP) techniques such as linear programming and integer programming are widely used for supply-chain activities in the oil industry. The MP techniques support the decision-making in the crude oil supply chain [44], refineries [24] and the refined oil product supply chain [37] for mostly the strategic and tactical planning.

These techniques solve large optimization problems and the outcome is broadly the tactical or strategic planning with operational guidelines. Product flows and transfer prices are usually given inputs in the optimization for single time periods with these techniques [2]. However, the prices and product flows change over time within the considered period. Dynamic modelling of the supply chain incorporates time. Complementing the current MP techniques for planning purposes with dynamic modelling and operational decision-making creates competitive advantages as more anticipation on the future dynamics becomes part of the decision-making.

As control strategy for supply-chain scheduling we use MPC [9]. MPC is illustrated schematically in Figure 2-9, it can be seen as the dynamic application of MP techniques. MPC is a control technique for processes and uses an internal model for predicting system behaviour over a predefined prediction horizon. The features of the MPC method to optimize in receding time windows where new solutions are often calculated and to include constraints and predictions in these calculations makes it different from other control methods. The MPC controller uses measurements from the system and to determine the optimal control inputs for the system.

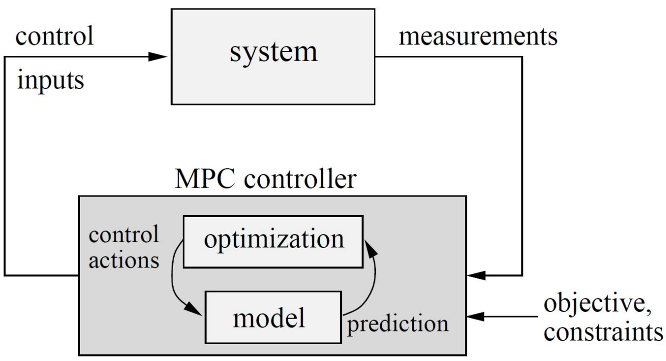


Figure 2-9: Schematic illustration of Model Predictive Control for a system [4].

In the supply-chain scheduling application, the MPC uses the model and measured inventory levels at the depots to model product flow and transfer price dynamics. Then, it determines using optimization which possible shipments towards depot gives the best predicted performance according to the objective function. MPC algorithms use objective functions as the criteria for the optimization or stabilization of the controlled system. This optimization is not done in applications for classical control, so they cannot be specified as profit-maximizing. For economic systems, the optimization refers to maximization of one or more payoffs, such as profits or utility at the end of the planned horizon [29]. This feature to specify the objective function for maximizing profits or minimizing operating costs is an important reason to choose MPC for supply-chain scheduling.

An issue for most control techniques in supply-chain management is the requirement to perform properly in the presence of constraints [21]. MPC has the ability to handle constraints directly, since they can be included in the design [3]. This constraint handling property of MPC is one of the most important contributors to its success in industrial applications [31, 41]. Both the ability to specify an objective function and the constraint-handling property are the main reasons to use an MPC algorithm for supply chain scheduling in this thesis. Section 4-3-1 discusses the constraints in the MPC for scheduling shipments at a storage depot in the supply chain.

2-4 Conclusions

Supply chains often experience rapid changing conditions and disruptions, the resulting supply-chain dynamics are difficult to interpret with only professional expertise. Most supply-chain modeling techniques use static product flows and transfer pricing assumptions, this is not a realistic representation of supply chains where product flows and prices change over time. In order to include the evolution of product flows and transfer prices over time in the modelling, we develop a modelling technique for supply-chain dynamics at a storage depot with Economic Engineering Systems Theory in Chapter 3. As a result, we can include the insights from the modelling about dynamically changing costs and revenues in the decision-making for supply-chain management.

Schedulers in the supply chain face large logistical optimization problems which are time-consuming and prone to human error and missed opportunities. The numerous choices and considerations make the task of scheduling a complicated process. The distribution of product in supply chains is a dynamic process that requires active coordination and more anticipation on what could happen to prevent undesirable behaviour in the supply chain and create more competitive advantages. Automation in combination with smart decision-making can relieve the schedulers of the problems they experience. In Chapter 4, we develop an MPC algorithm to schedule shipments towards depots in a profit-maximizing way that puts automation within reach. The tasks automated by the MPC algorithm are similar to the goals to be achieved in master scheduling and detailed scheduling and execution in Figure 2-7.

Chapter 3

Dynamic Supply-Chain Modelling with Economic Engineering Systems Theory

3-1 Introduction

Rapid change and disruptions lead to dynamic behaviour in the supply chain which is difficult to interpret with only professional expertise. In this chapter, we use Economic Engineering Systems Theory [33] to model supply-chain dynamics at storage depots. Economig Engineering (EconE) models are based on analogies with principles in engineering and use dynamical relationships to describe economic systems. The EconE models are interpretable and we use them to model the dynamics of product flows, resulting inventory levels and transfer prices at storage depots in the supply chain.

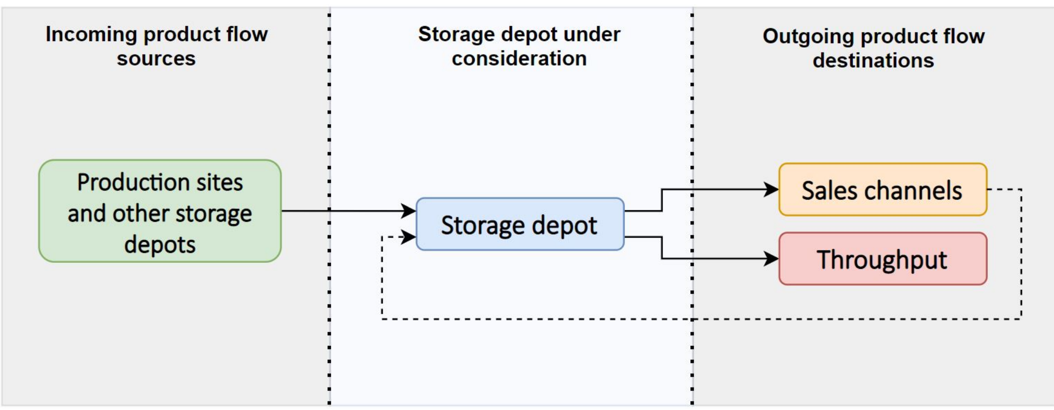


Figure 3-1: Simplified version of Figure 2-4 for the structure that we use for the modelling technique with Economic Engineering Systems Theory in this chapter.

In Section 3-2, we use Economic Engineering Systems Theory for the development of a storage depot model. Figure 3-1 shows the simplified version of Figure 2-4 that we use for the modelling technique for supply-chain dynamics around a storage depot in the supply chain. All blocks in Figure 3-1 for incoming and outgoing flows at the depot have their own unique transfer price and costs that we include in the modelling. As a result, the dynamically changing revenues and costs become part of the decision-making with model-driven objective function in the Model Predictive Control (MPC) algorithm in Chapter 4.

In Section 3-3, we describe the modelling technique for the dynamics in the supply chain with more detail on the basis of building blocks. The building blocks correspond with the blocks in Figure 3-1. They allow the modelling approach to have the unique characteristics of storage depots covered in their models. The model for a storage depot in Section 3-2 contains one of each building blocks in this chapter. When a storage depot has multiple sources for incoming product flows as in Figure 2-4, there can be more building blocks connected to the depot with unique transfer prices and costs. This adds to the modelling that unique models for different storage depots can be made in a general way.

Section 3-4 performs a qualitative analysis of the generalized storage depot model introduced in this chapter. We assess in a qualitative way that the modeling technique with Economic Engineering Systems Theory is suitable for modelling and analyzing supply-chain dynamics.

The modelling technique is discussed with bond graph theory, Karnopp [25] provides more explanation on dynamic modelling with bond graph theory.

3-2 Economic Engineering Supply-Chain Modelling Technique with Transfer Pricing

Figure 3-2 depicts the EconE model for a storage depot in a supply chain. The building blocks in the model correspond with Figure 3-1. Table 3-1 gives an overview of the interpretations and corresponding units of the elements, input- and state variables. A compliance element (C-element) stores the allocated volume for transactions or the inventory level at the storage depot. An inertia element (I-element) stores a transfer price that is paid by the buyer of product to the seller. An resistance element (R-element) does not store a variable. The elements, variables and signals on the bonds for each building block are further discussed in Section 3-3.

The product flow that comes in from production sites or other depots at the storage depot is modelled with the flow source $S_{\mathrm{f},1}$. The product flow that goes out as throughput to other depots is modelled with the flow sink $S_{\mathrm{f},2}$. Here, the shipment volumes on the shipment days are the inputs to the storage depot model. The output is the state variable q_7, which represents the inventory level at the storage depot. The sum of the flows at the 0-junction of the storage depot is equal to zero. For this model, this implies that the net flow towards the sales channels is the difference between the incoming and outgoing flows at the storage depot and the marginal change in inventory.

There are three transfer prices in the model: the source price p_4, the sales price p_9 and the throughput price p_{16}. The source price is the price that is agreed on with the product source of the depot under consideration. The sales price is the price paid by the sales channels to get product from the depot under consideration in the model. The throughput price is paid by other depots further in the supply chain that need supply from the storage depot to fulfill the demand of customers. The volumes that are agreed on with the source and throughput price are placed in the schedule to be transferred to the depots. These allocated volumes are the inputs to the system at the days the transfer take place.

Table 3-1: Interpretation and units of elements and variables in the storage depot model of Figure 3-2.

Symbol	Interpretation	Units
Elements		
C_1	Cost accrual rate	[€/ (tonne^2· day)]
I_1	Source price elasticity	[tonne^2 / (€· day)]
R_1	Primary transfer premium rate	[€/ tonne^2]
C_2	Revenue accrual rate	[€/ (tonne^2· day)]
I_2	Sales price elasticity	[tonne^2 / (€· day)]
R_2	Secondary transfer premium rate	[€/ tonne^2]
C_3	Cost accrual rate	[€/ (tonne^2· day)]
R_3	Customer marketing rate	[tonne^2 / €]
R_4	Primary transfer premium rate for throughput	[€/ tonne^2]
I_3	Throughput price elasticity	[tonne^2 / (€· day)]
C_4	Cost accrual rate	[€/ (tonne^2· day)]
Input variables		
u_1	Volumes in schedule allocated to storage depot	[tonne]
u_2	Volumes in schedule allocated to other storage depot	[tonne]
State variables		
q_2	Allocated volume at source for storage depot	[tonne]
q_7	Inventory level at storage depot	[tonne]
q_{12}	Allocated volume at storage depot for sales channels	[tonne]
q_{18}	Allocated volume at storage depot for throughput	[tonne]
p_4	Transfer price paid by storage depot to source	[€/tonne]
p_9	Transfer price paid by sales channels to storage depot	[€/tonne]
p_{16}	Transfer price paid by throughput depots to storage depot	[€/tonne]

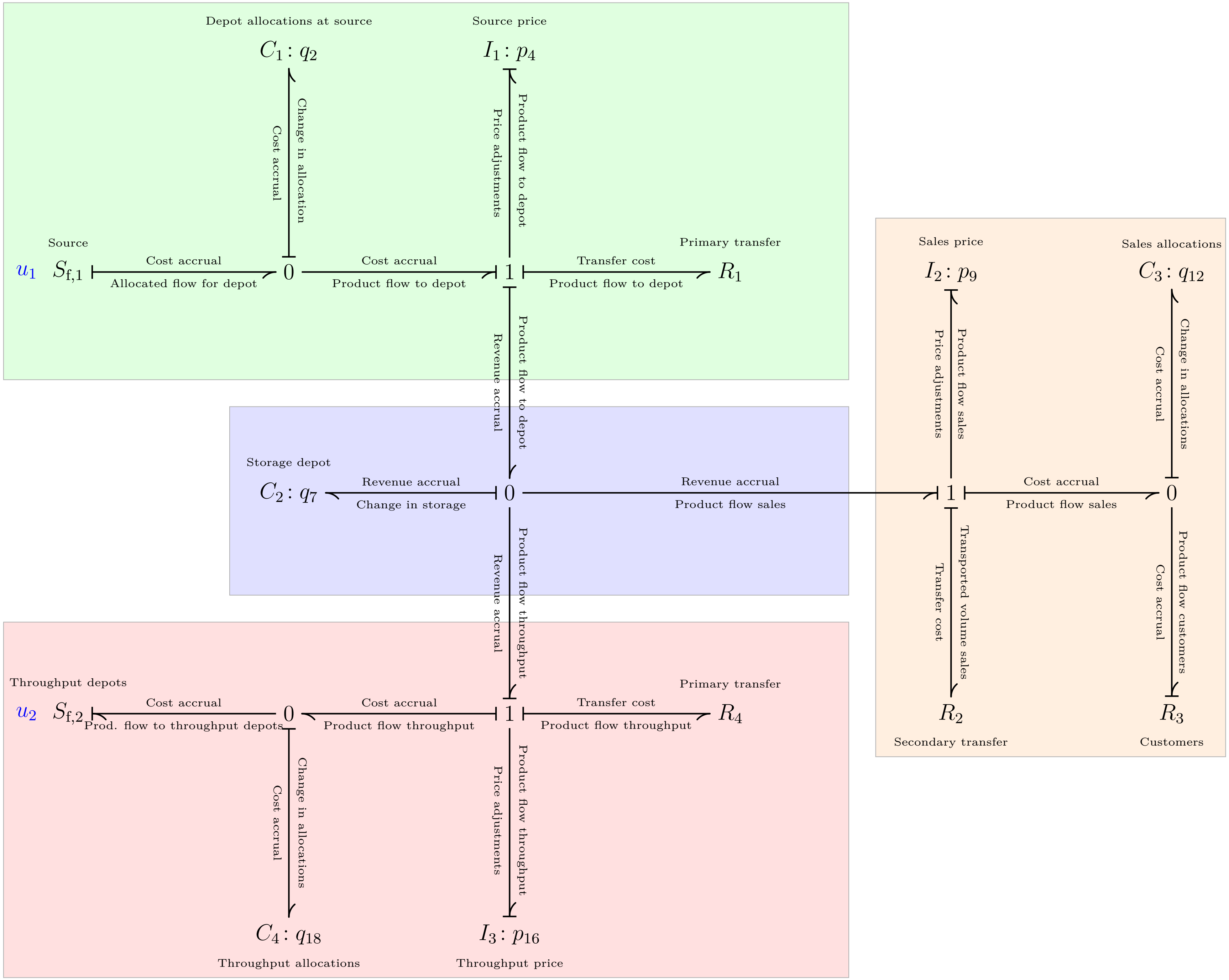


Figure 3-2: Economic Engineering model for a storage depot in the supply chain.

3-3 Building Blocks for the Supply-Chain Modelling Technique

The EconE model for a storage depot in the supply chain shown in the previous section consists of four different building blocks. The central point in the model is the building block for the storage depot itself where the inventory level is stored. From this building block, the other building blocks can be connected. In this section, we elaborate on each of the buildings blocks.

Building block for the storage depot under consideration

Figure 3-3 shows the building block the storage depot. Supply and demand come together at the storage depots. Depots are modelled with a C-element. In mechanics, a C-element is a spring and functions as a storage of distance which remembers its neutral position. For supply chains, this distance represents the inventory level at the depot. Figure 3-4 depicts a typical evolution of inventory levels over time in a storage depot, this is similar to what you expect by a spring that returns to its neutral position.

Elasticity is the property of a spring to return to its initial position after forces that altered the length have been removed. Analogously, the C-element is used to model the inventory which functions as the storage of product with a certain rigidity as property of the inventory returning to its neutral position in stock.

In engineering, C-elements store potential energy in terms of the state variable q. In this building block, the variable q represents the amount of product at the storage depot and the C-element stores the potential cash flow that might result from selling the stored amount of product. The amount of product stored at the depot changes over time with a change dq for every time step.

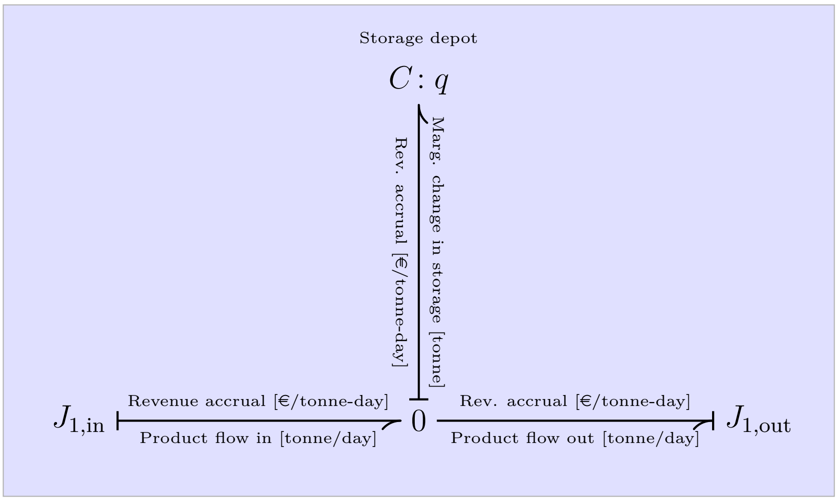


Figure 3-3: Building block for the storage depot.

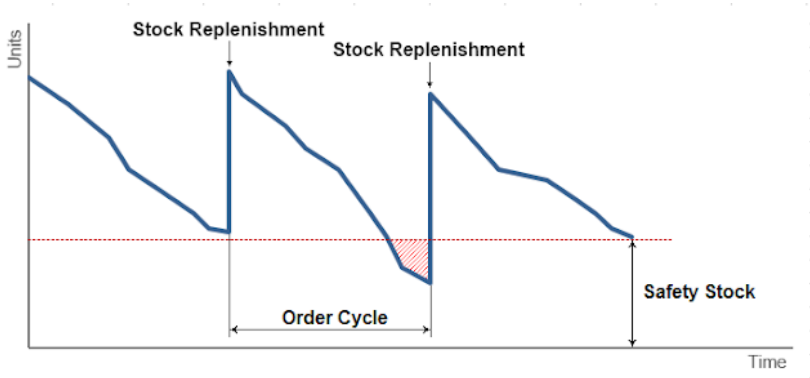


Figure 3-4: Typical evolution of inventory levels over time [34].

The C-element is connected to the 0-junction, which has an equal flow of product in and out. The amount of product towards the C-element is determined by the difference of in and outgoing flow of product at the junction as captured in Equation (3-1). Therefore, the causal stroke at the bond towards the C-element is at the junction side.

$$\text{product flow in} = \text{product flow out} + \text{marginal change in stock level} \tag{3-1}$$

The effort on the bonds in the storage depot building block is the revenue accrual, which can be seen as the force that is working on the inventory towards the neutral position. The potential revenues are stored in the storage depot. The transactions to receive the revenues have not taken place yet. The I-element in the building blocks for the sales channels and throughput realize these transactions. The revenue accrual is the same on all bonds connected to the 0-junction, which makes sense because all product pass through the depot before being further distributed.

For Shell in particular, there are storage facilities for refined oil products everywhere throughout the supply chain. Storing the oil products in inventories along the supply chain deals with different constraints for products and locations. These constraints on the element are not directly integrated in this Economic Engineering modelling approach. These constraints are taken into account when the model is used in a control setting which is topic of Chapter 4.

Building block for product flowing towards the storage depot from the source

The second building block for the model is shown in Figure 3-5, it represents the part of the supply chain where product flows enter the system of a storage depot. This product flow is an input to the system and is represented by a flow source S_f. This building block also contains other elements; a C-element for the amount of product allocated for the depot at the source, an I-element for the source price of the product and a R-element for the primary transfer to the depot.

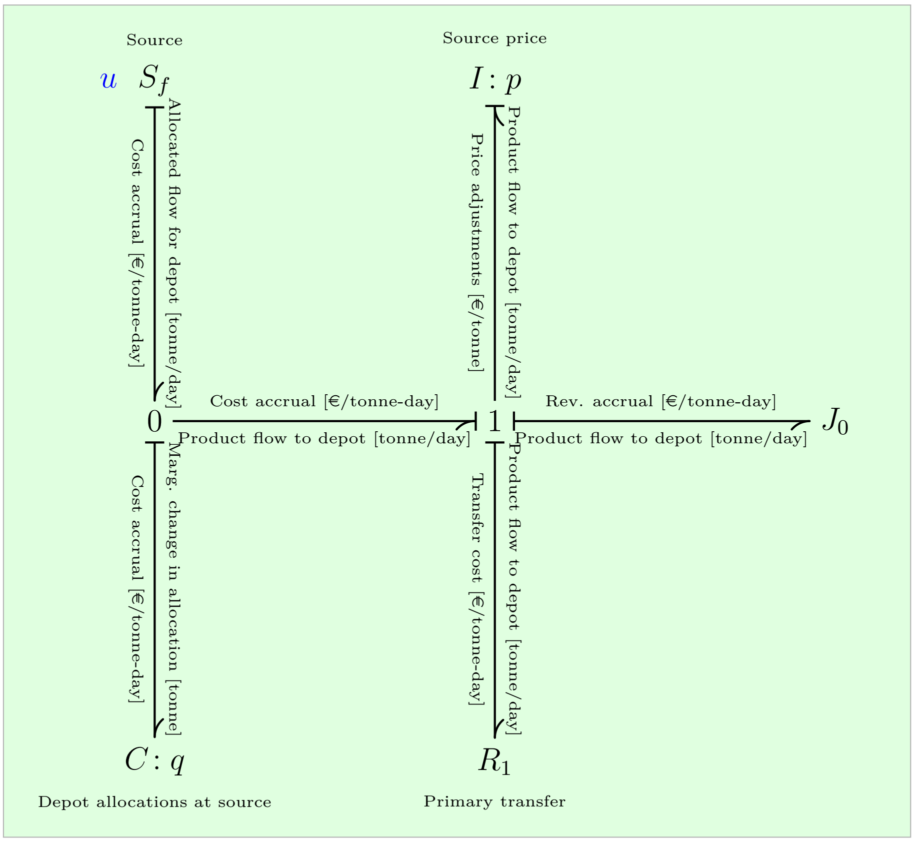


Figure 3-5: Building block for product flowing from the source towards the storage depot.

The shipments towards a depot are decisions made by the supply-chain managers based on current conditions in the supply chain and contractual agreements. The S_f-element is an input element, hence the input symbol u is placed below the flow source symbol S_f. In bond graph theory, the flow source represents an element where a flow is added to the system. Here, the causal stroke shows which side of the bond determines the magnitude of the flow. The causal stroke is at the side of the flow source since the magnitude is chosen by the supply-chain decision-maker.

Before product is arriving at the storage depot, agreements take place to allocate product from production sites to depots. At that moment, product is in ownership of the storage depot and that is modelled using the C-element. Allocations of product to parties are an useful concept in distribution networks that can provide cost savings [51].The allocated product changes over time and with shipments where variable amounts of product are collected every time. The cost accrual by the storage depot for the product allocated to the storage depot is the effort at the C-element. With the cost accruals, we account for the costs incurred for the product to be received [38].

In this building block, the I-element keeps track of the price paid by the storage depot to the supplier of the product that enters the system. The element let the transactions take place between the source and the storage depot where the product goes. The source price is the price paid to the manufacturers with which the product is lifted from the production sites to the storage depot. The source margin is the difference between the selling price of the products and the actual value of the product.

Entities like storage depots in the supply chain are connected by means of a supply-line, which means the transfer of the product to its destination. The transportation comes with a certain cost and hence these transfer costs are modelled with an R-element. The R-element returns a cost, given the flow of product in transport as given in Equation (3-2). In this element, cash flows are dissipated from the system and this is done by means of the transfer costs. Important to note is that the transportation of the products is modelled with an R-element connected to a 1-junction, the 1-junction implies that the flow of molecules does not dissipate as all bonds connected to the 1-junction have the same flow. Other transfer costs like packaging can also be taken care of in this R-element.

$$e = R_1 f \tag{3-2}$$

There are different ways of transportation possible for oil product in the supply chain of Shell and for every supply-line different ways of transportation are possible. Not all depots are accessible by train or shipping and the different modes of transport have different costs, sizes and duration. The assumption is needed to simplify these effects and this is done by modelling the transportation as one compound element.

Building block for product flowing towards sales channels

The building block for the product flowing from the storage depot to the sales channels is shown in Figure 3-6. This building block is connected to the building block for the storage depot at the junction J_0. The I-element generates the demanded flow by the sales channels based on the sales price, as a result the causal stroke at the bond is located at side of the I-element.

