Model Based Control of Reefer Container Systems

Ph.D. Dissertation

Kresten Kjær Sørensen

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7C
DK-9220 Aalborg

Sørensen, Kresten Kjær
Model Based Control of Reefer Container Systems
ISBN: 978-87-7152-063-7
Second Edition, February 2015

Lodam electronics
Kærvej 77
6400 Sønderborg
Denmark

Department of Electronic Systems
Aalborg University
Fredrik Bajers Vej 7C
9220 Aalborg Ø
Denmark

Typeset in LaTeX using the PhD thesis template made by Jesper Kjær Nielsen,
http://kom.aau.dk/~jkn/latex/latex.php.

Thesis Details

Thesis Title: Model Based Control of Reefer Container Systems
Ph.D. Student: M.Sc., Kresten Kjær Sørensen, Lodam Electronics
Supervisors: Prof. Ph.D. Jakob Stoustrup, Aalborg University
Prof. Ph.D. Thomas Bak, Aalborg University
M.Sc., Ph.D., Morten Juel Skovrup, IPU
M.Sc., Lars Mou Jessen, Lodam Electronics

The main body of this thesis consists of the following papers.

[A] Kresten K. Sørensen and Jakob Stoustrup, "Modular Modeling and Simulation Approach - Applied to Refrigeration Systems," *Proceedings of the 2008 IEEE Multiconference on Systems and Control* , pp. 983-988, doi:10.1109/CCA.2008.4629691.

[B] Kresten K. Sørensen, Jens D. Nielsen and Jakob Stoustrup, "Modular Simulation of Reefer Container Dynamics," *SIMULATION March 2014 vol. 90 no. 3, pp. 249-264*, doi:10.1177/0037549713515542.

[C] Kresten K. Sørensen, Morten Juel Skovrup, Lars M. Jessen, Jakob Stoustrup, "Modular Modeling of a Refrigerated Container," In Press, *International Journal of Refrigeration*, doi:10.1016/j.ijrefrig.2015.03.017.

[D] Kresten K. Sørensen, Jakob Stoustrup and Thomas Bak, "Adaptive MPC for a Refrigerated Container," In Press, *Control Engineering Practice*, doi:10.1016/j.conengprac.2015.05.012.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty of Engineering and Science at Aalborg Unversity. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Abstract

This thesis is concerned with the development of model based control for the Star Cool refrigerated container (reefer), with the objective of reducing energy consumption. The system has been available since 2005 and is currently the most energy efficient reefer available and with more than 150000 units in service worldwide and a yearly production that constitute one third of the total number of reefers produced. Traditionally reefers are governed by decentralized PID controllers with very little mutual coordination, leading to less than optimal control. This project has been carried out under the Danish Industrial PhD program and has been financed by Lodam together with the Danish Ministry of Science, Technology and Innovation. The main contributions in this thesis are on the subjects of modeling, simulation and control of a reefer, and experimental model validation.

A modular nonlinear simulation model is developed using a control oriented approach where accurately modeled dynamics of the metal in the heat exchangers are combined with steady state equations for the refrigerant circuit, resulting in a model that matches the states that are important for control very well, both statically and dynamically.

Different options for efficient simulation of the model are investigated using a modular simulation environment, developed to run in MATLAB®, showing that for modular simulation of this class of systems, the MATLAB® `ode15s` solver is slower but not more accurate than a modified variable step forward Euler solver. The difference between monolithic and modular simulation of the model is also investigated, revealing a large difference in speed when the `ode15s` solver is used and no significant difference when the variable step forward Euler solver is used.

A control structure consisting of a linearizing inner loop controller and an energy optimizing outer loop controller is presented. The outer loop model predictive controller saves energy through adaptation to daily variations in ambient temperature and a ventilation rate that is reduced to fit the actual demand. A parameter estimator is used to determine the latent variables of the cargo through measurements on the return and supply air temperatures, and the result is used to continuously update the model predictive controller such that it is able to adapt to different types of cargo. The controller is verified using the simulation model and energy savings of up to 21.9% are found when

both the adaptation to varying ambient temperatures and the reduced ventilation rate is applied.

Resumé

Denne afhandling omhandler udvikling af modelbaseret regulering til en Star Cool kølecontainer med henblik på at reducere energiforbruget. Containeren har været i produktion siden 2005 og er den mest effektive kølecontainer på markedet, med mere end 150000 enheder i drift over hele verden og med en årlig produktion der udgør en tredjedel af verdensmarkedet. Traditionelt foregår regulering af kølekontainere med decentrale PID regulatorer med ingen eller meget lidt indbyrdes koordinering, hvilket resulterer i en suboptimal regulering as systemet.

Projektet er blevet gennemført under det danske Erhvervs-PhD program og er blevet financeret af Lodam sammen med Ministeriet for Forskning, Innovation og Videregående Uddannelse.

De væsentligste bidrag i denne afhandling omhandler modellering, simulering og regulering af en kølecontainer, samt eksperimentel validering modellen.

Der er udviklet en modulær, ulineær simulerings model med henblik på model baseret regulering, hvor nøjagtigt modellerede dynamikker af metallet i varmevekslerne er kombineret med steady state ligninger for kølekredsen. Den resulterende model opnår god statisk og dynamisk nøjagtighed, for de states der er vigtige for test og udvikling af regulatorer til systemet.

Forskellige muligheder for effektiv simulering af modellen undersøges ved hjælp af et modulært simuleringsmiljø udviklet til at køre i Matlab® og resultaterne viser at for modulær simulering af denne klasse af systemer er Matlab®'s `ode15s` solver langsommere, men ikke mere præcis end en forward euler solver med en forbedret metode til at vælge skridtlængde. Forskellen mellem monolitisk og modulær simulering af modellen undersøges og viser en stor forskel i hastighed når `ode15s` solveren anvendes og ingen signifikant forskel når forward Euler solveren anvendes.

En regulator, der består af en lineariserende regulator i den indre løkke og en energioptimerende regulator i den ydre løkke præsenteres. Den ydre løkke består af en model prædiktiv controller som sparer energi gennem tilpasning til daglige variationer i omgivelsestemperatur og en reduceret ventilation af lasten i containeren, der er tilpasset det aktuelle behov. En parameter estimator anvendes til at estimere de latente variable for lasten gennem målinger af retur og tilluftstemperaturer og resultatet bruges til løbende at opdatere den model prædiktive regulator således at den er i stand til at tilpasse sig forskellige lastttyper. Regulatoren verificeres ved hjælp af simuleringsmodellen og en energibesparelse på op til 21,9% opnås.

Contents

Thesis Details iii

Abstract v

Resumé vii

Preface xv

I Introduction and Summary 1

1 Introduction 3

1 Background and Motivation . . . 3

2 The Reefer Container . . . 4

3 State of the Art and Related Work . . . 7

3.1 Reefer Systems . . . 7

3.2 Modeling of Refrigeration Systems . . . 8

3.3 Simulation of Refrigeration Systems . . . 11

3.4 Control of Refrigeration Systems . . . 13

4 Project Objectives . . . 16

5 Summary of Contributions . . . 18

5.1 Reefer Model . . . 18

5.2 Simulation . . . 18

5.3 Adaptive Model Predictive Control for a Reefer . . . 18

2 Summary of Work 21

1 Reefer Container Modeling . . . 21

1.1 Objectives . . . 21

1.2 Control Oriented Modeling . . . 22

1.3 Identification of Model Parameters . . . 25

1.4 Verification and Results . . . 26
1.5 Contributions . . . 31
2 Simulation . . . 32
2.1 Objectives . . . 32
2.2 Simulation Environment . . . 32
2.3 Results . . . 41
2.4 Contributions . . . 43
3 Energy Optimizing Control . . . 44
3.1 Objectives . . . 44
3.2 Controller Design . . . 44
3.3 Parameter Estimation . . . 48
3.4 Results . . . 50
3.5 Contributions . . . 53

3 Conclusion 55
1 Discussion . . . 55
2 Perspective and Future Work . . . 57
2.1 Perspective . . . 57
2.2 Future Work . . . 58
References . . . 60

II Papers 67

A Modular Modeling and Simulation Approach - Applied to Refrigeration Systems 69
1 Introduction . . . 71
2 Modelling . . . 73
2.1 Component Model Syntax . . . 73
2.2 Simulation Model Syntax . . . 75
3 Simulation . . . 77
4 Reefer Model . . . 78
4.1 Pipe Joining Junction . . . 79
4.2 Compressor . . . 80
4.3 Expansion Valve . . . 80
4.4 Economizer . . . 80
4.5 Evaporator . . . 80
4.6 Condenser . . . 81
4.7 Receiver . . . 81
4.8 Box . . . 81
5 Results . . . 81
5.1 Simulation Speed . . . 81

6 Discussion and Future Work . . . 83
6.1 Discussion . . . 83
6.2 Future Work . . . 84
7 Aknowledgements . . . 84
References . . . 84

B Modular Simulation of Reefer Container Dynamics 87
1 Introduction . . . 89
2 Methods . . . 93
2.1 Refrigeration System Model . . . 93
2.2 Modelling . . . 95
2.3 Simulation . . . 99
2.4 Experiments . . . 106
3 Results . . . 108
4 Conclusion . . . 113
References . . . 114

C Modular Modeling of a Refrigerated Container 117
1 Introduction . . . 120
2 Modeling . . . 121
2.1 Pipe Joining Junction . . . 123
2.2 Pipe Splitting Junction . . . 124
2.3 Compressor . . . 124
2.4 Condenser . . . 125
2.5 Receiver . . . 127
2.6 Expansion Valve . . . 130
2.7 Economizer . . . 130
2.8 Evaporator . . . 132
2.9 Box . . . 136
3 Results . . . 137
4 Conclusion and Future Work . . . 142
4.1 Conclusion . . . 142
4.2 Future Work . . . 142
5 Acknowledgment . . . 142
References . . . 142

D Adaptive MPC for a Refrigerated Container 147
1 Introduction . . . 150
2 Methods . . . 152
2.1 Refrigeration System Simulation Model . . . 152
2.2 Parameter and State Estimation . . . 155
2.3 Controller Setup . . . 161

3 Results 168
4 Conclusion 171
References 171

Todo list

Preface

This thesis is submitted as a collection of papers in partial fulfillment of the requirements for a Doctor of Philosophy at the Section of Automation and Control, Department of Electronic Systems, Aalborg University, Denmark. The work presented in this thesis has been supported by the Danish Ministry of Science, Technology and Innovation under the Industrial PhD program. The work was carried out in the period from January 2007 to December 2013 at Lodam electronics and at the Section of Automation and Control, Aalborg University.

I would like to thank my supervisors Professor Jakob Stoustrup, Morten Juel Skovrup and Lars Mou Jessen for their support, expert knowledge and guidance that have been invaluable to the project.

I would also like to thank all of my colleagues at Lodam for the support and help they have provided that removed all the practical obstacles regarding the Star Cool reefer. A special thanks to Peter Rosenbeck for continuous motivation and encouragement when the remaining tasks seemed overwhelming. I would like to thank MCI for their help to modify the test unit and for providing additional test units and facilities. Furthermore, I would like to thank Poul Kim Madsen from MCI for sharing his profound knowledge of the system details, needed during the modeling phase of the project. During the project I had the privilege to stay at the Automatic Control Laboratory at ETH Zürich and I would like to thank Professor Manfred Morari for giving me the opportunity to study in a such a stimulating environment and Assistant professor Colin Jones for expert guidance while I was there.

Finally, I would like to thank my wonderful wife for her love and support, that have made it possible for me to work intensively on the thesis during the last six months.

Kresten Kjær Sørensen
Aalborg University, June 11, 2015

Part I

Introduction and Summary

Chapter 1

Introduction

This thesis is concerned with modeling, simulation and control of a refrigerated container. The goal is to develop control strategies capable of handling the complex dynamics of the refrigeration system while seeking to minimize energy consumption. This must be achieved while maintaining the temperature within the cargo hold within operational limits.

1 Background and Motivation

This industrial PhD project was started by Lodam in order to enhance the competitive edge of the Star Cool refrigerated container (reefer) and ensure that it continues to be a market leader with respect to energy efficiency and reliability. The goal of the Industrial PhD program is to strengthen research and development in Danish business communities, by educating scientists with an insight into the commercial aspects of research and development, and by developing personal networks in which knowledge between companies and universities can be disseminated. The program includes requirements with respect to public-private cooperation where half the time is spent at a private company and the other half at the university. In order to fulfill the requirements, the thesis has a strong focus on integration with related innovation projects and applications at Lodam.

Lodam electronics has been developing solutions for Heating, Ventilation and Air Conditioning (HVAC) for more than 40 years with the ambition of being a leading global developer of energy-saving electronic controls for cooling, heating and air handling. Lodam strive to offer innovative and cost-effective solutions, enabling its customers to consistently outperform their peers in energy efficiency [1]. One of Lodam's most important products is a complete control solution for the Star Cool [2] reefer, which is an insulated container with a built-in refrigeration unit that is used to transport perishable cargo worldwide. The reefer is developed, built and marketed by Mærsk

Container Industri [2] and currently there is about 150.000 Star Cool containers in use all over the world, which corresponds to one third of the total amount of reefers in service. The Star Cool container was at its introduction in 2005 the most advanced and energy efficient container available, which gave it the lowest total cost of ownership (TCO) and this has forced the competitors to improve their designs in order to compete.

The most common cargo is food, and this market is growing [3] every year, due to increasing demand for exotic food from the growing middle class. It is very important that food is not subjected to harmful temperatures during transport and therefore there is an intense focus on reliability, ruggedness and ease of maintenance for reefers which leads to some stringent demands for any new functionality. A description of the Star Cool reefer system is given in the following section.

2 The Reefer Container

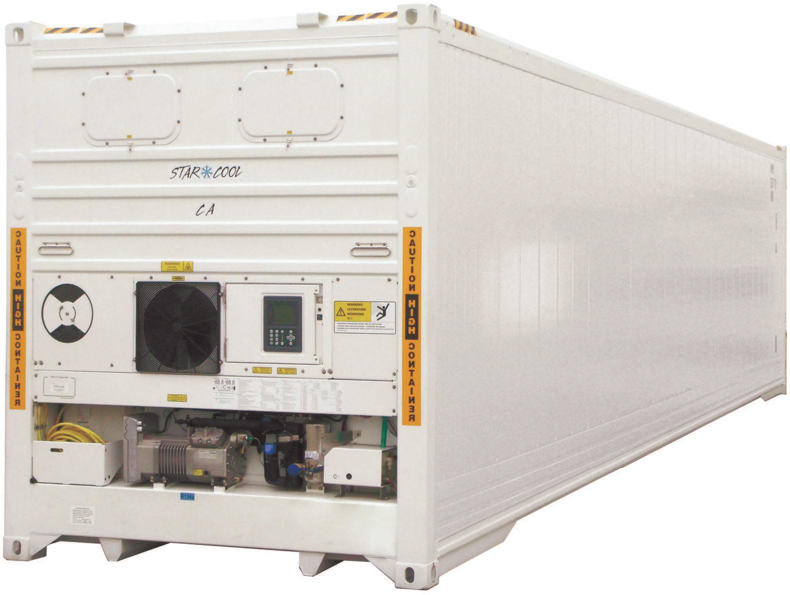


Fig. 1.1: Star Cool Reefer Container

A reefer container is an insulated rectangular box with a loading door in one end and a refrigeration unit in the other. A picture of the refrigeration unit end of a Star Cool reefer is shown in Figure 1.1. The outer dimensions are compatible with those of a regular 40 foot cargo container and that makes it possible to transport perishable cargo worldwide using the existing infrastructure, as long as power for the refrigeration unit is available. The container has a steel frame, the walls are made from aluminum sheets with foam insulation in between and the floor consists of aluminum in a T-profile such that air can flow from the refrigeration unit to the entire cargo hold through the floor. In order to provide the largest possible cargo hold that fits as many pallets as possible the walls and the refrigeration unit must be kept as thin as possible resulting in a wall thickness of 75mm which gives a 43W/K U-value for the container. Figure 1.2 shows a schematic of the airflow inside the cargo hold of a container. Cold air from the refrigeration unit is injected into the T-floor where it travels underneath the cargo until it rises up between the pallets or along the walls to the ceiling of the container. The air is heated throughout its circulation in the cargo hold, by heat from the cargo and heat from the surroundings and the heated air then flows back to the refrigeration unit. The air is circulated by the evaporator fans that are located in the refrigeration unit, above the evaporator. In order to have correct distribution of air throughout the cargo hold it is very important that the cargo has been packed correctly in the container because any vacant spots would cause a lot of air, that was destined for pallets further down, to rise to the ceiling and return. The evaporator fans can be off, running at low speed or at high speed, where high speed provides an airflow that is roughly twice the air flow at low speed. Correct ventilation of the cargo is important to prevent hot-spots, where cargo isn't properly ventilated from emerging and the normal approach to prevent this in chilled goods is to let the fans run at high speed for part of the time or all of the time, depending on the cargo and the program used. A program defines the mode of operation for the reefer controller and the available programs are listed here:

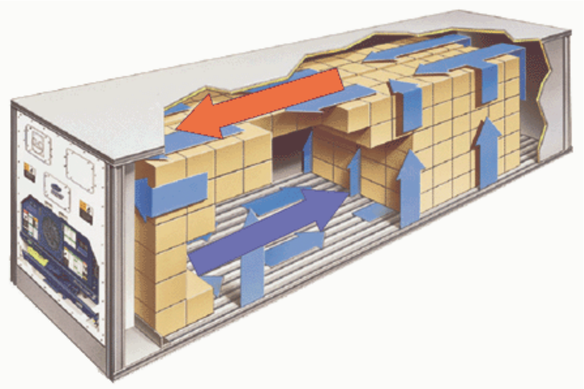


Fig. 1.2: Airflow in the Refrigeration Container

- Automatic Ventilation (AV+): Is used to regulate the CO2 level that slowly increases due to respiration of the cargo, by opening a fresh air valve when the CO2 level exceeds the CO2 set-point. This is used for fruit and vegetables.

- Controlled Atmosphere (CA): Is an improved version of AV+ where both CO2 and O2 levels may be regulated using the fresh air valve and a vacuum pump that removes CO2 through a special polymer membrane.
- Multiple Temperature Set-point (MTS): Is used to change the temperature set-point automatically during a trip if the cargo requires a special temperature trajectory.
- Cold Treatment (CT): Is used to kill insects in the cargo by lowering the cargo temperature for a fixed period.
- QUEST: Is an energy saving program [4] that exploits the fact that most cargo can withstand pulsed cooling at a temperature lower than the set-point. It enables better utilization of the compressor and the fans, resulting in reduced energy consumption

In addition to these programs it is also possible to regulate the relative humidity inside the cargo hold, but in this project the focus has been on temperature control only. The temperature range where the container is expected to function is with an ambient temperature from -30°C to +50°C and with a set point ranging from -30°C to +40°C. The point of operation with respect to temperature has a big influence on the capabilities of the refrigeration system which is clearly visible in the following table, [2] that shows the maximal cooling capacities at different set-points:

Set-point	Capacity
+1.7°C	11500W
-18°C	6500W
-29°C	4000W

From the capacity table above it can be seen that the cooling capacity drops almost linearly with the set-point and this is due to the drop in refrigerant gas density at the compressor inlet. The refrigeration system used on the Star Cool container is made with reduced energy consumption as the most important design parameter and therefore it is also more complex than the system found in competitor products. A schematic of the refrigeration system is shown in Figure 1.3. The refrigeration system is a two stage system with an economizer and a semi-hermetic reciprocating two-stage compressor. There are three main pressure levels in the system; The suction pressure in the evaporator; The intermediate pressure between the two compressor stages; And the discharge pressure in the condenser. The condenser and evaporator is of the fin-and-tube type where the refrigerant flows through a parallel set of tubes that are embedded in perpendicular fins which increase the effective surface area, and thereby compensating for the low heat transfer from metal to air. The receiver is a buffer tank for refrigerant and it is needed because the different operating points require different amounts of refrigerant to run efficiently. The compressor is equipped with a variable frequency drive (VFD)

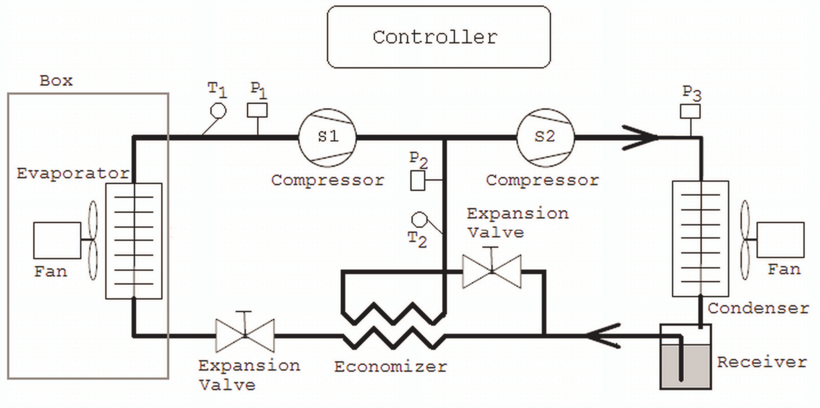


Fig. 1.3: Schematic of the Refrigeration Unit for the Star Cool Reefer Container

that enables the compressor to run at speeds between 20Hz and 110Hz. The expansion valves are pulsed on/off valves that are driven with a six second period. The condenser is located between the compressor and the condenser fan that are both visible in Figure 1.1, where also the fresh air valve, receiver and control panel can be seen. The economizer increase refrigeration systems efficiency and operational range. A small part of the high pressure refrigerant is evaporated to the intermediate pressure and chills the main refrigerant flow before it enters the evaporator, which adds extra cooling potential to the refrigerant and thereby the necessary mass flow through the first stage of the compressor is reduced [5]. The addition of the economizer increase the refrigeration system operational envelope and that enables the refrigeration system to meet the demands for the temperature set-point and ambient temperature that was mentioned earlier. This wide temperature range is not only a challenge for the refrigeration system but also for the system control that must be able to handle a system whose dynamics change considerably with the operating point.

3 State of the Art and Related Work

In this section the state of the art for modeling, simulation and control of reefer systems are described and important related work within the field of thermal systems are summarized.