The building block for modelling the market to customers contains a 0-junction with elements. The product that is allocated for transactions to the sales channels is stored in a C-element, just like the allocations are done for the depot at the source in Figure 3-5. The C-element also makes it possible to reverse the process and being able to get additional product at the market. Buying the product can be beneficial when the future prices of the product are higher than the spot price [7]. The R_0-element connected to the 0-junction takes care of the flow that is picked up at the depot and extracted from the system with the relation described in Equation (3-3). The cost accrual for the customers is the effort on the bonds that takes care of the product pick-ups. This cost accrual is paid by the customers and received by the storage depot.

$$f = R_0 e \tag{3-3}$$

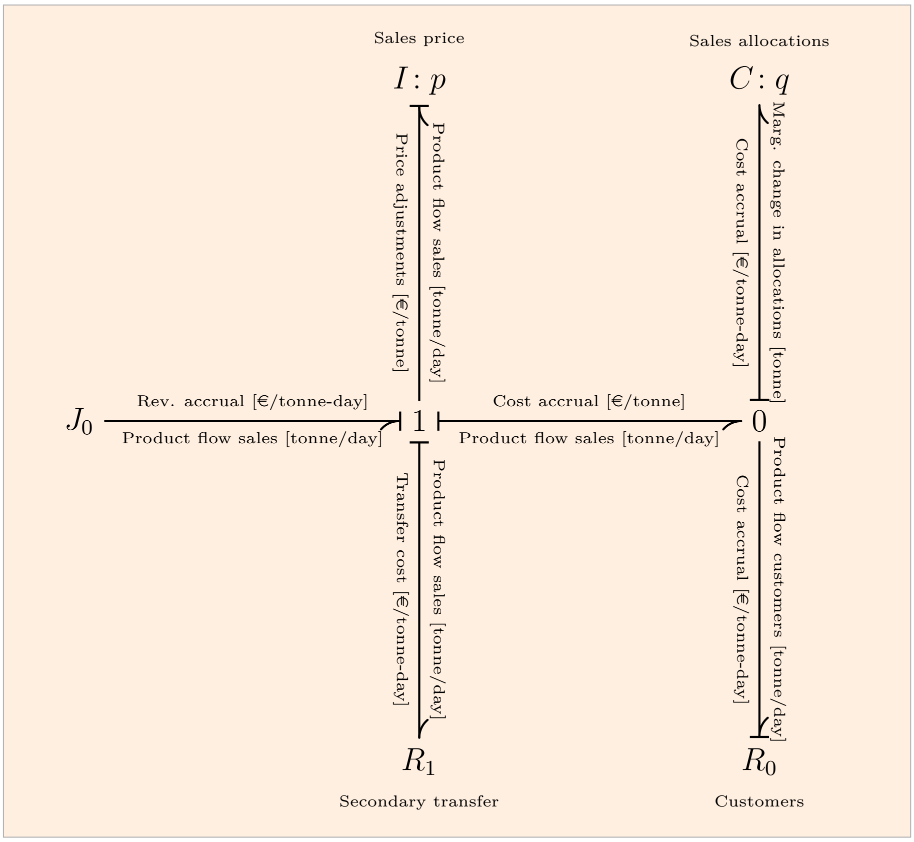


Figure 3-6: Building block for product flowing from the storage depot towards sales channels.

The transportation of the product to the sales channels is referred to as the secondary transfer and is modelled with the R_1-element on the 1-junction. The secondary transport is usually done by trucks in the refined oil product supply chain. Every depot has a number of trucks that drive back and forth from the depot to sales channels like retail stations. The associated costs for secondary transfer are accounted for in the R_1-element with the same relation as described for the primary transport in Equation (3-2).

Building block for product flowing towards other depots

The building block where flow of product leaves the system of the modelled depot to other depots is shown in Figure 3-7. This building block is developed in a similar way as for the building block where product is entering the system but in the reversed direction. When the storage depot functions as a serving hatch where significant amounts of product are put through to another depot, the flow is extracted from the system with a flow sink $S_{f,out}$.

At junction J_0, this building block is connected to the building block for the storage depot. The throughput price is stored at the I-element. The cost accrual for the throughput depots is the effort on the bonds at the 0-junction and are based on the throughput price. These costs are paid by the throughput depot to the storage depot.

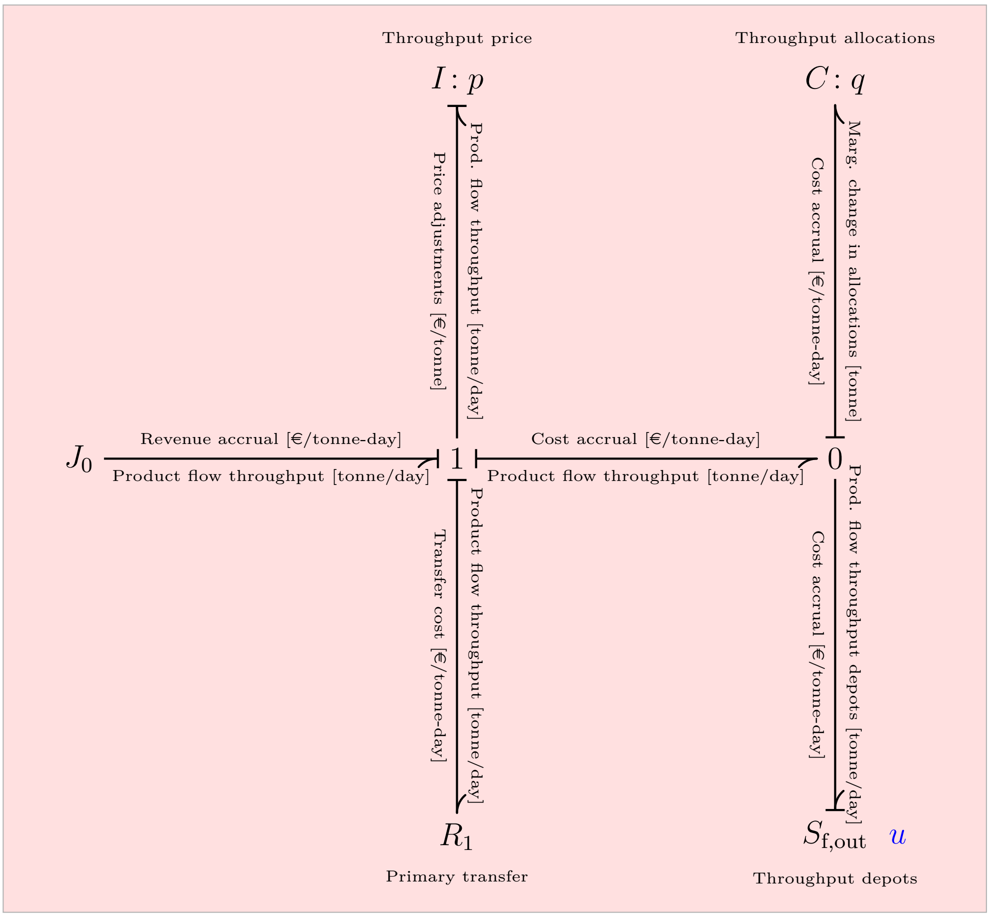


Figure 3-7: Building block for product flowing from the storage depot towards other depots.

The R-element in this building block represents the primary transfer for the other depots and thus the costs will be borne by the other depot. The contractual agreed volumes are modelled with the C-element. These agreed amounts of product are then in ownership of and transported to the other depot.

3-4 Qualitative Analysis of the Storage Depot Model

The bond graph model of the storage depot in Figure 3-2 is convertible to a state-space representation. The state-space representation of the generalized model for the storage depot in a supply chain of Figure 3-2 is given in Equation (3-4). The full derivation for the state-space representation of the model is done with the mathematical relations in Table 3-2 and shown in Appendix B.

In Equation (3-4), $x^T = \begin{bmatrix} q_2 & p_4 & q_7 & p_9 & q_{12} & p_{16} & q_{18} \end{bmatrix}$ is the state vector with the inventory level, allocated volumes and transfer prices as given in Table 3-1. $u^T = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$ is the input vector with the allocated incoming and outgoing products flows for the storage depot under consideration in the shipment schedule.

$$\dot{x} = \begin{bmatrix} 0 & -\frac{1}{I_1} & 0 & 0 & 0 & 0 & 0 \\ C_1 & -\frac{R_1}{I_1} & -C_2 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_1} & 0 & -\frac{1}{I_2} & 0 & -\frac{1}{I_3} & 0 \\ 0 & 0 & C_2 & -\frac{R_2}{I_2} & -C_3 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_2} & -\frac{C_3}{R_3} & 0 & 0 \\ 0 & 0 & C_2 & 0 & 0 & -\frac{R_4}{I_3} & -C_4 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_3} & 0 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u \tag{3-4}$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x$$

In Figure 3-8, the response of the system is shown for an input signal that represents shipments coming in at the depot on the first three days and a shipment going out to other depots at day 25. The parameters in the storage depot model for the qualitative analysis are tuned by hand. The corresponding Matlab code used for the qualitative analysis is given in Appendix F.

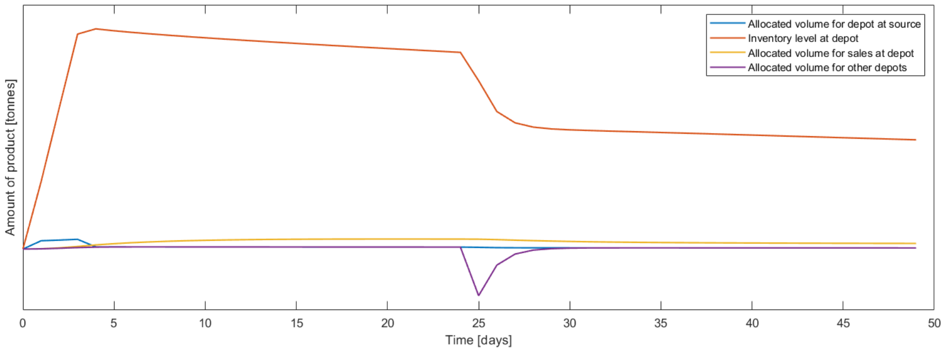
The evolution of the states q at the C-elements is shown in the upper graph. The inventory level at the depot increases as a result of the incoming shipments. After the increase in inventory at the depot, the volume allocated to the sales channels increases and remains present as long as there is product at the depot. The allocated volume for throughput is zero until the shipment going out as input is given to the system. At that moment, the depot has a negative allocated volume for throughput which is fulfilled afterwards with product from the inventory at the depot.

The sales price decreases while the depot is being supplied from the source. Thereafter, the prices increases while the inventory decreases until the the throughput of product to another depot takes place. The sales price then increases a bit more because of the drop in inventory. Both the source price and the throughput price are at level at the moment the products is transferred to or from the depot.

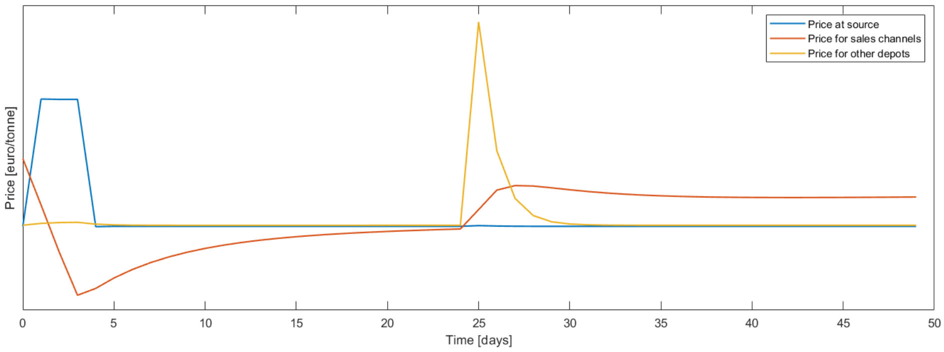
Figure 3-9 partly shows the pole-zero map for the storage depot model with transfer pricing. Two poles and a zero close to -100 are not visible in the pole-zero map. All poles are located

Table 3-2: Mathematical relations in bond-graph elements to derive state-space representation.

Bond-graph element	Interpretation	Mathematical relation
0-junction	Allocation of flows	$\sum f = 0$
1-junction	Market clearing	$\sum e = 0$
R-element	Transfer of product / product outlet	$e = R_1 f,\ f = R_0 e$
I-element	Price elasticity	$f = Ip$
C-element	Storage of inventory or allocations	$e = Cq$



(a) Quantity state responses for in- and outgoing shipments.



(b) Price state responses for in- and outgoing shipments.

Figure 3-8: Time response of the storage depot model for an input signal that represents shipments coming in at the depot on the first three days and a shipment going out to other depots at day 25.

in the left-half plane, which ensures stability for the system model [12]. The complex pole pairs generate a response component that is a decaying sinusoid [11]. This can explain that there is cyclical behaviour in the supply chain. The presence of some cycle is no coincidence, for example, sales channels recurrently pick up product at the depot. The cyclical behavior in the supply chain is analyzable in the frequency-domain. The frequency-domain analysis of supply chains is out of the scope of this research, but it is a promising research topic and further discussed in Section 7-3.

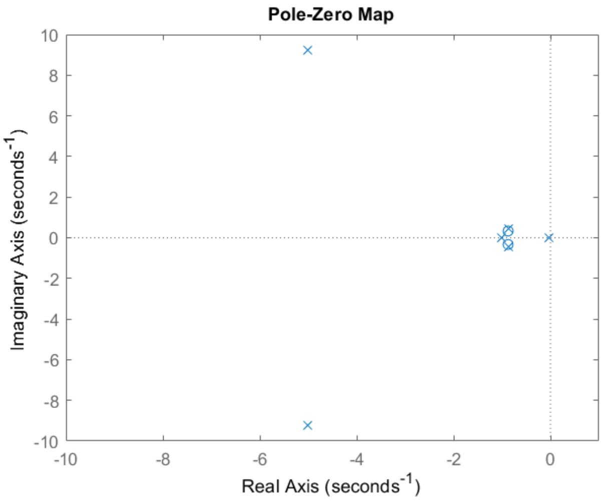


Figure 3-9: Pole-zero map of the storage depot model.

3-5 Conclusions

We use Economic Engineering Systems Theory for modelling the dynamic behaviour in the supply chain at storage depots. We give economic interpretation to the engineering concepts used in the dynamic modelling technique. The technique includes the dynamics of product flows and transfer prices in the modelling for a storage depot. This makes it possible to extract the revenues and costs from the model such that we can make the dynamics part of the decision-making for supply-chain operations, this is done for scheduling in Chapter 4.

The generalized approach with building blocks allows the user to make storage depot models as detailed as is required for the application. When more information is known about allocated volumes and transfer prices for product that is shipped to or from the storage depot, multiples of the same building blocks can be connected to the storage depot under consideration. This means that the modeling technique is scalable and adaptable, which makes it possible to customize the storage depot models.

Finally, not all supply chains deal with transfer pricing; an alternative modelling approach for storage depots in supply chains without transfer pricing is described in Appendix A.

Chapter 4

Model-Driven Objective Functions with Economic Engineering in MPC

4-1 Introduction

In this chapter, we use Model Predictive Control (MPC) for the scheduling of shipments at a storage depot. With MPC, we can specify a profit-maximizing objective function and we automate parts of the decision-making in the scheduling process. Figure 4-1 visualizes the scheduling process with MPC in a semi-automated setting.

We construct the objective function in the MPC algorithm with the Economic Engineering storage depot model. As a result, the future dynamics of product flows and transfer prices become part of the trade-offs in the decision-making process. The MPC scheduling algorithm and objective function are part of the MPC block in Figure 4-1 and replace professional expertise in the decision-making for scheduling shipments at a storage depot.

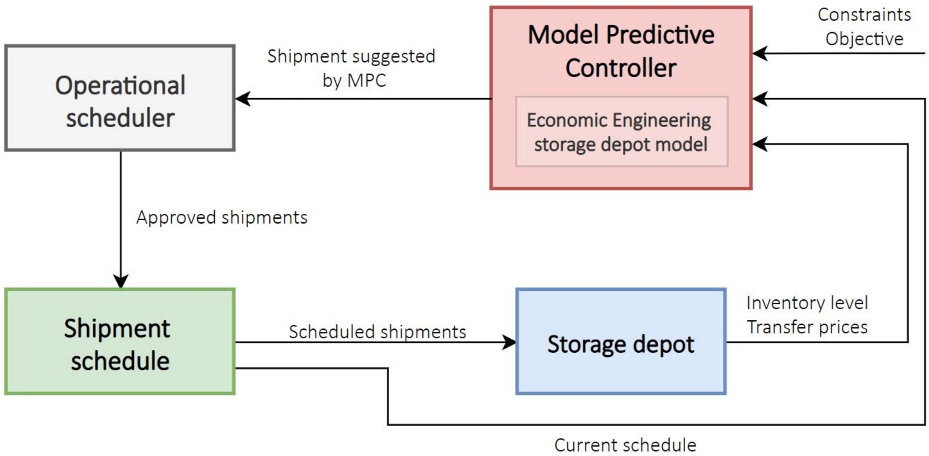


Figure 4-1: Model-driven and semi-automated decision-making process for scheduling shipments towards and from the storage depot in a supply chain.

In Section 4-2, we develop the objective function for the MPC algorithm based on the Economic Engineering storage depot model which we develop in Chapter 3. We thereby include the modelled dynamics of product flows and transfer prices at the storage depot in the decision-making. This is different from the more static approaches in which price changes over time are not included.

In Section 4-3, we build the MPC algorithm for the semi-automated scheduling process. Figure 4-1 shows how the MPC algorithm is interacting with the storage depot, operational scheduler and the shipment schedule for the decision-making. To incorporate the expertise of the scheduler, the MPC algorithm is setup as a semi-automated process. The schedulers assess the feasibility and have to approve the shipments that are suggested by the MPC algorithm. Ultimately, the MPC algorithm is able to replace the operational scheduler for completely automated supply-chain scheduling.

4-2 Economic Engineering Model-Driven Objective Functions

In this section, we use the Economic Engineering modelling technique developed in Chapter 3 to construct the objective function for the MPC algorithm. We thereby express the terms in the objective function completely with parameters and variables of the storage depot model. With this model-driven objective function, the modelled dynamics of product flows and transfer prices are taken into account in the scheduling optimization. In order to derive the terms for the objective function in a clear way, we use the numbered version of the bond-graph model for the storage depot in Figure 4-2.

In the general problem formulation in Equation (4-1), we choose the profit-maximization for the storage depot as the target of the objective function. This decision is made since supply-chain companies usually have the target to maximize profits in their operations [36]. The profits are built up by the revenues and the costs that are made to match the supply and demand at the depot. We use the storage depot model to construct the expressions for the revenue- and cost-elements of the objective function in the remainder of current section. In the end, we combine all expressions in the optimization problem formulation.

$$\begin{aligned}
\max_u \text{profit} &= \max_u \sum_{t=1}^{N_p} \text{profits}(p(t), q(t), u(t)) \\
&= \max_u \sum_{t=1}^{N_p} \text{revenues}(p(t), q(t), u(t)) - \text{costs}(p(t), q(t), u(t))
\end{aligned} \tag{4-1}$$

The objective of the MPC algorithm is to output control actions that maximize the profits over the period til the prediction horizon N_p. In Equation (4-1), the revenues and costs are functions of the tranfer prices $p(t)$ and the product flows $q(t)$ that are state variables in the

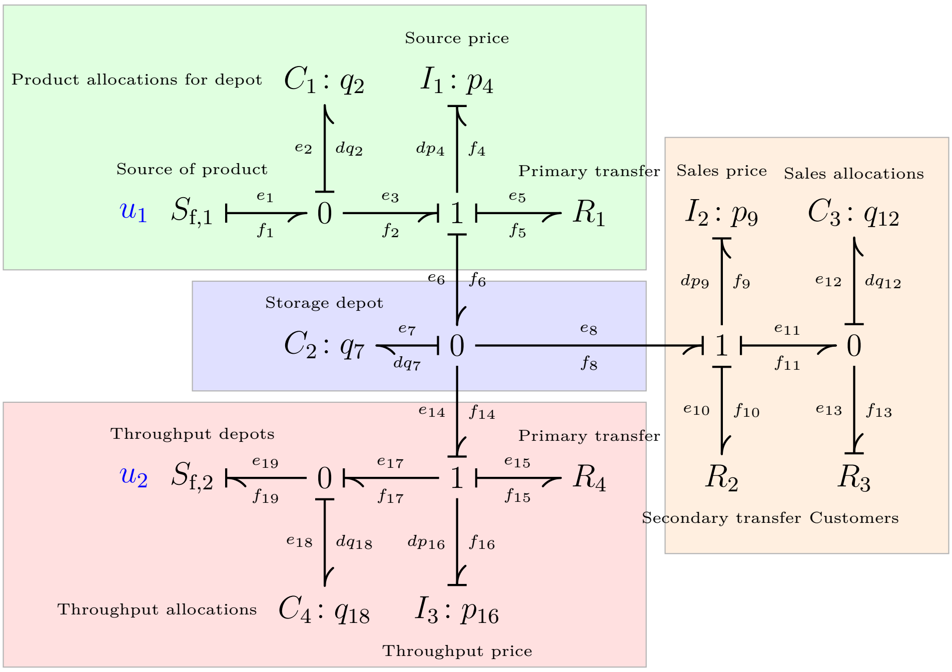


Figure 4-2: Numbered bond graph model for a storage depot in the product supply chain.

Economig Engineering (EconE) storage depot model. Equation (4-2) shows how we compose the input for the model $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$. $u_0(t)$ is the shipping volume going in and out at the storage depot in the shipment schedule at day t. $u_{\text{ex}}(t)$ is the additional shipping volume that can be added to the schedule. Every additional considered input $u_{\text{ex}}(t)$ is selected from the set of available shipments $U_{\text{ex}}(t)$ that could be added at day t to the schedule. The MPC algorithm outputs $u_{\text{ex}}(t)$ that has to be approved by the operational scheduler, this is further discussed in Section 4-3.

$$u(t) = u_0(t) + u_{\text{ex}}(t). \tag{4-2}$$

The revenue for the depot is the cash flow they receive from selling product to the sales channels and throughput depots. We use the numbered bond-graph and mathematical relations in Table 3-2 to express the revenue in Equation (4-3). The revenue is equal to the price paid by the buyer to get the product times the flow of product they receive from the storage depot. The first term represents the revenue from the sales to the sales channels. The second term is the revenue that follows from the throughput of product to other depots. The transfer price for the throughput of product p_{16} differs from the transfer prices that are used for the sales channels p_9. The reason for this difference is the fact that the storage depot charges different prices for sales to sales channels than to other depots. The dynamical transfer prices ensure that we use time-evolving prices for the product in the objective function, this reflects reality.

$$\begin{aligned} \text{revenues}(p(t), q(t), u(t)) &= f_8 p_9(t) + f_{14} p_{16}(t) \\ &= I_2 p_9(t)^2 + (u_2(t) - \frac{dq_{18}(t)}{dt}) p_{16}(t). \end{aligned} \tag{4-3}$$

The costs in the objective function are composed by the product costs for getting the product from the source and the transfer costs.

$$\text{costs}(p(t), q(t), u(t)) = \text{product costs}(p(t), q(t), u(t)) + \text{transfer costs}(p(t), q(t), u(t)) \tag{4-4}$$

The expression for the product costs in Equation (4-5) is similar to the expression for the product revenues from throughput sales. The product cost equals the flow of product that flows into the depot times the price paid to the source. Here, both the flow and price change over time. The flow of product varies with the different modes of transport that can be used to ship the product or the availability of product at the source.

$$\begin{aligned} \text{product costs}(p(t), q(t), u(t)) &= f_6 p_4(t) \\ &= f_3 p_4(t) \\ &= (u_1(t) - \frac{dq_2(t)}{dt}) p_4(t) \end{aligned} \tag{4-5}$$

The transfer costs are based on the primary transfer and secondary transfer. As shown in Equation (4-6), the primary transfer costs are determined with the product flow towards the depot, which is the same flow as used in Equation (4-5). The secondary transfer costs are determined with the same product flow as used in the calculation for the revenues from the sales channels. For both transfer costs, the flows are multiplied with the premium paid for the transfer of the amounts of product. The premium is determined with the resistance element (R-element) that represents the transfer of product in the storage depot model.