3.1 Reefer Systems

With respect to refrigeration systems in reefers, Star Cool is state of the art. It is the only brand that use a two stage compressor with a VFD for speed control and that gives Star

Cool an advantage with respect to efficiency in part load situations, where it is running most of the time. The main competitors, Carrier [6] and Thermo King [7], both use digital scroll compressors which is a scroll compressor with an unloading mechanism. In the digital scroll compressor unloading is done by separating the scroll sets that compress the refrigerant momentarily, which stops compression but leaves the motor running at low torque. While not optimal it is much better than the alternative methods for capacity modulation, which is throttling or start/stop operation. Start/stop operation causes fluctuations in the cooling capacity and air temperature that is unacceptable when a precise temperature is required but energy-wise it is an excellent alternative to digital scroll. Throttling that uses a choke valve on the suction line or a hot-gas bypass from the compressor discharge to the evaporator inlet can be compared to driving with full throttle in a car while controlling the speed by applying the brakes, which is very inefficient.

3.2 Modeling of Refrigeration Systems

The second law of thermodynamics states that heat cannot flow from a cold reservoir to a hot reservoir by itself, and therefore cooling of goods to a temperature that is lower than that of the ambient surroundings, must be achieved through artificial means. An example of a sub-critical vapor compression cycle that is able to move energy from a cold to a hot reservoir is shown in Figure 1.4. Because the temperature of evaporation depends on the pressure, it is possible to evaporate a liquid refrigerant below the temperature of the cold reservoir while absorbing heat from the cold reservoir, if the pressure is low enough. When all refrigerant has evaporated to vapor it is compressed which increase the pressure and temperature of the vapor above the temperature of the hot

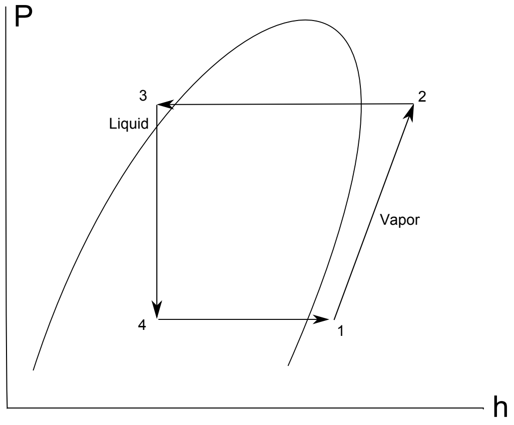


Fig. 1.4: Single Stage Sub-critical Refrigeration Cycle

reservoir. The heated refrigerant vapor is then condensed while releasing energy that was absorbed from the cold reservoir to the hot reservoir. The pressure of the liquid refrigerant may then be decreased using an expansion device which causes the refrigerant to evaporate as it enters the evaporator and this closes the cycle. The efficiency, or coefficient of performance (COP), of the refrigeration cycle is defined as the ratio of cooling provided to the amount of consumed electrical energy and therefore the system should be designed to reduce the amount of electricity consumed by the compressor. The COP is highly dependent on the operating conditions of the refrigeration system and as the temperature difference between the cold and the hot reservoir is increased the COP is decreased. The Star Cool reefer use a two stage cycle as shown in Figure 1.5 in order to ensure a better COP at large temperature differences. In the two-stage cycle the compressor consists of a high and a low pressure stage that each have a displacement volume that is optimized for their nominal working pressures. Furthermore, a refrigerant to refrigerant heat exchanger, called an economizer, is used to remove additional energy from the refrigerant after the condenser and this result in a larger enthalpy difference from the inlet to the outlet of the evaporator. Because of this, a larger amount of cooling can be delivered by the evaporator without increasing the flow of refrigerant. The economizer evaporates a fraction of the refrigerant from the condenser and it is then compressed by the second compressor stage. Modeling of vapor compression systems are well covered i the literature and in [8] it is shown that a low order black box model can be obtained through system identification. The black-box model is shown to be a good match to the real system but is difficult to relate to the physical characteristics of the real system. A physics based models have the advantage of being more useful with respect to controller design and system analysis, because the equations of the model are related directly to the physical properties of the real system. The primary approaches

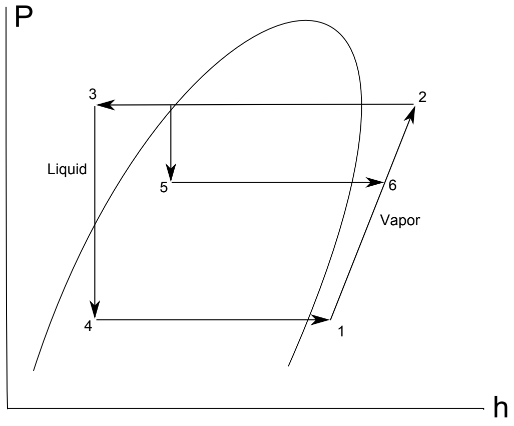


Fig. 1.5: Two-stage Sub-critical Refrigeration Cycle

for physical modeling are finite difference methods and lumped parameter methods. In [9, 10] finite difference methods are used to simulate heat exchangers with good results but due to the large amount of control volumes this method results in models with a very high order, which is undesirable with respect to controller development. Lumped parameter models results in low order models and are therefore better suited for controller development, and will usually also be less computationally intensive to simulate.

In the following an overview of the different modeling approaches is given.

Evaporator and Condenser

The black-box approach to modeling of heat exchangers, that was demonstrated in [8, 11] showed that the salient dynamics, which are important from a controller development perspective, are closely related to the thermal inertia of the metal in the heat exchangers. For physics based models of heat exchangers a prevalent idea is that of the moving boundary model, [12, 13, 14] where two or more control volumes of varying size is used to accurately model refrigerant properties in different phases from vapor to sub-cooled liquid. Reasonably accuracy may be obtained by a lumped parameter model with a moving boundary between each of the control volumes [15].

Compressor

The compressor is a determining factor for system capacity because it drives the flow of refrigerant and any error in the predicted mass flow will be propagated to the other components of the system. Therefore, a compressor model with an accurate prediction of mass flow rate is essential for system level simulation, but also for control because the compressor model determines the important system gain from compressor speed to capacity. According to [16] there are three categories of reciprocating compressor models; Detailed physics based dynamical models; Efficiency based models that are based on an ideal compression and isentropic efficiency; The last category is map based models which is a table lookup of performance data or a polynomial fit to the data table. Detailed physics based models are able to reflect the dynamics for each stroke of the compressor and the heat transfers between the refrigerant and each part for the compressor. Such models are well suited for compressor development but generally too computationally demanding for system level simulation. Efficiency based models assume adiabatic compression and may include physical details [17] such as displacement and clearance volume to improve accuracy. Most compressor manufacturers supply performance data for their compressor in the form of 10 coefficient polynomials [18] that may be used to predict the compressor performance in map based models, within limits for the data.

Expansion valve

The most important property of an expansion valve model is the ability to accurately reflect the mass flow because it has a large impact of system capacity. An expansion valve is basically an orifice that limits the flow of a substance which results in a pressure drop, and this has been described well in the literature [19, 20, 21]. In [22] it is shown that the general orifice model is accurate for expansion valves when manufacturer data is used to parametrize the expansion valve model.

Economizer
The economizer is a plate heat exchanger, and modeling of these are described well in the literature [23, 24] with model fidelity ranging from very detailed models that accurately reflects flow dynamics and distribution to simple steady state models that are based on the logarithmic mean temperature difference (LMTD) [25].

Receiver
A receiver is a container where excess refrigerant is stored when load conditions of the system requires a smaller refrigerant charge than what is actually loaded into the system. An accurate physics based dynamic model is described in [15] where also the receiver interaction with the condenser is described. A receiver model matched against experimental results is described in [26] where it is found that a lumped receiver model is a good match to the real system, at high refrigerant charges where phase separation was observed in the receiver.

3.3 Simulation of Refrigeration Systems

In [27] a survey of different simulation techniques for refrigeration systems is carried out and it is found that a wide range of very different techniques are used.

Energy system simulation is a mature field where tools such as TRNSYS [28], that provides a library of energy system components and its own system description language, have been used for more than 35 years. The Department of Energy Engineering at Technical University of Denmark have developed WinDali [29] for simulation of refrigeration systems where models are compiled to native machine code which make the simulations very fast.

Dymola [30] is a relatively new tool that may be used to simulate almost any type of dynamic system from a set of symbolic equations that describe the system behavior. The tool depends on the free Modelica [31] language, to describe the physical system with a high layer of abstraction. A range of classes that describes a physical property such as voltage, flow or pressure, may be used for component model I/O. It is not necessary to define the direction of signals between components when the models are developed because Dymola has automatic ordering and reduction of the final set of equations, before the simulation is run. This is very convenient for the user because equations may be listed in arbitrary order and arranged in the most comprehensible way. The final set of equations may be simulated by a range of different solvers and in

[32] the importance of choosing the correct solver, with respect to model properties, is demonstrated. In [26] DYMOLA is shown to be well suited for simulation of refrigeration systems due to the ease of model formulation and large library of pre-defined models.

The tool that is most commonly used for controller development is MATLAB®, which also includes state-of-the-art toolboxes for many scientific and engineering disciplines, including modeling and controller design. It has an active on-line community that contributes with tools and helper functions. It provides a range of solvers for solving DAE's, PDE's and ODE's and it has good integration with other languages such as C, C++ and Java which can be used as a flexible external interface. SIMULINK® is a tool for creating and simulating complex models from built-in or custom blocks, which may be added to the MATLAB® suite. The models may be simulated using a range of different solvers and if algebraic loops exist SIMULINK® tries to break them by inserting zero-order-holds. The solvers that are available in SIMULINK® are a set of different order stiff and non-stiff solvers [33], and theses solvers are also available inside the standard MATLAB® environment.

According to [34] simulation of a system, that includes states that have very different dynamical speeds, as a monolithic block has the drawback that during changing conditions the step size for the slow states are much smaller than what is needed to achieve satisfactory accuracy. It is shown in [34, 27] that the fast dynamical states in many cases can be converted into steady state equations, resulting in a faster simulation. Another way of reducing the computational load is presented in [35, 36] where a multi-rate simulator is used to simulate the fast and slow parts of a model separately which gives a substantial increase in simulation speed because an appropriate step size is used for all states. In [37] a modular multi-rate simulation environment is presented. Different solvers can be used for each of the components, making it possible to simulate mechatronics systems using original models that are made for different simulation tools. It introduces methods for solving algebraic loops that can occur when different numerical models are combined in the simulation environment. The modular approach has several advantages; It is faster to build a new simulation model from a library of components than starting from scratch and it is also easier to maintain because the code is naturally divided and therefore less likely to be entangled across component models. A drawback with the modular approach is that in order to run sub-models at different step sizes it is necessary to handle the exchange of data between component models. This may be done by inserting zero order holds (ZOH) between the component models [35], but this effectively introduces a delay between components, which may affect stability and accuracy of the simulation. In a simulation, the impact of inserted ZOH's depends on the frequency of data exchange between components and the dynamical speed of the components. When data exchange happens at a low frequency compared to component dynamics, oscillatory behavior and possibly simulation breakdown is likely to occur. In [38, 39] different methods for handling data exchange between components are introduced and shown to give good results, in special cases. An iterative method for data exchange between component models, that guarantees stability, is presented in [37].

Methods for numerical simulation of systems of differential equations are plentiful and the field very diverse, in the sense that specialized solvers exist for almost any scientific field where numerical simulation is used. In [40, 41, 42] the methods are divided in two overall categories; Runge-Kutta methods and linear multi-step methods. A numerical solution to a set of ordinary differential equations is an approximation, and not an exact solution to the problem. Generally there is a trade off between the time it takes to reach a solution and the accuracy of the solution, and most solvers are supplied with an error tolerance that the local truncation error of the solution must stay within. Runge-Kutta methods are a family of explicit and implicit methods that are very common due to their accuracy and versatility, and solvers of this type are available in both MATLAB®, SIMULINK® and DYMOLA. Explicit methods calculate the future state of a system from the current state only, while implicit methods solves an equation that includes both the current and future state. Explicit methods are well suited for non-stiff systems, and for same order methods, explicit solutions to non-stiff systems are faster than implicit solutions. For stiff systems explicit methods may be very slow or even unstable and in this case the use of implicit methods are required. For literature on the exact algorithms, stability analysis and applications, see [33, 43, 44, 45].

3.4 Control of Refrigeration Systems

The reefer is a multiple-input multiple-output (MIMO) system for which applicable control methods are single-input single-output (SISO) and MIMO controllers. A SISO controller is based on classical control theory and a very common implementation is in the form of PID controllers [46], which are used in the majority of industrial control systems. The theory of design and tuning of PID controllers are described in [47]. For nonlinear systems, PID controllers can only guarantee asymptotic stability around the operation point that was used in the design, [48] and this can be addressed in a number for ways.

One method is detuning the controller to improve global robustness, but this can severely impact performance of the controller. Gain scheduling is a method where the gains are varied according to the operating point, and in this way the impact on performance is reduced while global robustness is improved. Gain scheduling is frequently used in industrial applications and an overview of gain scheduling methods and their applications are given in [49]. Today the most advanced controller used in reefers is PID's with gain scheduling to compensate for system nonlinearities. The drawbacks of gain scheduling are that stability is only guaranteed around the design points and in order to achieve global stability the system must be linearized in a fine mesh in terms of scheduling variables [50]. This scales poorly as the number of scheduling variables is increased and for MIMO systems it can be very time consuming. Furthermore, the rate of change in the scheduling variables must be slow, compared to system dynamics and this place fundamental limitations on the performance that may be achieved.

Linear Parameter Varying (LPV) control is a method that incorporates robustness, performance and bandwidth limitations into a unified framework [50, 51, 52] and therefore global stability and performance can be guaranteed.

When a good model of the nonlinearity in a nonlinear system is available, feedback linearization may be applied in order to enable control with a linear controller [48]. The feedback linearization is a transformation that is inserted in the control loop, which in combination with the plant, forms a linear system that can be controlled with good performance using traditional linear methods.

If the nonlinear nature of the system is uncertain, control performance using feedback linearization may be significantly degraded. The problem of uncertainties in models and disturbances is addressed by robust control design methods, [53] that guarantee stability and performance within given bounds for disturbances and system uncertainties.

Reduced Energy Consumption for Reefers

In order to increase efficiency at part load the QUEST control scheme was developed by Wageningen UR [4], resulting in a very large increase in efficiency for reefers without good capacity regulation. Based on laboratory and field trials the temperature fluctuation limits and ventilation requirements for typical cargo such as apples and bananas were mapped, and this knowledge was implemented into QUEST. The scheme exploits the temperature fluctuation limits to run the compressor in on/off operation without unloading or throttling, which greatly improves system efficiency. The QUEST program also reduces power consumption, by reducing the amount of ventilation, which saves power on the fans. All electrical power consumed by the evaporator fans ends up as heat inside the container and this heat must then be removed by the refrigeration system. Therefore, a reduced ventilation rate also results in lower power consumption by the compressor. In the scientific literature the amount of work specific to reefers is limited to the studies that laid the foundation for the QUEST control scheme, which is a PhD thesis [54] and some papers [55, 56, 57, 58], that has focus on cargo quality. They all have good and detailed models of the cargo with respect to quality, but the refrigeration system is not treated in detail. For reefers the QUEST scheme [4] is currently the most advanced approach but while it is based on experiences gathered through the use of Model Predictive Control (MPC) [54] to improve product quality control and reduce energy consumption it does not adapt to external disturbances.

In [54] MPC was used to control the product quality of chilled cargo in refrigerated containers with the focus on modeling and control of the cargo quality, resulting in a reduction of mass loss in the cargo due to less evaporation and lower energy consumption due to reduced ventilation rate.

Model Predictive Control was introduced in the petro-chemical industry in order to control difficult processes with long delays and unknown states but today it is used in a wide range of applications such as power plant control and the automotive industry

[59]. MPC is used for optimizing control of processes with respect to known future demands or known future changes in external conditions, while keeping within a given set of constraints. The performance of MPC is highly dependent on the quality of the model on which it is based because it is used to predict the behavior of the system over the prediction horizon without the possibility of error corrections from measurements. For nonlinear systems where the model dynamics may change, either a nonlinear or adaptive linear approach must be used in order to keep performance and avoid violating constraints. The drawback of using MPC is that it is computationally heavy and therefore not very well suited for embedded systems such as the Star Cool reefer controller. Methods for reducing the quadratic problem of an MPC to a simpler explicit piece-vise linear function have however been described in the literature and in [60, 61] this is demonstrated.

In [62] MPC is used to exploit daily variations in ambient temperature by cooling more when the COP is higher, during the night where temperature is low, using the cargo to store some cooling which reduces the amount of cooling needed during the day where the COP is lower. Another study where MPC is used for quality control is [63] where future spikes in ambient temperature, that is too large for the refrigeration system to handle, is countered by cooling the food in a supermarket display case in advance. In [64] saturation of a refrigeration system due to high ambient temperature is countered by a learning based algorithm that pre-cools the foodstuffs in the display cases a supermarket based on previous saturations of the refrigeration system. Over time the system "learns" when to apply extra cooling in order to counter anticipated high temperatures in the future.

4 Project Objectives

Objectives
The overall objective of this project is to investigate the potential for reduced energy consumption on the Star Cool reefer by the introduction of modern control methods, without compromising the quality of the transported goods. A model based control design approach is chosen because the system is well known and variations between individual reefers are small. Therefore, a dynamic model of the refrigeration system that has adequate accuracy for control design is needed. This means that the model should accurately reflect the measurements that are used as controller inputs, both statically and dynamically.

Test and verification of control algorithms on reefers can be very expensive, especially when realistic load profiles are required and the conditions inside and around the reefer needs to be controlled accurately. This requires a test cabin that can contain a reefer, equipped with a HVAC system large enough to suppress fluctuations in air temperature around the reefer, while the reefer is running. Furthermore, such tests can be very time consuming, because of the slow dynamics of the cargo. The cost of running these tests can be reduced by simulating the reefer system, using a simulation model. It is important that the speed of simulation is significantly faster than testing on a real system and that the results of the simulation are accurate. A simulation method that possesses these properties must therefore be identified.

A reduction in energy consumption of the Star Cool reefer is desired and could be obtained through modern control. The available measures for reducing the energy consumption are the thermal inertia of the cargo that can be used as a buffer, the ventilation rate inside the container and it may also be possible to optimize control set-points. Appropriate control methods should therefore be investigated and a controller setup that address the different challenges present on a reefer, should be selected.

Commercial Scope
For Lodam an important outcome of the project is to reduce the time it takes to develop, test and verify new controllers. This go hand in hand with the research objectives for modeling and simulation, but it also means that the simulation tool must be easy to work with and that it must have easy access to good controller development capabilities. Such capabilities are found in MATLAB® and, for commercial reasons, it was decided that the methods developed in this study should be implemented in MATLAB®. This has the additional advantage that a wide range of tools for data analysis and modeling already are available in MATLAB® and these could be useful to the project as well.

When hardware components on the reefer are changed it may require that the model is changed as well in order to reflect the dynamics of the new hardware and this should be easy. Therefore, it is required that the simulation method is flexible with respect to different hardware configurations.

A controller that reduces energy consumption, through the use of the thermal inertia of the cargo, requires optimization over a long period which can be computationally heavy and this does not fit well with an eventual implementation in the embedded controller of the Star Cool reefer. Therefore, methods that reduce the computational load of the controller should be investigated.

The objectives can be summarized as below:

- **Objective I: Control model**
  To create a non-linear dynamical model of the reefer that is suitable for development of controllers. The model should be accurate to within 1 Kelvin on the measurements that are important for control, and this must be verified against data from a real system.
- **Objective II: Simulation model**
  To investigate if it is possible to formulate a non-linear dynamical model such that it can also be used for test and verification of controllers through simulation. The simulation model is required to have an accuracy such that controllers can be tested on the model and subsequently used on the real system without further tuning. Furthermore, it is required that the model is able to simulate a period of 21 days, which corresponds to a normal trip of a container.
- **Objective III: Simulation method**
  To investigate appropriate methods that enables simulation of the non-linear model with emphasis on accuracy and simulation speed. The speed should be at least an order of magnitude faster than real time and the error introduced by the simulation should not exceed the variance in system tolerances on the reefers in the field.
- **Objective IV: Reduced energy consumption**
  Investigate the potential for reducing energy consumption during transport of frozen cargo in reefer containers, by using the thermal inertia of the cargo as a buffer, and by reducing the ventilation rate.

The work presented in this thesis can be related to one or more of the above objectives.

5 Summary of Contributions

The main contributions in this project are listed in this section and divided into three parts.

5.1 Reefer Model

Proposal for a control oriented nonlinear dynamical model of a reefer container that is verified against data from a real container at multiple capacity set-points with an error of less than 1K on the states important for control. The model is an extension of the ideas presented in [11] where it is shown that, for a car air conditioning system, a model that includes the dynamics of the metal in the heat exchangers and steady-state equations for the refrigerant control volumes has adequate accuracy to be used for controller design. The proposed model is a collection of components, which have been found in the literature and adapted, that together form an accurate representation of the Star Cool reefer. It has been shown that accuracy of the presented model is adequate for controller design and therefore the ideas presented by [11] also holds for the more complex reefer refrigeration system. A description of the development, test and verification of the model can be found in and the equations for the model are presented in [Paper C Section 2].

The model is currently being used by Lodam for development, test and verification of control and fault detection algorithms and it has proved to be a valuable tool that significantly reduces development time.

5.2 Simulation

A proposal for an efficient and accurate simulation method for the nonlinear model of the refrigeration container. An analysis of solver choice demonstrates that, for modular simulation, using a simple solver can be faster than more advanced solvers, without sacrificing accuracy. The simulation tool provides a solid and necessary base for development and test of model based controllers.

The contributions from this work related to the commercial interests are the simulation environment and its associated tools which will enable Lodam to improve and speed up the efforts for model based controller development for the Star Cool reefer.

The modular multi-rate simulation of the reefer is presented in [Paper A Section 3] and the comparison of methods and solvers is presented in [Paper B Section 2].

5.3 Adaptive Model Predictive Control for a Reefer

Proposal for an adaptive controller strategy that is able to reduce the energy consumption of the reefer with up to 21% by exploitation of daily variations in ambient temperature and a reduction of the rate of ventilation such that it matches the actual demand. A cargo estimator is used to estimate important unknown states and cargo parameters that are used to continuously update the linear model for the MPC, which allows the controller to adapt to cargo with different dynamics and heat transfer rates. A series of optimizations on the MPC reduces the computational load and makes it commercially interesting. The control strategy is presented in [Paper D Section 2.3].