There is no distinction for the different modes of transport in the model. For the refined oil product supply chain the cost differentiation amongst the transport options is important for the calculation of profits and decision-making. A manner to have distinction between different modes of transport is separate input channels from the source towards the depot and from the depot towards sales channels, this results in multiple different building blocks with all unique elements.

$$\begin{aligned} \text{transfer costs}(p(t), q(t), u(t)) &= f_6 e_5 dt + f_9 e_{10} dt \\ &= I_1 p_4(t) R_1 f_5 dt + I_2 p_9(t) R_2 f_{10} dt \\ &= R_1 I_1 p_4(t)(u_1(t) dt - dq_2(t)) + R_2 I_2^2 p_9(t)^2 dt \end{aligned} \tag{4-6}$$

We combine the expressions for revenues and costs in the profit-maximization statement in Equation (4-7). The terms in the objective function for the MPC algorithm are completely expressed with the parameters and variables from the developed Economic Engineering model in Chapter 3. There are variables that change over time in all expressions for revenues and costs. By including these price and flow dynamics in the objective function based on the model predictions, the potential revenues and costs for different decisions are dynamically determined.

$$\begin{aligned}
&\max_u \text{profit} = \max_u J(x(t), u(t)) && (4\text{-}7)\\
&\text{subject to} \qquad \dot{x} = Ax + Bu \\
&\qquad\qquad\qquad \text{where } x = \begin{bmatrix} q_2 & p_4 & q_7 & p_9 & q_{12} & p_{16} & q_{18} \end{bmatrix}^T \\
&\qquad\qquad\qquad 0 \le q_7(t) \le q_{7\,\max} \\
&\qquad\qquad\qquad u(t) = u_0(t) + u_{\text{ex}}(t) \qquad \text{where } u_{\text{ex}}(t) \in U_{\text{ex}}(t)
\end{aligned}$$

Here, $J(x(t), u(t))$ is the objective function:

$$\begin{aligned}
J(x(t), u(t)) = \sum_{t=1}^{N_p} &(I_2 p_9(t)^2 + (u_2(t) - \frac{dq_{18}(t)}{dt})p_{16}(t)) \qquad (4\text{-}8)\\
&- ((u_1(t) - \frac{dq_2(t)}{dt})p_4(t) + R_1 I_1 p_4(t)(u_1(t)dt - dq_2(t)) + R_2 I_2^2 p_9(t)^2 dt),
\end{aligned}$$

and the future states in x that are part of the objective function are predicted by the model. The initial states are updated after each time step, such that the inventory level can be aligned with the actual measurements at the depot. For the state evaluations in the objective function, we use the system responses of continuous storage depot model sampled at the same times t as the input such that the sample time is equal to the time step dt of one day.

The discrete optimization on the nonlinear objective function can be done with a brute-force search. Then, the optimization has for every search a global solution since there is a finite set of input possibilities for which the calculations are made.

4-3 Model Predictive Control Algorithm for Scheduling Shipments in the Supply Chain

We develop the MPC algorithm for scheduling shipments towards storage depots in a semi-automated setting such that the professional expertise of the operational scheduler remains part of the decision-making. The setup of the MPC algorithm is given in pseudo-code in

Algorithm 1. At a day of operations, the algorithm outputs the optimal additional shipments for the shipment schedule. These optimal shipments are selected from a pre-determined set of available shipments U_{ex} that could be added to the existing schedule. In practice, all the available additional shipments for the different modes of transport could be extracted from the different transport scheduling platforms to generate the set U_{ex}.

Algorithm 1: Model Predictive Control algorithm for scheduling shipments towards storage depots in supply chain at day of operations.

```
Result: Set of additional shipments for the schedule.
Data: Set of available shipment options U_ex, the existing schedule, measured inventory
      level and constraints.
for day of scheduling operations do
    Update inventory level for initial conditions.
    for u_ex ∈ U_ex do
        Combine existing schedule with u_ex over prediction horizon N_p.
        Simulate system for N_p with updated initial state and possible schedule.
        Calculate value objective function with simulation response.
    end
    Sort u_ex on value objective function in U_sort.
    while no shipment action u_ex is approved do
        for u_s ∈ U_sort do
            Check feasibility of shipment action.
            if suggested shipment action is feasible. then
                Send suggestion towards scheduler.
                Wait for approval scheduler.
            else if suggested shipment action is not feasible then
                Shipment action u_s is rejected.
            end
        end
    end
    Update schedule and states with approved additional shipments.
end
```

As example, the input u, that is controlled for the scheduling, is visualized in Figure 4-3. At a day of scheduling operations, there is an initial existing schedule and a scheduler can add the additional shipments to that schedule. These additional shipment can be added on days when there are no shipments scheduled yet, or when there are already shipments in the schedule and then they are added in total shipment volume on that day.

For every day of scheduling operations, the algorithm combines the existing schedule for the prediction horizon N_p with each possibility in the set of additional shipments separately. The algorithm simulates the system with every combination and calculates the potential profits according to the objective function. After the most profitable set is found, the feasibility is

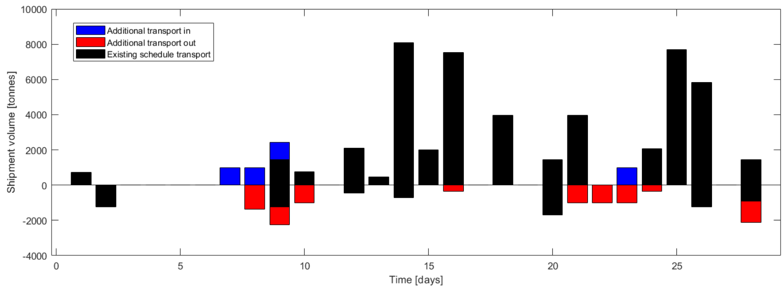


Figure 4-3: Visualized example of controlled input consisting of existing transport schedule and additional shipments that could be added. The positive values are the shipments coming in at the storage depot. The negative values are the shipments going out at the depot to other depots.

checked with the constraints discussed in Section 4-3-1. The MPC setup is developed as a semi-automated process to incorporate the expertise of the scheduler. Eventually the scheduling could be a fully-automated process, wherein schedulers do not have to approve shipments but only have to monitor and observe the feasibility of the actions of MPC algorithm. Technically, the fully automated setting is almost the same as the semi-automatic process, but the semi-automatic process is closer to what it can be in reality in the short-term.

4-3-1 Constraints and Considerations for Supply-Chain Scheduling

Table 4-1 shows the constraints that are implemented in the scheduling algorithm. There are inventory, transportation and storage location constraints acting on the process that are assessed in the scheduling algorithm. The constraints are assessed to ensure the feasibility of the control action. When the suggested control action is feasible, it can be approved by the scheduler and placed in the shipment schedule. Otherwise, the action is rejected and the next most optimal in the sorted list is picked.

The first inventory constraint is the maximum capacity at depot. Every depot has an unique capacity for the storage or number of tanks available in case of the refined oil storage depots. The capacity might change over time due to maintenance or sudden disruptions. Minimum stock is a constraint that has to be considered as well. The absolute minimum stock for a depot would be zero, but in practice there are agreements that emergency stock should be maintained [10].

The minimum and maximum capacity at the depot are hard constraints, they cannot be exceeded. In the scheduling algorithm, they are implemented with penalties in the calculation of the value for the objective function. This can be seen as a soft constraint, but the penalty is that high that it acts like a hard constraint. There is a point where a depot has too little inventory and a point where it has too much inventory as depicted in Figure 4-4. Tactical considerations with respect to the capacity used at depots can be taken into account with warning levels while scheduling. The soft constraints on these warning levels would be less

Table 4-1: Constraints accounted for in scheduling algorithm with MPC.

Constraint	Type
Inventory constraints	
Maximum capacity	Hard
Minimum capacity	Hard
Warning levels	- (Soft)
Transportation constraints	
Transport capacity	Input
Scheduling flexibility	Input
Planning cycle	Input
Storage location constraints	
Loading slots	Soft
Unloading slots	Soft

strict and can be further assessed by the operational scheduler.

The transportation constraints are for example the different modes of transport available at a depot, routes and the sizes of shipments. The shipments sizes are dependent on the transportation mode it is loaded in. For the product supply chain of Shell, pipeline, train, barge and truck are the four transportation options. However, not all options are available for every storage depot, depot locations might be only accessible by train and truck.

Another important aspect for the scheduling is the extent to which the scheduling can take place for the different modes of transport. In the refined oil product supply chain, barges are flexible and often used to supply depots, they are more flexible than trains and pipelines. Shipment plannings for trains and pipeline are made monthly for the month ahead, thereafter they are roughly scheduled two weeks in advance. These transportation constraints are not directly included in the MPC algorithm, but they are the key parameters for the generation of the set of available shipment options U_{ex}. Hence, they are specified as input constraints.

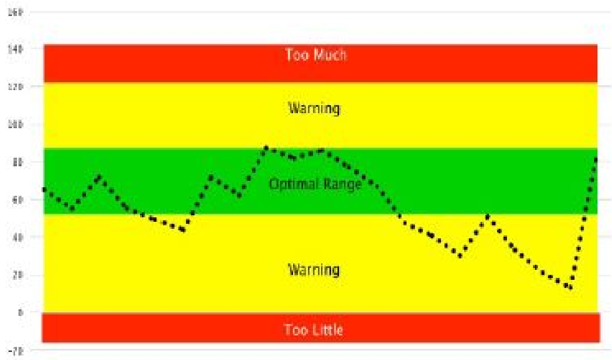


Figure 4-4: Illustration of optimal range in inventory levels [40].

In practice, these transportation constraints are directly implemented by the extraction of possible available shipments from the different transport scheduling platforms to generate the set U_{ex}.

For the storage location constraints, depots often have loading and unloading constraints. For example, at some depots barges cannot be unloaded at the same time and hence there is a maximum per day. These constraints are like the capacity constraints implemented with penalties. In the semi-automated scheduling process, they can also be assessed by the scheduler.

Finally, there are also production constraints, these include the availability of product at the production site, working days or minimum or maximum batch sizes for getting the product. The production constraints are not part of the scheduling algorithm. Eventually, all constraints must be taken into account in order to end up with a feasible schedule for shipments in the supply chain. As long as all constraints are not fully included in the semi-automated scheduling process, professional expertise will be required.

4-4 Conclusions

The model-driven objective functions in MPC algorithms with Economic Engineering models gives the opportunity to include the predicted dynamics by the model in the decision-making process. The integration of model-driven objective function with Economic Engineering Systems Theory in an MPC algorithm for scheduling is new. Different from the current approaches in which changes of transfer prices and product flows over time are not directly part of the decision-making, we use the future changes in product flows and price dynamics to calculate the optimal decisions in the scheduling algorithm.

In the scheduling process as visualized in Figure 4-1, the MPC suggests additional shipments to the operational scheduler. Since the MPC algorithm is able to take constraints into account, we use both the MPC and the operational scheduler to assess the feasibility of the additional shipments. Collaboration between the algorithm and the scheduler remains indispensable until developments make the input from the operational scheduler unnecessary.

Before practical implementation and full automation of the scheduling algorithm can take place, there are a couple of issues that have to be addressed. The implementation of all the constraints for the scheduling at storage depots is important, professional expertise must remain part of the process otherwise. The extraction of possible additional shipments for the modes of transport from the different transport scheduling platforms must be realized to have the real options in the optimisation. Lastly, the increase in computational complexity when the set of available shipments increases becomes a problem with the brute-force search as optimization method. The use of alternative optimization methods for the scheduling algorithm is recommended as future research.

Chapter 5

Application of the Scheduling Technology to Shell's Product Supply Chain

5-1 Introduction

This chapter aims to show the potential of the modelling technique and scheduling algorithm in application to Shell's refined oil product supply chain. The model-driven objective function in the Model Predictive Control (MPC) algorithm for supply-chain scheduling is based on the dynamics of product flows, inventory levels and transfer prices at the storage depot. In order to make the right decisions with the scheduling algorithm, the model has to be properly identified.

In Section 5-2, we perform the system identification for estimating the parameters of the storage depot model of Figure 3-2. We describe the system identification process for the storage depot model with historical supply-chain data for multiple storage depots in the refined oil product supply chain. The ability of the storage depot model to capture supply-chain dynamics is shown with identification and validation results.

In Section 5-3, we apply the scheduling algorithm to the Flörsheim depot in the Rhine river supply chain to show how supply-chain scheduling can be done with MPC. The results of the application and a modelled low water level scenario can be seen as the proof of concept of the scheduling algorithm.

Section 5-4 presents the corresponding Graphical User Interface (GUI) for the semi-automated scheduling process. With the GUI, we demonstrate what scheduling would look like in a setting where the MPC algorithm performs the decision-making; the job of the operational scheduler is to monitor and approve the shipments.

5-2 System Identification for Supply-Chain Modelling Technique

This section discusses the system identification process and results for the storage depot modelling technique with Economic Engineering Systems Theory. We estimate the parameters in the Economig Engineering (EconE) model in Chapter 3 with system identification.

By developing the EconE models, we define the underlying structure of the modelled system. The resulting state-space representation makes that the method for identifying the parameters of the EconE models is grey-box identification. In order to conduct a parameter identification for the storage depot model; the selection, collection and processing of input and output data is essential [53]. The data preparation is done for both identification and validation by demeaning the data.

The grey-box identification is done using Matlab [32] with the following steps:

1. Create function with A, B, C and D-matrices of state-space representation.
2. Set initial values of the parameters to be identified.
3. Load and process data.
4. Set search options.
5. Perform grey-box identification with *greyest()*.

The corresponding Matlab codes used for the identification process is given in Appendix G.

The C, I and $R-$parameters in the state-space representation in Equation (3-4) have to be identified. The correctness of the identified model is assessed with the Variance Accounted For (VAF) score [53], which is calculated with Equation (5-1). The VAF score of a model is given with a percentage that reflects the fit of the model output with the actual data. A good model has a low prediction error and a high VAF score.

$$\text{VAF} = \Big(1 - \frac{\text{var}(y_{\text{data}} - y_{\text{model}})}{\text{var}(y_{\text{data}})}\Big)100\% \tag{5-1}$$

We base the size of the identification and validation set on the decision-making process. In the refined oil product supply chain, when scheduling pipeline or train shipments, the schedulers look for the month ahead. During the scheduling, the shipment are placed in the schedule at least two weeks in front. Considering that the effects from these shipments are visible the days and weeks after they arrive, the dynamics are modelled for four weeks in the MPC algorithm. Taking this into account for the system identification process, we use an identification and validation period of two months.

Historic data for the inventory levels and primary transportation of product is used to perform the system identification. We use the data for Automotive Gas Oil (AGO), also called diesel fuel. We use the data for seven different depots (Flörsheim, Ludwigshafen, Würzburg, Altmannshofen, Linz, Wien Lobau and Salzburg) in the DACH region for the period June

2019 - September 2019. Inventory levels are measured daily at all depots. The data for primary transport represents the shipments towards depots with the quantities, dates, sources and destinations.

In the following section, we compare the system identification results for the EconE modelling approach with the actual data. We perform and show the results of the system identification for both the EconE model with and without Transfer Pricing (TP). The EconE modelling approach without TP is discussed in Appendix A.

5-2-1 Predictive Performance of Storage Depot Model

We use the primary transport schedule as input and the inventory levels as output during the system identification. The parameters of the model are identified during the training process and thus the time responses of the models to the primary transport as input can be simulated.

The VAF scores for both EconE models at all seven modelled depots are shown in Table 5-1. In this chapter, the system identificaion results are shown for the depot Flörsheim. The system identification results of the other depots for the same period are shown in Appendix C.

The VAF scores for the models at the different locations in Table 5-1 shows that the modelling technique for the dynamical behaviour around storage depots is promising. Despite the relatively lower VAF scores for some configurations, they seem to model the dynamics well when looking at the corresponding output responses in Appendix C.

	EconE model without TP		**EconE model with TP**	
Depot location	Identification	Validation	Identification	Validation
Florsheim	90.12	88.56	89.28	86.31
Ludwigshafen	72.03	52.39	74.85	62.36
Wurzburg	79.40	38.00	86.08	31.20
Altmannshofen	67.05	53.85	79.27	73.46
Linz	83.55	78.55	90.24	81.19
Wien Lobau	69.12	50.34	78.96	63.81
Salzburg	49.53	32.14	57.24	28.36

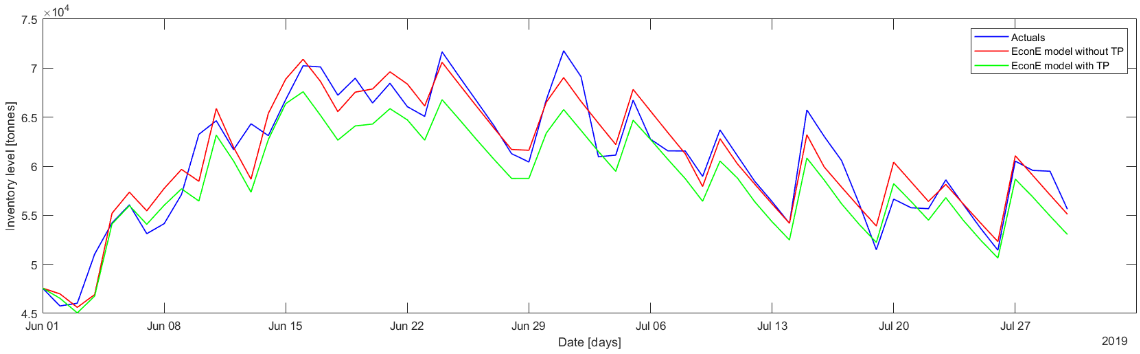
Table 5-1: Identification and validation VAF-scores for the EconE model with and without transfer pricing (TP) for the different storage depots.

Figure 5-1 shows the output responses of the identified EconE models and the measured inventory levels at the Flörsheim depot. The output responses are shown for the identification and validation set. The models are fit on the identification data set, where the actual inventory levels was the target variable (i.e. the output variable). The models for the depots perform well in reproducing these stock levels in the identification.

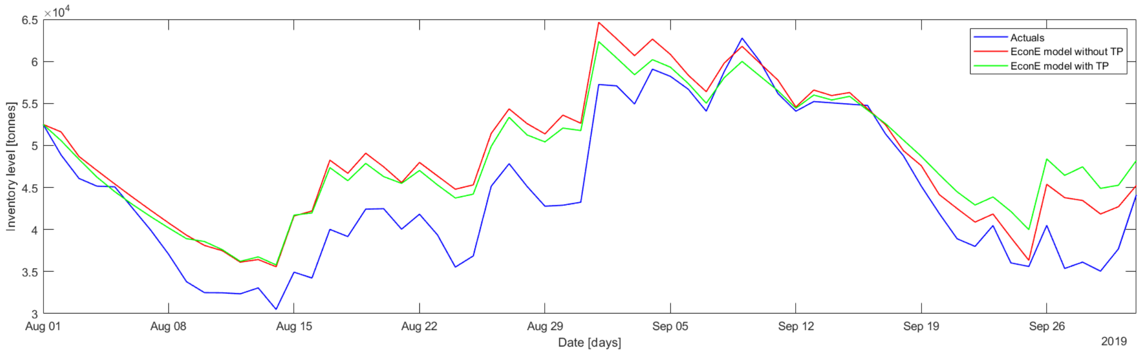
Successively, the identified model is used to predict the responses for the validation data set. The output responses for this validation set of both the EconE model with and without transfer pricing are again similar to the actual inventory levels. This indicates that the model has been properly identified and can accurately model the dynamics with the primary transport schedule as input.

The responses of the models and inventory levels evolve similarly. There is a small difference with the measured inventory levels at the beginning that affects the subsequent predicted inventory levels. After the stock levels align again, another unevenness develops in the end that continues to evolve; however these are minor details, the system dynamics are well modelled by both models.

Appendix D shows the evolution of the other state variables in the identification and validation phase except for the output state which is shown here in Figure 5-1.



(a) Inventory levels by EconE models and actuals for identification data.



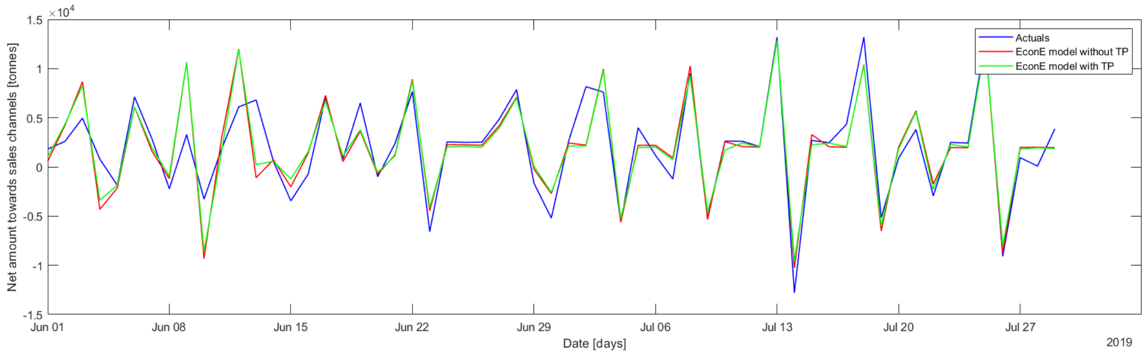
(b) Inventory levels by EconE models and actuals for validation data.

Figure 5-1: System identification results for the storage depot at Florsheim.

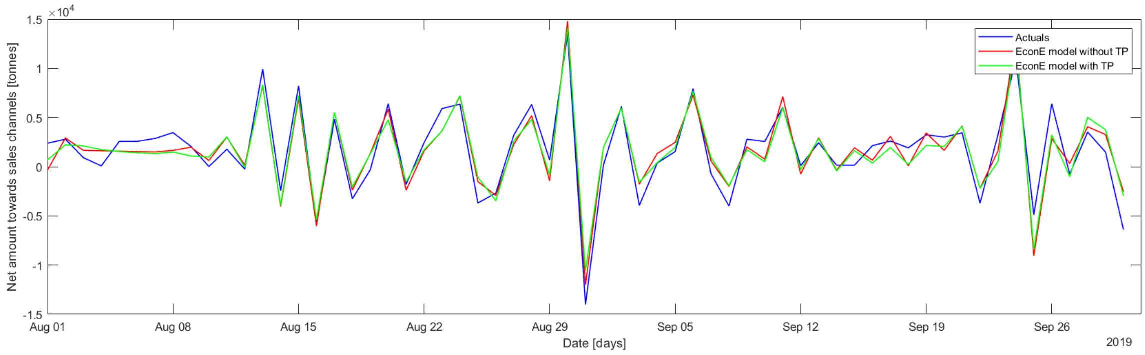
In Figure 5-2, we observe the identification and validation results for the net flow of product to the sales channels modelled by the EconE models compared with the net flow derived from the actual inventory levels for the identification data set.

The net flow of product to the sales channels is internally modelled in the systems' model. Correctly modelling the net flow towards the sales channels means that the inventory levels will also be well-predicted. The differences in inventory levels in Figure 5-1 are mainly caused by the differences in net flow of product in Figure 5-2. In the validation set, the differences in inventory levels between the models and the actual flows are made at the beginning because the net flow is less well predicted at the start of the validation set. After that, the models continue to accurately predict this net flow and this is reflected in the VAF scores for both models.

Given the VAF scores and the similar development of the inventory levels and net flow towards sales channels predicted by the model and the actuals, the EconE technique for modelling the dynamics of product flows around the storage depot works well.



(a) Net flow towards sales channels modelled by EconE models and actuals for identification data.



(b) Net flow towards sales channels modelled by EconE models and actuals for validation data.

Figure 5-2: System identification results for the storage depot at Florsheim.

5-3 Scheduling Shell's Product Supply Chain with Model Predictive Control

This section discusses the application of the MPC algorithm for supply-chain scheduling to the refined oil product supply chain of Shell. The goal is to show the potential of model-driven MPC for supply-chain scheduling. The generation of the results is done with the idea to show how the application of the MPC algorithm in supply chains would work.

For the simulation that is done with the MPC algorithm, the used data is the primary transportation schedule of the validation data set as shown in Section 5-2. It is assumed that the primary transport schedule that serves as input in the EconE model already exists partially in the scheduling simulation. This way, it looks more like the real-life scenario in which you have a certain schedule on which the extra shipments are added.

The set is generated based on a planning cycle parameter representing the planning cycles for the modes of transport and the size of shipments for that mode of transport. The values for these parameters used in the MPC simulation are shown in Table 5-2. For the pipeline it means that at day t the MPC algorithm can add an additional pipeline shipment of 5000 tonnes towards the depot in the schedule for day $t + 14$. We also insert a constraint for the maximum number of loading and discharging slots for the primary transport shipments, which we set to three. This excludes the loading slots for secondary transportation to sales channels.

Mode of transport	Quantity	Scheduling flexibility parameter
Barge	2000	7,8
Pipeline	5000	14
Train	1200	10

Table 5-2: Parameters to generate the set of possible additional shipments for the scheduling simulation.

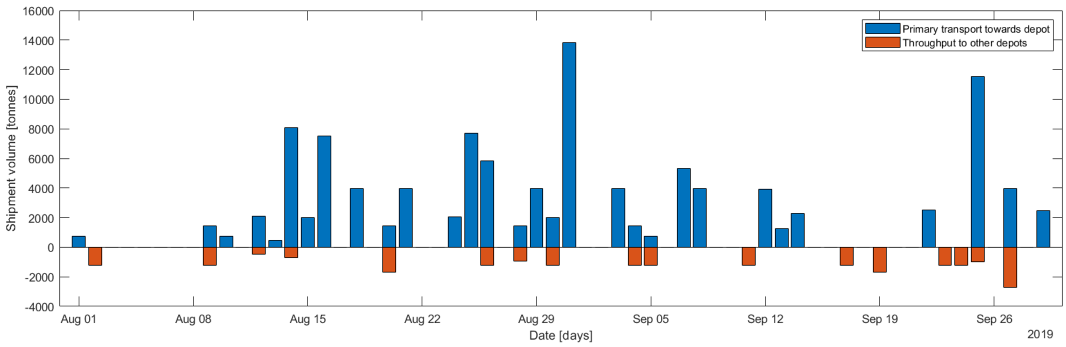
The technology needs further development and research before being applied to actual scheduling in supply chains. Furthermore, an important note is that the values for profits in the MPC simulation result from the objective function and are not the actual profits.

5-3-1 Results of the Scheduling Simulation with Model Predictive Control

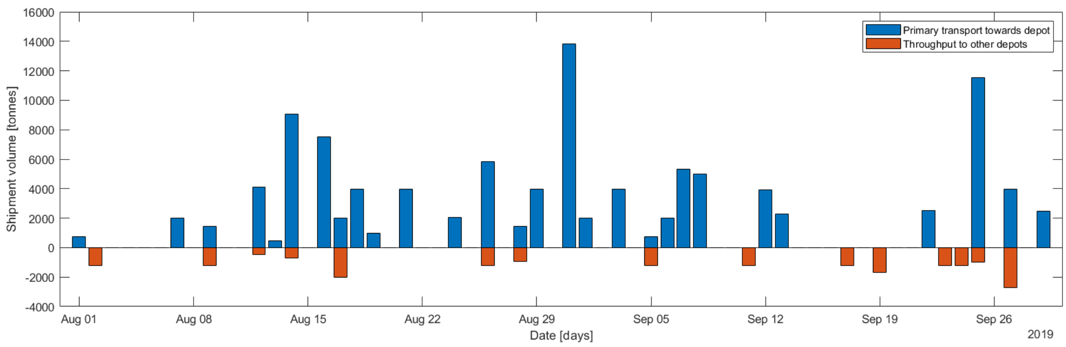
Figure 5-3-a shows the actual primary transport schedule for the validation set. The time response for the EconE models were modelled with this input. For the MPC simulation, parts of the primary transport schedule has been removed. Based on this adjusted schedule the MPC algorithm comes up with new shipments and the new schedule made by the MPC algorithm is shown in Figure 5-3-b. Parts of both primary transport schedules corresponds because these have not been removed. Furthermore, it seems that the MPC sometimes opts for multiple shipments on the same days instead of spreading shipments over multiple days. The inventory levels anticipated on the basis of the Economic Engineering model given the controlled schedule by the MPC algorithm are shown in Figure 5-3-c.

Figure 5-4-a shows the accumulated value of the profit objective function for the MPC algorithm over the simulation time. The accumulated value of the profit objective function is also determined for a simulation with the initial schedule which is not manipulated by the MPC algorithm. Figure 5-4-b presents the revenues and costs from which the profit is composed.

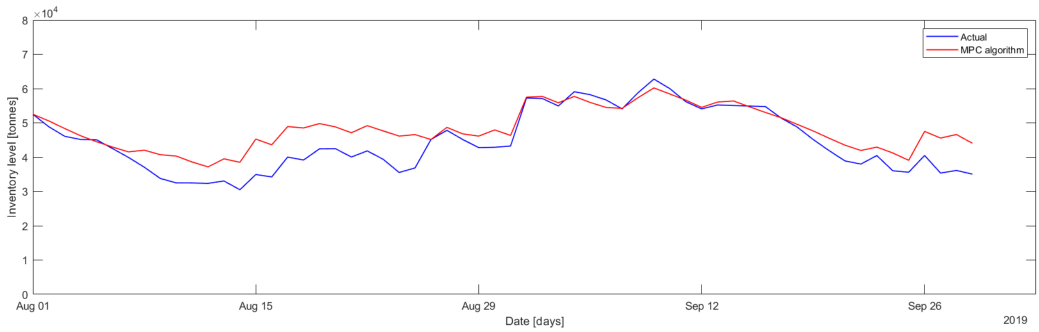
At the end, the profit that would have been obtained is higher when the schedule is created by the MPC algorithm than the original one. There is no difference in profit for the first days



(a) Original schedule for primary transport.



(b) Schedule for primary transport controlled by MPC.

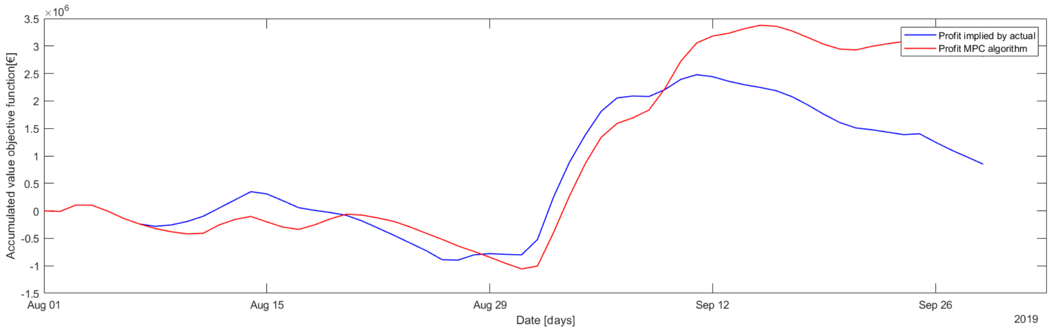


(c) Schedule for primary transport controlled by MPC.

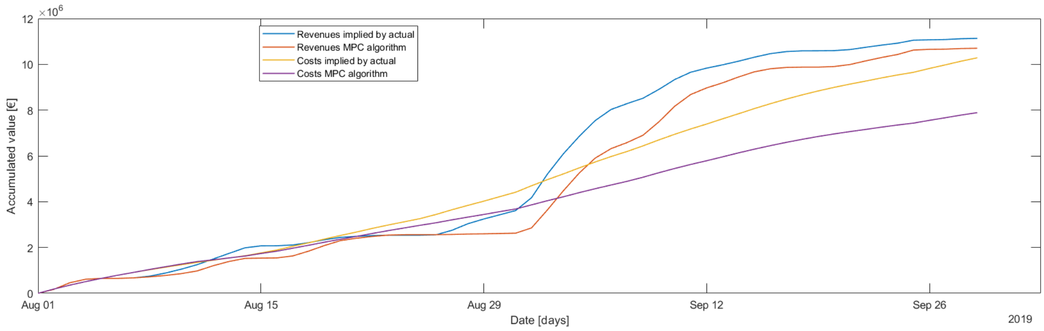
Figure 5-3: Shipment volumes in the shipments schedule for the storage depot and the resulting inventory levels in the simulation.

as the MPC algorithm cannot manipulate the schedule until seven days after the start of the simulation as explained about the planning cycle in Section 4-3.

Figure 5-4-b shows that the total revenues is lower for the MPC algorithm than the implied revenues by the actual schedule. We observe the same for the costs. The total volume sold is similar for both approaches, this means that the MPC found a way to supply the same demand with lower costs ending up slightly more profitable.



(a) Accumulated value of profit in objective function.



(b) Accumulated value of revenue and cost parts in objective function.

Figure 5-4: Accumulated values of profit objective function over simulation time for original schedule and MPC manipulated schedule.

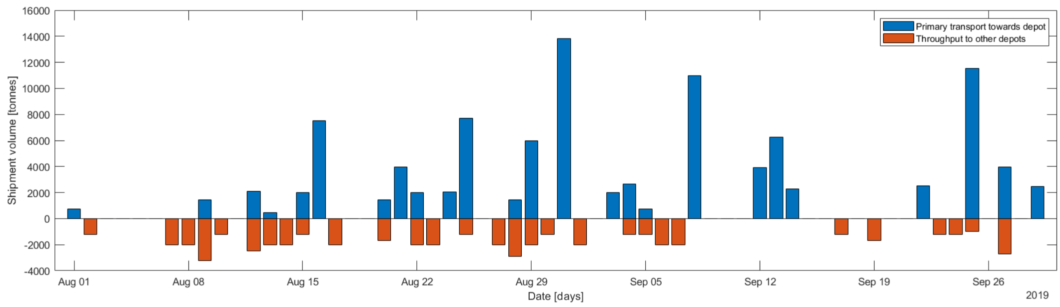
Scenario Modelling

In order to show the potential of the EconE modelling technique and the model-driven MPC algorithm in a disrupted supply chain, we perform a scenario analysis on how the technology deals with low water levels. In the scenario of low water levels, freight rates can be much more expensive as usual and barges might be loadable for small percentages. To give an idea how the MPC algorithm makes decision in such a situation, we compare the MPC simulation results for two situation:

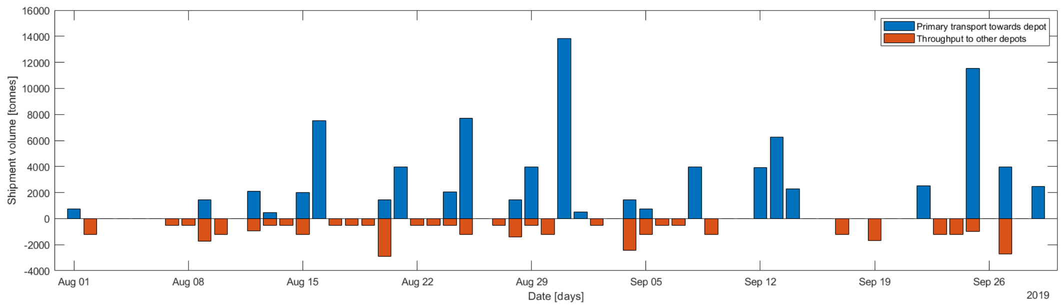
- The more usual situation where barges can be fully loaded (2000 tonnes) and no transfer cost correction.
- A low water levels scenario where barges can only be loaded for 25% (500 tonnes) and is three times more expensive.

We also adjust the scheduling flexibility parameter for barges to $\{7, 8, 9, 10\}$ and maximum loading and discharging time slots to four, this leads to a unrealistic situation with a lot more shipments but it emphasizes the decision-making of the MPC algorithm in both situations. Figure 5-5 presents the manipulated schedules by MPC for both scenarios.

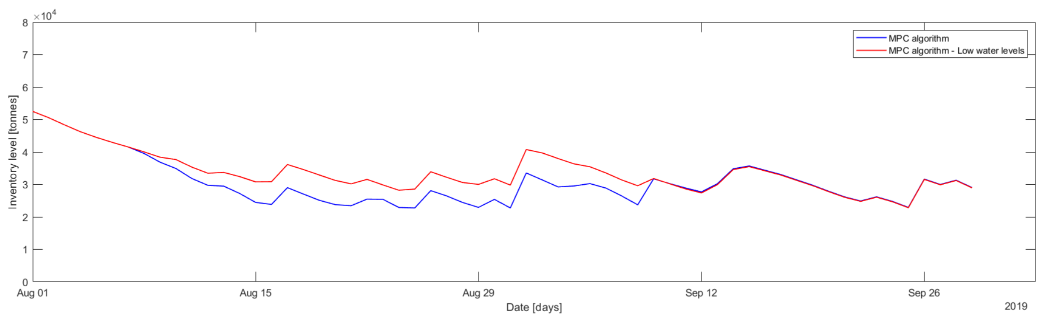
In both situations the MPC ends up with much more shipments as expected. The schedule in the usual situation has much more shipments coming in at the depot, where the low water situation has only some larger (pipeline) shipments added to the initial schedule. The shipments going out in both situations are just as frequent, but with lower quantities for the low water level situation. The resulting inventory levels in Figure 5-5-c show that the low water level situation has almost no effect on the inventory levels. Despite the fact that only the throughput depots receive less product from the storage depot, it seems that the demand from sales channels is being met equally well.



(a) Schedule for primary transport controlled by MPC for the more usual situation.



(b) Schedule for primary transport controlled by MPC for the low water level scenario.



(c) Resulting inventory levels from the simulation for both scenarios.

Figure 5-5: Simulation results of the decision-making by the MPC algorithm for a usual and a low water level scenario.

5-4 Graphical User Interface for Semi-Automated Supply-Chain Scheduling

The GUI serves to give an idea of how the modeling and decision-making with MPC adds value once it is further developed and how the decision-making process benefits from it in practice. The GUI shows the results from modelling the dynamics of product flows around the storage depot and the simulation results from the MPC algorithm. Furthermore, the scheduling tool is developed as a semi-automated process that requires input from the scheduler to approve or refuse shipments.

The first screen in the GUI functions to analyse how well the identification of the storage depot model is done for all depot locations and gives the VAF-scores.

On the second screen in Figure 5-7, we demonstrate the process where the scheduler assesses and approves the shipments based on the resulting dynamics and performance measures resulting from the simulation of the MPC algorithm. The MPC algorithm simulates the dynamics in the supply chain for four weeks with the already existing schedule and the suggest additional shipments. The assessment of the shipments by schedulers will mainly be to check the feasibility and to make it a semi-automated process by keeping the connection with the algorithm. Tactical considerations that are not included in the MPC could also be taken into account in this step. Therefore, a selection of economical performance measures are included in the interface to give the scheduler an indication of the effect of the shipments.

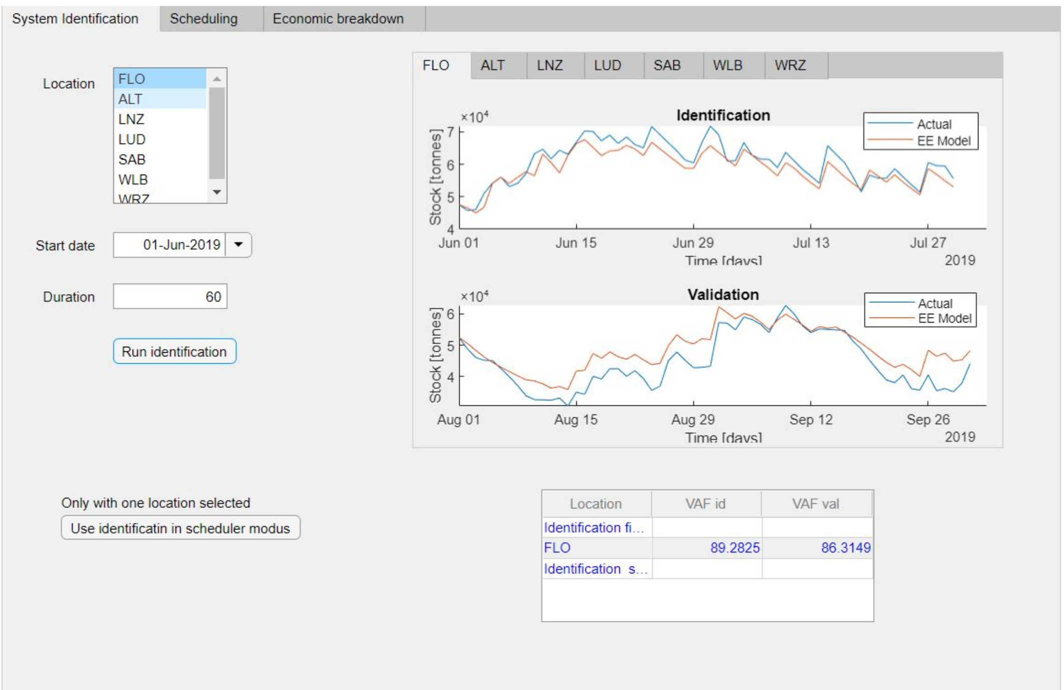


Figure 5-6: GUI tab for systems identification of the storage depot model for different locations.

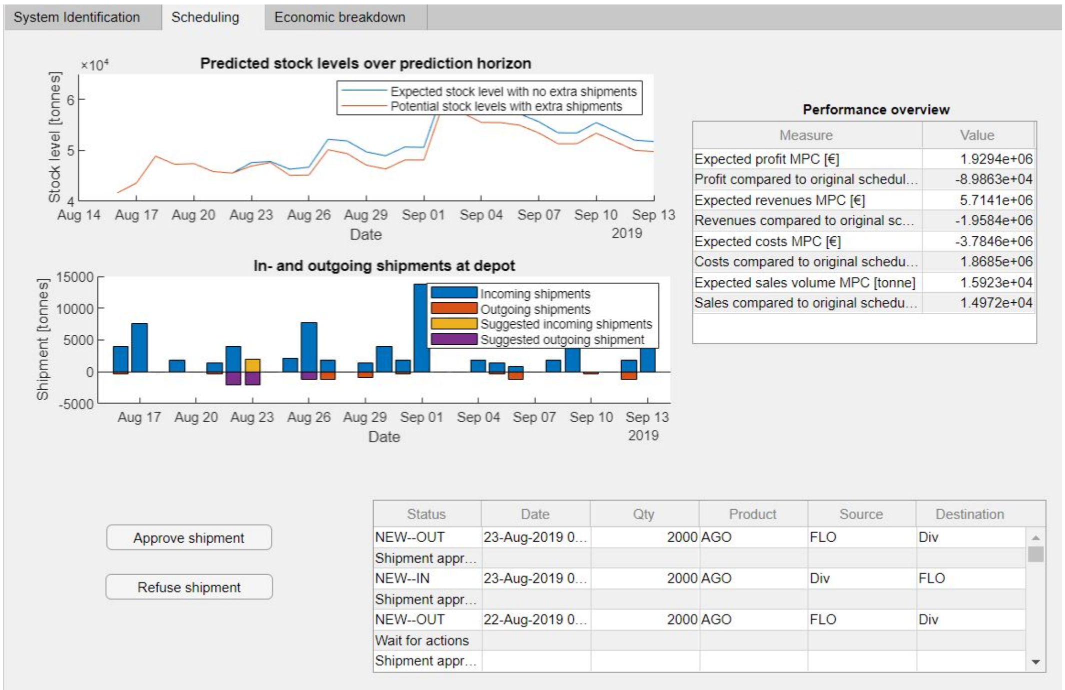


Figure 5-7: GUI tab for semi-automated scheduling of shipments towards depots with relevant performance measures.

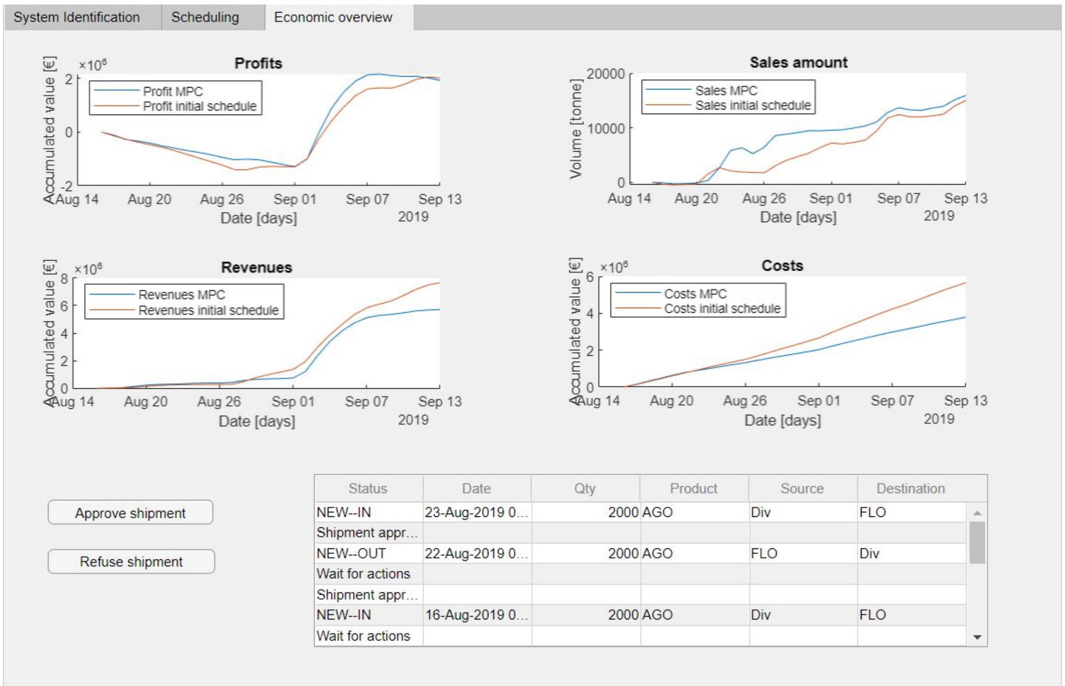


Figure 5-8: GUI tab for semi-automated scheduling of shipments towards depots with additional information on economical impact of decisions.

The last screen of the GUI in Figure 5-8 serves as additional information on the effect of the shipments. The difference in evolution of the potential profits, revenues and costs as effect of the additional suggested shipments is shown. Also, the predicted sales amounts are shown in this screen. Eventually, other appropriate performance measures for supply-chain scheduling could be included in the GUI.

5-5 Conclusions

In this chapter, we applied the modelling technique and scheduling algorithm to actual supply-chain data. The identified storage depot model shows the potential of the modelling technique with Economic Engineering Systems Theory. The developed modelling technique is the basis to add more anticipation on future product flows and transfer prices in the decision-making process for supply-chain management.

The results from the scheduling simulation indicate that the decision-making by schedulers would be different using the MPC algorithm. Continuously changing product flows and transfer prices influence the potential revenues and costs in the supply chain, the developed scheduling algorithm provides the possibility to incorporate predicted dynamic behaviour in the decision-making processes. For the scheduling process, this could lead amongst others to more economic benefits and additional time for strategic activities by supply-chain experts.

Altogether, the implementation of the scheduling technology is possible but complicated in practice. It is conceivable that the systems for the scheduling of the modes of transport are all on different platforms. Automating this entire process from reading the availability for transport options from these systems to actually booking time slots or shipments is a complex process. Further research and dedication from industry will be needed to get the full potential out of the technology.

Chapter 6

Conclusion

For supply-chain scheduling, the theory of model-driven objective functions in MPC with Economic Engineering Systems Theory incorporates the future dynamics of product flows and transfer prices into the decision-making. Supply-chain schedulers at storage depots experience changes in product flows and transfer prices that affect the revenues and costs. With the observed supply-chain concepts at Shell, we developed a dynamic model for product flows and transfer prices at a storage depot using Economic Engineering Systems Theory.

With the development of an Economic Engineering model for the storage depot, we are able to approach the scheduling operations as a Systems and Control problem. As a result, we can use techniques from this field to optimize and automate decision-making from the perspective of control engineers. We use MPC to schedule shipments at storage depots because of its constraints-handling property and the ability to specify an objective function. The interpretability of the Economic Engineering storage depot model is used to construct the objective function based on revenues and costs. With this model-driven objective function, the scheduling algorithm uses predictions for the future dynamics in the supply chain to optimize the shipments schedule in a profit-maximizing way.

On the other hand, the theory for model-driven objective function in MPC with Economic Engineering Systems theory is applicable outside the field of supply-chain scheduling. The ability to construct the objective function in MPC from Economic Engineering models is useful for economic optimization problems. Together with the development of an Economic Engineering model for an economic system, the theory allows allows for the inclusion of the predicted future dynamics of the economic system in the decision-making processes.

Chapter 7

Recommendations

7-1 Introduction

During the research and work on the developments in this thesis, interesting research opportunities were identified which we recommend in this chapter. Section 7-2 describes a follow-up study that has potential within the refined oil product supply chain in more detail. Section 7-3 discusses other application areas that are of interest for supply-chain management with Economic Engineering Systems Theory and Model Predictive Control (MPC) algorithms.

7-2 Flow Reallocations to Sales Channels in the Supply Chain with MPC

It occurs often that a disruption in the supply chain leads to infeasibilities in the supply chain. For example, a route cannot be used with a mode of transport. At that moment, a channel optimizer has to come up with alternative routes, but the best alternative options available are not known. It is hard to find the best combination of transportation and getting the product to its destination. The estimated economical impact is not visible for the decisions that have to be made.

Figure 7-1 shows how an MPC algorithm can be used to determine optimal reallocations given the constraints and related costs. The MPC algorithm uses the dynamic models for the storage depot and retail stations to suggest the reallocations to the channel optimizer. The channel optimizer assesses the feasibility and approves or rejects the reallocations.

The MPC can be developed such that it is capable of changing product flow options based on an objective such as profit or stock optimization. Plans can be made with alternatives while taking into account the constraints of locations, modes of transport and planned disruptions. By including freight rates for routes and other costs, the economical impact of these decisions can be evaluated.

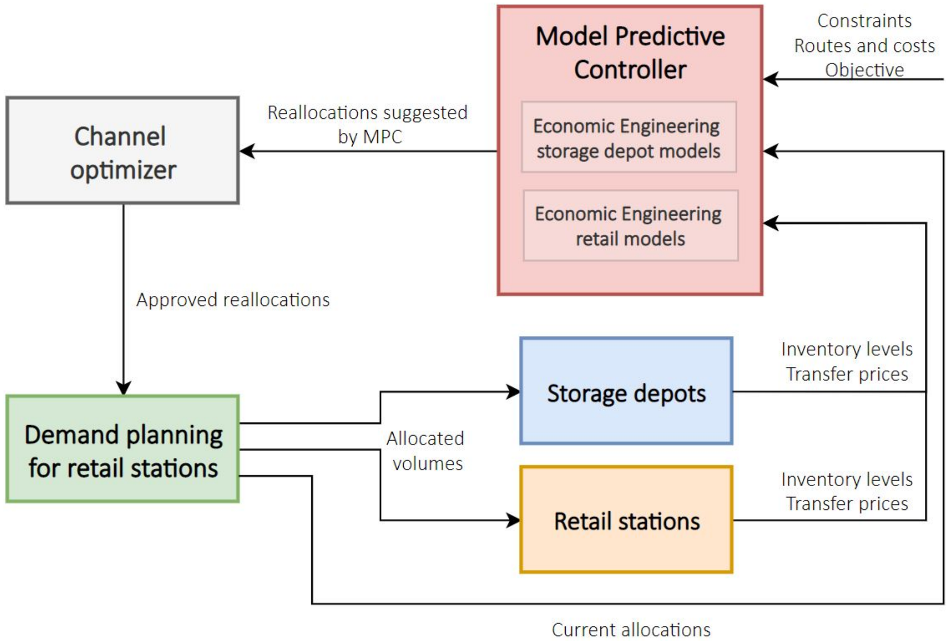


Figure 7-1: Decision-making process for the reallocation of product flows towards retail stations with MPC.