Chapter 2

Summary of Work

In this chapter the contributions from the project are described in detail. The description is divided in three sections concerning modeling, simulation and control of the reefer, respectively. The details of the reefer model are described in Section 1, the reefer simulation tools and experiments are described in Section 2 and the proposed adaptive controller for the reefer is described in Section 3.

1 Reefer Container Modeling

This section describes the important characteristics of the reefer system and summarizes the most important aspects of the work on the reefer simulation model. The equations of the model are described in detail in [Paper C].

1.1 Objectives

Development, test and verification of model based controllers require a good system model and therefore a large effort has been put into the development and verification of a simulation model for the reefer container. In order to meet the objectives of the project the model must be a good match to the dynamics of the refrigeration system, in order to support the development of a controller that is able to keep the system close to the energy optimal set-points. The model must also match the steady-state properties of the real system closely, in order to support development of controllers for long term energy optimization. The model should be designed such that it can serve as a tool for future development and test of controllers in order to meet Lodam's ambition of reducing the development time of new controllers. This requires that the model is flexible and that it can be reconfigured for other tasks if needed. One such task could be simulation of faults such as lack of refrigerant or reduced compressor capacity due to a mechanical fault and therefore the model must include the refrigerant circuit.

1.2 Control Oriented Modeling

Creating a model that meet the objectives requires identification of the system characteristics that impacts energy efficiency and for refrigeration systems this has been investigated in a range of different works. In [65] a simple refrigeration system with variable speed fans and compressors are analyzed with the conclusion that the relative speed of the compressor and the fans are very important for the overall power consumption of the system. This means that the model at least should be able to accurately reflect the mass flow through the compressor, the condensing pressure and the suction pressure, in dynamic and steady state conditions. The superheat of the evaporator is also an important parameter, because it has a big impact on the energy efficiency of the system, and therefore the dynamics and steady state properties of this parameter should also be matched by the model.

A simple way to achieve a good match on these parameter dynamics are presented in [11] where the authors show that the salient dynamics of a refrigeration system is determined by the thermal time constants of the metal in the heat exchangers and this observation is a major inspiration for the development of the model presented here. Achieving a good match of steady state conditions for a refrigeration system where the range of operation is as large as it is on the reefer is challenging because the capacity of the system changes with the difference between the ambient temperature and the temperature inside the container. This effect is caused by the fact that the saturated suction temperature, T_0, must be lower than the air inside the container in order to cool the air and due to a reduction in compressor efficiency at higher differential pressure. Both effects are related to non-linear refrigerant characteristics and therefore the refrigerant circuit in the model should match these characteristics. One way to achieve this is to make a first principles model of the full system that accurately models the state of the refrigerant in all control volumes, but this requires a lot of work and would yield a model that was more advanced than needed, so a simpler approach was pursued. The interesting dynamics from a control perspective is covered by modeling the thermal time constants of the heat exchangers metal and therefore it was found that the approach presented in [66], where the refrigeration circuit is modeled by steady state equations, could be used. So to summarize, the model presented here use first principles steady state equations for the refrigerant circuit and first principles dynamical equations for the larger thermal capacities in the system, such as the metal in the heat exchangers, the air in the container and the cargo. First principles are substituted by assumptions where it reduces model complexity and has a small effect on model accuracy. If it is found at a later point in time, that the component models that have been selected for the system model are inadequate with respect to accuracy, the modular modeling approach ensures that components can be easily substituted with better alternatives from the literature.

The schematic of the refrigeration system is shown on Figure 2.1 where the components of the system can be seen. It was chosen to create the following sub-models,

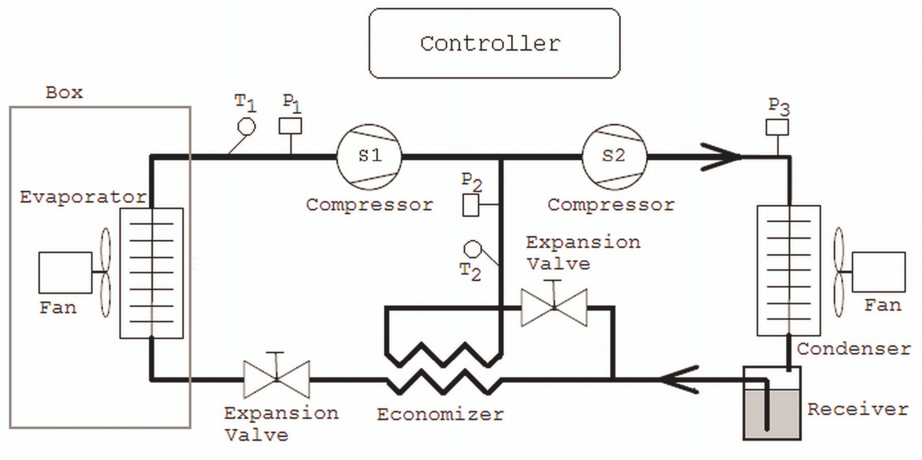


Fig. 2.1: Schematic of the refrigeration system

combining some components into one model:

- **Compressor:** The compressor model includes the two compressor stages and models the internal volume of the compressor. The compressor stages are modeled by steady state equations that give a mass flow and an output enthalpy based on speed, input enthalpy, input pressure and output pressure. The model is an efficiency based model that assumes adiabatic compression, equal mass flow on the inlet and the outlet and a steady flow of refrigerant vapor. It was shown in [66, 67] that this is a reasonable assumption because the dynamics of the individual piston strokes are fast compared to the evaporator and condenser dynamics. The important aspect of the internal volume model is that the state of the refrigerant in the control volume gives the evaporation pressure for the economizer and therefore it influences the sub-cooling of refrigerant to the evaporator. The equations for the two compressor stages are described in detail in [Paper C Section 2.3] and the equations for he internal volume are described in [Paper C Section 2.1].

- **Splitting Junction:** This is a very simple model for a pipe T-junction that splits a refrigerant flow in two, and it models the T-junction between the receiver and the economizer. The equations for this component are described in detail in [Paper C Section 2.2].

- **Condenser:** This component removes energy from the refrigerant that exits second stage of the compressor, thereby condensing it into a liquid. It was attempted to use a simple model for the condenser with only one control volume and a rough assumption for the pressure drop due to flow resistance, making the input pressure at the compressor equal to the pressure in the receiver plus a variable delta pressure that depends on the mass flow. The equations and assumptions for the condenser are described in detail in [Paper C Section 2.4].

- **Receiver:** The receiver is a buffer tank for excess refrigerant that is not currently located in other components. It is needed because the amount of refrigerant required for the evaporator and condenser to run efficiently varies with temperature and load. The receiver has two control volumes; One for the liquid refrigerant and one for the refrigerant vapor where the state of the vapor determines the pressure. In [Paper C Section 2.5] the equations and assumptions of the receiver is explained in detail.

- **Economizer with Expansion Valve:** The economizer is a counterflow plate heat exchanger that has evaporating refrigerant on the cold side and liquid refrigerant on the hot side. It sub-cools the refrigerant going to the evaporator and it helps boost system performance by increasing the refrigeration systems efficiency and capacity. The thermal time constant of the metal in the economizer is small and therefore it is not modeled for this component. Instead, it models the steady state heat transfer from the liquid refrigerant to the evaporating refrigerant with one control volume for each side, using a logarithmic mean temperature difference (LMTD). The expansion valve for the economizer is included in this model because it is a simple steady state calculation of mass flow from the pressure difference over the valve and the density of the liquid that enters the valve. The equations of the economizer are available in [Paper C Section 2.7] and the equations and assumptions for the expansion valve are described in [Paper C Section 2.6].

- **Evaporator with Expansion Valve:** Controlling the evaporator is often the most difficult part of refrigeration system control due to the highly nonlinear behavior of the superheat when it approaches zero where the sensitivity of the measurement vanishes. The superheat is the difference between the saturated suction temperature and the actual suction temperature and it is a measure for excess energy added to the refrigerant vapor after full evaporation. A higher superheat is achieved by lowering the pressure, but this reduces the efficiency of the compressor so the theoretical optimal point of operation is a superheat of zero kelvin. It is important that the refrigerant reaching the compressor is dry because liquid refrigerant can wash away the oil film that protects the pistons against wear and tear. When the superheat is zero the system is very close to liquid slugging, where the refrigerant is no longer completely dry. If liquid slugging occurs the superheat cannot be measured because the sensitivity of the superheat measurement has dropped to zero. Therefore, the normal approach is to run with a superheat that is as small as possible, and if a superheat controller is to be designed from the model it is required that the accuracy of the modeled pressure and suction temperature is high. This is achieved by modeling the evaporator as two control volumes separated by a moving boundary that depends on the amount of liquid refrigerant in the evaporator. In the first control volume the refrigerant evaporates and in the second it is superheated which means that the superheat depends on the position of the boundary, or the filling of the evaporator. As for the economizer, the expansion valve model is included in the evaporator model. The equations of the evaporator are available in [Paper C Section 2.8] and the equations and assumptions for the expansion valve are described in [Paper C Section 2.6].

- **Container and Cargo:** The container and the cargo are the largest thermal capacitances of the system and the temperature of the return air from the container is closely tied to the temperatures in the container. The cargo hold is modeled as two lumped volumes, one is the cargo and the other is the aluminum floor which is the main mass of metal inside the container. It is assumed that the air passing over the floor has the supply air temperature and that the temperature of the air passing over the cargo is the average of the supply and return temperatures. The cargo in the model that is used in this thesis is a load of frozen pork meat that was kindly provided by MCI for modeling experiments, which led to estimates of the heat transfer coefficient from the air to the cargo. In these experiments the mass and type of the cargo was known, making it easy to estimate the heat transfer coefficient but normally this is not knowledge that the controller has and that presents a challenge that must be overcome if a model based controller that includes the cargo dynamics is to be used. A temperature measurement of the cargo is not always available and this complicates the estimation of cargo dynamics further. This issue is treated is Section 3.3. The equations for the model of the cargo and the cargo hold are described in [Paper C Section 2.9]

The structure of the reefer model is defined by the collection of sub-models which have been refined and adapted to the current needs during the project.

1.3 Identification of Model Parameters

The structure is an important part of achieving good model performance but another very important part is the model parameters which were identified from measurements from the reefer in different operating points. The important parameters for this model are the heat transfer coefficients and the parameters that influence the mass flow of the compressor stages and expansion valves.

The mass flows are important to get right because they define the amount of refrigeration that is achieved from a given set of control values and they were calibrated from direct mass flow measurements on the reefer in different operating points. For the compressor model the adjustment were done by adapting the valve loss such that the mass flow at low suction pressure were correct. The expansion valve model was adapted by adjusting the characteristic constant.

The heat transfer coefficients were initially calculated from the cooling capacity and relevant steady state temperatures at different operating points, but due to inaccuracies in the model assumptions some subsequent minor adjustments was needed for the best result. The heat transfer coefficients influence the dynamics because they are determining for the amount of energy transferred between the main thermal capacitances that models the system dynamics and the adjustments done to the heat transfer coefficients were aimed at getting a better match of the system dynamics.

1.4 Verification and Results

In this section the verification of the full system and the component models is described. The requirements for the model that was described in Section 4, states that the model must be accurate to within 1K on the measurements that are used for control, in order to be suitable for development of controllers. The objective for the simulation model requires an accuracy that would allow most of the development of a controller to be done on the model, with only verification done on the real system. A controller developed for a fleet of reefers must be designed to be robust against the variations in dynamical response and capacity that exists from unit to unit. This means that the error of the model should be smaller than the error caused by variation between reefers in the field. Accurately measuring what the error is on reefers in the field is difficult but it was judged to be significantly larger than what is required for the control model. Therefore, both models may be verified by checking that the error on measurements that are important for control is below 1K in steady state and dynamic situations. The measurements, that are normally used by the reefer controller, are the supply air temperature T_{sup}, the return air temperature T_{ret}, the suction temperature T_{suc}, the saturated suction temperature T_0 and the saturated discharge temperature T_C. The cooling capacity may be calculated from the evaporator fan speed and the difference between T_{sup} and T_{ret}, and this can be used to verify that the cooling capacity of the model match the real system. This is significant because a matching cooling capacity indicates that the static properties of the compressor and the expansion valve are accurate.

A reefer with a cargo of frozen pork meat was kindly made available for the test by MCI and verification of the system model performance was done by comparing the measurements relevant for control, from an open loop simulation on the model and data recorded from a reefer. The open loop simulation were driven with the control signals recorded from the reefer.

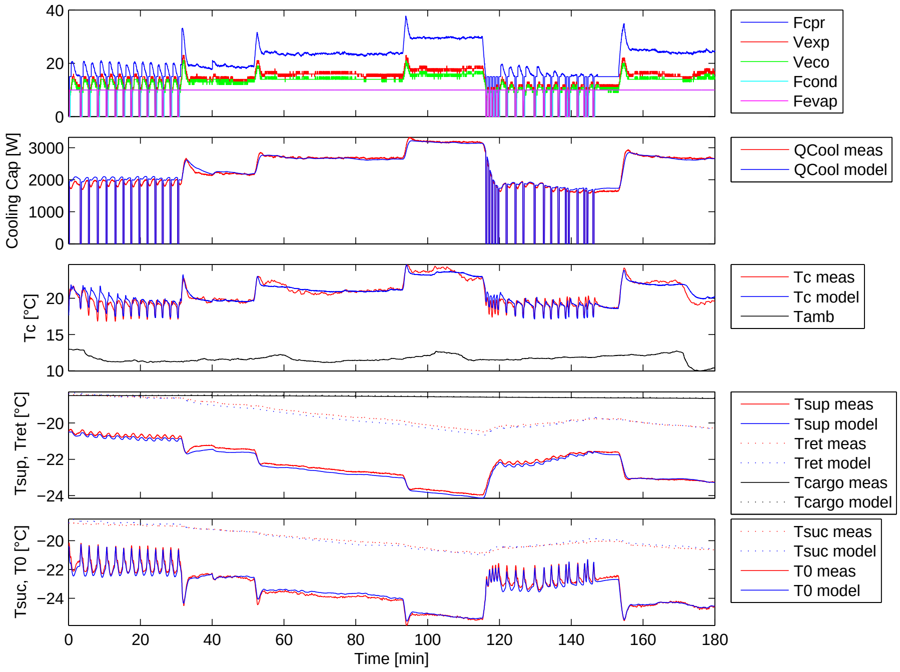


Fig. 2.2: Model verification results

Running the simulation in open loop mode will further help verify the models of the expansion valve and the compressor because errors in these components will integrate up over time and cause the amount of refrigerant in the evaporator to drift which would then cause either liquid slugging or a too low T_0. If the dynamics due to changes in capacity are still accurate at the end of the simulation, the models of the compressor and the expansion valves are also accurate. The test sequence consists of a range of steps in cooling capacity that ensures that the unit is run in both PWM and continuous mode. This is important in order to verify that the dynamics of the evaporator and the condenser, with regards to change on the control inputs, are accurate. A test period of three hours is used and the full sequence can be seen on Figure 2.2, while a closeup of the first 30 minutes where the reefer is running in PWM mode can be seen in Figure 2.3. In order to perform the model verification the start conditions of the model were adjusted to fit the initial state of the recorded data. The control signals from the recorded data are then used as input to the model while it is simulated for three hours. From the close-up on Figure 2.3 it is clear that the model of the evaporator is accurate because the model results for T_0 and T_{suc} have a dynamical response that is almost identical to the real system. For the condenser the results are not as good and it can be

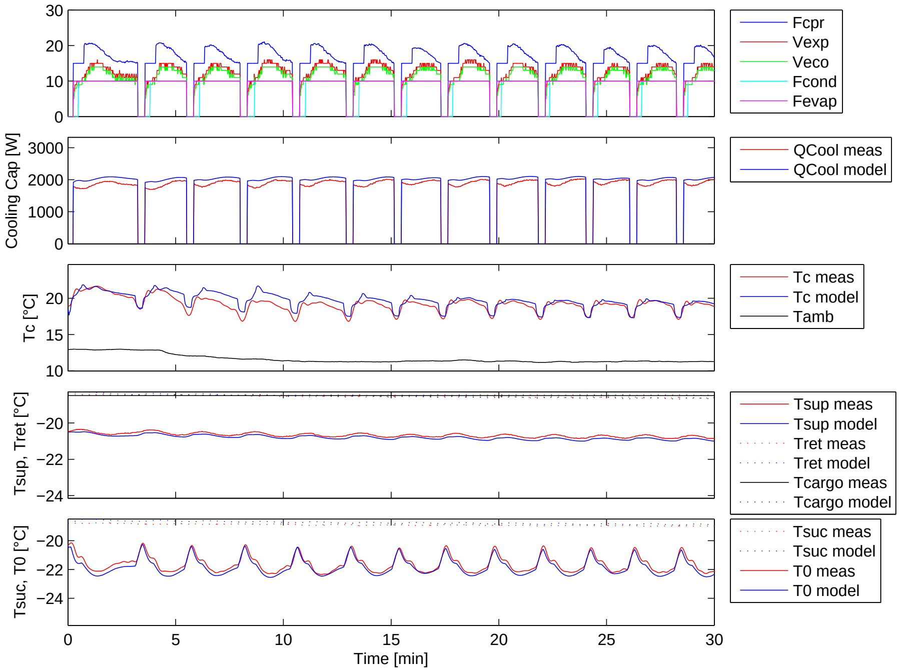


Fig. 2.3: Close-up of the start/stop phase

seen that the error on T_c exceeds 1K 10 minutes in to the simulation. Furthermore, the dynamics on start of the compressor cause a small pressure spike just after start where the model have a more soft pressure increase. This indicates that some dynamics may be missing, and considering the simple model used for the condenser, this is entirely possible. It seems that the condenser model lacks sensitivity to changes in ambient temperature, which is especially clear in the end where there is a sudden drop in ambient temperature. The cooling capacity matches very well throughout the simulation and this indicates that the steady state properties of the model are excellent. The supply and return air temperatures show a good match to the reefer but there seems to be a small phase shift on T_{sup}, which could be caused by missing sensor dynamics.

Verification of component performance

In the following the performance of the individual component models are rated, based on observations from the open loop test.

- **Compressor:** The compressor is deemed to be very accurate because the cooling capacity of the system and the model are accurate to within $\pm 100W$ for the majority of the test time. A set of measurements of refrigerant flow through the evaporator in steady state were provided by MCI and the results in Figure 2.4 demonstrates that the model of the compressor is reasonably accurate over the full range in speed and suction pressure. The flow was measured over in the liquid line, just before the evaporator expansion valve, and due to the pulsating modulation of the expansion valve the measurements vary, especially at high flow where the flow-meters internal averaging of the flow were challenged the most.

- **Condenser and Receiver:** Verification of the receiver and condenser models are difficult because the only available measurement is T_c, but it can be seen that the steady state pressure level is reasonably accurate but also that some fast dynamics are missing. The steady state pressure determines the enthalpy on the refrigerant that is forwarded to the economizer and drives the mass flow through the expansion valves. Therefore, when the steady state pressure is accurate, the impact of the errors on the fast dynamics is seen as limited.

- **Economizer with Expansion Valve:** The economizer on the Star Cool unit are normally controlled in open loop and therefore there are no measurements available for direct verification of the economizer model. Verification of the steady state properties of the economizer must therefore be done through observations on the full system. The refrigerant entering the economizer is on the bubble point at the discharge pressure and the enthalpy is then lowered through sub-cooling of the refrigerant by the economizer. The amount of sub-cooling have a direct impact on the cooling capacity provided by the evaporator, which can be measured and have been verified to closely match the real system. Because the compressor mass flow is accurate, it can be deduced from the energy balance of the evaporator that te economizer model provides the correct amount of sub-cooling.

- **Evaporator with Expansion Valve:** The evaporator model includes the following important states: T_{sup}, T_{suc} and T_0, which are all measured and therefore can be verified directly. The results on Figure 2.2 show that the static and dynamic response of the evaporator model closely match the real system. Verification of the expansion valve is done against the flow measurements shown on Figure 2.4, from which it can be concluded that the accuracy of the expansion valve model is adequate.

- **Container and Cargo:** The model of the container cargo-hold and the cargo models a single state, T_{ret}, with slow dynamics that depends on the thermal inertia of the cargo and some faster dynamics from the thermal inertia of the cargo hold. Verification of the slow dynamics are difficult from only three hours of simulation, but due to their dependency of the nature of the cargo the slow dynamics are expected to vary in the field and therefore an accurate model less important. An accurate model of the fast dynamics are however important because they are needed to design a long-term energy optimizing controller. On Figure 2.2 the results show that the cargo temperature changes very little during the test and the dynamic response on T_{ret} is therefore mainly from the cargo-hold. The results show that the dynamic response on T_{ret} is a good match to the real system.

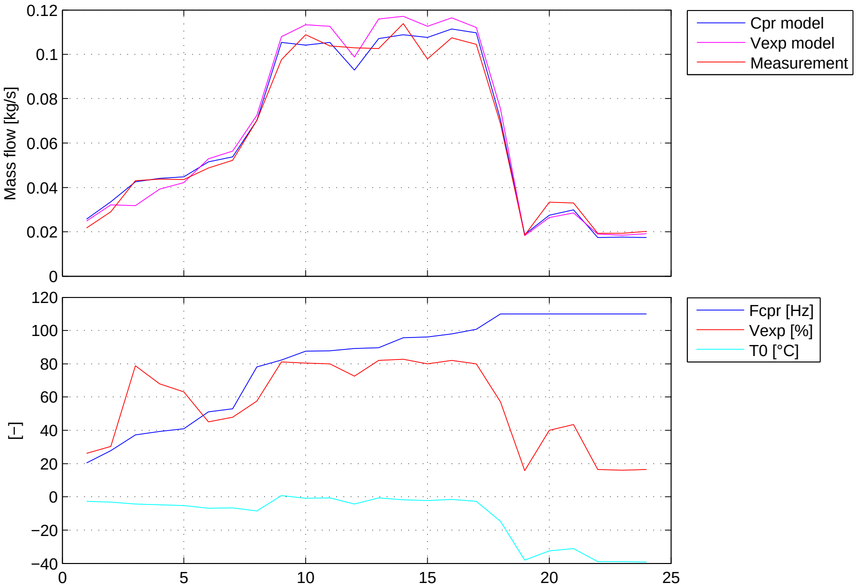


Fig. 2.4: Compressor and expansion valve mass flows

The absolute error for each of the signals considered in the model verification are shown in Figure 2.5, where it can be seen that the error for the signals on the evaporator are below 1K, which is deemed adequate for development of model based controllers for the Star Cool reefer.