The modelling technique for a storage depot in Chapter 3 does not use the known information about the demanded product flow by retail stations or other sales channels. The introduction of a new building block is necessary to include this additional information in the model. The Figure 7-2 shows the building block for the retail sales channel that can be used for known amounts of product demanded by that sales channels. The building block has an equivalent structure as the building block for the throughput to other depots in Figure 3-7.

The building block of Figure 7-2 can be connected at the junction J_0 to the 0-junction of the depot under consideration from which the flow of product is picked up, this is shown in Appendix E. The transfer price paid by the retail station to get the product is stored in the inertia element (I-element). The input at the flow sink $S_{\mathrm{f,out}}$ is the total expected demanded flow picked up by the retail stations at the storage depot. These known demanded volumes are part of the demand planning for retail stations. In Figure 7-1, the MPC algorithm modifies the volume in the demand planning when necessary.

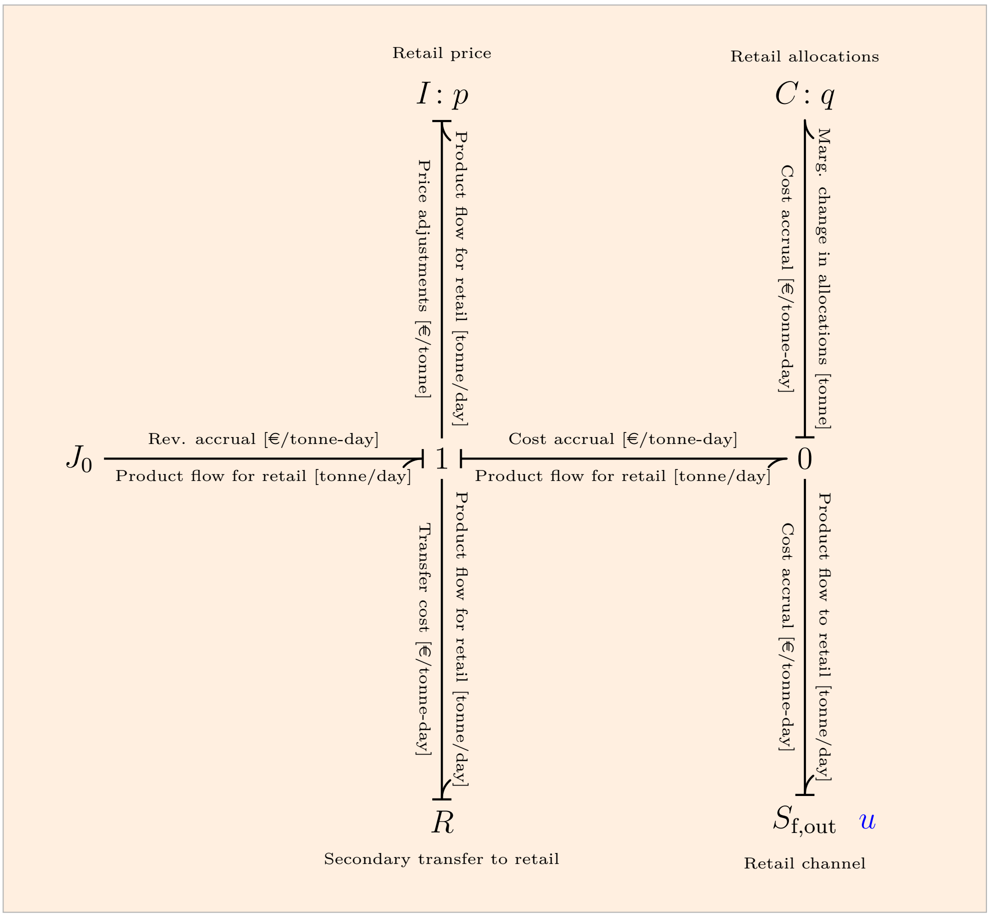


Figure 7-2: Building block for the flow of product to retail stations where the demanded product flow is known.

With the possibility of adding building block in Figure 7-2 in the storage depot model, the amount of product towards sales channels can be used as a control variable. In order to implement a MPC algorithm, it is useful to model the current and future state of the inventory at the sales channel locations. For example, when the state of the inventory at the retail station is known, the MPC algorithm can give priority to retail stations where stocks are expected to run out sooner.

Figure 7-3 shows the concept for an Economic Engineering model for retail stations. The shipment volumes that usually arrive at retail stations by truck are input to the model. The storage of the retail station is modelled with a compliance element (C-element) with the inventory as its state. The fuel station price is stored at the I-element, where the demanded flow by customers is determined. The amount of product that is refueled by consumers is stored in the rightmost C-element from where it is consumed with the resistance element (R-element).

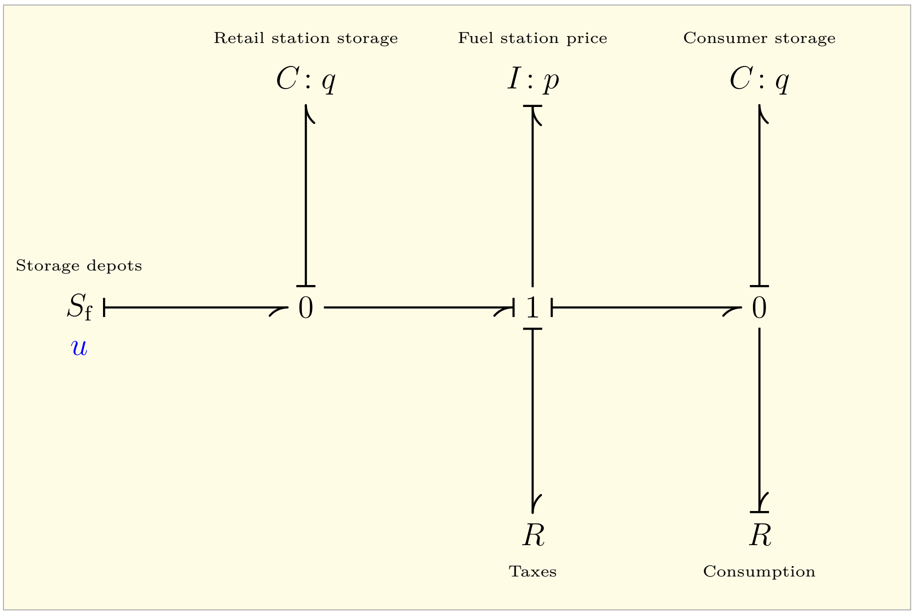


Figure 7-3: Concept for a model for a retail station in the refined oil product supply chain.

7-3 Other Recommendations

Different architectures for the Model Predictive Control algorithm

The scheduling algorithm is developed for one individual storage depot in the supply chain. For the scheduling of shipments, we could extend this to an MPC algorithm that takes into account other storage depots. In this thesis, the MPC algorithm for the storage depot is said to be decentralized, the algorithm does not directly use any information from other storage depots or MPC algorithms in the optimization.

A possible extension is one MPC algorithm for all storage depots together in a centralized architecture. However, for large-scale systems, the centralized approach requires computational efforts that are too large, making it difficult to implement in practice [13]. Another possibility is extending the algorithm to a distributed approach. The advantage of the distributed approach is the trade-off between computational complexity and performance that can be made. In a distributed approach, the MPC includes information for other elements in the supply chain in the decision-making at one storage depot. Both approaches offer the potential for improved decision-making in the scheduling algorithm, we recommend these for further research.

As is already mentioned in Chapter 4, the increase in computational complexity when the set of available shipments increases is a drawback of the developed scheduling algorithm. A trade-off between performance and computational complexity can be achieved by using alternative optimisation methods for the discrete optimisation problems. Search strategies like genetic algorithms or a branch-and-bound algorithm could be suitable candidates that reduce the computational complexity[28, 6]. Further research is recommended to use alternative optimisation methods in the scheduling algorithm.

Model Predictive Control approach for the entire supply chain

The developed theory is also applicable in a setting where all entities in the supply chain have their own MPC algorithm. All production sites, storage depots and sales channels that are part of the supply chain can have their own unique model and MPC algorithm that collaborate to control the supply chain. Figure 7-4, visualizes what this could look like for a three-stage supply chain according to Brown [8].

We recommend to develop models for all stages in the supply chain with Economic Engineering Systems Theory. The dynamics of product flows and transfer prices between the entities can be included in the modelling such that dynamically changing costs and revenues become part of the decision-making. Similar to the approach of Figure 7-4, the setup with models and MPC algorithms for all entities in the supply chain can be custom made dependent on the application and objective. The Economig Engineering (EconE) model and MPC algorithm for storage depots can be used where it may need adjustments. The proposed EconE model in Figure 7-3 can be used as a starting point for the retailer side.

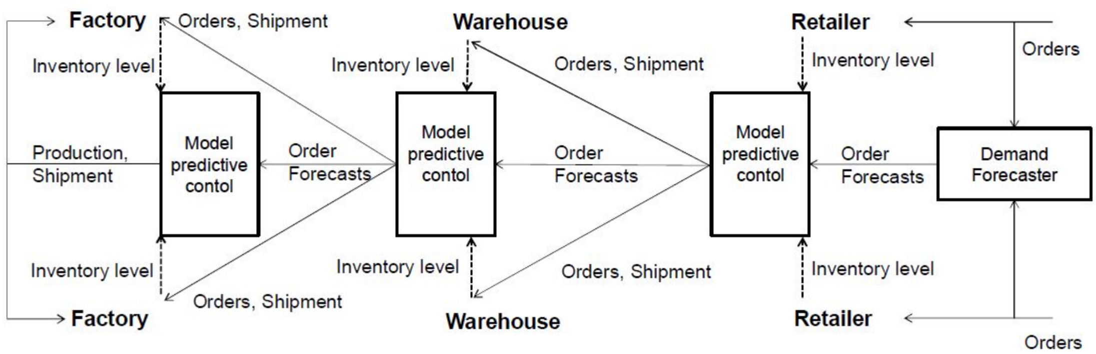


Figure 7-4: MPC approach for a three-stage supply chain according to Brown [8, 21].

Supply-chain management in the frequency domain

This thesis focused on supply-chain management in the time-domain for relatively short-term tasks. Different from day-to-day scheduling on the short-term, planning in supply-chain management is about the more long-term tasks. In the long-term, cyclical behaviour in supply chains is closely related to how inventories are managed [47]. The presence of cycles follow from seasonality, order cycles, inventory replenishment and many other aspects [16].

Cyclical behavior of systems is analyzable in the frequency-domain. We recommend further research into the use of frequency-domain tools for supply chains. The use of frequency-domain tools may be a useful extension to current approaches in supply-chain management. They can be used for performing scenario analysis or defining performance measures [50]. We refer to the research by van Ardenne into the use of frequency-domain tools for business valuations and scenario analysis as starting point.

Application of the supply-chain modelling technique and scheduling algorithm to other supply chains

In this thesis, we applied the developed modelling technique and scheduling algorithm to the refined oil product supply chain. The technology can also be applied to other product supply chains. These supply chains can have the need for modelling the dynamical behaviour of product flows and transfer prices, optimal decision-making and automated processes. Since we developed the modeling technique in a generalized way with building blocks, the models for storage depots in other supply chains can be easily created when identifying the incoming and outgoing product flows.

Finally, we recommend including transfer price data in the system identification process to add more realism. This thesis uses the primary transport data and inventory levels to show the potential of the modelling and scheduling algorithm. By identifying the EconE model with transfer price data of the products in the supply chain, the model could become more accurate for the transfer prices which will also benefit the decision-making.

Supply chains for products other than the refined petroleum products may be more appropriate to include this transfer pricing data. The refined oil product supply chain is a margin business where price margins are different for all sales channels and locations. As a result, the transfer prices are different for all parties in the supply chain which makes it complicated to include the transfer price data. Therefore, we recommend to apply this first to other supply chains with less price differentiation between sales channels.

Appendix A

Economic Engineering Modelling Technique without Transfer Pricing

A-1 Storage Depot Model without Transfer Pricing

Figure A-1 depicts the developed model for modelling the dynamics around a product depot without transfer pricing. The flow of product enters the system from production or other depots as input at $S_{\mathrm{f},1}$ and they go out to throughput depots at $S_{\mathrm{f},2}$. The flow source and sink are the inputs to the model. The output is the state variable q_4, which represents the stock level at the depot. The sum of the flows at the 0-junction is equal to zero and that implies that the flow towards the depot is the difference of in- and outgoing flows.

The building block for the storage depot and sales are the same for both this modelling technique and the modelling technique with transfer pricing. The difference between the modelling techniques are the building blocks with the S_f-elements. In these building blocks, there are no transfer prices and product allocations. The transfer costs with respect to the R_1- and R_2-element do not affect the potential energy that flows in or out the depot due to the connection with the flow source and sink. Hence, these elements do not have any effect on the dynamics of the inventory level in the model.

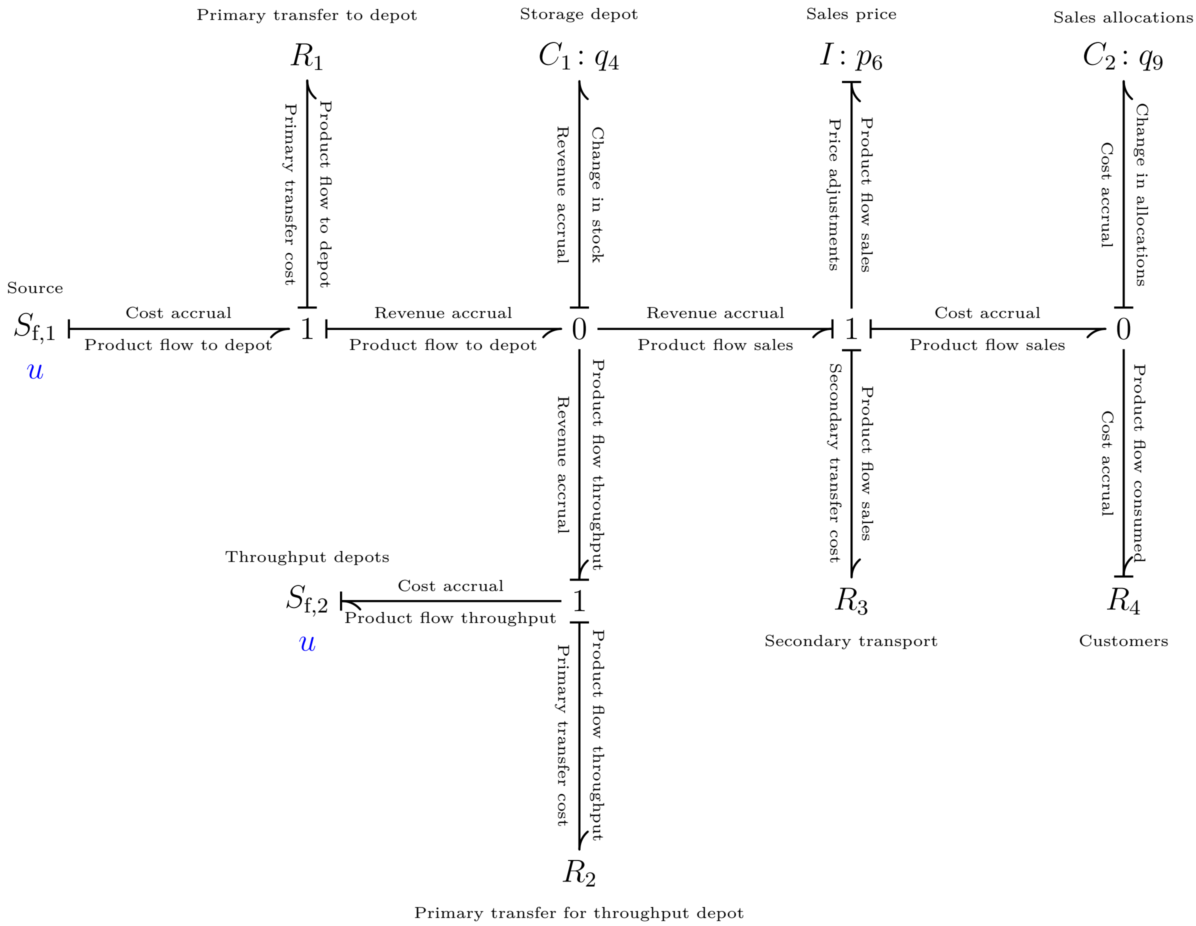


Figure A-1: Economic Engineering model for a storage depot in a supply chain without transfer pricing.

A-2 Derivation State-Space Representation for Economic Engineering Model

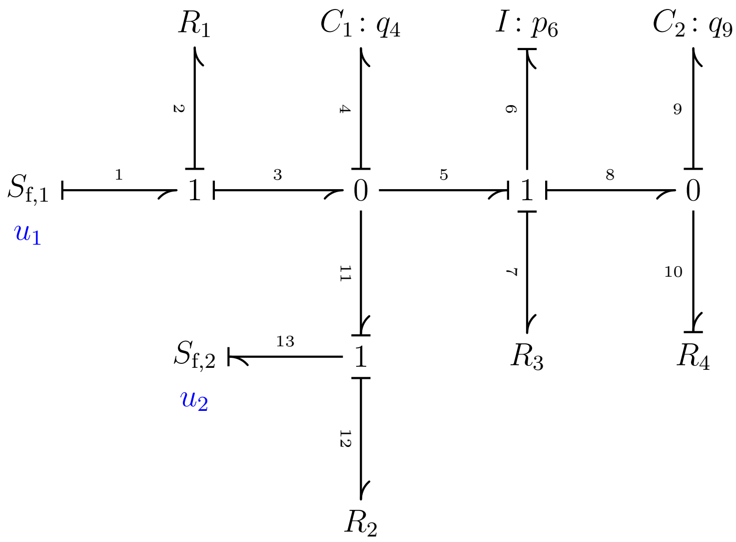


Figure A-2: Numbered version of Economic Engineering model for a storage depot in a supply chain without transfer pricing.

The equations for the flows in the model at the junctions are:

$$
\begin{aligned}
u_1 &= f_1 = f_2 = f_3 \\
f_3 &= \dot{q}_4 + f_5 + f_{11} \\
f_5 &= f_6 = f_7 = f_8 \\
f_8 &= \dot{q}_9 + f_{10} \\
f_{11} &= f_{12} = f_{13} = u_2
\end{aligned}
$$

The equations for the efforts in the model at the junctions are:

$$
\begin{aligned}
e_1 &= e_2 + e_3 \\
e_3 &= e_4 = e_5 = e_{11} \\
e_5 &= \dot{p}_6 + e_7 + e_8 \\
e_8 &= e_9 = e_{10} \\
e_{11} &= e_{12} + e_{13}
\end{aligned}
$$

From the elements, the relationships lead to the following equations:

$$
\begin{aligned}
e_2 &= R_1 \cdot f_2 \\
e_4 &= C_1 \cdot q_4 \\
p_6 &= I \cdot f_6 \\
e_7 &= R_3 \cdot f_7 \\
e_9 &= C_2 \cdot q_9 \\
f_{10} &= R_4 \cdot e_{10} \\
e_{12} &= R_2 \cdot f_{12}
\end{aligned}
$$

Then the state equations become:

$$
\begin{aligned}
\dot{q}_4 &= f_3 - f_5 - f_{11} \\
&= u_1 - \frac{p_6}{I} - u_2
\end{aligned}
$$

$$
\begin{aligned}
\dot{p}_6 &= e_5 - e_7 \\
&= C_1 q_4 - C_2 q_9 - R_3 f_7 \\
&= C_1 q_4 - C_2 q_9 - \frac{R_3}{I} p_6
\end{aligned}
$$

$$
\begin{aligned}
\dot{q}_9 &= f_8 - f_{10} \\
&= \frac{1}{I} p_6 - \frac{1}{R_4} e_{10} \\
&= \frac{1}{I} p_6 - \frac{C_2}{R_4} q_9
\end{aligned}
$$

This results, with q_4 as output of the model, in the state-space representation:

$$
\begin{aligned}
\begin{bmatrix} \dot{q}_4 \\ \dot{p}_6 \\ \dot{q}_9 \end{bmatrix} &= \begin{bmatrix} 0 & -\frac{1}{I} & 0 \\ C_1 & -\frac{R_3}{I} & -C_2 \\ 0 & \frac{1}{I} & -\frac{C_2}{R_4} \end{bmatrix} \begin{bmatrix} q_4 \\ p_6 \\ q_9 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_4 \\ p_6 \\ q_9 \end{bmatrix}
\end{aligned}
$$

Appendix B

Derivation State-Space Representation Economic Engineering Model

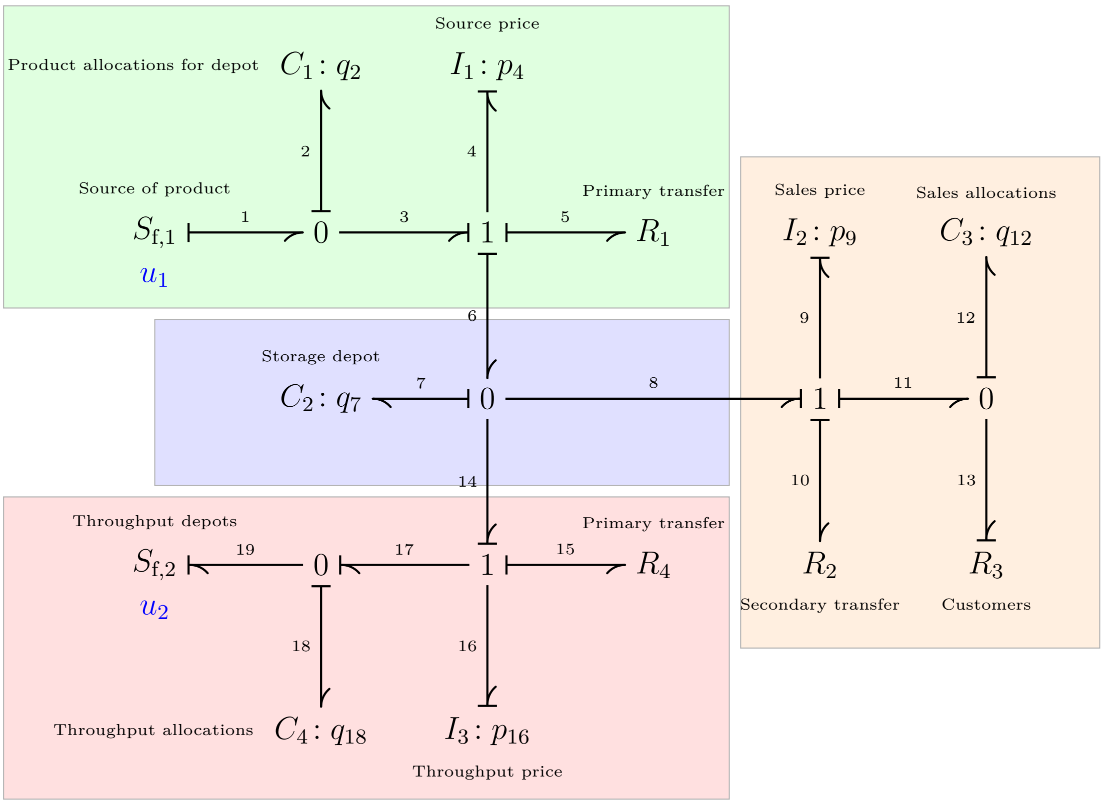


Figure B-1: Bond graph model for a storage depot in the supply chain used for the derivation of the state-space representation.

The equations for the flows f in the model at the junctions are:

$$
\begin{aligned}
u_1 = f_1 &= \dot{q}_2 + f_3 \\
f_3 &= f_4 = f_5 = f_6 \\
f_6 &= \dot{q}_7 + f_8 + f_{14} \\
f_8 &= f_9 = f_{10} = f_{11} \\
f_{11} &= \dot{q}_{12} + f_{13} \\
f_{14} &= f_{15} = f_{16} = f_{17} \\
f_{17} &= \dot{q}_{18} + f_{19} = \dot{q}_{18} + u_2
\end{aligned}
$$

The equations for the efforts e in the model at the junctions are:

$$
\begin{aligned}
e_1 &= e_2 = e_3 \\
e_3 &= \dot{p}_4 + e_5 + e_6 \\
e_6 &= e_7 = e_8 = e_{14} \\
e_8 &= \dot{p}_9 + e_{10} + e_{11} \\
e_{11} &= e_{12} = e_{13} \\
e_{14} &= e_{15} + \dot{p}_{16} + e_{17} \\
e_{17} &= e_{18} = e_{19}
\end{aligned}
$$

From the elements, the relationships lead to the following equations:

$$
\begin{aligned}
e_2 &= C_1 \cdot q_2 \\
p_4 &= I_1 \cdot f_4 \\
e_5 &= R_1 \cdot f_5 \\
e_7 &= C_2 \cdot q_7 \\
p_9 &= I_2 \cdot f_9 \\
e_{10} &= R_2 \cdot f_{10} \\
e_{12} &= C_3 \cdot q_{12} \\
e_{13} &= R_3 \cdot f_{13} \\
e_{15} &= R_4 \cdot f_{15} \\
p_{16} &= I_3 \cdot f_{16} \\
e_{18} &= C_4 \cdot q_{18}
\end{aligned}
$$

Then the state equations become:

$$\begin{aligned}
\dot{q}_2 &= f_1 - f_3 \\
&= u_1 - \frac{p_4}{I_1}
\end{aligned}$$

$$\begin{aligned}
\dot{p}_4 &= e_3 - e_5 - e_6 \\
&= e_2 - R_1 f_5 - e_7 \\
&= C_1 q_2 - R_1 \frac{p_4}{I_1} - C_2 \cdot q_7
\end{aligned}$$

$$\begin{aligned}
\dot{q}_7 &= f_6 - f_8 - f_{14} \\
&= \frac{p_4}{I_1} - \frac{p_9}{I_2} - \frac{p_{16}}{I_3}
\end{aligned}$$

$$\begin{aligned}
\dot{p}_9 &= e_8 - e_{10} - e_{11} \\
&= C_2 q_7 - R_2 f_{10} - C_3 q_{12} \\
&= C_2 q_7 - \frac{R_2}{I_2} p_9 - C_3 q_{12}
\end{aligned}$$

$$\begin{aligned}
\dot{q}_{12} &= f_{11} - f_{13} \\
&= \frac{1}{I_2} p_9 - \frac{1}{R_3} e_{13} \\
&= \frac{1}{I_2} p_9 - \frac{C_3}{R_3} q_{12}
\end{aligned}$$

$$\begin{aligned}
\dot{p}_{16} &= e_{14} - e_{15} - e_{17} \\
&= C_2 q_7 - R_4 f_{15} - C_4 q_{18} \\
&= C_2 q_7 - \frac{R_4}{I_3} p_{16} - C_4 q_{18}
\end{aligned}$$

$$\begin{aligned}
\dot{q}_{18} &= f_{17} - f_{19} \\
&= \frac{p_{16}}{I_3} - u_2
\end{aligned}$$

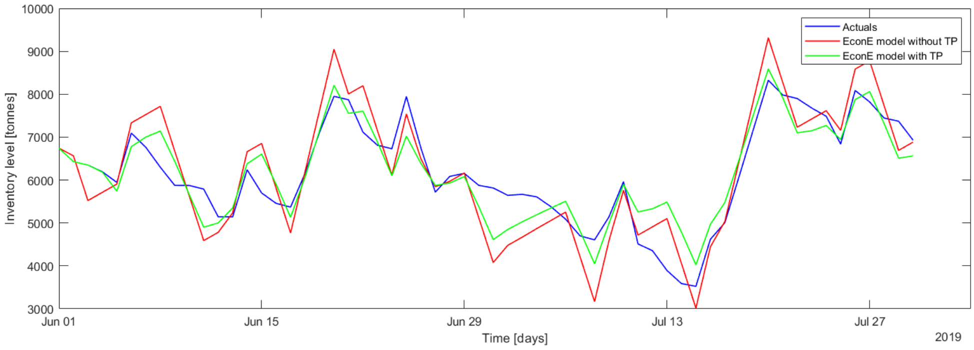
This results, with q_4 as output of the model, in the state-space representation:

$$\begin{bmatrix} \dot{q}_2 \\ \dot{p}_4 \\ \dot{q}_7 \\ \dot{p}_9 \\ \dot{q}_{12} \\ \dot{p}_{16} \\ \dot{q}_{18} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{I_1} & 0 & 0 & 0 & 0 & 0 \\ C_1 & -\frac{R_1}{I_1} & -C_2 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_1} & 0 & -\frac{1}{I_2} & 0 & -\frac{1}{I_3} & 0 \\ 0 & 0 & C_2 & -\frac{R_2}{I_2} & -C_3 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_2} & -\frac{C_3}{R_3} & 0 & 0 \\ 0 & 0 & C_2 & 0 & 0 & -\frac{R_4}{I_3} & -C_4 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_3} & 0 \end{bmatrix} \begin{bmatrix} q_2 \\ p_4 \\ q_7 \\ p_9 \\ q_{12} \\ p_{16} \\ q_{18} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

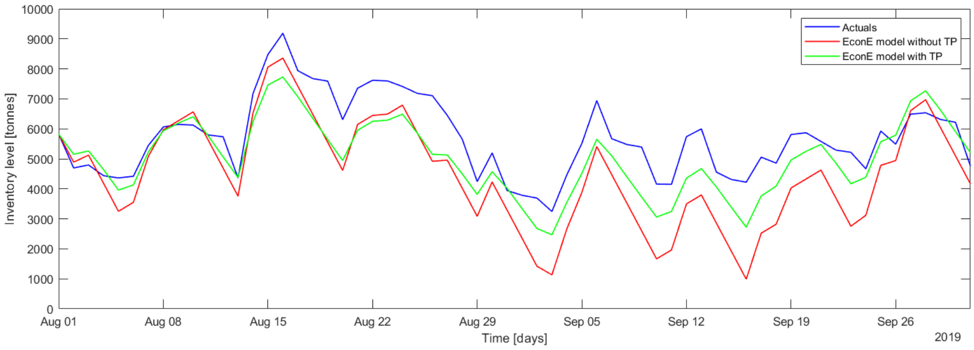
$$y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_2 \\ p_4 \\ q_7 \\ p_9 \\ q_{12} \\ p_{16} \\ q_{18} \end{bmatrix}$$

Appendix C

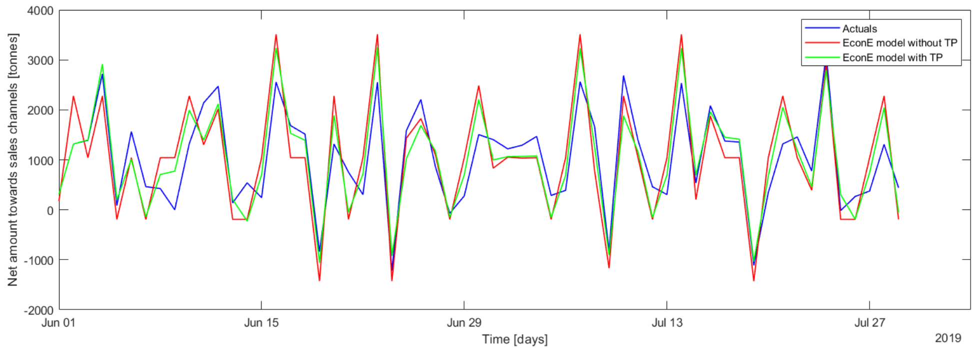
Identification Results Depots DACH Region



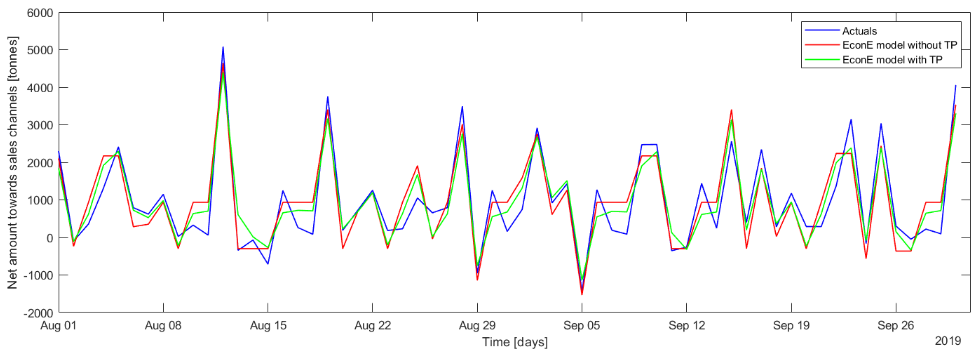
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.

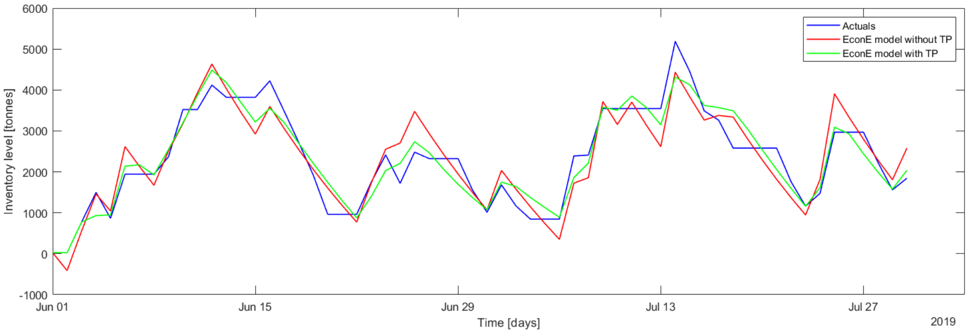


(c) Net flow towards sales channels by EconE models and actuals for identification data.

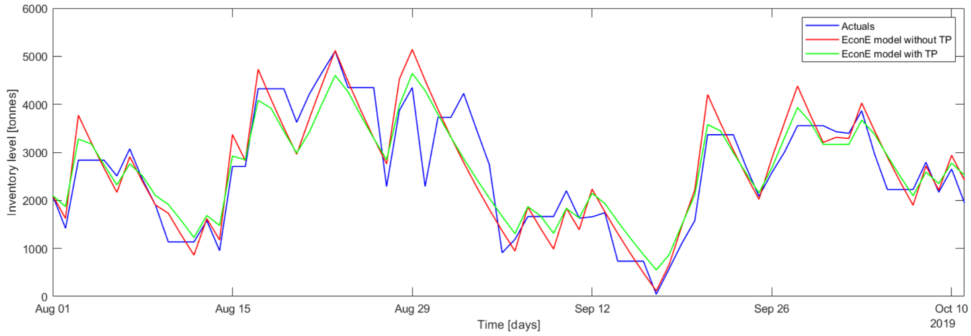


(d) Net flow towards sales channels by EconE models and actuals for validation data.

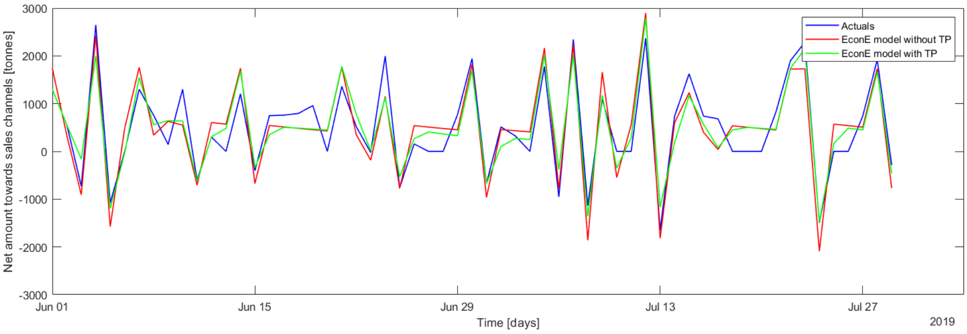
Figure C-1: System identification results for the storage depot at Altmannshofen.



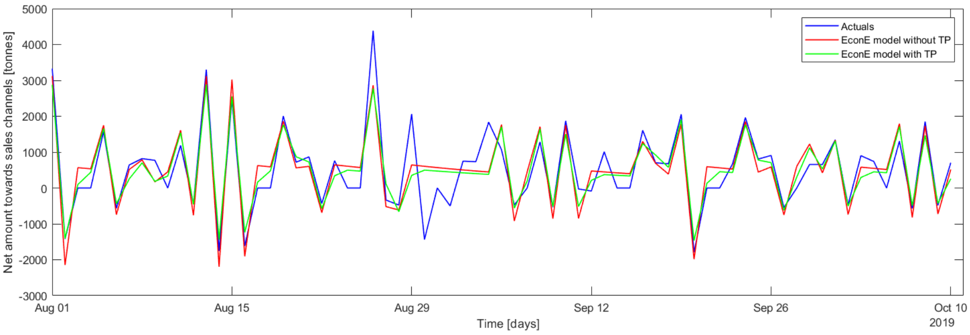
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.

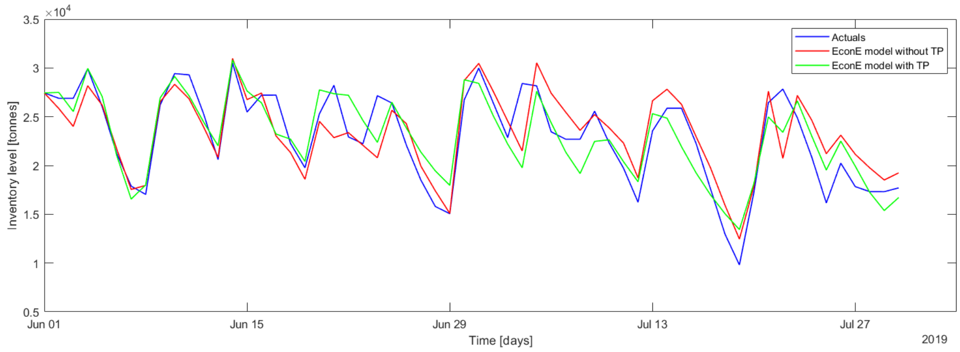


(c) Net flow towards sales channels by EconE models and actuals for identification data.

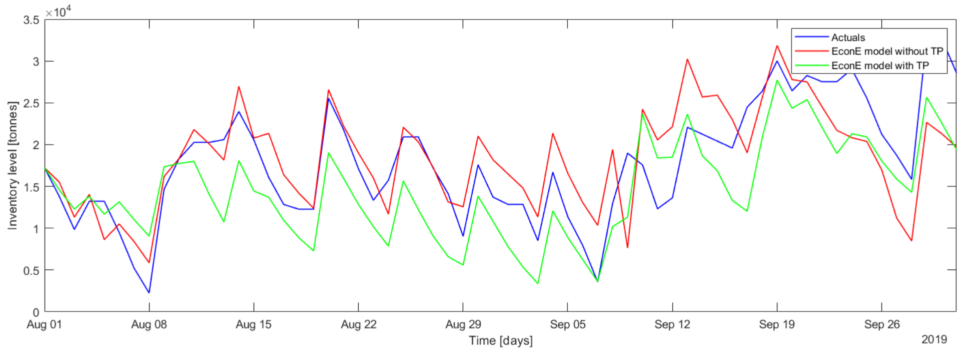


(d) Net flow towards sales channels by EconE models and actuals for validation data.

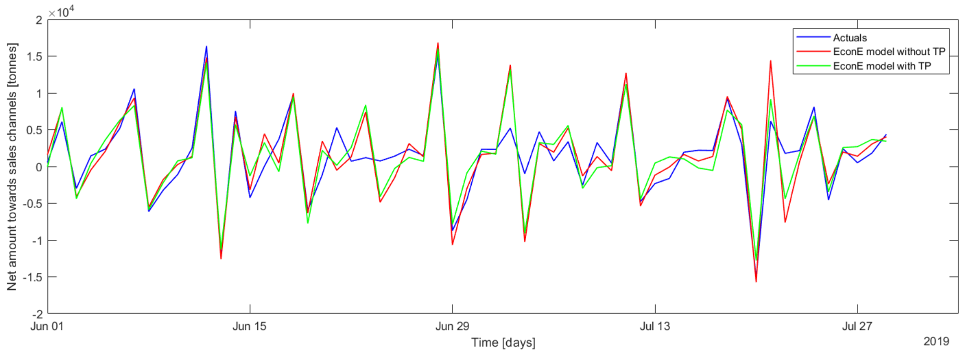
Figure C-2: System identification results for the storage depot at Linz.



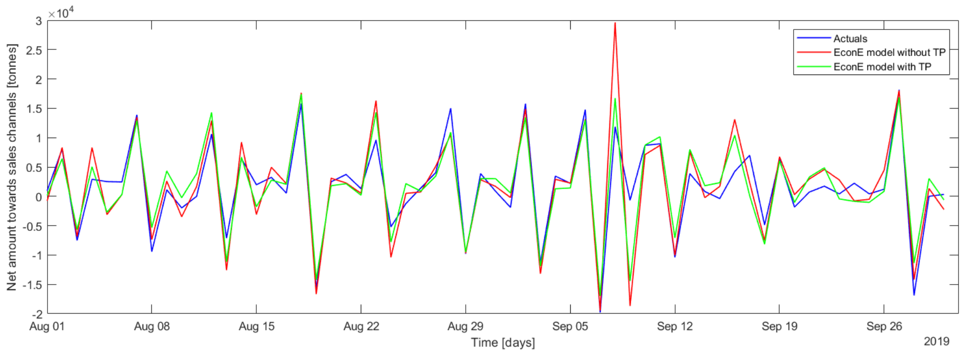
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.

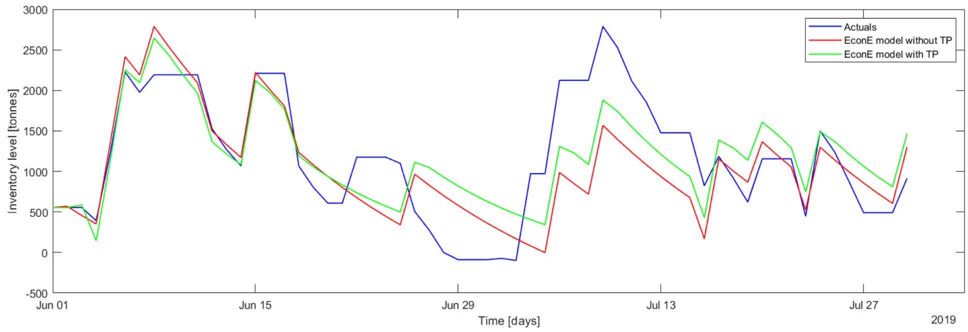


(c) Net flow towards sales channels by EconE models and actuals for identification data.

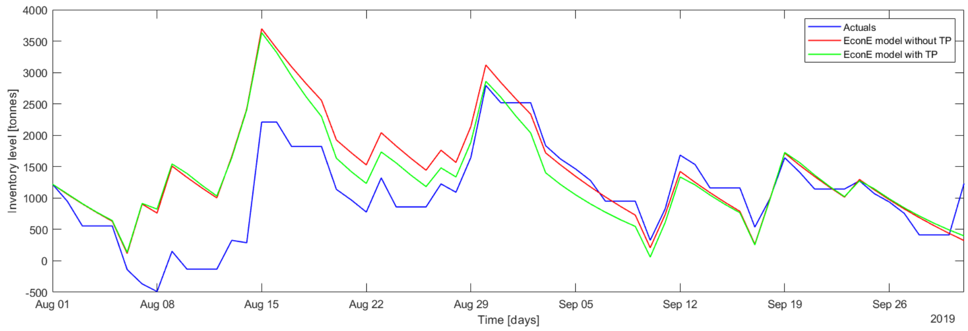


(d) Net flow towards sales channels by EconE models and actuals for validation data.

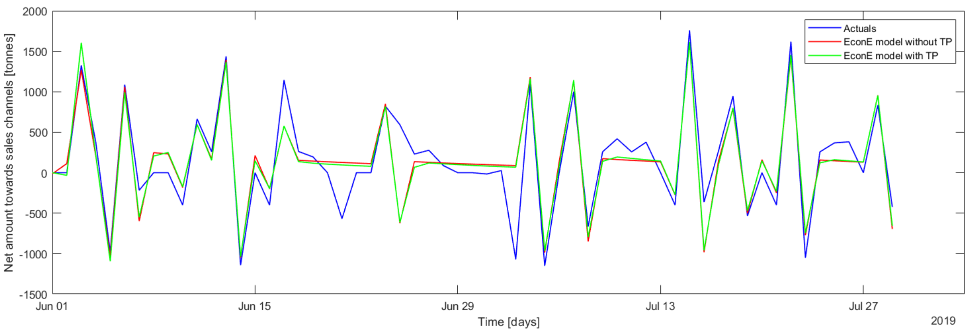
Figure C-3: System identification results for the storage depot at Ludwigshafen.



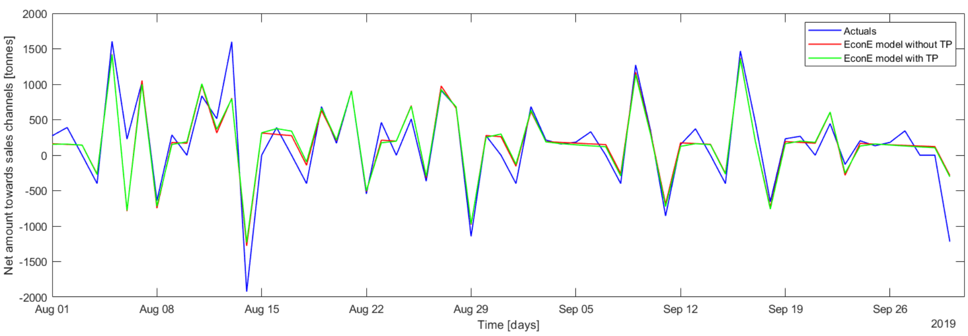
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.

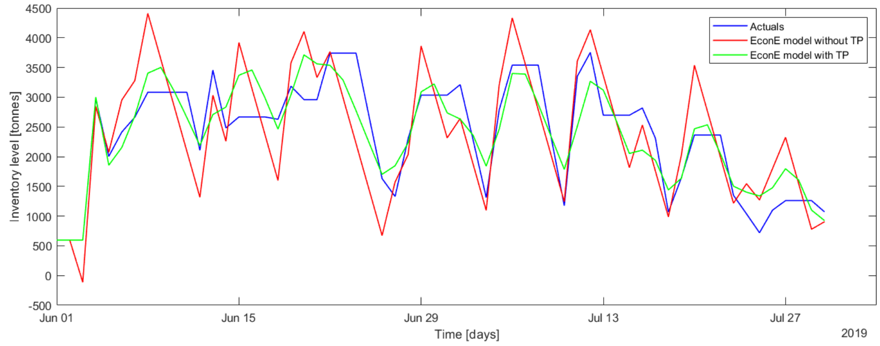


(c) Net flow towards sales channels by EconE models and actuals for identification data.

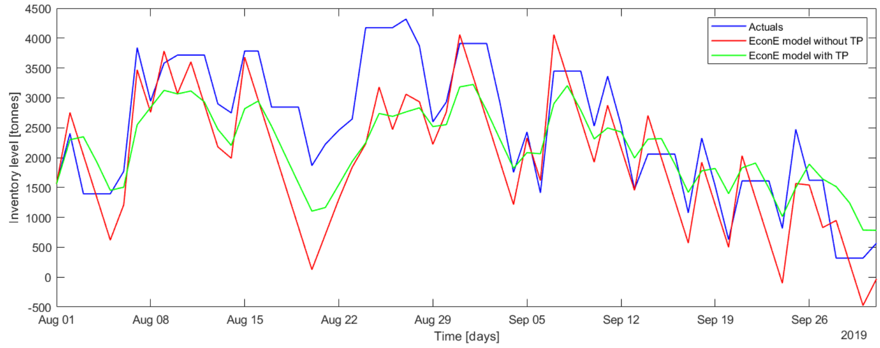


(d) Net flow towards sales channels by EconE models and actuals for validation data.

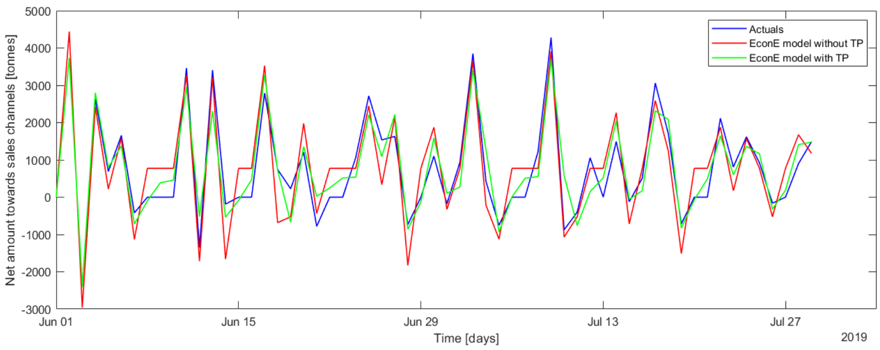
Figure C-4: System identification results for the storage depot at Salzburg.