Therefore, objectives I and II have been achieved.

The results obtained in this work are comparable to the results shown by Rasmussen et Al. in [8, 68, 69] where control oriented models of refrigeration system components for residential air-condition are developed and verified against measurements from real systems. In [11] it was shown that the salient dynamics of heat exchangers mainly depends on the metal. In this work dynamic models of the heat exchangers metal surfaces, combined with a non-linear refrigeration circuit, have been shown to result in excellent static and dynamic accuracy.

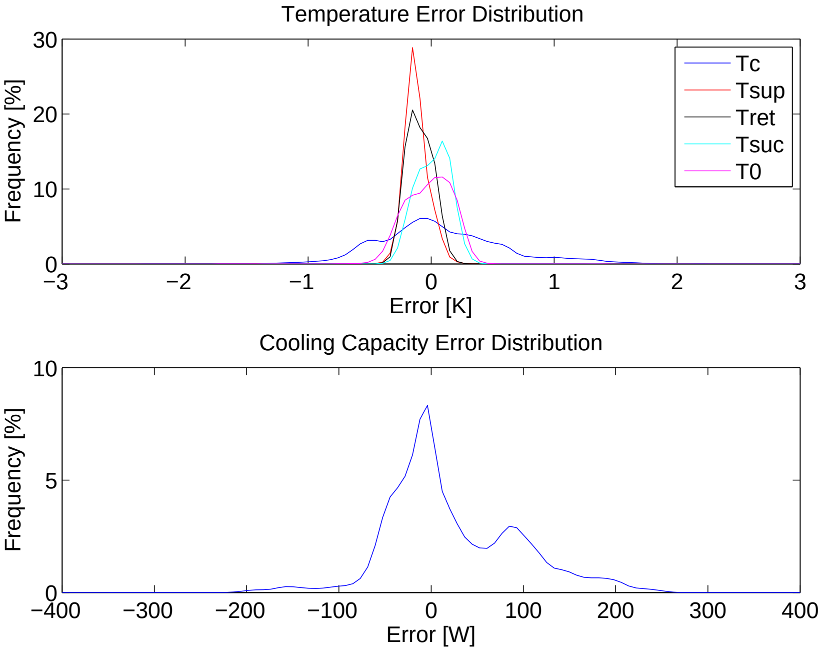


Fig. 2.5: Model error distribution

1.5 Contributions

The contribution from this part of the work is a proposal for control oriented model of a Star Cool reefer container, that verifies that the methods for control oriented modeling of refrigeration systems in car air conditioners presented in [11] also holds for more complex systems. The proposed model is a collection of components, which have been found in the literature and adapted, that together form an accurate representation of the Star Cool reefer.

The model is currently being used by Lodam for development, test and verification of control and fault detection algorithms and it has proved to be a valuable tool that significantly reduces development time. It enables the development of model based controllers that are capable of reaching the set-point faster and have better disturbance rejection than existing controllers. The modular nature of the model and the integration with MATLAB® have proved to be valuable every time the model have needed to be changed or updated to meet the demands of different development tasks.

2 Simulation

This section describes the investigations into different simulation methods for the reefer model described in [Paper C] which was supported by the simulation environment and associated tools that are described in [Paper A] and [Paper B]. Modular and monolithic simulation methods are evaluated and compared with respect to speed and accuracy, using MATLAB®'s built in `ode15s` solver and a simple Variable Step Forward Euler (VS-FE) solver. This reveals that for the reefer model, the VS-FE solver is significantly faster than `ode15s` solver, while keeping the same accuracy.

2.1 Objectives

The work carried out with respect to simulation was initially aimed at achieving the commercial goal of enabling Lodam to develop cutting edge control for the Star Cool reefer and improve the time to market. This required a way to simulate the reefer model and a way to use the model as a basis for development of controllers in an automated way, in order to speed up the development process and it was found that the best option for controller development was by utilizing MATLAB®'s control toolbox. The resulting objective was to make it possible to simulate a modular model of the reefer analogous to WinDali [29] or TRNSYS [28], but inside MATLAB®, in order to achieve synergy with MATLAB®'s controller development tools. This resulted in the simulation environment described in [Paper A], where it was found that there was a theoretical increase in simulation speed, when simulating the model in a modular configuration, using the multi-rate method, instead of simulating the model as a monolithic block. This resulted in the scientific objective of investigating the theoretical increase in simulation speed and this work is described in [Paper B].

2.2 Simulation Environment

The most important item among the modeling tools is the simulation environment that consists of a model loader and a range of different simulation functions that can simulate the model in monolithic or modular configurations. The model loader parse model definitions written in Extensible Markup Language (`XML`) and verify the integrity of the connections between component models. Built into the simulation functions are profilers that allow the user to retrieve and plot information on the simulation time and the number of steps required for each of the component models, after a simulation has been completed. The essential features and results for these tools are described in the following.

Model Loader and Simulation Model Description Language

The simulation model for the reefer consists of a collection of component models where each model consists of an `m` file containing the input/output equations, an `XML` file describing the input/output properties of the model, its execution mode, and the name of the corresponding `m` file. The interface of the `m` file is compatible with the solvers of MATLAB®'s ODE suite [33] and have the form $\dot{x} = f(x, u, t, p)$ such that they may be simulated using all of MATLAB®'s built-in solvers [70]. In the `XML` file it can be specified which solver to use, allowing each component to be simulated by the solver best suited for the job. A similar approach was demonstrated for electro mechanical systems in [36] and [37] where a system is divided into sub systems by their time constants and simulated with a solver and step size that is appropriate for each sub system. The syntax for writing the component model definition `XML` file is shown in Listing 2.1.

Listing 2.1: Component Model Syntax

```
 <?xml version="1.0" encoding="UTF-8"?>
<component name=["Component Name]">
<inputs>
  <input name=["Input1 Name"] type=["Input1 Type"] description=["Input1 Description"]/>
  .
  .
  <input name=["InputN Name"] type=["InputN Type"] description=["InputN Description"]/>
</inputs>

<states>
  <state name=["State1 Name"] type=["State1 Type"] default=["State1 Start Value"]
    description=["State1 Description"]/>
  .
  .
  <state name=["StateN Name"] type=["StateN Type"] default=["StateN Start Value"]
    description=["StateN Description"]/>
</states>

<connectors>
  <connector type=["type"] conid=["Conn. Name"] >
    <{input, state} name=["Name"] type="Physical Entity"/>
    <{input, state} name=["Name"] type="Physical Entity"/>
   </connector>
  .
  .
</connectors>

<control_inputs>
  <input name=["Input Name"] type=["Input Type"] description=["Input Description"]/>
  .
  .
</control_inputs>

<simulation method={"ode15s", "call"} call=[".m File Function Name"]/>
<filename>[.m File Name]</filename>
</component>
```

The `<inputs>` section contains a list of the inputs to the component model, and the inputs must be listed in the same order as they occur in the input vector of the model function in the `m` file. Each input has a name, a type used for type-checking when inputs are connected, and a description. The `<states>` section is similar to the input section but it describes the states of the function and a default value must be declared for each state. The default values are used as the initial value for the states in simulations environment, but it is also possible to use a custom set of initial values if needed.

The inputs of each sub component must be connected to a state in another component and this is done in the `XML` file that defines the structure of the simulation model, shown in Listing 2.2.

Listing 2.2: Simulation Model Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<simulation_model name=["Simulation Model Name"]>
<component_path>[Component Library Path]
</component_path>

<components>
  <component name=["Component Model Name"] file=["Component Model XML File Name"]/>
  .
  .
  <component name=["Component Model Name"] file=["Component Model XML File Name"]/>
</components>

<connections>
  <connector name=["Connector Name"]>
    <component name=["Component Name"] conid=["Connection Name"]/>
    <component name=["Component Name"] conid=["Connection Name"]/>
  </connector>

  <connection name=["Connection Name"]>
     <component name=["Component Name"] type="Input" input=["Input Name"]/>
     <component name=["Component Name"] type="Output" output=["Output Name"]/>
  </connection>

</connections>
</simulation_model>
```

For refrigeration systems the number of connections needed to connect the component models are large. Each refrigerant pipe requires three states; Pressure, enthalpy and mass flow and in order to avoid making a connection for three states on every pipe in the simulation model definition signals may be grouped together in connectors in the component model definitions. They allow the user to connect a set of signals from one component model to a similar set on another component model in one operation. Type checking is performed by the model loader such that signals of the same type are connected and such that inputs always are connected to states. If the user tries to connect two incompatible connectors the model loader returns an error and a description of the nature of the problem.

During simulation a global state vector $\mathbf{X}$ is used to store the collection of states from the component models and the model loader creates mapping matrices for each of the component models that maps its state and input vectors to the global state vector. The advantage of this approach is that matrix multiplications are very fast in MATLAB® and therefore the exchange of information between the component models are fast and efficient during simulation which keeps the overhead low. The matrix $\mathbf{Z}_k$ maps from $\mathbf{X}$

to the component model x-vector $\mathbf{x}_k$ such that

$$\mathbf{X} = \mathbf{Z}_k \cdot \mathbf{x}_k \tag{2.1}$$

$$\mathbf{x}_k = \mathbf{Z}_k^T \cdot \mathbf{X} \tag{2.2}$$

where the index k denotes the component model number. Component model inputs $\mathbf{u}_k$ are mapped from $\mathbf{X}$ by the connection matrix $\mathbf{CM}_k$ such that

$$\mathbf{u}_k = \mathbf{CM}_k \cdot \mathbf{X} \tag{2.3}$$

Because $\mathbf{CM}_k$ maps from state variables that reflect a physical value to inputs that take a physical value, it is a zero-one matrix and must have exactly one "1" in each row. This is checked by the model loader when it has created the mapping matrices such that all inputs are connected to exactly one state, thus ensuring the integrity of the model. The information extracted from the model definition files are combined in a structure by the model loader and returned to the MATLAB® workspace.

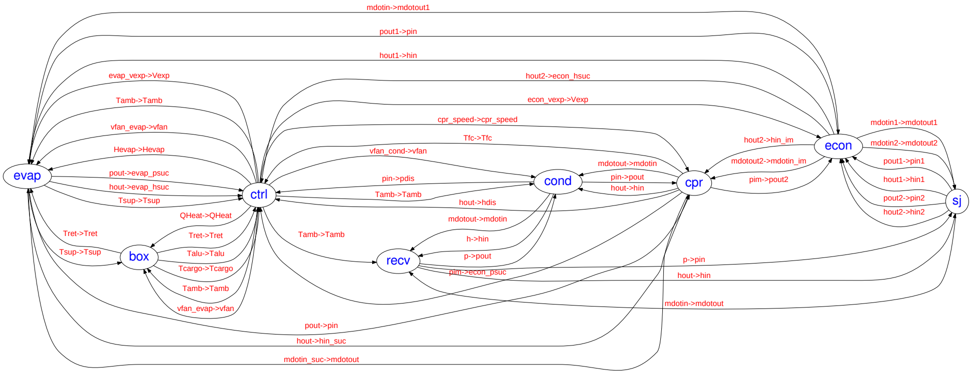


Fig. 2.6: The detailed layout of the simulation model.

An image with the model structure is created every time the model is loaded in order to provide the user with a comprehensible overview of the connections between the component models. The structure for the reefer simulation model is shown in Figure 2.6. The component models are annotated with their names and the connections are annotated with the names of the state and input on either end of the connection and an arrow that indicates the direction of the connection.

Modular and Monolithic Simulation

The difference between modular multi rate simulation and monolithic simulation of the reefer model is investigated in [Paper A] and [Paper B], and this work focus on the difference in speed and accuracy of the two methods. This research is enabled by a monolithic wrapper function that combines the output of the component model functions into on large output vector, using the mapping matrices in the model. The interface of the wrapper has the same form as the component models and is compatible with standard ODE solvers such that it can be simulated by a solver.

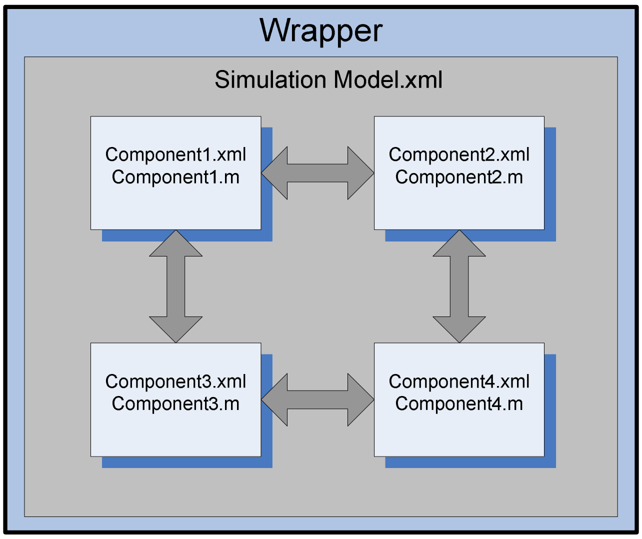


Fig. 2.7: The monolithic wrapper.

Each time the monolithic wrapper function is called, it iterates through the list of component models and calls each of them with their respective input and state vectors. The outputs are then combined in the global state gradient vector $\dot{\mathbf{X}}$ and returned to the solver. This means that the model can be simulated as a monolithic block without changing the model definition or rewriting the model equations and that ensures that a true comparison between the two methods can be achieved. Figure 2.7 shows a diagram of how the monolithic wrapper encapsulates the component models and handles the connections between them.

The model may also be simulated as a modular system, using the multi rate method, where the component models are simulated individually by a numerical solver over a fixed period after which the global state vector is updated. This corresponds to inserting zero order holds between the component models as shown on Figure 2.8, but the advantage is that each component can be simulated using a solver and a step size that is just right for that particular component. In a system where there is a large difference between the time constants in the component models, significant increases in simulation speed can be achieved because slower components do not have to be simulated with an

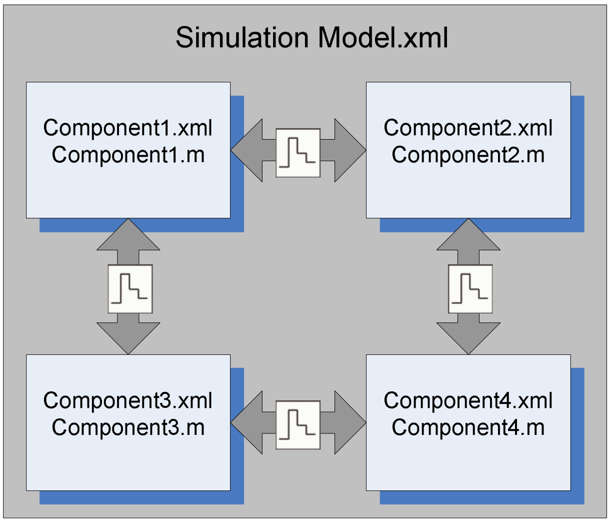


Fig. 2.8: .

unnecessarily small step size, which reduces the computational load. The drawback of the approach is that the zero order holds introduces abrupt changes on the inputs to the component models, which results in larger gradients on the states that requires a smaller step size in the simulation in order to achieve the necessary accuracy. This can be remedied by using a higher order extrapolation algorithm than the zero order hold that is currently used but this can result in numerical instability for tightly coupled fast and slow component models [35].

This means that there exists a trade-off between accuracy, stability and speed that must be considered when applying the multi-rate method. An example of the impact of inserting ZOH's in series with the component models is shown on Figure 2.9 where the frequency responses of a simple linear system has been augmented with varying delays. The delay very significantly reduce the gain margin of the system close to the frequency $\frac{D}{2}$ where the delay alone have decreased the phase by 180°. Consequently, it is necessary to ensure that the systems gain crossover frequency is lower than $\frac{D}{2}$ by a safe margin, in order to ensure stability, but also to ensure that the impact on accuracy is low. Theoretical analysis of the problems regarding optimal selection of the data exchange rate in modular simulation can be found in the literature [71, 36, 35], but for the reefer model it has not been necessary to apply advanced analysis, because the dynamics of the system are slow compared to the data exchange rate. This means that it is probable, that the simulation speed can be increased through the application of more advanced data exchange scheduling, but that is outside the scope of this project.

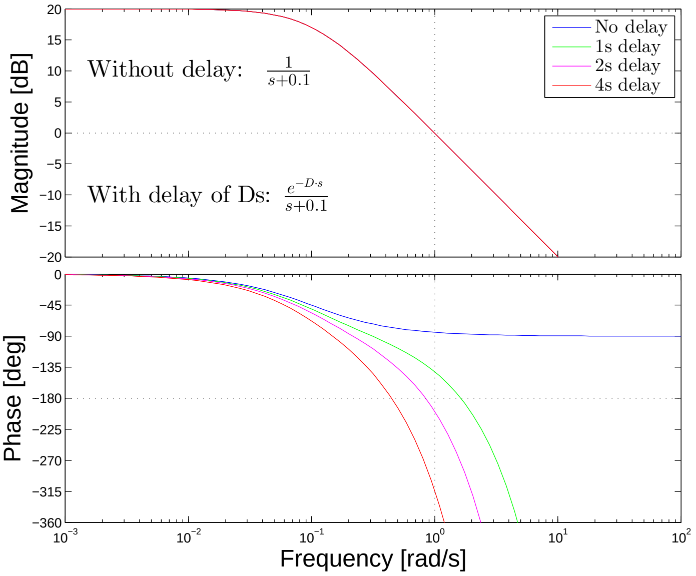


Fig. 2.9: Impact of ZOH delays on a linear system.

Linearization

When using the model for controller development, it is necessary to have a linear representation of the system model, because it is required by controller development methods such as H_∞ [72] methods or Linear Parameter Varying methods [73, 74]. In fact, if the system is highly nonlinear the linear model will only be valid for small perturbations and in this case it is necessary to linearize the nonlinear model over a wide range of operating points. Doing that by hand would be too time consuming so an automatic method is required and therefore the simulation environment is supplemented with a linearization tool, that can linearize the model in an operating point. The algorithm used is the same as the one used to linearize SIMULINK® models [75], where the model states and inputs are perturbed around an operating point, resulting in a linear state space model. The monolithic wrapper is used during linearization in order to avoid the delays caused by the zero order hold between the component models. Normally the model includes a controller and as such the model has no inputs, so it is possible to exclude one or more component models from the linearization. This can also be used if a state space model for a subset of the components in the model is needed. For example, for design of a controller for the injection of refrigerant in the evaporator, all components except the evaporator can be excluded.

VS-FE Solver

During the work on modular and monolithic simulation of the model it was discovered that the `ode15s` solver has a significant startup overhead which impacted the results in modular simulations where the solver was called for each of the overall multi rate steps during the simulation. In order to investigate this impact a simple Variable Step Forward Euler (VS-FE) solver, that is an adaptation of text book solvers presented in [76] and [44], was introduced. The step size selection algorithm was adapted to work on systems with large numerical differences of the states, by using a normalized second order derivative for step size calculation. The local error E for the first order explicit Euler method is given by

$$E = \frac{1}{2}h^2|\ddot{y}| \tag{2.4}$$

where h is the step size and $\ddot{y}$ is the second order derivative of the state. Substituting E with the acceptable local error tolerance TOL and isolating h yields the step size that limits how much the first order gradient is allowed to change in each step and therefore also the magnitude of the local error.

$$h = \sqrt{\frac{2 \cdot TOL}{|\ddot{y}|}} \tag{2.5}$$

TOL is the absolute tolerance for the second order derivative of the system state, thus bounding the step size with respect to how fast the first order derivative changes and thereby bounding the local error.

The states in the container model are numerically very different and therefore the second order derivatives are also numerically very different. This causes the states that are numerically small to have the smallest second order derivatives, even during fast changes for that particular state. The consequence of this is that states that are small may have a relatively large second order derivative, but it will still be smaller than the second order derivatives of the larger states, and that can result in numerical instability for states of small magnitude. For the reefer model the notion of an absolute error tolerance is therefore impractical and this is addressed by normalizing the second order derivative with respect to the size of the states as shown in (2.6).

$$\ddot{\mathbf{y}}_{norm} = \frac{|\ddot{\mathbf{y}}|}{|\mathbf{y}| + 1} \tag{2.6}$$

$$h = \sqrt{\frac{2 \cdot TOL}{max(\ddot{\mathbf{y}}_{norm})}} \tag{2.7}$$

The drawback of normalizing the second order derivative is that as a state approaches zero the normalized second order derivative will approach infinity and result in very small steps. This issue is addressed by adding one to the state vector and therefore as a state approaches zero, its normalized second order derivative will approach the real second order derivative. Due to the normalization of the second order derivatives, TOL is a measure of error relative to the state size for large states and it approaches a measure of absolute error as the states approach zero. It was expected that the proposed VS-FE solver would be less accurate than the built-in MATLAB® solvers but it was found that there was no loss of accuracy but instead a significant increase in simulation speed.

2.3 Results

The work on simulation of the reefer model focused on the differences between modular and monolithic configurations and solver choice. A range of experiments was conducted to uncover the differences in speed and accuracy, and the nature of the errors introduced by the zero order holds in the modular approach. The main results can be seen in Figure 2.10 where the mean error, the maximum error and the simulation times are shown. The

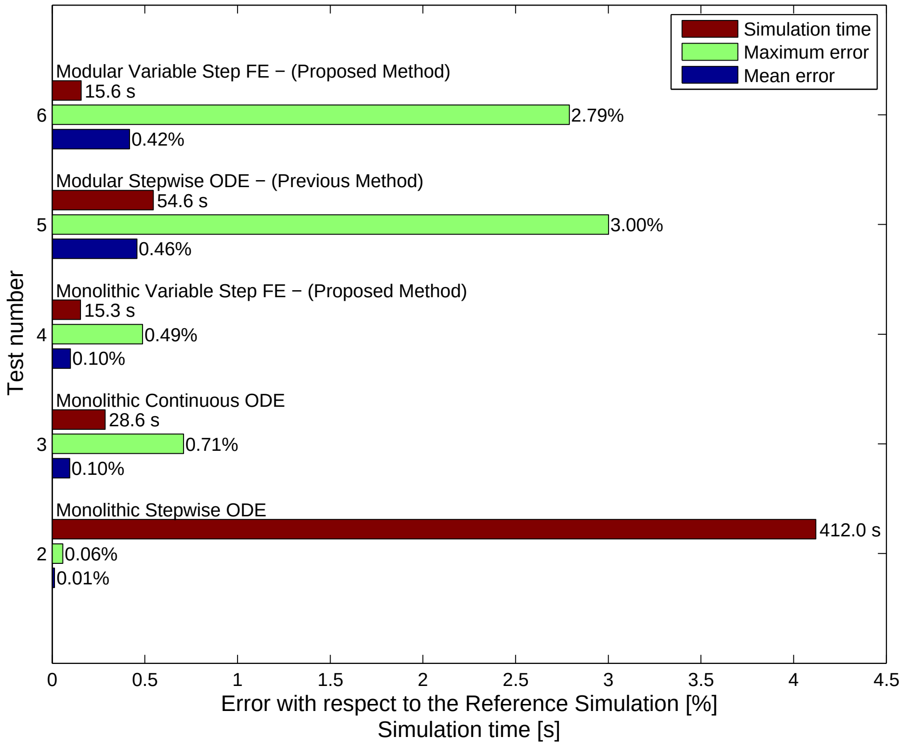


Fig. 2.10: Comparison of test results for monolithic and modular methods.

model is simulated using different methods with a sequence that changes the cooling capacity every 100 seconds to make sure that the model is properly excited. The *Modular Stepwise ODE* method was presented in [Paper A] and the *Modular Variable Step FE* method is the improved method using the VS-FE solver that was presented in [Paper B]. The results show that the proposed VS-FE solver is able to maintain accuracy while simulating the model at higher speed and the same is true for the monolithic configuration illustrated by *Monolithic Variable Step FE* and *Monolithic Continuous ODE*. The large startup overhead for the `ode15s` solver can be seen from the comparison of the *Monolithic Continuous ODE* and the *Monolithic Stepwise ODE* test where the only difference is that the ODE solver is called once for the first test and for every of the 1000 time steps in the second test. The large overhead is caused by the internal linearization that is done in `ode15s` to obtain the Jacobian of the system [33] before simulation can begin. While the `ode15s` solver achieves higher accuracy with fewer steps than the VS-FE solver the overhead exceeds this advantage when used for many successive short simulations. This can be seen for the detailed results shown in Table 2.1 where it can be seen that the VS-FE solver uses approximately twice the steps compared to the ODE solver in the modular configuration. Another interesting result is that the VS-FE solver manages to complete the monolithic simulation of the system at a lower error and a higher speed than the ODE solver even though the ODE solver is only called once and therefor the startup overhead is not significant. This is caused by the `ode15s` solver's automatic update of the Jacobian when it is deemed too inaccurate. The implicit nature of the `ode15s` solver makes it well suited for stiff systems, but also slow, compared to explicit methods like the VS-FE solver. The stiff solver was used because the model, under certain operating conditions, can exhibit stiff behavior and in this case the choice of a stiff solver, instead of an explicit method such as `ode45`, ensures a robust simulation. While `ode45` not normally is used to simulate the model, it can run reliably in the present test scenario and the results can be seen in Table 2.1. The accuracy of `ode45` is comparable to `ode15s` and the VS-FE solver and it is faster than `ode15s` but slower than the VS-FE solver. The number of steps used by `ode45` is close to the number of steps used by the VS-FE solver, and that indicates that there is a larger overhead in `ode45` than in the VS-FE solver.

No	Description	Solver	Time	Mean Err.	Max Err.	Steps
2.	Monolithic Stepwise ODE	`ode15s`	412s	0.013%	0.058%	109683
3.	Monolithic Cont. ODE	`ode15s`	28.6s	0.095%	0.709%	7621
4.	Monolithic VS-FE	VS-FE	15.3s	0.099%	0.488%	4236
5.	Modular ODE	`ode15s`	54.6s	0.457%	3.002%	15247
6.	Modular VS-FE	VS-FE	15.6s	0.418%	2.790%	34072
10.	Modular Runge-Kutta(4,5)	`ode45`	29.9s	0.433%	2.828%	34409

Table 2.1: Simulation results compared to the monolithic reference.

The modular simulation approach has been found to be efficient for simulation of the reefer container model and considering the nature of the system this is in line with other results from the literature [71, 35, 36]. The speed of the VS-FE simulation in this example is roughly 64 times faster than real time which is better than the requirement that was set in Section 4, of a speed that was at least an order of magnitude faster than real time. Furthermore, it was required that the error introduced by the chosen simulation method was smaller than the variation between refrigeration units in the field. This variation is due to hardware tolerances and wear and tear os the system, and while it is difficult to ascertain the exact size of the error it is deemed to be larger than 5%. In [Paper D] the model was simulated for 21 days at different loads, fulfilling the last requirement from objective III.

Therefore, objective III has been achieved.

2.4 Contributions

A proposal for an efficient and accurate simulation method for the nonlinear model of the refrigeration container. The analysis of solver choice demonstrates that, for modular simulation, using a simple solver can be faster than more advanced solvers, without sacrificing accuracy. The simulation tool provides a necessary and solid base for development and test of controllers in the last phase of the project.

The contributions from this work related to the commercial interests, are the simulation environment and its associated tools, which will enable Lodam to improve and speed up the efforts for model based controller development for the Star Cool reefer. The tools are flexible and extensible, which enables integration with current and future controller development tools.

3 Energy Optimizing Control

In this section the work regarding development of an adaptive temperature controller, that aims to reduce the amount of energy used by the reefer during normal operation, is described. The controller is described in [Paper D] and it consists of a model predictive controller used for long term optimization, a linearizing low level controller, a parameter estimator that estimates the unknown parameters and states of the cargo and a predictor for future ambient temperature.

3.1 Objectives

The primary objective of the controller is that is must be able to keep the temperature of the cargo within its limits such that damage to the cargo can be avoided and the secondary objective is that the controller should minimize energy consumption. Two potential methods for reduced energy consumption must be investigated; The first method is adaptation to daily variations in ambient temperature [62] and the second method is a reduced ventilation rate [54]. Exploitation of the daily variations in ambient temperature has the potential to reduce energy consumption through the utilization of the thermal inertia of the cargo as a storage for cooling. The efficiency of the refrigeration system is inversely proportional to the difference between the ambient temperature and the temperature inside the container, and by cooling extra during low temperature periods and less during high temperature periods the overall energy consumption of the system can be reduced. A reduced ventilation rate has the potential to reduce the energy consumption because the fans would use less power but also because the energy used by the fans eventually ends up as heat inside the container and must be removed by the refrigeration system. This means that a reduced ventilation rate will result in less energy consumed by both the refrigeration system and the evaporator fans.

3.2 Controller Design

Model predictive control is used to exploit the daily variations in ambient temperature because it is able to consider the trajectory of the predicted future ambient temperature. The required length of the prediction horizon is at least one period of the variation which is 24 hours that must be divided into discrete steps for the controller to optimize over. The total number of steps should be kept as low as possible in order to minimize the optimization problem but if the MPC has to control the refrigeration system directly, the required resolution would result in an optimization problem that is infeasible to solve on a normal computer. Therefore, the proposed controller is a two level design with a lower level linearizing controller that controls the compressor and expansion valves such that the system delivers the requested cooling capacity and a high level MPC that outputs a cooling capacity request. The structure of the controller can be seen on Figure 2.11. It was chosen to use cooling capacity as the interface between the two controllers because it has a linear influence on the temperature in the box and that enables the use of a linear model for the MPC. The model was created using a modified version of the component model of the cargo hold that has cooling capacity as an input instead of supply air temperature and a simple PI controller for testing. The modular modeling framework was used in order to make simulation, testing and linearization simple and the structure of the reduced model can be seen on Figure 2.12. When linearized without the controller, the resulting model have three inputs which are fan speed, cooling capacity and ambient temperature, and three outputs which are the

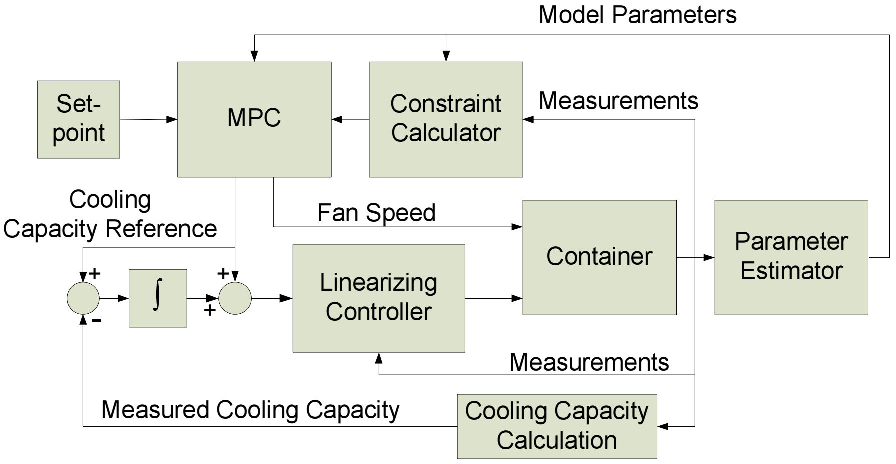


Fig. 2.11: Controller Block Diagram

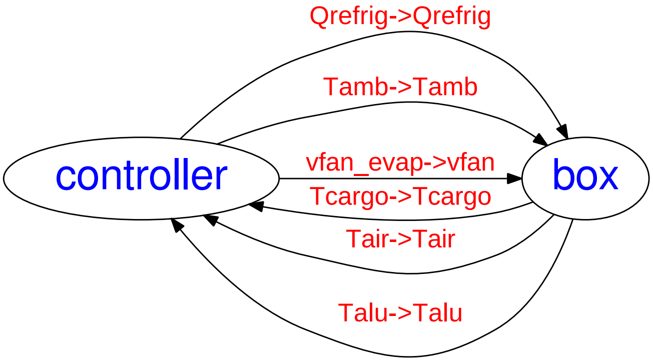


Fig. 2.12: Structure of the reduced model used in the MPC

temperatures of the air, the floor and the cargo. The linearizing controller between the MPC and the container shown on Figure 2.11 must follow the cooling capacity reference as accurately as possible such that the system seen by the MPC resembles the reduced model. This is achieved through a nonlinear feed-forward that gives the compressor speed and expansion valve opening degree that is required to provide the requested cooling capacity and integrators to remove any errors. Because the cooling capacity reference is discrete, it is impossible to follow due to the time constants of the system and therefore any accumulated cooling capacity error that is due to lag is removed by the integrator shown in Figure 2.11.

The MPC minimize an objective function that must represent the cost of running the system at its current operating point and therefore the system COP was mapped for different ambient temperatures, giving the result visible on Figure 2.13. The dotted line is the COP of the system when the compressor is running in PWM mode, where the compressor is stopped and started with a duty cycle that matches the required cooling

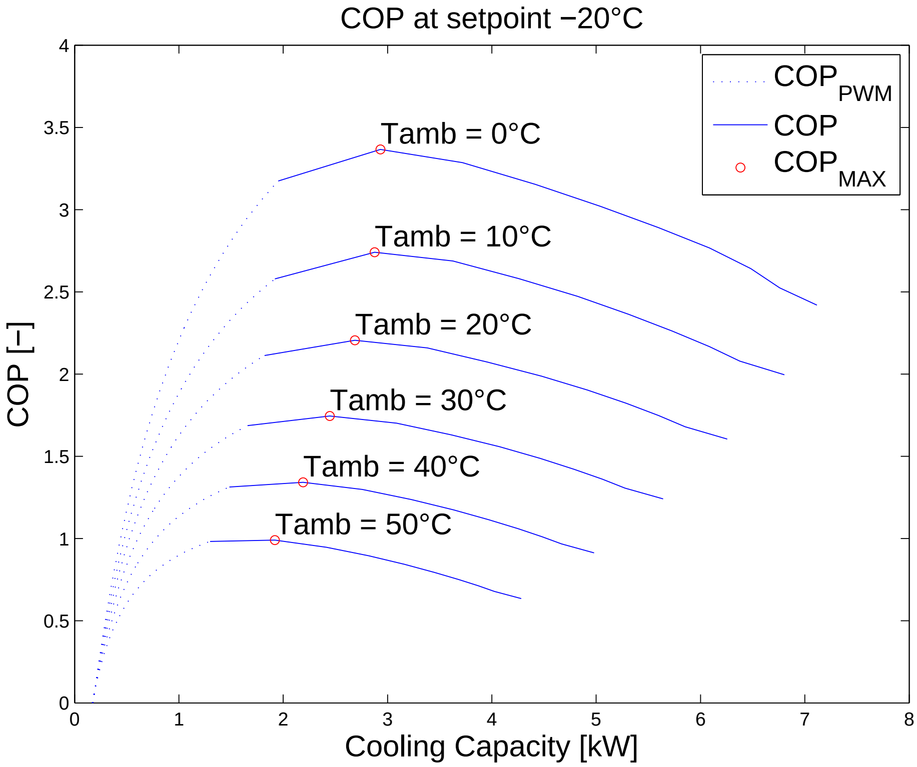


Fig. 2.13: The COP of the refrigeration system at varying ambient temperatures

capacity. The reason for the sharp decline of COP in PWM mode is that the evaporator fan continues running, while the compressor is off and therefore the contribution from the fan compared to the cooling capacity is increased as the duty cycle approaches zero. The ambient temperature has a big impact on both the level and the shape of the COP curve and in order to effectively exploit this, the objective of the MPC must accurately reflect the cost over the length of the prediction horizon. This requires knowledge of the future ambient temperature which is not available, and therefore the predictor outlined in section 3.3 is used.

The predicted ambient temperature is used to select the appropriate COP curve during operation and the COP data is converted to a series of linear segments in the form $y = ax + b$ that is used in an epigraph representation of the cost of running the compressor in the constraints. The epigraph representation of the nonlinear nature of the COP ensures that a linear method can be used to solve the problem and this contributes to keeping the computational load of running the controller low. This is important in order to enable the controller to be run on embedded hardware, thereby making the controller commercially interesting. Multi rate optimization step length [77] is introduced, to further reduce the computational load, while fine grained control of the present step is ensured.

$$P_c(k) + V_{fan}(k) \cdot 195 + T_s(k) \cdot 10^4 \tag{2.8}$$

The objective function in Equation (2.8) consists of the epigraph variable that represents the compressor power P_c, the cost of running the fan on low speed V_{fan}, and the slack variable T_s that is introduced as a soft constraint in order to ensure that the problem can still be solved if the temperatures drift outside the constraints due to model inaccuracies.

$$V_{fan}(k) \cdot Q_{min}(k) \leq Q_{cool}(k) \leq V_{fan}(k) \cdot Q_{max}(k) \tag{2.9}$$
$$V_{fan-min} \leq V_{fan}(k) \leq 1 \tag{2.10}$$
$$Q_{cool}(k) \cdot a_1(k) + b_1(k) \leq P_c(k) \tag{2.11}$$
$$Q_{cool}(k) \cdot a_2(k) + b_2(k) \leq P_c(k) \tag{2.12}$$
$$Q_{cool}(k) \cdot a_3(k) + b_3(k) \leq P_c(k) \tag{2.13}$$
$$Q_{cool}(k) \cdot a_4(k) + b_4(k) \leq P_c(k) \tag{2.14}$$
$$0 \leq T_s(k), \tag{2.15}$$
$$T_{cargo-min} - T_s(k) \leq T_{cargo}(k) \leq T_{cargo-max} + T_s(k), \tag{2.16}$$
$$T_{air}(k) < T_{air-max} + T_s(k) \tag{2.17}$$

Equations (2.9) to (2.17) are the constraints where (2.9) and (2.10) are the actuator limits, and (2.11) to (2.14) are the constraints used in the piecewise affine epigraph representation of the COP. The temperature constraints for the cargo and the air are given by Equation (2.16) and (2.17).

3.3 Parameter Estimation

Ambient Temprature

The selection of the correct parameters for the controller COP constraints depends on the predicted ambient temperature, and because the reefer has no access to weather forecasts the ambient temperature must be predicted by the reefer itself. A simple phase locked loop (PLL) is used to align the phase to the measured ambient temperature from the past 24 hours and using this phase the next 24 hours is predicted, assuming a sinusoidal temperature trajectory. In Figure 2.14 the predictions for the first 100 hours of a container's journey from a Danish port are shown, with five hours between each of the predictions. In the beginning the predictions are inaccurate because the phase cannot be found from the few samples that are available but within the first 24 hours of the trip the estimates begin to match. It is not an accurate match, but the most important thing is that the phase is right, because that will enable the MPC to schedule extra cooling at the right time.

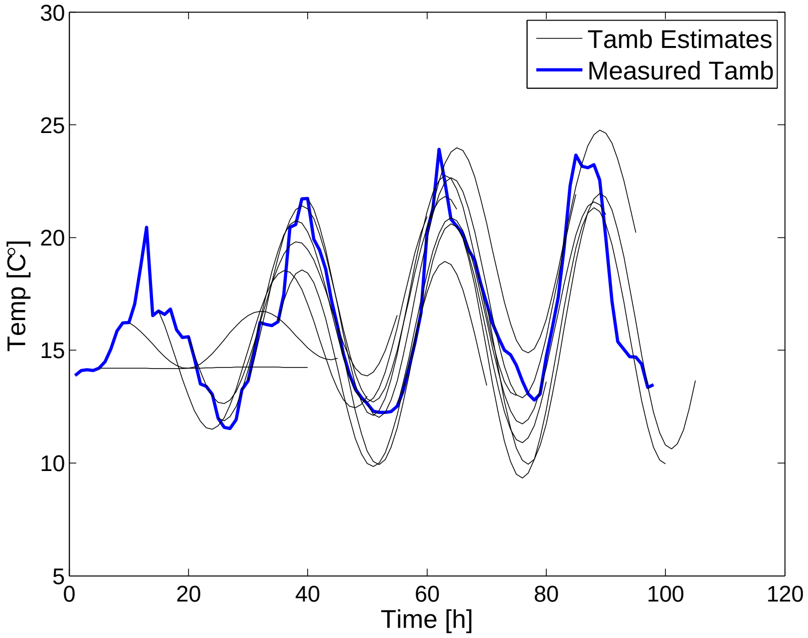


Fig. 2.14: Prediction of future ambient temperature

Cargo State and Paramter Estimation

Reducing the energy consumption of the reefer is the focus of this research, but the most important objective of the controller is to ensure that the temperature of the cargo stays within its constraints, and while the parameters of the refrigeration system are well known, there is no direct reliable temperature measurement of the cargo and there are large uncertainties on its parameters. This makes it very difficult to ensure that the constraints are not violated without reducing the allowed temperature range in order to create safety zones which would impact the potential energy savings negatively. Therefore, an estimator for the latent cargo parameters was introduced. The estimator is based on the model equations for the cargo hold and the cargo, and it estimates the following latent variables:

Description	Unit
Cargo heat capacity, C_{cargo}	J/K
Cargo heat transfer coefficient, α_{cargo}	W/K
Cargo temperature, T_{cargo}	°C

The estimator is based on the assumption that the system is Linear Time Invariant (LTI) and it derives that value of the parameters C_{cargo} and α_{cargo} from two measurements of the cargo at either end of a long period of sustained cooling at high capacity. The

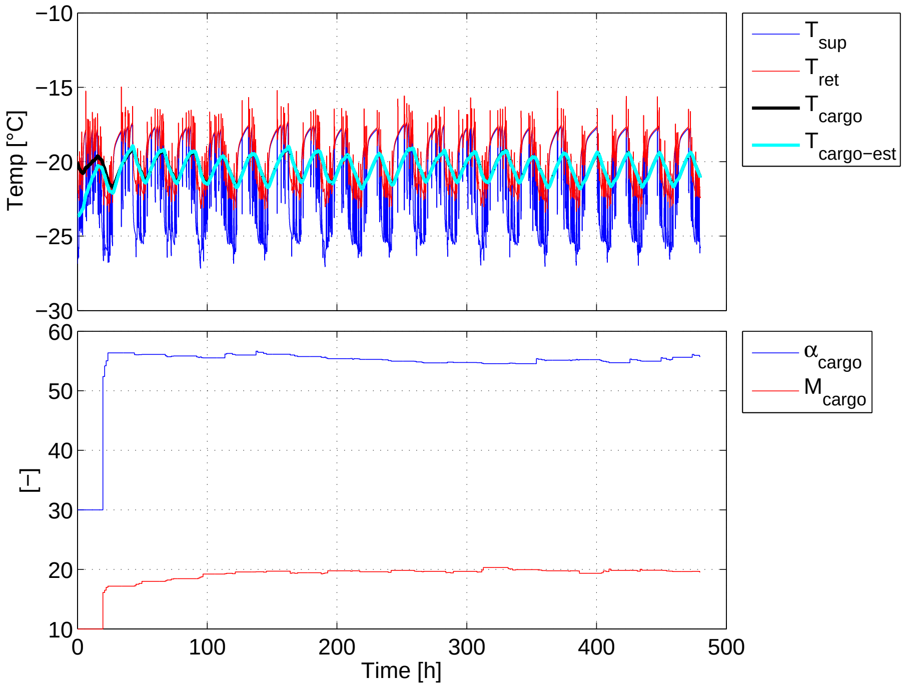


Fig. 2.15: Startup of the cargo parameter estimator

reason for this is that in order to estimate the cargo dynamics they must be excited, and since the thermal inertia of the cargo is very large, a significant amount of cooling must be applied before the dynamics can be estimated reliably through the air temperature. A first order filter with the air temperature as input and a time constant calculated from C_{cargo} and α_{cargo}, was used to estimate the cargo temperature. Therefore, the estimate of the cargo temperature will only be accurate if the parameters C_{cargo} and α_{cargo} are correct. For a thorough description of the parameter estimator, please see [Paper D Section 2.2]. Figure 2.15 shows the result of the estimator running on data from a simulation of the model, with added measurement noise. The estimated heat capacity of the cargo has been converted to mass in metric tons, using the specific heat capacity for bacon which is $1050\frac{J}{kg \cdot K}$ [78]. The values of α_{cargo} and M_{cargo} have been scaled by 10 and 1000 respectively, in order to better show the details of the results. In this test, it is expected that α_{cargo} levels out at 55 and that M_{cargo} level out at 20, and from the first update of the estimate the values are close to the real value. The resulting estimated cargo temperature tracks the real value well and while there is some inaccuracy in the parameter estimates, it is deemed good enough for feedback control.

It is important that the estimator finds a set of parameters that makes the linear model behave the same way as the real system. If the model is inaccurate the estimated parameters may also be inaccurate to compensate for the model inaccuracy, but as long as the behavior of the model matches the real system this is not important. The parameters from the estimator is applied to the linear model of the cargo hold before every execution of the MPC and that ensures that the controller is able to perform well on different types of cargo.

3.4 Results

The controller was tested in two different configurations, such that the energy savings from reduced ventilation and exploitation of ambient temperature cycles could be isolated. Each configuration was run at three different ambient temperatures and the result from the first 50 hours of the reference test and the two MPC configurations at 20°C ambient temperature can be seen on Figure 2.16. The reference test is a PI controller that sets the cooling capacity reference to the linearizing controller and it is shown in the two upper panels. This shows how the reefer would normally run with its current capacity controller; the cooling capacity is increased during the hotter periods, which is the opposite of the desired behavior for reduced energy consumption.

The two panels in the middle show the results from the MPC where the fans are running continuously and it can be seen that the MPC uses the cargo's thermal inertia to move cooling from cold to hot periods. The air and cargo temperature rise to its upper constraint during the period where the ambient temperature is at its highest. The compressor is running in PWM most of the time because there is nothing to gain by running the compressor faster, at a lower efficiency, if it is not possible to turn off

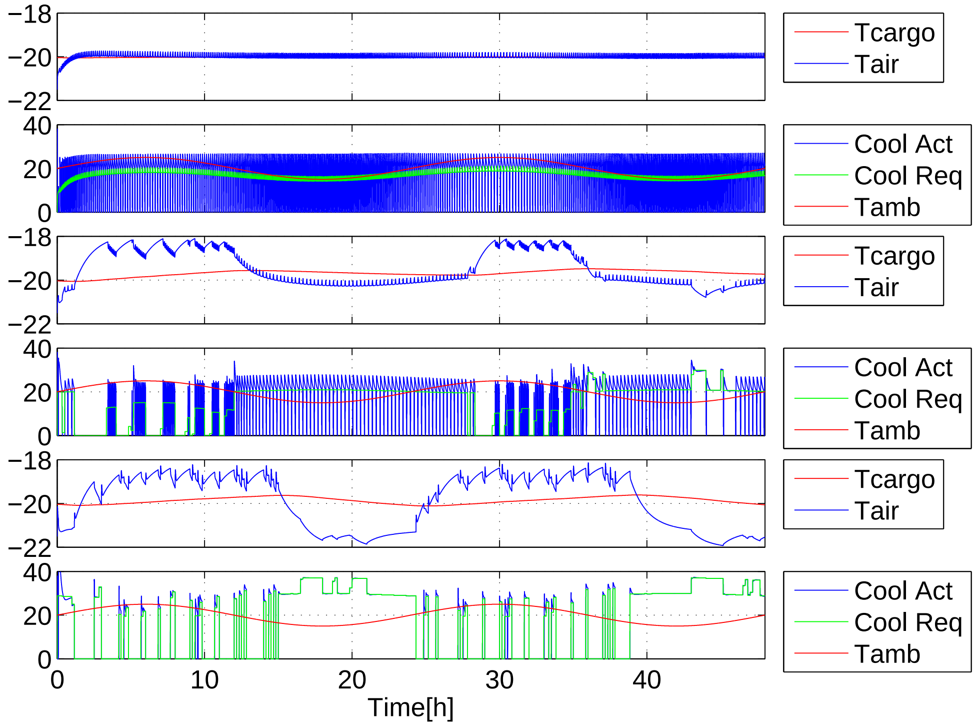


Fig. 2.16: Test results for the reference and the two test scenarios at 20°C ambient temperature.

the fans for a longer period afterwards.

The result from the test where the MPC is allowed to turn the fan off when the compressor is not running, is shown in the two panels in the bottom of the figure. The compressor is no longer running PWM but instead at a higher capacity that allows the compressor and fans to be turned off for longer periods, thus saving power. During the periods of high ambient temperature the compressor is only turned on to keep the air temperature within the constraints, while the cargo temperature is slowly increasing and this show that the controller behaves as intended.

The tests were all executed using the full simulation model of the container during a 21 day simulation, in order to minimize the impact of any errors at the start of the simulation. The model was simulated in a modular configuration and the full set of tests was completed in four days, which saved a lot of time compared to a real life test. The power savings found in the six tests are listed in the following table:

Ambient Temperature	Fans Always ON	Fans ON/OFF
10 ± 5°C	2.53%	21.9%
20 ± 5°C	3.07%	11.1%
30 ± 5°C	2.70%	3.96%

From the results it can be seen that the potential power savings are dependent on the operating conditions and the reason for this is found in the way the refrigeration system is designed. It must be able to cool down a hot cargo within reasonable time and therefore it runs at a small fraction of the available capacity when the cargo is at the set-point, which is also where the system is most efficient. The COP curve for the compressor alone is monotonically decreasing as the speed increases but when the power from the fans is added, the COP curves shown in Figure 2.13 with a maximum in the lower capacity range emerge. This means that when the amount of cooling needed to keep the set-point matches the most efficient capacity, the potential energy reduction from turning off the fans is zero and this is reflected by the results, that shows that the energy savings decrease as the ambient temperature increases. When the most efficient mode of operation is running the system continuously, the cargo can still be used as a cooling storage and therefore there will always be something to gain from using this control strategy.

Studies of exploitation of large thermal inertias for reduction of energy consumption or handling of peak demands in other applications have shown that it is often possible to store significant amounts of cooling, leading to savings up to 30% or much improved prevention of refrigeration system saturation. In [64] a learning based algorithm was used to prevent saturation of system capacity in a supermarket refrigeration system based on previous saturation events. This method has the advantage of being simple and easily applied to an embedded system, but it would require some modifications if it were to be used on a reefer due to the large variation in set points and cargo type that is transported. Different sets of learning data for each set point would have to be stored and maintained by the controller and the method should be enhanced to consider the economical aspect of the control problem. It has been shown in many works that the combination of MPC and large thermal inertias can yield considerable savings [63, 62, 79], and this is also reflected by the findings in this study.

In the present study the potential saving for the Star Cool reefer was investigated and it was found that a saving between 2.5% and 3% is possible with a 10K variation in ambient temperature. With a consumption of 8.9MW [80] for all the containers on a ship a 2.5% reduction yields 403.7GJ over a 21 day trip which is roughly 10000kg of heavy fuel oil. This results in a significant reduction in pollution and cost for transportation of refrigerated cargo, that will be even larger if the reduced ventilation rate is applied as well. The six tests presented here show a big difference in potential savings and therefore further experiments should be run to test the entire range of operation for the container, but it is expected that the largest savings will be found where the difference between temperature set-point and ambient temperature is low.

The investigation have uncovered the potential for energy savings in frozen reefer cargo, by using the thermal inertia of the cargo as a buffer, and thereby objective IV is fulfilled.

3.5 Contributions

Proposal for a control strategy that is able to reduce the power consumption of a reefer through the use of the cargo as storage for cooling, and by being able to estimate important cargo parameters and adapt the controller accordingly. A series of optimizations on the MPC reduces the computational load and makes it commercially interesting. Adaptive model predictive control has not previously been used on a reefer, but this work shows that it is feasible, and that significant savings can be achieved.

Chapter 3

Conclusion

The work in this thesis concerns model based control for reefer containers used to transport perishable cargo worldwide. The aim of the project was to reduce the amount of energy consumed by the refrigeration system, without compromising the quality of the transported goods. The results of this work have been summarized in the previous chapters and the details are presented in the four papers that may be found in [Part II]. In this chapter the conclusions of the research are presented and recommendations for future work and possible applications of the research are given.

1 Discussion

A model based controller was developed to handle the problems that exist on refrigerated containers and tested using an accurate nonlinear simulation model of the system. The contributions in this project are divided in three categories: Modeling and Verification, Simulation and Controller Design and Validation. Objectives and requirements were defined in Chapter 1 and subsequently verified in Chapter 2.

Modeling and Verification