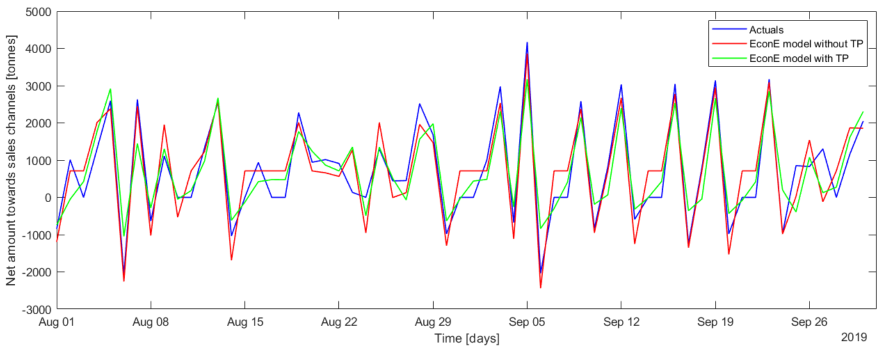
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.

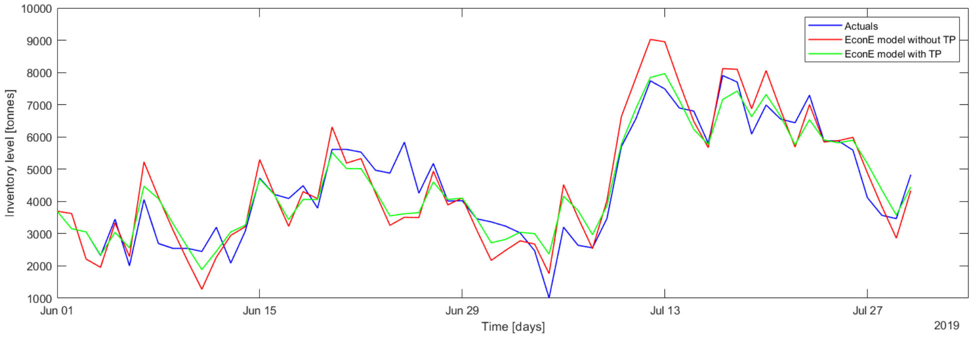


(c) Net flow towards sales channels by EconE models and actuals for identification data.

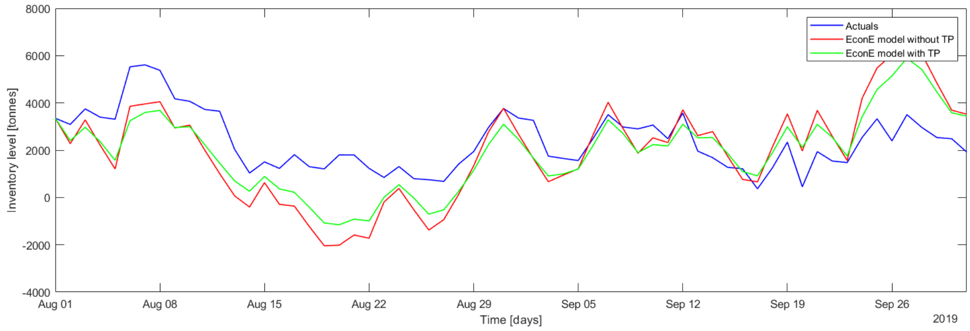


(d) Net flow towards sales channels by EconE models and actuals for validation data.

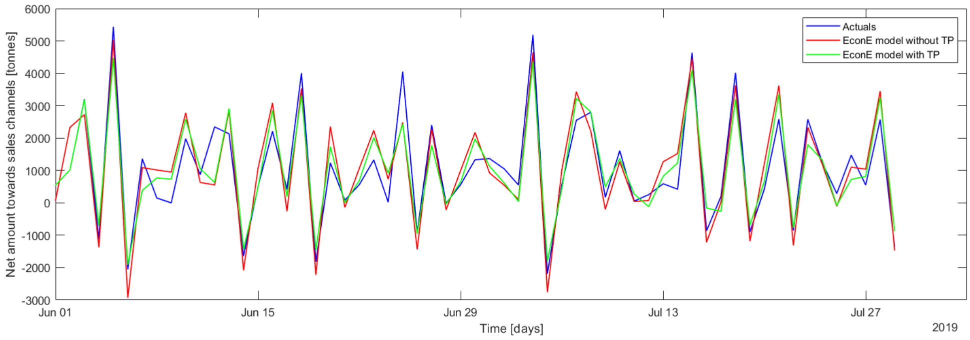
Figure C-5: System identification results for the storage depot at Wien Lobau.



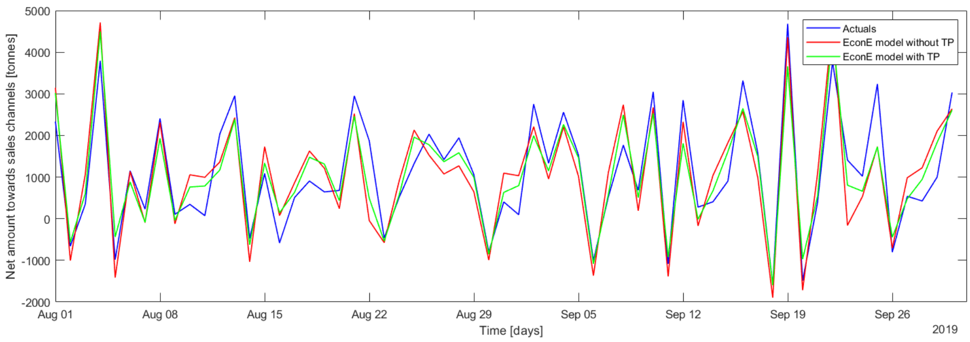
(a) Inventory levels by EconE models and actuals for identification data.



(b) Inventory levels by EconE models and actuals for validation data.



(c) Net flow towards sales channels by EconE models and actuals for identification data.



(d) Net flow towards sales channels by EconE models and actuals for validation data.

Figure C-6: System identification results for the storage depot at Wurzburg.

Appendix D

State Responses of the Storage Depot Model in the System Identification

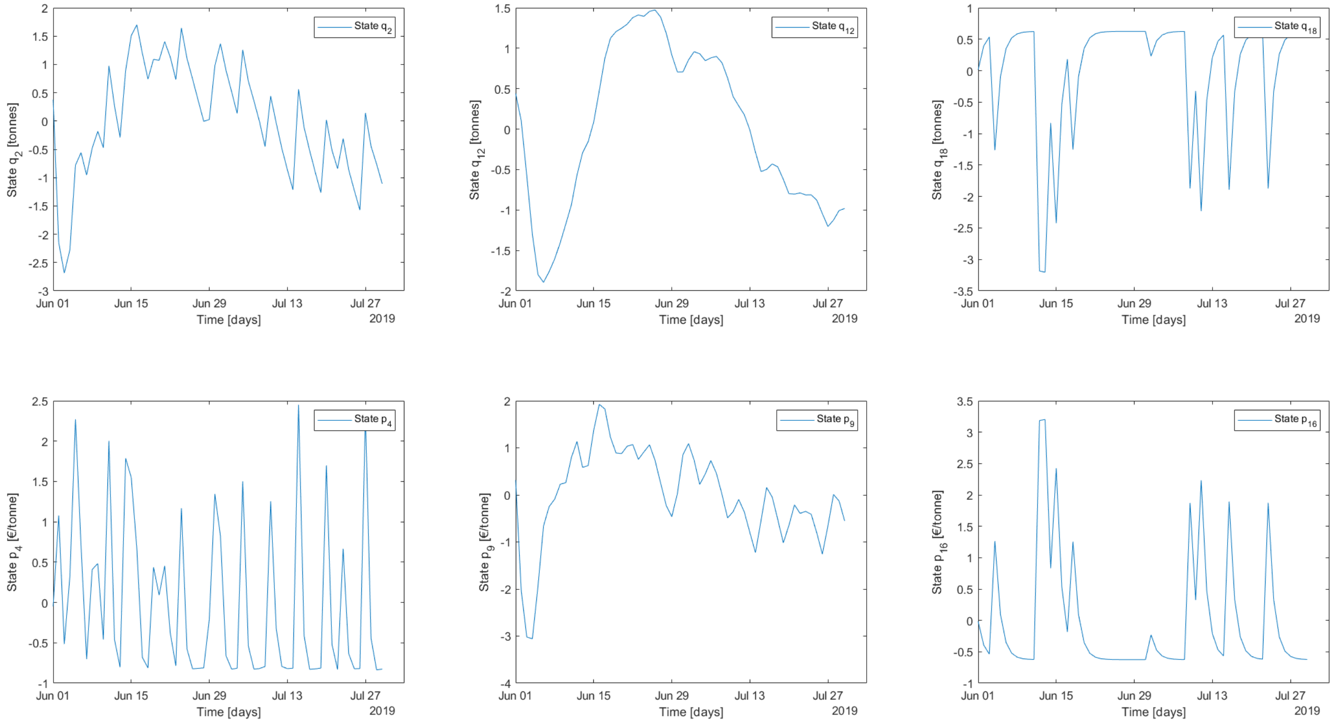


Figure D-1: Normalized state responses of the storage depot model in the identification phase.

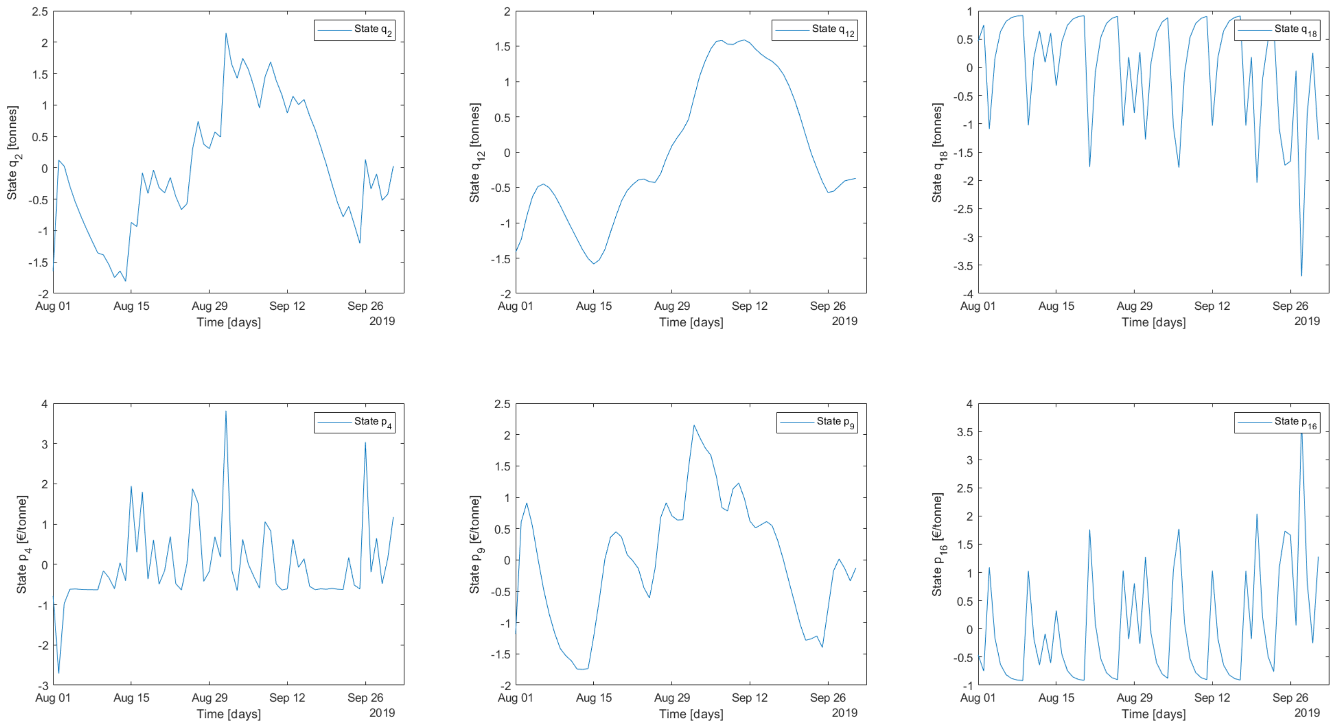


Figure D-2: Normalized state responses of the storage depot model in the validation phase.

Appendix E

Economic Engineering Storage Depot Model with Additional Building Block for Sales Channels

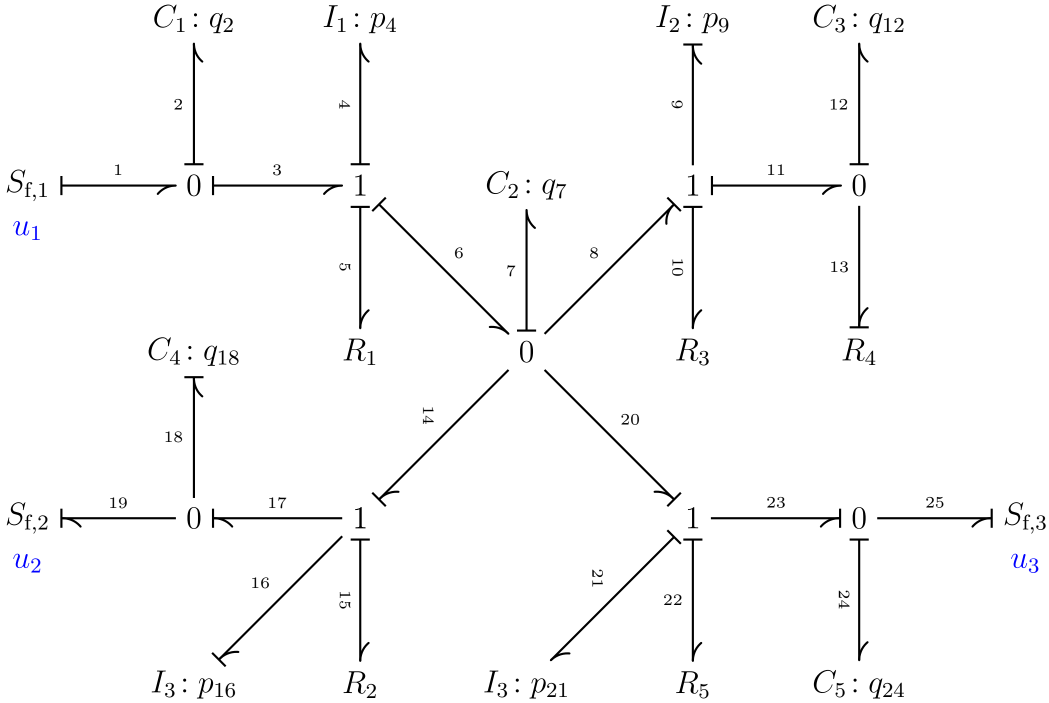


Figure E-1: Bond graph model for a storage depot in the supply chain with an additional building block for known volumes for sales channels.

Using the same steps in the derivation as done in Appendix A an Appendix B, the state-space representation results in:

$$
\begin{bmatrix} \dot{q}_2 \\ \dot{p}_4 \\ \dot{q}_7 \\ \dot{p}_9 \\ \dot{q}_{12} \\ \dot{p}_{16} \\ \dot{q}_{18} \\ \dot{p}_{21} \\ \dot{q}_{24} \end{bmatrix}
=
\begin{bmatrix}
0 & -\frac{1}{I_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C_1 & -\frac{R_1}{I_1} & -C_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{I_1} & 0 & -\frac{1}{I_2} & 0 & -\frac{1}{I_3} & 0 & -\frac{1}{I_4} & 0 \\
0 & 0 & C_2 & -\frac{R_3}{I_2} & -C_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{I_2} & -\frac{C_3}{R_4} & 0 & 0 & 0 & 0 \\
0 & 0 & C_2 & 0 & 0 & -\frac{R_2}{I_3} & -C_4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{I_3} & 0 & 0 & 0 \\
0 & 0 & C_2 & 0 & 0 & 0 & 0 & -\frac{R_5}{I_4} & -C_5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_4} & 0
\end{bmatrix}
\begin{bmatrix} q_2 \\ p_4 \\ q_7 \\ p_9 \\ q_{12} \\ p_{16} \\ q_{18} \\ p_{21} \\ q_{24} \end{bmatrix}
+
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 0 \\
0 & 0 & -1
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}
$$

$$
y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\begin{bmatrix} q_2 \\ p_4 \\ q_7 \\ p_9 \\ q_{12} \\ p_{16} \\ q_{18} \\ p_{21} \\ q_{24} \end{bmatrix}
$$

Appendix F

Matlab Code for Qualitative Analysis Storage Depot Model

```
function [A,B,C,D] = ...
    depot_model_tp(R_1,R_2,R_3,R_4,C_1,C_2,C_3,C_4,I_1,I_2,I_3,¬)

A=[0 -I_1 0 0 0 0 0; ...
    C_1 -R_1*I_1 -C_2 0 0 0 0; ...
    0 I_1 0 -I_2 0 -I_3 0; ...
    0 0 C_2 -R_2*I_2 -C_3 0 0; ...
    0 0 0 I_2 -C_3*R_3 0 0; ...
    0 0 C_2 0 0 -R_4*I_3 -C_4; ...
    0 0 0 0 0 I_3 0];
B=[1 0; 0 0; 0 0; 0 0; 0 0; 0 0; 0 -1];
C=[0 0 1 0 0 0 0];
D=[0 0];

end
```

```
R_1 = 1;
R_2 = 10;
R_3 = 10;
R_4 =0.1;
C_1 =10;
C_2 =0.1;
C_3 =1;
C_4 =10;
I_1   = 10;
```

```
I_2   = 10;
I_3   = 10;

[A,B,C,D] = depot_model_tp(R_1,R_2,R_3,R_4,C_1,C_2,C_3,C_4,I_1,I_2,I_3);
sys = ss(A,B,C,D);
u = zeros(2,50);
u(1,1:3) = 0.5;
u(2,25) = 0.5;
t = linspace(0,length(u)-1,length(u));

[y,¬,x] = lsim(sys,u,t,[0 0 0 0 0 0 0]);

pole(sys)
zero(sys)
figure
pzmap(sys)
```

Appendix G

Matlab Code for Functions System Identification

```
function [A,B,C,D] = ...
    id_depot_model_tp(R_1,R_2,R_3,R_4,C_1,C_2,C_3,C_4,I_1,I_2,I_3,¬)

A=[0 -I_1 0 0 0 0 0; ...
    C_1 -R_1*I_1 -C_2 0 0 0 0; ...
    0 I_1 0 -I_2 0 -I_3 0; ...
    0 0 C_2 -R_2*I_2 -C_3 0 0; ...
    0 0 0 I_2 -C_3*R_3 0 0; ...
    0 0 C_2 0 0 -R_4*I_3 -C_4; ...
    0 0 0 0 0 I_3 0];
B=[1 0; 0 0; 0 0; 0 0; 0 0; 0 0; 0 -1];
C=[0 0 1 0 0 0 0];
D=[0 0];

end
```

```
function [results, par, data] = model_identification_tp(par,location)

%% Data
filename = append('data_table_',location,'_AGO.mat');
load(filename);

% Take identification data
data_table_id = data_table(data_table.timedate(par.('timerange_id')),:);

```

```
% Setup identification data for time period
data.u_id = [double(data_table_id.q_depot_in) ...
    double(data_table_id.q_depot_out)];
data.t_id = linspace(0,length(data.u_id)-1,length(data.u_id))';
data.y_id = data_table_id.stocklevel;

%% Prepare identification data

% Fill missing data
data_table_id.stocklevel = ...
    fillmissing(data_table_id.stocklevel,'linear','SamplePoints',data.t_id);
data_table_id.Price = ...
    fillmissing(data_table_id.Price,'linear','SamplePoints',data.t_id);

% Take trend in data, mean for polyfit(¬,¬,0)
q_supply_in_id=polyfit(data.t_id,double(data_table_id.q_depot_in),0);
q_supply_out_id=polyfit(data.t_id,double(data_table_id.q_depot_out),0);
q_stock_id=polyfit(data.t_id,double(data_table_id.stocklevel),0);

supply_in_trend_id=polyval(q_supply_in_id,data.t_id);
supply_out_trend_id=polyval(q_supply_out_id,data.t_id);
data.stock_trend_id=polyval(q_stock_id,data.t_id);

% Detrended identification data
supply_in_detrend_id=double(data_table_id.q_depot_in)-supply_in_trend_id;
supply_out_detrend_id=double(data_table_id.q_depot_out)-supply_out_trend_id;
stock_detrend_id=double(data_table_id.stocklevel)-data.stock_trend_id;

% Identification input-output data
data.u_id = [supply_in_detrend_id supply_out_detrend_id];
data.y_id = stock_detrend_id;

data_id=iddata(data.y_id,data.u_id,1);

%% Identification

% Call function and initialize greybox identification
odefun='id_depot_model_tp';
parameters={'transport',par.R_1;'transport',par.R_2;'transport',par.R_3;...
    'consumption',par.R_4;'inventory_1',par.C_1;'inventory_2',par.C_2;...
    'inventory_3',par.C_3;'inventory_4',par.C_4;'elasticity1',par.I_1;...
    'elasticity2',par.I_2;'elasticity3',par.I_3};

fcn_type='c';
Ts = 0;
init_sys=idgrey(odefun,parameters,fcn_type);

% Set greyest options
opt_gb = greyestOptions;
opt_gb.SearchMethod = 'auto';
```

```
opt_gb.SearchOptions.MaxIterations = 500;
opt_gb.SearchOptions.Advanced.MaxFunctionEvaluations = 500;
opt_gb.InitialState = 'backcast';

% Identify model
id_model = greyest(data_id,init_sys,opt_gb);

% Simulate with identification data
x0_id = [0;0; stock_detrend_id(1); 0;0;0;0];
[data.y_id_model, ¬, data.x_model] =lsim(id_model,data.u_id,data.t_id, x0_id);

% VAF Identification calculation
VAF_id = var(data.y_id - data.y_id_model) / var(data.y_id) ;
results.VAF_id = 100 * ( 1 - VAF_id );

%% Validation data
data_table_val = data_table(data_table.timedate(par.timerange_val),:);
data.t_val = linspace(0,length(data_table_val.stocklevel)-1,...
                                    length(data_table_val.stocklevel))';

% Fill missing data
data_table_val.stocklevel = ...
    fillmissing(data_table_val.stocklevel,'linear','SamplePoints',data.t_val);
data_table_val.Price = ...
    fillmissing(data_table_val.Price,'linear','SamplePoints',data.t_val);

% Prepare data
q_supply_in_val = polyfit(data.t_val,double(data_table_val.q_depot_in),0);
q_supply_out_val = polyfit(data.t_val,double(data_table_val.q_depot_out),0);
q_stock_val = polyfit(data.t_val,double(data_table_val.stocklevel),0);

data.supply_in_trend_val = polyval(q_supply_in_val,data.t_val);
data.supply_out_trend_val = polyval(q_supply_out_val,data.t_val);
data.stock_trend_val = polyval(q_stock_val,data.t_val);

supply_in_detrend_val = ...
    double(data_table_val.q_depot_in)-data.supply_in_trend_val;
supply_out_detrend_val = ...
    double(data_table_val.q_depot_out)-data.supply_out_trend_val;
stock_detrend_val = double(data_table_val.stocklevel)-data.stock_trend_val;

% Validation input-output data
data.u_val = [supply_in_detrend_val supply_out_detrend_val];
data.y_val = stock_detrend_val;

%% Validation

% Simulation
```

```
x0_val = [data.x_model(end,1); data.x_model(end,2); stock_detrend_val(1); ...
    data.x_model(end,4); data.x_model(end,5);data.x_model(end,6); ...
    data.x_model(end,7)];
[data.y_val_model, ¬, data.x_val_model] = ...
    lsim(id_model,data.u_val,data.t_val, x0_val);

% VAF
VAF_val = var(data.y_val - data.y_val_model) / var(data.y_val);
results.VAF_val = 100 * ( 1 - VAF_val );

id_par = id_model.Report.Parameters.ParVector;
par.R_1 = id_par(1);par.R_2 =id_par(2);par.R_3 = id_par(3);
par.R_4 =id_par(4);par.C_1 =id_par(5);par.C_2 =id_par(6);
par.C_3   = id_par(7);par.C_4   = id_par(8);par.I_1   = id_par(9);
par.I_2   = id_par(10);par.I_3   = id_par(11);

end
```