Objective I described the need to create a non-linear dynamical model of the reefer with adequate accuracy for development of controllers and objective II described that the non-linear dynamical model of the reefer was required to be formulated such that it could be used as a simulation model for test and verification of controllers. An accurate model of the relevant parts of the container was created using a modular approach that allows components to be changed easily when needed. The used methodology was based on observations done on car air conditioning systems where it was found that the salient dynamics of the refrigeration system are defined by the thermal inertia of the metal in the heat exchangers, and this was shown to also be true for the larger refrigeration system

in the Star Cool reefer. The model was verified against data from a real reefer with cargo. A collection of component models was selected from the literature and combined into a system model of the Star Cool reefer that showed good accuracy on the states important for control. The modeling of the reefer and subsequent verification of the model is presented in [Paper C]. In Section 1.4 it was concluded that the requirements were fulfilled and objectives I and II had been reached.

Simulation

Objective III stated that a simulation method, which met certain demands for speed and accuracy, had to be identified in order to enable a reduction in controller development time. Therefore, it was investigated how to most efficiently simulate the nonlinear model of the reefer, using a purpose made simulation framework for MATLAB®, that enabled simulation experiments in different configurations. An adaptation to a variable step forward Euler solver was made, and when the resulting method was compared to MATLAB®'s built-in solvers it was found to be significantly faster, while keeping comparable accuracy for simulations on the reefer model. It was investigated how monolithic and modular methods impacted the speed and accuracy of simulation on the model, and it was found that the modular method is faster when using MATLAB® solvers. When the adapted variable step forward Euler solver were used, there was no significant difference between the modular and monolithic methods, but they were both faster than when using MATLAB®'s built-in solvers. The simulation framework and its associated tools were used to verify the reefer model, and supported the development and test of the model based controller. The research has shown that it is possible to speed up simulations of refrigeration systems using the modular method and shed light on the implications of solver choice for modular simulation. Today the model is actively used for test and verification of controllers at Lodam, which have resulted in much shorter development time, due to the accuracy of the model and the high speed of the simulation. In Section 2.3 it was concluded that the requirements for simulation were fulfilled and that Objective III had been achieved.

Controller Design and Validation

Objective IV required an investigation of the potential energy savings by using the cargo as a storage for cooling and by reducing the ventilation rate. An adaptive model predictive controller was developed and tested using the simulation model and showed that power savings of up to 21% can be achieved, by reducing the ventilation rate and using the cargo thermal inertia to store cooling. The largest savings were found to originate from the reduced ventilation rate and found to be very dependent on the operating point, but it is assessed that a typical container vessel will be able to save at least 10000 kg of heavy fuel oil during a 21 day trip, if the proposed controller is applied to the reefers on board. A parameter estimator for the unknown parameters and states of the cargo was introduced, and enabled the model predictive controller to fully exploit the maximum allowed temperature variations of cargo and the air in the container. The proposed controller consists of an inner loop controller that handles the fast refrigeration system dynamics and follows a cooling capacity reference that is given by the model predictive controller in the outer loop. This linearizes the nonlinear refrigeration system from the perspective of the model predictive controller and greatly reduces the problem that must be optimized, leading to a lower computational load. A range of techniques were used to further reduce the computational load of the MPC, in order to make it commercially interesting. The non-linear nature of the system COP were embedded into the constraints using a piecewise affine epigraph representation, which ensured that the MPC problem could be solved using linear methods and multi rate optimization step lengths were used to ensure fine grained control, while limiting the size of the problem. In Section 3.4 it was verified that the proposed controller fulfills the requirements and therefore objective IV, and the main objective of reducing the energy consumption through model based control, have been reached.

2 Perspective and Future Work

In this section the perspectives of the scientific work in relation to other fields and recommendations on future work are given.

2.1 Perspective

The nonlinear simulation model is very specific to the Star Cool reefer but due to the modular design it can easily be reconfigured to match other types of vapor compression systems, in for example heat pumps or cold storage facilities. The modeling methodology, where steady state equations for the refrigeration circuit are combined with a dynamical model of the metal in the heat exchangers, has proven to be a good trade off between physical detail and accuracy and it is assessed that this methodology can be used advantageously, for a large class of energy systems that have a phase change in the working fluid.

The work done on simulation has highlighted some implications of the choice of simulation method and solver that may be useful in applications where large systems with very different time constants are simulated. This type of system can be found within transportation, aviation, process industry and energy systems.

The principle of the proposed controller structure, with an inner loop cooling capacity controller and an outer loop optimizing controller, works well for a reefer but this structure could also be used on other refrigeration systems with variable capacity. The parameter and state estimator for the cargo requires that a model of the cargo hold is available to achieve acceptable accuracy. For other types of cold stores where the dynamics of the store itself are known and the details of the goods are unknown, an estimate may be acquired using the proposed estimator and thereby the possibility for optimal exploitation of varying ambient temperature is achieved.

The tools that were developed to support the research will be beneficial to the task of developing a unified, modern controller for the Star Cool reefer and the work has shown that a flexible simulation environment can be built directly in MATLAB®, resulting in strong synergies with the available toolboxes.

2.2 Future Work

A model for the latent heat in moist air and buildup of ice on the evaporator would make the simulation model of the reefer a more useful tool, because it would then be possible to use it for optimization of defrost scheduling, and the injection controller could be tested under more realistic conditions where the dynamics of the evaporator change as ice builds up. It would also be beneficial to add the ability of tracking the location of the oil used to lubricate the compressor that circulates the refrigeration system. It would be particularly useful to be able to predict oil accumulation in the evaporator and the discharge of oil from the compressor, because this could be used to make a controller that ensure proper oil return to the compressor, at all times. The condenser model is too simple and added physics could help improve the accuracy of the modeled discharge pressure.

The modeling and simulation tools would benefit from integration with an automatic parameter tuning tool such as Senstools, [81] that can tune the parameters of a model, based on recorded data from a real system. Advanced data exchange scheduling methods are described in the literature and application of such a method, could provide guaranteed stability of the modular simulation and possibly also a higher simulation speed. Alternatively, the current ZOH data exchange could be replaced by a simpler extrapolation method, such as a first or second order hold, which would increase speed and accuracy but not guarantee stability.

Development of a robust controller to replace the current feedback linearized inner loop capacity controller would improve system reliability under varying system characteristics. Fault tolerant control methods that are capable of self-reconfiguration should be investigated such that the system may handle larger mechanical faults that cannot be handled by the robust controller. Implementing a model predictive controller on an embedded system is deemed to be very demanding for the existing Star Cool controller CPU, and therefore the problem should be reduced further or simpler methods should be investigated. This could be in the form of a self learning controller that was adapted to the present problem or a through a further reduction of the MPC problem. Explicit solutions that reduce the quadratic problem of an MPC to a simple linear function has been described in the literature and could be used to enable implementation of the developed control laws on the current reefers in the field.

References

[1] Lodam, "Lodam electronics homepage," 2013. [Online]. Available: http://www.lodam.com

[2] MCI, "Maersk container industry, starcool refrigerated container specifications," 2013. [Online]. Available: http://www.maerskbox.com/

[3] D. S. C. Limited, "Reefer shipping market annual report," Drewry House, 213 Marsh Wall, London, E14 9FJ,England, 2009. [Online]. Available: http://www.drewry.co.uk/get_file.php?id=1161

[4] Wageningen, "Agrotechnology & food sciences group, quest-leaflet," 2015. [Online]. Available: https://www.wageningenur.nl/upload_mm/d/6/b/2b3cd77e-4c34-4a62-95a9-1142267427de_Questleaflet2008.pdf

[5] S. Jonsson, "Performance simulations of twin-screw compressors with economizer," *International Journal of Refrigeration*, vol. 14, no. 6, pp. 345 – 350, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/014070079190031B. doi:10.1016/0140-7007(91)90031-B

[6] Carrier, "Carrier: Specifications for the primeline reefer," 2006. [Online]. Available: http://www.container.carrier.com/StaticFiles/Carrier.com/Carrier%20Brand%20Sites%20Content/Carrier-Container/Files/Product_Solutions/Container_Equipment/PrimeLine.pdf

[7] ThermoKing, "Thermoking: Specifications for the magnumplus reefer," 2011. [Online]. Available: http://www.na.thermoking.com/content/dam/tki-na/pdf/Marketing%20Literature/60281_MagnumPlus_2011.pdf

[8] B. Rasmussen, *Control-oriented Modeling of Transcritical Vapor Compression Systems*. University of Illinois at Urbana-Champaign, 2002. [Online]. Available: http://books.google.dk/books?id=dQ0jywAACAAJ

[9] E. W. Grald and J. MacArthur, "A moving-boundary formulation for modeling time-dependent two-phase flows," *International Journal of Heat and Fluid Flow*, vol. 13, no. 3, pp. 266 – 272, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0142727X9290040G. doi:http://dx.doi.org/10.1016/0142-727X(92)90040-G

[10] P. Mithraratne and N. Wijeysundera, "An experimental and numerical study of the dynamic behaviour of a counter-flow evaporator," *International Journal of Refrigeration*, vol. 24, no. 6, pp. 554 – 565, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700700000530. doi:http://dx.doi.org/10.1016/S0140-7007(00)00053-0

[11] B. Rasmussen, A. Musser, and A. Alleyne, "Model-driven system identification of transcritical vapor compression systems," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 444–451, 2005. doi:10.1109/TCST.2004.839572

[12] M. Willatzen, N. Pettit, and L. Ploug-Sørensen, "A general dynamic simulation model for evaporators and condensers in refrigeration. part i: moving-boundary formulation of two-phase flows with heat exchange," *International Journal of Refrigeration*, vol. 21,

no. 5, pp. 398 – 403, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000911. doi:10.1016/S0140-7007(97)00091-1

[13] N. Pettit, M. Willatzen, and L. Ploug-Sørensen, "A general dynamic simulation model for evaporators and condensers in refrigeration. part ii: simulation and control of an evaporator," *International Journal of Refrigeration*, vol. 21, no. 5, pp. 404 – 414, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000923. doi:10.1016/S0140-7007(97)00092-3

[14] W.-J. Zhang and C.-L. Zhang, "A generalized moving-boundary model for transient simulation of dry-expansion evaporators under larger disturbances," *International Journal of Refrigeration*, vol. 29, no. 7, pp. 1119 – 1127, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700706000570. doi:10.1016/j.ijrefrig.2006.03.002

[15] R. Shah, *Dynamic Modeling and Control of Single and Multi Evaporator Subcritical Vapor Compression Systems*. University of Illinois at Urbana-Champaign, 2003. [Online]. Available: http://books.google.dk/books?id=Tni2JgAACAAJ

[16] L.-X. Zhao, C.-L. Zhang, and B. Gu, "Neural-network-based polynomial correlation of single- and variable-speed compressor performance," *HVAC&R Research*, vol. 15, no. 2, pp. 255–268, 2009. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10789669.2009.10390836. doi:10.1080/10789669.2009.10390836

[17] H. Qiao, R. Rademacher, and V. Aute, "A review for numerical simulation of vapor compression systems," in *Proceedings of 2010 International Refrigeration Conference*, Purdue University, Indiana, USA, 2010. [Online]. Available: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2089&context=iracc

[18] H. Air-Conditioning and R. Institute, "Standard for positive displacement refrigerant compressors and compressor units." [Online]. Available: http://www.ari.org/App_Content/ahri/files/standards%20pdfs/ANSI%20standards%20pdfs/ANSI-ARI-540-2004%20latest.pdf

[19] C. Co, *Flow of fluids through valves, fittings, and pipe*, ser. Technical paper. Crane Co., 1988. [Online]. Available: http://books.google.dk/books?id=-fdRAAAAMAAJ

[20] D. Green and R. Perry, *Perry's Chemical Engineers' Handbook, Eighth Edition*, ser. McGraw Hill professional. McGraw-Hill Education, 2007. [Online]. Available: http://books.google.dk/books?id=tH7IVcA-MX0C

[21] R. Miller, *Flow measurement engineering handbook*, ser. Chemical engineering books. McGraw-Hill, 1996. [Online]. Available: https://books.google.dk/books?id=0e9RAAAAMAAJ

[22] H. Li and J. E. Braun, "A method for modeling adjustable throat-area expansion valves using manufacturers rating data," *HVAC&R Research*, vol. 14, no. 4, pp. 581–595, 2008. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10789669.2008.10391027. doi:10.1080/10789669.2008.10391027

[23] T. Zaleski and K. Klepacka, "Approximate method of solving equations for plate heat exchangers," *International Journal of Heat and Mass Transfer*, vol. 35, no. 5, pp. 1125 –

1130, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/001793109290173P. doi:10.1016/0017-9310(92)90173-P

[24] S. K. Al-Dawery, A. M. Alrahawi, and K. M. Al-Zobai, “Dynamic modeling and control of plate heat exchanger,” *International Journal of Heat and Mass Transfer*, vol. 55, no. 23-24, pp. 6873 – 6880, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001793101200525X. doi:10.1016/j.ijheatmasstransfer.2012.06.094

[25] J. Kay and R. Nedderman, *Fluid Mechanics and Transfer Processes*. Cambridge University Press, 1985. [Online]. Available: https://books.google.dk/books?id=43U7AAAAIAAJ

[26] K. Prölss, G. Schmitz, D. Limperich, and M. Braun, “Influence of refrigerant charge variation on the performance of an automotive refrigeration system,” in *International Refrigeration and Air Conditioning Conference, Paper 843*, 2006. [Online]. Available: http://docs.lib.purdue.edu/iracc/843

[27] Guo-liang Ding, “Recent developments in simulation techniques for vapour-compression refrigeration systems,” *International Journal of Refrigeration*, vol. 30, no. 7, pp. 1119–1133, 2007. doi:10.1016/j.ijrefrig.2007.02.001

[28] “Trnsys.” [Online]. Available: http://sel.me.wisc.edu/trnsys/

[29] M. Skovrup, “Windali - an open-structured component modeling and simulation program based on standard programming languages,” in *Proceedings of SIMS Conference. Lyngby*, 2000, pp. 157–172.

[30] “Dymola.” [Online]. Available: http://www.dynasim.se/index.htm

[31] P. Fritzson and P. Bunus, “Modelica – a general object-oriented language for continuous and discrete-event system modeling,” in *IN PROCEEDINGS OF THE 35TH ANNUAL SIMULATION SYMPOSIUM*, 2002, pp. 14–18.

[32] L. Liu, F. Felgner, and G. Frey, “Comparison of 4 numerical solvers for stiff and hybrid systems simulation,” in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, Sept 2010, pp. 1–8. doi:10.1109/ETFA.2010.5641330

[33] L. F. Shampine and M. W. Reichelt, “The matlab ode suite,” *Journal on Scientific Computing*, vol. 18, pp. 1–22, 1997. doi:10.1137/S1064827594276424

[34] C. Rosen, D. Vrecko, K. Gernaey, and U. Jeppsson, “Implementing adm1 for benchmark simulations in matlab/simulink,” *Journal of Water science and technology*, vol. 54, no. 4, pp. 11–19, 2006. doi:10.2166/wst.2006.521

[35] R. M. Howe, “Accuracy and stability tradeoffs in multirate simulation,” in *Proceedings of SPIE*, vol. 4367, 2001, pp. 113–126. doi:10.1117/12.440011

[36] J. G. Pearce, R. E. Crosbie, J. J. Zenor, R. Bednar, D. Word, and N. G. Hingorani, “Developments and applications of multi-rate simulation.” in *UKSim*, D. Al-Dabass, Ed. IEEE, 2009, pp. 129–133. [Online]. Available: http://dblp.uni-trier.de/db/conf/uksim/uksim2009.html#PearceCZBWH09. doi:10.1109/UKSIM.2009.23

[37] R. Kübler and W. Schiehlen, “Modular simulation in multibody systems,” *Multibody System Dynamics*, vol. 4, pp. 107–127, 2000. doi:10.1023/A:1009810318420

[38] J. Läckel, F. Junker, and S. Toepper, "Block-oriented modelling of rigid multibody systems with regard to subsystem techniques," in *Advanced Multibody System Dynamics*, ser. Solid Mechanics and Its Applications, W. Schiehlen, Ed. Springer Netherlands, 1993, vol. 20, pp. 49–66. [Online]. Available: http://dx.doi.org/10.1007/978-94-017-0625-4_3. doi:10.1007/978-94-017-0625-4_3

[39] R. Kübler and W. Schiehlen, "Two methods of simulator coupling," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 6, no. 2, pp. 93–113, 2000. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1076/1387-3954%28200006%296%3A2%3B1-M%3BFT093. doi:10.1076/1387-3954(200006)6:2;1-M;FT093

[40] J. Butcher, "General linear methods for ordinary differential equations," *Mathematics and Computers in Simulation*, vol. 79, no. 6, pp. 1834 – 1845, 2009, applied and Computational MathematicsSelected Papers of the Sixth PanAmerican WorkshopJuly 23-28, 2006, Huatulco-Oaxaca, Mexico. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378475407001462. doi:http://dx.doi.org/10.1016/j.matcom.2007.02.006

[41] J. C. Butcher, "General linear methods," *Acta Numerica*, vol. 15, pp. 157–256, 5 2006. [Online]. Available: http://journals.cambridge.org/article_S0962492906220014. doi:10.1017/S0962492906220014

[42] J. Butcher, "General linear methods," *Computers & Mathematics with Applications*, vol. 31, no. 4-5, pp. 105 – 112, 1996, selected Topics in Numerical Methods. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0898122195002227. doi:http://dx.doi.org/10.1016/0898-1221(95)00222-7

[43] U. Ascher and L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1998. [Online]. Available: http://books.google.dk/books?id=AqQ6QMthXkMC

[44] J. C. Butcher, *Numerical methods for ordinary differential equations*. J. Wiley, 2003. [Online]. Available: http://www.worldcat.org/isbn/9780471967583

[45] K. Atkinson, *An introduction to numerical analysis*. Wiley, 1978. [Online]. Available: http://books.google.dk/books?id=ByPpQ1nt3esC

[46] K. J. AAström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed. Instrument Society of America, Research Triangle Park, NC, 1995.

[47] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[48] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, 2002, vol. 3.

[49] W. Leithead, "Survey of gain-scheduling analysis design," *International Journal of Control*, vol. 73, pp. 1001–1025, 1999.

[50] J. Shamma and M. Athans, "Gain scheduling: potential hazards and possible remedies," *Control Systems, IEEE*, vol. 12, no. 3, pp. 101–107, June 1992. doi:10.1109/37.165527

[51] F. Wu, "Control of linear parameter varying systems," Ph.D. dissertation, University of California, 1995.

[52] G. J. Balas, “Linear, parameter-varying control and its application to a turbofan engine,” *International Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 763–796, 2002. [Online]. Available: http://dx.doi.org/10.1002/rnc.704. doi:10.1002/rnc.704

[53] I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design.* New York, NY, USA: John Wiley & Sons, Inc., 1996.

[54] G. J. C. Verdijck, “Model-based product quality control applied to climate controlled processing of agro-material,” 2003. [Online]. Available: http://alexandria.tue.nl/extra2/200310443.pdf

[55] G. J. C. Verdijck, M. Weiss, and H. A. Preisig, “Model-based product quality control for a potato storage facility,” in *American Control Conference, 1999. Proceedings of the 1999*, vol. 4, 1999, pp. 2558–2562 vol.4. doi:10.1109/ACC.1999.786522

[56] G. J. C. Verdijck and G. van Straten, “A modelling and control structure for product quality control in climate-controlled processing of agro-material,” *Control Engineering Practice*, vol. 10, no. 5, pp. 533 – 548, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066102000035. doi:10.1016/S0967-0661(02)00003-5

[57] G. J. C. Verdijck, G. van Straten, and H. Preisig, “Optimisation of product quality and minimisation of its variation in climate controlled operations,” *Computers and Electronics in Agriculture*, vol. 48, no. 2, pp. 103–122, 2005. doi:10.1016/j.compag.2005.02.009

[58] R. G. M. van der Sman and G. J. C. Verdijck, “Model predictions and control of conditions in a ca-reefer container,” in *ISHS Acta Horticulturae 600: VIII International Controlled Atmosphere Research Conference*, Agrotechnological Research Institute ATO, P.O. box 17, 6700 AA Wageningen, the Netherlands, 2003, pp. 163–171. [Online]. Available: http://www.actahort.org/members/showpdf?booknrarnr=600_20

[59] J. Maciejowski, *Predictive Control: With Constraints*, ser. Pearson Education. Prentice Hall, 2002. [Online]. Available: http://books.google.dk/books?id=HV_Y58c7KiwC

[60] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, “The explicit solution of model predictive control via multiparametric quadratic programming,” in *American Control Conference, 2000. Proceedings of the 2000*, vol. 2, 2000, pp. 872–876 vol.2. doi:10.1109/ACC.2000.876624

[61] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109801001741. doi:http://dx.doi.org/10.1016/S0005-1098(01)00174-1

[62] L. S. Larsen, C. Thybo, and H. Rasmussen, “Intelligent control - optimizing the operation of refrigeration systems under daily variations in ambient temperature,” in *Danske Køledage, March 15th-16th, 2007. Proceedings of Danske Køledage 2007.*, 2007.

[63] J. Cai, J. Stoustrup, and J. Jorgensen, “Preventing refrigerated foodstuffs in supermarkets from being discarded on hot days by mpc,” in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, 2008. [Online]. Available: http://www.control.aau.dk/~jakob/selPubl/papers2008/ifacwc_2008_3.pdf. doi:10.3182/20080706-5-KR-1001.01880

[64] K. Vinther, H. Rasmussen, R. Izadi-Zamanabadi, J. Stoustrup, and A. Alleyne, "Learning-based precool algorithms for exploiting foodstuff as thermal energy reserve," *Control Systems Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014. doi:10.1109/TCST.2014.2328954

[65] A. Jakobsen, B. Rasmussen, and M. Skovrup, "Development of energy optimal capacity control in refrigeration systems," in *Proceedings of 2000 International Refrigeration Conference*. Purdue University, Indiana, USA: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1498&context=iracc, 2000, pp. 329–336.

[66] R. Zhou, T. Zhang, J. Catano, J. T. Wen, G. J. Michna, Y. Peles, and M. K. Jensen, "The steady-state modeling and optimization of a refrigeration system for high heat flux removal," *Applied Thermal Engineering*, vol. 30, no. 16, pp. 2347 – 2356, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S135943111000219X. doi:10.1016/j.applthermaleng.2010.05.023

[67] R. Koury, L. Machado, and K. Ismail, "Numerical simulation of a variable speed refrigeration system," *International Journal of Refrigeration*, vol. 24, no. 2, pp. 192 – 200, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700700000141. doi:10.1016/S0140-7007(00)00014-1

[68] B. P. Rasmussen and A. G. Alleyne, "Control-oriented modeling of transcritical vapor compression systems," *Journal of dynamic systems, measurement, and control*, vol. 126, no. 1, pp. 54–64, 2004.

[69] B. Rasmussen and A. Alleyne, "Dynamic modeling and advanced control of air conditioning and refrigeration systems," *Air Conditioning and Refrigeration Center. College of Engineering. University of Illinois at Urbana-Champaign.*, 2006. [Online]. Available: http://hdl.handle.net/2142/12355

[70] "MATLAB®- ordinary differential equation solvers." [Online]. Available: http://www.mathworks.se/help/matlab/ordinary-differential-equations.html

[71] C. B. Brosilow, Y.-C. Liu, J. Cook, and J. Klatt, "Modular Integration Methods for Simulation of Large-Scale Dynamic Systems," *Modeling, Identification and Control*, vol. 6, no. 3, pp. 153–179, 1985. doi:10.4173/mic.1985.3.4

[72] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Wiley, 2005. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-047001167X.html

[73] F. Wu, "Control of linear parameter varying systems," Ph.D. dissertation, University of California, 1995. [Online]. Available: http://www.mae.ncsu.edu/wu/paper/PhDthesis.ps

[74] D. Petersson and J. Löfberg, "LPV H_2-controller synthesis using nonlinear programming," in *Proceedings of the 18th IFAC World Congress :*, 2011, pp. 6692–6696. doi:10.3182/20110828-6-IT-1002.02028

[75] "MATLAB® SIMULINK® linearization algorithm." [Online]. Available: http://www.mathworks.se/help/slcontrol/ug/exact-linearization-algorithm.html

[76] E. Kreyszig, *Advanced Engineering Mathematics*, 9th ed. Wiley, Nov. 2005. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471488852

[77] U. Halldorsson, M. Fikar, and H. Unbehauen, “Nonlinear predictive control with multirate optimisation step lengths,” *IEE Proceedings - Control Theory and Applications*, vol. 152, no. 3, pp. 273–285, 2005. doi:10.1049/ip-cta:20041310

[78] “Specific heats of common food and foodstuff,” 2015. [Online]. Available: http://www.engineeringtoolbox.com/specific-heat-capacity-food-d_295.html

[79] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jorgensen, “Economic model predictive control for building climate control in a smart grid,” in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, IEEE. http://orbit.dtu.dk/fedora/objects/orbit:74519/datastreams/file_6543920/content, 2012, pp. 1–6.

[80] Container-Handbook, “8.1.2 actual power consumption,” 2009. [Online]. Available: http://www.containerhandbuch.de/chb_e/wild/index.html?/chb_e/wild/wild_08_01_02.html

[81] M. Knudsen, *Senstools: A Matlab Toolkit for Parameter Estimation in Linear and Nonlinear Systems Using a Sensitivity Approach*. Department of Control Engineering, Aalborg University, 1996. [Online]. Available: http://books.google.dk/books?id=ygHPQwAACAAJ

Part II

Papers

Paper A

Modular Modeling and Simulation Approach - Applied to Refrigeration Systems

Kresten K. Sørensen[a,b] and Jakob Stoustrup[a]

[a]*Section for Automation and Control, Department of Electronic Systems*
Aalborg University, Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
[b]*Lodam electronics a/s, Kærvej 77, 6400 Sønderborg, Denmark*
E-mail: kresten@es.aau.dk jakob@es.aau.dk

The paper has been published in the proceedings of the
2008 IEEE Multi-conference on Systems and Control , pp. 983-988.

Abstract

This paper presents an approach to modeling and simulation of the thermal dynamics of a refrigeration system, specifically a reefer container. A modular approach is used and the objective is to increase the speed and flexibility of the developed simulation environment. The refrigeration system is divided into components where the inputs and outputs are described by a set of XML files that can be combined into a composite system model that may be loaded into MATLAB®. *A set of tools that allows the user to easily load the model and run a simulation are provided. The results show a simulation speed-up of more than a factor of three by partitioning the model into smaller parts, and thereby isolating fast and slow dynamics. As a cost there is a reduction in accuracy which in the example considered is less than one percent.*

1 Introduction

Numerical simulation is extensively used for experiments within the field of control engineering, and with the increasing power of computers it has become possible to simulate very large dynamical systems on a normal desktop PC at reasonable speed. But larger system models results in larger and more complex equation sets that are difficult to handle and therefore a range of simulation tools capable of handling large system models is available. When choosing a simulation tool it is worth considering the ease of modeling, simulation speed and accuracy because these parameters vary from tool to tool.

A common tool utilized within control engineering is SIMULINK® [1], which allows the user to create and simulate large models from built-in or user developed component libraries through a Graphical User Interface (GUI). The simulation model composed of component models may be solved by a range of block oriented input/output solvers that automatically adjusts the size of the numerical integration step. SIMULINK® variable-step solvers change the step size during simulation [2], reducing the step size to increase accuracy when the states of a simulation model are changing rapidly and increasing the step size to avoid taking unnecessary steps when the models states are changing slowly. Computing the step size adds to the computational overhead at each step but can reduce the total number of steps, and hence simulation time, required to maintain a specified level of accuracy for models with rapidly changing or piecewise continuous states. The selected integrations step is however inherited by blocks further down the signal path and if the components further down the path have slow dynamics compared to the preceding blocks they will be simulated with unnecessarily small integration step sizes, leading to a computational overhead. Such a system is referred to as stiff and while MATLAB® and SIMULINK® have specific solvers for these types of problems they do not address the problem with computational overhead in stiff systems.

Another approach is used by DYMOLA [3] that implements the MODELICA [4] language which is an equation-based object-oriented modelling language. In MODELICA symbolic equations that define the dynamical behavior of a component may be entered in a non-causal way, leaving the task of ordering and reducing the final set of equations to the simulation engine before a simulation can be run. Stiff problems may be solved using implicit methods that allow larger step-sizes at the cost of solving a set of nonlinear equations at each time step. DYMOLA uses mixed-mode integration [5] that takes a middle course where the system is split up into fast and slow states. Only the fast states are discretized implicitly leaving a smaller set of nonlinear equations to solve at each time step and thus a faster simulation. To speed up the simulation even further inline integration [6] is supported. The discretization formulas are inserted (in-lined) into the problem and DYMOLA's symbolic engine is applied to the resulting equations [5]. The automatic model reduction and partitioning approach used by DYMOLA does not, however, take advantage of a-priori system knowledge that already exists.

In this paper an attempt was made to build a small and fast simulation environment that provides easy modular modelling and rapid prototyping of refrigeration systems from a library of component models. A modelling and simulation environment for MATLAB® has been developed to enable simulation experiments on nonlinear models consisting of a mix of models based on Ordinary Differential Equations (ODE), Differential Algebraic Equations (DAE) and purely algebraic equations. The objective of the tool is to provide rapid and flexible refrigeration system model development from a predefined set of refrigeration component models. A refrigeration container model has been chosen as an example for this study.

Refrigeration containers are used to move many different types of cargo between all areas in the world and this puts some unusual requirements on the refrigeration system with respect to the temperature range on both the cold and the hot side. The goods transported may require a stable temperature between -30C° and +20C° and the temperature of the air around the container can be between -30C° and +50C°. Because of the nonlinear nature of the refrigeration system and the large temperature range it is infeasible to use a linear model for simulation experiments and therefore a nonlinear model must be used. The requirements for such a model is that it should match the real system close enough to be used for closed loop control experiments and simulations should run at least one order of magnitude faster than experiments on the real system. Another important requirement is that it must be possible to change the model configuration without having to rewrite or reorganize the entire set of equations for the model by hand. Therefore a modular approach is selected where the model of the refrigeration system is composed of a set of interchangeable component models that are based on first principles and assumptions where appropriate.

A model of a refrigeration container is developed and used as a test case for the simulation environment. The system has both fast and slow states but the fast states are isolated in a single component leading to a potential speed-up of the simulation with a modular approach. Therefore it was attempted to decouple component models with slower dynamics from models with fast dynamics by simulating each of the component models separately and only exchange input/output values between the component models at fixed discrete times. The number of steps and average step time for the solvers for each of the components is used to calculate the difference in simulation time of the example system, as either a modular model or a monolithic model.

2 Modelling

The model of the refrigeration system is divided into components that each represent a physical component of the system, i.e. a condenser or an evaporator. Each **component model** is described by two files; an `m` file that holds the input/output equations of the model and an `XML` file that describes the properties of the inputs and outputs of the `m` file. The **simulation model** is the overall model for the refrigeration plant and its properties are described by an `XML` file that holds a list of included component models and the connections between them. Therefore the structure of the simulation model is defined by the simulation model definition file, and from this the model loader can create a simulation object that is used by the simulator.

2.1 Component Model Syntax

Modelling of component models is basically the same as for normal ODE model functions that may be solved by MATLAB®'s built-in `ode` solvers, but additional info is needed by the model loader in order to do type-checking when connecting the inputs and outputs of the component models. Each component model is described by an `m` file containing the input/output equations, an `XML` file describing the input/output properties of the model, its execution mode, and the name of the corresponding `m` file. The syntax for writing the component model `XML` file is shown in the box below:

Listing A.1: Component Model Syntax

```
 <?xml version="1.0" encoding="UTF-8"?>
<component name=["Component Name]">
<inputs>
  <input name=["Input1 Name"] type=["Input1 Type"]
    description=["Input1 Description"]/>
  .
  .
  <input name=["InputN Name"] type=["InputN Type"]
    description=["InputN Description"]/>
</inputs>

<states>
  <state name=["State1 Name"] type=["State1 Type"]
    default=["State1 Start Value"]
    description=["State1 Description"]/>
  .
  .
  <state name=["StateN Name"] type=["StateN Type"]
    default=["StateN Start Value"]
    description=["StateN Description"]/>
</states>

<connectors>
  <connector type=["type"] conid=["Conn. Name"]>
    <{input, state} name=["Name"]
      type="Physical Entity"/>
    <{input, state} name=["Name"]
      type="Physical Entity"/>
   </connector>
  .
  .
</connectors>

<control_inputs>
  <input name=["Input Name"] type=["Input Type"]
    description=["Input Description"]/>
  .
  .
</control_inputs>

<simulation method={"ode15s", "call"}
  call=[".m File Function Name"]/>
<filename>[.m File Name]</filename>
</component>
```

The <inputs> section contains a list of the inputs to the component model, and it is important that the inputs are listed in the same order as they occur in the input vector of the model function. Each input has a name, a type used for type-checking when inputs are connected, and a description. There must be the same number of inputs in the input list as the length of the input vector of the corresponding model function. The `<states>` section is similar to the input section except that it describes the states or outputs of the function and that a default value must be declared. The default value is used as initial value in simulations when the simulation tool is not given an initial state vector to start from. Signals may be grouped together in connectors that allows the user to connect a set of signals from one component model to a similar set on another component model in one operation.

Because this environment is used for refrigeration systems it has a built-in connector class for refrigeration pipe interfaces but obviously, for other applications, other connections will be relevant - see Simulation Model Syntax in Subsection II.B. On the refrigeration pipe interface three variables exist; a mass flow $\dot{m}$, a pressure p, and an enthalpy h. The model loader will return an error if each refrigeration pipe interface does not contain exactly one of each of the mentioned aforementioned types. The individual variables may be either an input or a state but the model loader will check that each of the inputs can be connected to a state of the correct type when two refrigeration pipe interfaces are connected. This saves the user the work of having to connect the variables manually, but when building the component models attention must be given to where the different states that are shared between models are located.

2.2 Simulation Model Syntax

The syntax for the simulation model `XML` file are listed below:

Listing A.2: Simulation Model Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<simulation_model name=["Simulation Model Name"]>
<component_path>[Component Library Path]
</component_path>

<components>
  <component name=["Component Model Name"]
    file=["Component Model XML File Name"]/>
  .
  .
  <component name=["Component Model Name"]
    file=["Component Model XML File Name"]/>
</components>

<connections>
  <connector name=["Connector Name"]>
    <component name=["Component Name"]
      conid=["Connection Name"]/>
    <component name=["Component Name"]
      conid=["Connection Name"]/>
  </connector>

  <connection name=["Connection Name"]>
     <component name=["Component Name"]
      type="Input" input=["Input Name"]/>
     <component name=["Component Name"]
      type="Output" output=["Output Name"]/>
  </connection>

</connections>
</simulation_model>
```

The simulation model is composed of a set of component models and their connections described by the simulation model `XML` file, containing a list of the included component models and a description of how the components are connected. The `<components>` section lists the component models used in the simulation model and it is allowed to use a component model more than once if they are given unique names. In the `<connections>` section all the connections in the simulation model are listed. A connector connection is established as in the `<connector>` sections by giving the name of the two component models and the connector on each of the components. Inputs and outputs that are not associated with connector interfaces, such as control inputs to actuators, are connected in a `<connection>` section by listing the two components one by one. In each of the `<component>` sections it must be stated whether the signal is an input or an output, and what the name of the signal is. The model loader provides type checking on connections between components and gives a precise description, with the names of the implicated components and signals, in case of an eventual error in the set of connections between components.

The model may be loaded into MATLAB® with a model loader function that loads each of the components and creates a struct containing all the information necessary for simulation. During simulation the states of the component models are kept in a single vector, denoted $\mathbf{X}$, and therefore two matrices that maps between $\mathbf{X}$ and component model I/O are generated for each component model. The matrix $\mathbf{Z}_n$ maps from $\mathbf{X}$ to the component model x-vector $\mathbf{x}_n$ such that

$$\mathbf{X} = \mathbf{Z}_n \cdot \mathbf{x}_n \tag{A.1}$$

$$\mathbf{x}_n = \mathbf{Z}_n^T \cdot \mathbf{X} \tag{A.2}$$

where the index n denotes the component model number. Component model inputs $\mathbf{u}_n$ are mapped from $\mathbf{X}$ by the matrix $\mathbf{CM}_n$ such that

$$\mathbf{u}_n = \mathbf{CM}_n \cdot \mathbf{X} \tag{A.3}$$

Because $\mathbf{CM}_n$ maps from state variables that reflect a physical value to inputs that take a physical value, it is a one-zero matrix and must have exactly one "1" in each row. When the model loader has created the mapping matrices they are used to check the connection integrity of the system model such that all inputs are connected to exactly one state.

3 Simulation

Simulation of the system model is done in discrete time steps defined in the time vector given in the simulation function call. In each of the time steps the component models are simulated separately, according to the method defined in the component model `XML` file and the results are then combined into the X vector containing the states for all of the component models. Component models that are purely algebraic are evaluated in one operation like a normal MATLAB® function, and dynamical models are solved by one of MATLAB®'s built-in `ode` functions. Fig. (A.1) shows a possible structure for a simulation model consisting of four component models.