Appendix H

Matlab Code for Functions Model Predictive Control Algorithm

```
function [myStruct] = MPC_algorithm_tp(myStruct,loc,routes,par)

%% -------- MAIN FILE MPC --------- %%
for idx = 1:length(loc)
    i = char(loc{idx});

    % Initialize lifting counter
    count_shipments.in =  ...
        [myStruct.(i).planning_MPC_0.pipeline_in(1:par.T_sim+par.N_p+2,:)';...
         myStruct.(i).planning_MPC_0.barge_in(1:par.T_sim+par.N_p+2,:)';...
         myStruct.(i).planning_MPC_0.train_in(1:par.T_sim+par.N_p+2,:)'];
    count_shipments.in(:,r) = 0;

    count_shipments.out = ...
        [myStruct.(i).planning_MPC_0.pipeline_out(1:par.T_sim+par.N_p+2,:)';...
         myStruct.(i).planning_MPC_0.barge_out(1:par.T_sim+par.N_p+2,:)';...
         myStruct.(i).planning_MPC_0.train_out(1:par.T_sim+par.N_p+2,:)'];
    count_shipments.out(:,r) = 0;
    myStruct.(i).count_shipments_con = count_shipments;

    % Model output arrays
    myStruct.(i).x_con_model = zeros(par.T_sim*7+7,par.N_p+1);
    myStruct.(i).y_con_model = zeros(par.T_sim*2+2,par.N_p+1);
    myStruct.(i).t = linspace(0,par.T_sim+par.N_p,par.T_sim+par.N_p+1);

    % Set initial planning for MPC Simulation
    myStruct.(i).u_con = myStruct.(i).data_large.u_val';
```

```
    r = linspace(10,45,10);    r2 = linspace(10,45,10);
    myStruct.(i).u_con(1,r) = 0;    myStruct.(i).u_con(2,r2) = 0;
    myStruct.(i).u_con(:,r) = ...
        [-myStruct.(i).data_large.supply_in_trend_val(r)' ; ...
        -myStruct.(i).data_large.supply_out_trend_val(r)' ];

    %Initial conditions
    myStruct.(i).x_con_model(1:7,2) = ...
        [myStruct.(i).data_large.x_val_model(1,:)];
end

% Start simulation
for j=1:par.T_sim
     for idx = 1:length(loc)
        i = loc{idx};
        i = char(i);

        % Update state and inputs
        x0_con_new = myStruct.(i).x_con_model(7*j-6:7*j,2);
        u_con_planning = myStruct.(i).u_con(:,j:j+par.N_p);

        c_time = mod(j+4,5);
        if c_time == 0
            % Determine optimal planning
            [u_con_opt_extra, I_max] = ...
                optimal_u_determination_tp(x0_con_new,u_con_planning,...
            myStruct.(i),routes.(i),j);
            % Write extra operations in planning
            myStruct.(i).u_con(:,j:j+par.N_p) = ...
                myStruct.(i).u_con(:,j:j+par.N_p) + u_con_opt_extra;
            myStruct.(i) = update_planning_and_count(myStruct.(i), ...
                routes.(i), I_max, j);
        end
        % Update states
        [myStruct.(i).y_con_model(j,:), ...
            myStruct.(i).x_con_model(7*j+1:7*j+7,:)] = ...
            update_function_tp(x0_con_new,u_con_planning,myStruct.(i).par_large);
     end
end

for idx = 1:length(loc)
    i = loc{idx};
    i = char(i);

    % Simulate model for full simulation time
    x0_sim = [myStruct.(i).data_large.x_val_model(1,:)];
    myStruct = model_simulation_tp_mpc(x0_sim,myStruct,i);
end
end
```

```
function [u_opt_extra, I_max] = ...
    optimal_u_determination_tp(x0,u,myStruct_i,routes_i,con,ix)
% This function determines the optimal set of extra operations that should
% be added to the operational planning.

% Iterate over all possible extra shipments in (out)
for i=1:length(routes_i.all_in)
    % Count shipments on day t over prediction horizon with possible new
    % shipments i'th option
    [cnt, counter_in, counter_out] = count_shipments_Np(myStruct_i, ...
        routes_i, ix, i);

    u_ex = [routes_i.all_in(i,1:myStruct_i.par.N_p+1); ...
        routes_i.all_out(i,1:myStruct_i.par.N_p+1)];
    u_tot = u + u_ex;
    profit(i) = profit_calculator(x0,u_tot,con,myStruct_i,cnt);
end

% Determine optimal shipments where profit is maximized
[maxp, I_max] = max(profit)
u_opt_extra = [routes_i.all_in(I_max,1:myStruct_i.par.N_p+1); ...
    routes_i.all_out(I_max,1:myStruct_i.par.N_p+1);];
end
```

```
function [profit] = profit_calculator(x0,u,myStruct_i)

% Initialize state space
[A,B,C,D] = depot_model_tp(myStruct_i.par_large);

% Simulate
sys = ss(A,B,C,D);
t = linspace(0,myStruct_i.par.N_p,myStruct_i.par.N_p+1);
[y_p,¬,x_p] = lsim(sys,u,t,x0);

y_p = y_p';
x_p = x_p;

% Capacity constraints
x_cap = myStruct_i.max_cap - myStruct_i.data_large.stock_trend_val(1);
x_caplow = 0.2*myStruct_i.max_cap - myStruct_i.data_large.stock_trend_val(1);

% Set profit
profit = 0;

% Calculate profit for N_p
for i=1:myStruct_i.par.N_p
    if x_p(i+1,3) > x_cap
        penalty = 99e50;
```

```
    elseif x_p(i+1,3) < x_caplow
        penalty = 99e50;
    else
        penalty = 0;
    end


     dq2 = x_p(i+1,1)-x_p(i,1);
     dq7 = y_p(1,i+1)-y_p(1,i);
     dq18 = x_p(i+1,7)-x_p(i,7);

     u1 = u(1,i+1)+myStruct_i.data.supply_in_trend_val(1);
     u2 = u(2,i+1)+myStruct_i.data.supply_out_trend_val(1);
     q7 = y_p(1,i+1)+myStruct_i.data.stock_trend_val(i+1);

    profit = profit + I_2*x_p(i+1,4)^2 + x_p(i+1,6) * (u2-dq18) ...
        - x_p(i+1,2) * (u1-dq2) ...
        - (u1-dq2)*R_1*I_1*x_p(i+1,2)...
        - R_3*I_2^2*x_p(i+1,4)^2 ...
        - penalty;
end

end
```

```
%% This function generates U_ex based on the parameters for the input ...
    constraints. It determines U_ex for the different storage depots.

t_pipe = [14];
t_barge = [7:8];
t_train = [10];

qty_pipe = 5000;
qty_barge = 2000;
qty_train = 1200;

loc = {'LUD','FLO','WRZ','WLB','LNZ','ALT','SAB'};
load('routes.mat')
load('parameters.mat')


pipe_options = permn([0 qty_pipe],size(t_pipe,2));
nnzInRow1 = sum(pipe_options ≠ 0, 2);
deleteRow1 = zeros(size(pipe_options,1),1);
for j=1:size(pipe_options,1)
    if nnzInRow1(j) > 2
        deleteRow1(j) = j;
    end
end
```

```
deleteRow1(deleteRow1==0) = [];
pipe_options(deleteRow1,:) = [];
pipe_counter = zeros(size(pipe_options));
for j=1:size(pipe_options,1)
    pipe_counter(j,:) = double(any(pipe_options(j,:),1));
end


barge_options = permn([0 qty_barge],size(t_barge,2));
nnzInRow2 = sum(barge_options ≠ 0, 2);
deleteRow2 = zeros(size(barge_options,1),1);
for j=1:size(barge_options,1)
    if nnzInRow2(j) > 3
        deleteRow2(j) = j;
    end
end
deleteRow2(deleteRow2==0) = [];
barge_options(deleteRow2,:) = [];
barge_counter = zeros(size(barge_options));
for j=1:size(barge_options,1)
    barge_counter(j,:) = double(any(barge_options(j,:),1));
end

train_options = permn([0 qty_train],size(t_train,2));
nnzInRow3 = sum(train_options ≠ 0, 2);
deleteRow3 = zeros(size(train_options,1),1);
for j=1:size(train_options,1)
    if nnzInRow3(j) > 2
        deleteRow3(j) = j;
    end
end
deleteRow3(deleteRow3==0) = [];
train_options(deleteRow3,:) = [];
train_counter = zeros(size(train_options));
for j=1:size(train_options,1)
    train_counter(j,:) = double(any(train_options(j,:),1));
end

 for idx = 1:length(loc)
    iloc = loc{idx};
    if isfield(routes.(iloc),'PI') && isfield(routes.(iloc),'RC') && ...
        isfield(routes.(iloc),'BA')
        alloptions = zeros(size(pipe_options,1)*size(train_options,1)*
                        size(barge_options,1),par.N_p+1);
        pipe_liftings1 = zeros(size(alloptions,1),size(pipe_options,2));
        barge_liftings1 = zeros(size(alloptions,1),size(barge_options,2));
        train_liftings1 = zeros(size(alloptions,1),size(train_options,2));

        ix = 1;
        for i=1:size(pipe_options,1)
```

```
                for j=1:size(barge_options,1)
                    for k=1:size(train_options,1)
                        alloptions(ix,t_pipe) = alloptions(ix,t_pipe)+ ...
                            pipe_options(i,:);
                        pipe_liftings1(ix,:) = pipe_counter(i,:);
                        alloptions(ix,t_barge) = alloptions(ix,t_barge)+ ...
                            barge_options(j,:);
                        barge_liftings1(ix,:) = barge_counter(j,:);
                        alloptions(ix,t_train) = alloptions(ix,t_train)+ ...
                            train_options(k,:);
                        train_liftings1(ix,:) = train_counter(k,:);
                        ix= ix+1;
                    end
                end
            end

            alloptions_out = ...
                zeros(size(train_options,1)*size(barge_options,1),par.N_p+1);
            ix2 = 1;
            for j=1:size(barge_options,1)
                for k=1:size(train_options,1)
                    alloptions_out(ix2,t_barge) = alloptions_out(ix2,t_barge)+ ...
                        barge_options(j,:);
                    alloptions_out(ix2,t_train) = alloptions_out(ix2,t_train)+ ...
                        train_options(k,:);
                    ix2= ix2+1;
                end
            end

            routes.(iloc).all_in = ...
                zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);
            routes.(iloc).all_out = ...
                zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);
            routes.(iloc).extra_liftings.pipe_liftings = ...
                zeros(size(routes.(iloc).all_in,1),size(pipe_options,2));
            routes.(iloc).extra_liftings.barge_liftings = ...
                zeros(size(routes.(iloc).all_in,1),size(barge_options,2));
            routes.(iloc).extra_liftings.train_liftings = ...
                zeros(size(routes.(iloc).all_in,1),size(train_options,2));

            routes.(iloc).extra_liftings.barge_liftings_out = ...
                zeros(size(routes.(iloc).all_out,1),size(barge_options,2));
            routes.(iloc).extra_liftings.train_liftings_out = ...
                zeros(size(routes.(iloc).all_out,1),size(train_options,2));

            ix3 = 1;
            for j=1:size(alloptions,1)
                for k=1:size(alloptions_out,1)
                    routes.(iloc).all_in(ix3,:) = alloptions(j,:);
```

```
                    routes.(iloc).extra_liftings.pipe_liftings(ix3,:) = ...
                        pipe_liftings1(j,:);
                    routes.(iloc).extra_liftings.barge_liftings(ix3,:) = ...
                        barge_liftings1(j,:);
                    routes.(iloc).extra_liftings.train_liftings(ix3,:) = ...
                        train_liftings1(j,:);
                    routes.(iloc).all_out(ix3,:) = alloptions_out(k,:);
                    routes.(iloc).extra_liftings.barge_liftings_out(ix3,:) = ...
                        barge_liftings1(k,:);
                    routes.(iloc).extra_liftings.train_liftings_out(ix3,:) = ...
                        train_liftings1(k,:);
                    ix3= ix3+1;
                end
            end

            routes.(iloc).extra_liftings.t_pipe = t_pipe;
            routes.(iloc).extra_liftings.t_barge = t_barge;
            routes.(iloc).extra_liftings.t_train = t_train;

        elseif isfield(routes.(iloc),'PI') && isfield(routes.(iloc),'RC') && ...
            (isfield(routes.(iloc),'BA')==0)
            alloptions = ...
                zeros(size(pipe_options,1)*size(train_options,1),par.N_p+1);
            pipe_liftings1 = zeros(size(alloptions,1),size(pipe_options,2));
            train_liftings1 = zeros(size(alloptions,1),size(train_options,2));

            ix = 1;
            for i=1:size(pipe_options,1)
                for k=1:size(train_options,1)
                    alloptions(ix,t_pipe) = alloptions(ix,t_pipe)+ ...
                        pipe_options(i,:);
                    pipe_liftings1(ix,:) = pipe_counter(i,:);
                    alloptions(ix,t_train) = alloptions(ix,t_train)+ ...
                        train_options(k,:);
                    train_liftings1(ix,:) = train_counter(k,:);
                    ix= ix+1;
                end
            end

            alloptions_out = zeros(size(train_options,1),par.N_p+1);
            ix2 = 1;
            for k=1:size(train_options,1)
                alloptions_out(ix2,t_train) = alloptions_out(ix2,t_train)+ ...
                    train_options(k,:);
                ix2= ix2+1;
            end

            routes.(iloc).all_in = ...
                zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);
```

```
        routes.(iloc).all_out = ...
            zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);
        routes.(iloc).extra_liftings.pipe_liftings = ...
            zeros(size(routes.(iloc).all_in,1),size(pipe_options,2));
        routes.(iloc).extra_liftings.train_liftings = ...
            zeros(size(routes.(iloc).all_in,1),size(train_options,2));
        routes.(iloc).extra_liftings.train_liftings_out = ...
            zeros(size(routes.(iloc).all_out,1),size(train_options,2));

        ix3 = 1;
        for j=1:size(alloptions,1)
            for k=1:size(alloptions_out,1)
                routes.(iloc).all_in(ix3,:) = alloptions(j,:);
                routes.(iloc).extra_liftings.pipe_liftings(ix3,:) = ...
                    pipe_liftings1(j,:);
                routes.(iloc).extra_liftings.train_liftings(ix3,:) = ...
                    train_liftings1(j,:);
                routes.(iloc).all_out(ix3,:) = alloptions_out(k,:);
                routes.(iloc).extra_liftings.train_liftings_out(ix3,:) = ...
                    train_liftings1(k,:);
                ix3= ix3+1;
            end
        end

        routes.(iloc).extra_liftings.t_pipe = t_pipe;
        routes.(iloc).extra_liftings.t_train = t_train;

    elseif (isfield(routes.(iloc),'PI')==0) && isfield(routes.(iloc),'RC') ...
        && isfield(routes.(iloc),'BT')
        alloptions = ...
            zeros(size(train_options,1)*size(barge_options,1),par.N_p+1);
        barge_liftings1 = zeros(size(alloptions,1),size(barge_options,2));
        train_liftings1 = zeros(size(alloptions,1),size(train_options,2));

        ix = 1;
            for j=1:size(barge_options,1)
                for k=1:size(train_options,1)
                    alloptions(ix,t_barge) = alloptions(ix,t_barge)+ ...
                        barge_options(j,:);
                    barge_liftings1(ix,:) = barge_counter(j,:);
                    alloptions(ix,t_train) = alloptions(ix,t_train)+ ...
                        train_options(k,:);
                    train_liftings1(ix,:) = train_counter(k,:);
                    ix= ix+1;
                end
            end

        alloptions_out = ...
            zeros(size(train_options,1)*size(barge_options,1),par.N_p+1);
        ix2 = 1;
```

```
        for j=1:size(barge_options,1)
            for k=1:size(train_options,1)
                alloptions_out(ix2,t_barge) = alloptions_out(ix2,t_barge)+ ...
                    barge_options(j,:);
                alloptions_out(ix2,t_train) = alloptions_out(ix2,t_train)+ ...
                    train_options(k,:);
                ix2= ix2+1;
            end
        end

        routes.(iloc).all_in = ...
            zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);
        routes.(iloc).all_out = ...
            zeros(size(alloptions_out,1)*size(alloptions,1),par.N_p+1);

        routes.(iloc).extra_liftings.barge_liftings = ...
            zeros(size(routes.(iloc).all_in,1),size(barge_options,2));
        routes.(iloc).extra_liftings.train_liftings = ...
            zeros(size(routes.(iloc).all_in,1),size(train_options,2));

        routes.(iloc).extra_liftings.barge_liftings_out = ...
            zeros(size(routes.(iloc).all_out,1),size(barge_options,2));
        routes.(iloc).extra_liftings.train_liftings_out = ...
            zeros(size(routes.(iloc).all_out,1),size(train_options,2));

        ix3 = 1;
        for j=1:size(alloptions,1)
            for k=1:size(alloptions_out,1)
                routes.(iloc).all_in(ix3,:) = alloptions(j,:);
                routes.(iloc).extra_liftings.barge_liftings(ix3,:) = ...
                    barge_liftings1(j,:);
                routes.(iloc).extra_liftings.train_liftings(ix3,:) = ...
                    train_liftings1(j,:);
                routes.(iloc).all_out(ix3,:) = alloptions_out(k,:);
                routes.(iloc).extra_liftings.barge_liftings_out(ix3,:) = ...
                    barge_liftings1(k,:);
                routes.(iloc).extra_liftings.train_liftings_out(ix3,:) = ...
                    train_liftings1(k,:);
                ix3= ix3+1;
            end
        end
        routes.(iloc).extra_liftings.t_pipe = t_pipe;
        routes.(iloc).extra_liftings.t_barge = t_barge;
        routes.(iloc).extra_liftings.t_train = t_train;
    end
end

save('routes.mat','routes')
```

Bibliography

[1] M. Agarwal, R. Sharma, and L. Mathew. Challenges in supply chain management in upstream sector of oil and gas industry. *Agro Supply Chain Conference (ASCC) 2016*, 2016.

[2] A. Alawneh, M. Alrefaei, A. Diabat, R. Al-Aomar, and M.N. Faisal. An LP model for optimizing a supply chain management system for steel company. *Engineering and Computer Science*, 2210, 2014.

[3] F. Allgöwer, R. Findeisen, and Z.K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35:299–315, 2004.

[4] M. Arnold, R.R. Negenborn, Göran Andersson, and Bart De Schutter. Multi-area predictive control for combined electricity and natural gas systems. *2009 European Control Conference, ECC 2009*, 2015.

[5] A. Arreola-Risa and G. DeCroix. Make-to-order versus make-to-stock in a production-inventory system with general production times. *IIE Transactions*, 30:705–713, 1998.

[6] J. Bard. *Engineering Optimization: Theory and Practice, Third Edition*, volume 29. John Wiley & Sons, Inc, 1997.

[7] S. Benghida and D. Benghida. Facts from the contango situation of gas and oil markets. *International Journal of Civil Engineering and Technology*, 9, 2018.

[8] M. Braun, D. Rivera, M. Flores, M. Carlyle, and K. Kempf. A model predictive control framework for robust management of multi-product multi-echelon demand networks. *Annual Reviews in Control*, 2003.

[9] E. F. Camacho and C. Bordons. *Model Predictive control.* Springer London, 2007.

[10] European Commission. *EU oil stocks.* 2020.

[11] D. L. Feucht. *Handbook of Analog Circuit Design.* Elsevier Science, 2014.

[12] G. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control Of Dynamic Systems.* Prentice Hall, 1994.

[13] J. R. D. Frejo and E. F. Camacho. Global versus local mpc algorithms in freeway traffic control with ramp metering and variable speed limits. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1556–1565, 2012.

[14] J. Gjerdrum, N. Shah, and L. Papageorgiou. Transfer prices for multienterprise supply chain optimization. *Industrial & Engineering Chemistry Research*, 40, 2001.

[15] A. Halley, J. Nollet, M. Beaulieu, J. Roy, and Y. Bigras. The impact of the supply chain on core competencies and knowledge management: Directions for future research. *International Journal of Technology Management*, 49, 2010.

[16] M. Hanus. *Customer order cycle of a production company, its bottlenecks and potential for improvements.* Master's thesis, Wageningen University, 2015.

[17] J.B. Hooper. Natural gas and refined products. *Analytical Chemistry (Washington)*, 67, 1995.

[18] M. Huhns, L. Stephens, and N. Ivezic. Automating supply-chain management. *The First International Joint Conference on Autonomous Agents & Multiagent Systems*, pages 1017–1024, 2002.

[19] H. Brian Hwarng and Na Xie. Understanding supply chain dynamics: A chaos perspective. *European Journal of Operational Research*, 184(3):1163 – 1178, 2008.

[20] A.C. Inkpen and M.H. Moffett. *The Global Oil & Gas Industry: Management, Strategy & Finance.* PennWell, 2011.

[21] D. Ivanov, S. Sethi, A. Dolgui, and B. Sokolov. A survey on control theory applications to operational systems, supply chain management, and industry 4.0. *Annual Reviews in Control*, 46, 2018.

[22] T. Jacobs. How oil innovation has benefited other industries. *Journal of Petroleum Technology*, 71:40–42, 2019.

[23] L. Jezuita. *Improving the overall customer service level: a case study at Philips.* Master's thesis, University of Twente, 2017.

[24] M. Joly. Refinery production planning and scheduling: The refining core business. *Brazilian Journal of Chemical Engineering*, 29(2):371–384, 2012.

[25] D. Karnopp, D. Margolis, and R. Rosenberg. *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems: Fifth Edition.* John Wiley & Sons, Inc, 2012.

[26] E. Koca, M. Sedaghat, and K.P. Yoon. Optimal supply & demand balance in service environments. *Journal of Service Science (JSS)*, 7:43, 2014.

[27] P. Kotler. Marketing during periods of shortage. *Journal of Marketing*, 38(3):20–29, 1974.

[28] E. Lawler and D. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.

[29] P. T. Liu. *Dynamic Optimization and Mathematical Economics.* Mathematical Concepts and Methods in Science and Engineering. Springer US, 2013.

[30] J. Maciejowski. *Predictive Control With Constraints.* Prentice Hall., England, 2002.

[31] M. Manfred and H. L. Jay. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667 – 682, 1999.

[32] MathWorks. *MATLAB Release 2019a.* The MathWorks, Inc., 2019.

[33] M.B. Mendel. *Principles of Economic Engineering.* Lecture Notes, Delft University of Technology, 2019.

[34] Microsoft Power BI Community. *Need help with a measure stock cycle.* 2019.

[35] J. Miller. *Volume and Variability in Demand Segmentation.* 2017.

[36] T. Miller and R. de Matta. *Profit Maximization Modeling for Supply Chain Planning*, pages 1910 – 1921. 2014.

[37] S. A. Mirhassani. An operational planning model for petroleum products logistics under uncertainty. *Applied Mathematics and Computation*, 196:744–751, 2008.

[38] K. Nelson, M. Barth, and D. Cram. Accruals and the prediction of future cash flows. *The Accounting Review*, 76, 2001.

[39] N. G. Orie. *Dynamic modelling and control of the oil market: an Economic-Engineering approach.* Master's thesis, Delft University of Technology, 2020.

[40] C. Ptak and C. Smith. *Demand Driven Material Requirements Planning (DDMRP).* Industrial Press Inc., U.S., 2016.

[41] J. Qin and T. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11:733–764, 2003.

[42] Royal Dutch Shell PLC. *Shell Investors' Handbook 2012-2016.* 2016.

[43] B. Sahay. Supply chain collaboration: The key to value creation. *Work Study*, 52:76–83, 2003.

[44] H. Sahebi, S. Nickel, and J. Ashayeri. Strategic and tactical mathematical programming models within the crude oil supply chain context - a review. *Computers and Chemical Engineering*, 68, 2014.

[45] D. Stratiev, R. Dinkov, Petkov K, and K. Stanulov. Evaluation of crude oil quality. *Petroleum and Coal*, 52, 2010.

[46] The Engineering Concepts. *Petroleum Refinery & Its Products.* 2020.

[47] M. Udenio. *Inventory dynamics and the bullwhip effect : studies in supply chain performance.* PhD thesis, Industrial Engineering & Innovation Sciences, 2014.

[48] U.S. Energy Information Administration. *Low Rhine River water levels disrupt petroleum product shipments to parts of Europe.* 2018.

[49] U.S. Energy Information Administration. *Short-Term Energy Outlook Data Browser.* 2020.

[50] X. A. van Ardenne. *Business Valuation in the Frequency Domain: A Dynamical Systems Approach.* Master's thesis, Delft University of Technology, 2020.

[51] M. van der Heijden and E. Diks. *Verdeel en heers: voorraadallocatie in distributienetwerken.* Kluwer Bedrijfswetenschappen, 1999.

[52] S. van Houten. *The price and volume dynamics of inland Rhine shipping in relation to climate induced water level changes.* Master's thesis, Erasmus University Rotterdam, 2019.

[53] M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach.* Cambridge University Press, 2007.

[54] T.F. Wallace and R.A. Stahl. *Sales and Operations Planning The Executive Guide.* Steelwedge Software, 2006.

[55] B. Wang, Y. Liang, T. Zheng, M. Yuan, and H. Zhang. Optimisation of a downstream oil supply chain with new pipeline route planning. *Chemical Engineering Research and Design*, 145:300 – 313, 2019.

Glossary

List of Acronyms

AGO	Automotive Gasoil
C**-element**	compliance element
DACH	Germany, Austria and Switzerland
EconE	Economig Engineering
I**-element**	inertia element
IOCs	Integrated Oil Companies
GUI	Graphical User Interface
MP	Mathematical Programming
MPC	Model Predictive Control
MTO	Make-to-order
MTS	Make-to-stock
R**-element**	resistance element
TP	Transfer Pricing
VAF	Variance Accounted For