The discretization of the signals between the component models is equivalent to inserting a zero order hold between all models as shown on Fig. (A.1).

The input to the individual component models are calculated by applying (A.2) and (A.3) and given as arguments to the appropriate simulation function such that

$$\mathbf{x}_n(k) = f_n\left([\mathbf{t}(k-1) \quad \mathbf{t}(k)], \quad \mathbf{Z}_n \cdot \mathbf{x}_n, \quad \mathbf{CM}_n \cdot \mathbf{X}\right) \tag{A.4}$$

where f_n is the simulation function for the referenced component model and $[\mathbf{t}(k-1) \quad \mathbf{t}(k)]$ is the time interval in which to simulate. The results from the each of the

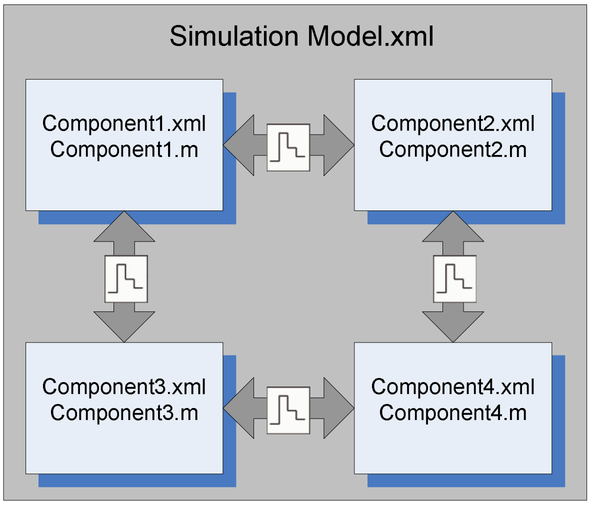


Fig. A.1: Example of a Simulation Model Structure

component simulations are then combined into the system state-vector X by

$$\mathbf{X}(k) \quad = \quad \sum_{n=1}^{N} \mathbf{Z}_n \cdot \mathbf{x}_n(k) \tag{A.5}$$

where N is the total number of components in the model. The simulation environment may then proceed and simulate the next time step with the same procedure as above.

4 Reefer Model

The refrigeration system used as an example here utilize an economizer to increase the efficiency of the system at high pressure differences between the cold and hot side. The compressor efficiency is lower at high pressure differences which is exist when there is a large difference between the evaporation and condensation temperatures.The economizer arrangement increases the refrigeration capacity and improves the coefficient of performance (COP) [7]. A schematic of the system can be seen in Fig. A.2. The elements of the model are described in the sequel. The explicit 44 equations of the model are not derived here due to space limitations, but they are available in [8].

The system consists of a two-stage piston compressor, a condenser, an evaporator, and an economizer [7] which is a counterflow plate heat exchanger. The compressor is equipped with a frequency converter which enables it to run at variable speed. The expansion valves are electromagnetically pulsed on/off valves and the fans may run at

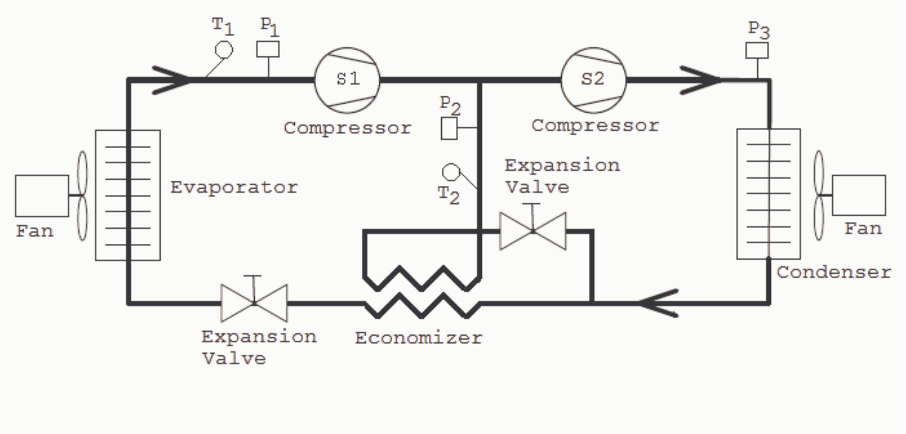


Fig. A.2: Refrigeration System for a Reefer Container

half speed, full speed, or be turned off entirely. There are three major pressure levels in this setup; p_1 is the evaporator pressure, p_2 is the intermediate pressure between the two compressor stages, and p_3 is the condenser pressure. Since p_2 is also the evaporation pressure on the cold side of the economizer it is coupled closely to the evaporation temperature of the economizer, and hence influential on the inlet temperature of the refrigerant to the evaporator.

According to [9] the dominant dynamics of a refrigeration system are the thermal time constants of the metal surfaces in the heat exchangers and refrigerant mass time constants, with respect to control applications. Also according to [9] some of the components have dynamics that are so fast compared to the dominant dynamics that they may be replaced with algebraic equations, thus reducing the model order while preserving the physical behavior of the model on the dominant dynamics. The components that may be modeled algebraically are the expansion valves and the compressors while the rest of the components are modeled using first principles or assumptions. Two pipe junction models are needed in order to model the joining and splitting of refrigerant flows that occur between the compressor stages and after the receiver, respectively.

4.1 Pipe Joining Junction

The pipe junction model has three states; Pressure p, internal mass M, and output enthalpy h_{out}. It also has five inputs; the mass flows on all three interfaces $\dot{m}_{in1}$, $\dot{m}_{in2}$, $\dot{m}_{out}$, and the input enthalpy for both refrigerant inputs h_{in1} and h_{in2}. The pressure of this component model has fast dynamics because it is a small volume containing vapor with a high mass flow and no boiling liquid to dampen pressure oscillations and therefore the exact dynamics are neglected and the absolute pressure simply calculated instead.

4.2 Compressor

The compressor has two almost identical stages where the only difference is that the displacement volume of the first stage is twice as large as that of the second stage. The compressor stages are modelled by two algebraic functions giving the mass flow and output enthalpy as a function of input pressure, output pressure, input enthalpy, the speed of the compressor, and the temperature of the compressor. The compression is assumed adiabatic and the physical behavior of the model includes harmful volume, and valve pressure loss. The mass of the refrigerant in the compressor is neglected and therefore the mass flows on the input and output are equal.

4.3 Expansion Valve

The expansion valve model is purely algebraic and modelled as a continuous valve giving the average mass flow of the electromagnetically pulsed on/off valves used on the reefer. A lookup table is used to find the mass flow at full opening as a function of the pressures on both sides and this is multiplied with the ON time, which is the fraction of time it is turned on. The expansion is assumed adiabatic end therefore there is no change to the enthalpy of the refrigerant.

4.4 Economizer

The hot side of the economizer is filled with liquid refrigerant running from the receiver to the evaporator expansion valve and is therefore modelled as a single region with uniform heat transfer from liquid to metal. The liquid volume is assumed to have uniform pressure and enthalpy and a constant pressure drop from input to output. The cold side, where refrigerant evaporates, is modelled as a single volume where the amount of energy transferred from the metal to the refrigerant is dependent on the difference in temperature between the refrigerant and the metal walls. The thermal capacitance of the metal walls acts as a damper on the dynamics and it is therefore included in the model.

4.5 Evaporator

The evaporator is modelled as in [10] which is a lumped model with a moving boundary between the two phase and the vapor volume.

4.6 Condenser

The condenser is modelled as in [11] and [12] which is a lumped model with a moving boundary between the two-phase volume and the vapor volume.

4.7 Receiver

The receiver is a buffer tank for excess refrigerant. The refrigerant is led from the condenser into the top of the receiver and liquid refrigerant to the expansion valves are taken from the bottom. The receiver is usually either neglected in dynamical models of refrigeration plants or not existing in the modelled plant. It is, however, not entirely without influence on the refrigeration system's dynamics, especially in startup situations and during fast pressure changes. The reason for this is that the liquid in the receiver acts as a buffer and has a dampening effect on pressure transients from the condenser, but this can also lead to problems with vapor bubbles in the feed line to the expansion valves which severely degrades the mass flow. The liquid in the receiver that goes to the expansion valves may be either sub-cooled or at the boiling point. If the liquid starts to boil it will turn into a two phase mixture of liquid and vapor with a quality that depends on how much it is boiling. When the condenser fan is switched on, the pressure in the receiver can drop rapidly, and if the temperature of the liquid in the receiver is close to the boiling point it will begin to boil until the temperature drops below the boiling point.

4.8 Box

The box is the largest thermodynamic capacitance in the reefer due to its mass, but in this example an empty container has been used and therefore the thermodynamic capacitance consists mainly of the aluminium T-floor and the air inside the container. Energy is exchanged by air circulating from the evaporator, over the floor, and back to the evaporator again along the sides and roof of the container. The air is heated by energy leaking through the insulated walls, floor and roof of the container.

5 Results

5.1 Simulation Speed

An experiment has been carried out in order to measure the increase in speed gained by simulating the system as separate component model functions instead of a large single-function model. The reefer model used in the experiment has 17 discrete and 35 continuous states and are divided into 10 component models, where four are purely algebraic and the remaining six are continuous. A simulation of a 4000 s period has been carried out using a laptop equipped with a 2.0 GHz Core 2 Duo processor and 2 GB of RAM.

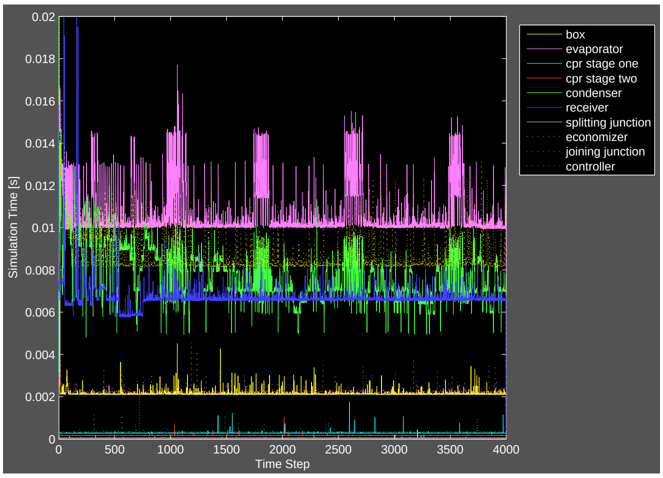


Fig. A.3: Simulation Time for Each of the Components

The discrete time step size of the simulation environment was set to one second and the simulation completed in 156.8 s, i.e., 25.5 times faster than real time experiments.

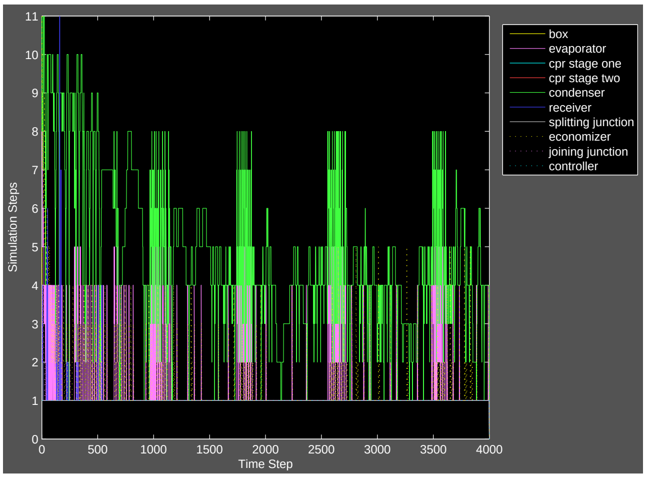


Fig. A.4: Number of Simulation Steps for Each of the Components

Fig. A.3 shows the time used to simulate each of the components at the discrete time steps during the 4000 s simulation, and Fig. A.4 shows the number of simulation steps used by each of the components for each discrete time step. The time, T_{uni}, that it would have taken a solver to simulate the model if it had been unified into a single, monolithic, function is calculated using the total number of simulation steps and the average time for a single simulation step for each model. The total number of simulation steps S_n is the number of times that a model component function has been called during the entire simulation, either by the ODE solver if it it is continuous or directly by the simulation environment if it is algebraic. The average time for a single simulation step t_n is found by

$$t_n = \frac{T_n}{S_n} \tag{A.6}$$

where T_n is the total time that the CPU has spent solving a particular model during the entire simulation.

The time saved by simulating in smaller components has been calculated by averaging over the entire simulation period as in

$$T_{uni} = \sum_{n=1}^{N} S_{max} \cdot t_n x \tag{A.7}$$

where N is the total number of component models, S_{max} is the number of simulation steps used by the component model that used the most simulation steps during the entire simulation. Equation (A.7) yields a total simulation time of 528.7 s for a monolithic model which corresponds to a speed-up of

$$\frac{528.7s}{156.8s} = 337.2\% \tag{A.8}$$

for the modular model compared to a monolithic model. The average error of the modular simulation is 0.734%, relative to a simulation of the same model unified into a monolithic function, which is acceptable when considering the improvement in simulation speed.

6 Discussion and Future Work

6.1 Discussion

There are both benefits and drawbacks to the modular approach; it is up to the user to determine a suitable size of the discrete time steps with respect to the fastest dynamics in the set of component model states that are used as inputs to other component models. It is however possible for a component model to have fast internal dynamics, that is, a state that is not used as input to other component models and thereby discretized by the simulation environment.

The separation of the component models, however, makes it possible to simulate systems with both fast and slow dynamics faster than it is possible with a unified system model, where all the equations of the composite model are rewritten as one function that can be used for simulation. The reason for this is that a solver for a unified model would need to evaluate all the equations for the entire model for each of the time steps, which would be sized with respect to the fastest dynamics of the unified model. By simulating the system as separate component models it is possible to have one or more components with fast internal dynamics that are simulated using smaller time steps than is necessary for components with slow dynamics. This results in an overall reduction in calculations needed to simulate the system for a given period of time and thereby a faster simulation.

The decentralized nature of the simulation has a potential cost on the achieved accuracy of the simulation result due to the sequential computation. On the other hand solving smaller algebraic equations might have a positive influence on the accuracy of the result. In fact, in some cases, a numerically infeasible simulation might be rendered feasible by decentralization for some systems.

6.2 Future Work

A mathematical formulation that describes the implications that the discretization has on the accuracy of the simulation results and provides a guideline, or even automatic identification of the largest possible discrete integration step that can be used for a given model without exceeding the target accuracy of the simulation.

7 Aknowledgements

The authors gratefully acknowledge the assistance from Kim Madsen, Maersk Container Industri and Lars Mou Jessen, Lodam electronics a/s, for counseling on modeling of the reefer container components.

References

[1] "Matlab® Simulink®." [Online]. Available: http://www.mathworks.com/products/simulink/

[2] "Matlab®- ordinary differential equation solvers." [Online]. Available: http://www.mathworks.se/help/matlab/ordinary-differential-equations.html

[3] "Dymola." [Online]. Available: http://www.dynasim.se/index.htm

[4] P. Fritzson and P. Bunus, “Modelica – a general object-oriented language for continuous and discrete-event system modeling,” in *IN PROCEEDINGS OF THE 35TH ANNUAL SIMULATION SYMPOSIUM*, 2002, pp. 14–18.

[5] M. T. Soejima S, “Application of mixed mode integration and new implicit inline integration at toyota,” in *In proceedings of 2nd International Modelica Conference, Oberpfaffenhofen, Germany*, 2002, pp. 65–1 – 65–5. [Online]. Available: http://www.modelica.org/events/Conference2002/papers/p09_Soejima.pdf

[6] H. Elmqvist, M. Otter, and F. E. Cellier, “Inline integration: A new mixed symbolic/numeric approach for solving differential-algebraic equation systems,” in *In Proceedings of ESM 1995, European Simulation Multiconference*, 1995.

[7] S. Jonsson, “Performance simulations of twin-screw compressors with economizer,” *International Journal of Refrigeration*, vol. 14, no. 6, pp. 345 – 350, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/014070079190031B. doi:10.1016/0140-7007(91)90031-B

[8] K. Sørensen, M. Skovrup, L. Jessen, and J. Stoustrup, “Modular modelling of a reefer container,” *Submitted for publication*, 2013.

[9] B. Rasmussen, A. Musser, and A. Alleyne, “Model-driven system identification of transcritical vapor compression systems,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 444–451, 2005. doi:10.1109/TCST.2004.839572

[10] W.-J. Zhang and C.-L. Zhang, “A generalized moving-boundary model for transient simulation of dry-expansion evaporators under larger disturbances,” *International Journal of Refrigeration*, vol. 29, no. 7, pp. 1119 – 1127, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700706000570. doi:10.1016/j.ijrefrig.2006.03.002

[11] M. Willatzen, N. Pettit, and L. Ploug-Sørensen, “A general dynamic simulation model for evaporators and condensers in refrigeration. part i: moving-boundary formulation of two-phase flows with heat exchange,” *International Journal of Refrigeration*, vol. 21, no. 5, pp. 398 – 403, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000911. doi:10.1016/S0140-7007(97)00091-1

[12] N. Pettit, M. Willatzen, and L. Ploug-Sørensen, “A general dynamic simulation model for evaporators and condensers in refrigeration. part ii: simulation and control of an evaporator,” *International Journal of Refrigeration*, vol. 21, no. 5, pp. 404 – 414, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000923. doi:10.1016/S0140-7007(97)00092-3

Paper B

Modular Simulation of Reefer Container Dynamics

Kresten K. Sørensen[a,b] , Jens D. Nielsen[a] and Jakob Stoustrup[a]

[a]*Section for Automation and Control, Department of Electronic Systems*
Aalborg University, Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
[b]*Lodam electronics a/s, Kærvej 77, 6400 Sønderborg, Denmark*
E-mail: [kresten,jdn,jakob]@es.aau.dk

The paper has been published online before print in *Simulation*,
December 29, 2013, doi:10.1177/0037549713515542.

Abstract

The amount of food transported long distances in reefer containers is constantly increasing and so is the cost per mile because of rising fuel prices. One way to reduce the cost is to minimize the energy consumed by reefer containers through a better controller but in order to achieve this a fast and flexible simulation model is needed for controller development. The simulation model may also be used for developing fault diagnosis methods for the reefer container and thereby further lowering costs by reducing the amount of functioning spare parts that is replaced and by providing early warning for faults enabling preventive maintenance. In this paper the feasibility of using different simulation methods is assessed with the goal of identifying a fast but accurate method that works well in a multi-rate environment. A modular multi-rate simulation environment for a dynamical system consisting of components with different dynamical speeds is presented with an improvement of previous results. The simulation speed is improved by 350% with no reduction in accuracy of the solution, by substituting the MATLAB® `ode15s` *solver with an explicit first order solver with a step size calculation algorithm that ensures numerical stability and that the error is bounded using a minimum of calculations. The reefer container model is simulated using both* `ode15s` *and the proposed method both in multi-rate and monolithic configurations. The results are analyzed and compared with respect to speed and accuracy.*

1 Introduction

Reefer containers are used extensively to transport food all over the world and by mid 2008 there was a worldwide fleet of 11.4 million TEU (Twenty Foot Equivalent) predicted to grow by 69% to 19.3 million TEU by 2013, according to [1]. The container cargo market is highly competitive and therefore it is interesting to examine any possible means of lowering the Total Cost of Ownership (TCO), which covers the cost of initial procurement and operation during the lifetime, of a reefer container. The average lifetime expectancy for a reefer is 12 years and the cost of procurement is small compared to the cost of inspection, repairs and energy over the lifetime of the container. This of course means that there is a potential for lowering the TCO through a reduction of these three factors; that will be clarified in the following. Before every trip a reefer must complete a Pre Trip Inspection (PTI) test which is a self test where the container tests that its cooling capacity is as specified and that there is no other obvious problems. This requires that the container is taken to a special PTI area on the harbor where it is plugged in and the PTI test is started by an operator, for a fee that covers handling and power. The PTI test is executed in this way because it is a programmed sequence that requires the container to be empty such that it may change the temperature set point and cooling capacity without risk of damaging the cargo. But if this self check could be done by examining the relationship between control signals and sensor inputs without changing set point or cooling capacity it would be possible to avoid the PTI test because the container controller would be aware of the capabilities of the container at all times. This self check may be extended further from covering just the available cooling capacity and elementary faults to accurate detection and identification of the majority of likely failures on the container, thereby lowering the time and effort needed to carry out maintenance during the trip. It can also enable early warning on errors that increase over time, which opens up the possibility to carry out preventive maintenance between trips where parts that is close to failing can be replaced and that is always preferable over repairs that has to be carried out at sea while the container is in use. These measures would increase reliability due to better and in-time maintenance of the container and thereby also the chance of loosing a cargo due to system failure. But in order to accurately identify faults it is necessary to use model or observer based FDI techniques and for that an accurate model that can be embedded in the container controller is needed. In general, a fairly high-fidelity model can be required in order to detect certain types of faults. Some faults might require a good static model fit, whereas other faults might require the model to fit well dynamically [2]. In this paper, we will mainly focus on the latter class of faults.

The average power consumption of a reefer is 3.6kW per TEU [3] and assuming an idle time of 50% the total power consumption of the worlds reefer container fleet is above 20GW. In recent years the shipping business have been looking into ways to cut costs and the energy consumed by reefers, that earlier were deemed insignificant compared to the main engine energy consumption, has now come into focus. There are two ways to lower energy consumption; By changing the mechanical design of the reefer and the cooling system or by optimizing the way it is controlled. This paper use the Star Cool container [4] as an example but the mechanical design of this reefer is already nearly optimal, leaving control optimization as the best option for efficiency improvements. The refrigeration unit is controlled by a microprocessor with a control algorithm already optimized for energy efficiency, on the short term. On the long term, however, the potential for energy savings is large if daily variations in ambient temperature are exploited by cooling more when the ambient temperature is low and less when it is high [5, 6]. This method "stores" some cooling in the cargo during low ambient temperature periods where refrigeration system efficiency is higher and takes it back when the ambient temperature is high and refrigeration system efficiency is lower. The cargo in a reefer container is the single largest thermal capacitance of the system with a time constant that is several orders of magnitude higher than the smallest time constant of the refrigeration system dynamics, which yields a very stiff system. Controller development is an iterative process where a design is tested, evaluated, modified and tested over again and again until a satisfactorily result is achieved. On a system like this the time consumed can be very long because a test must last several days, due to daily temperature variations and since the system in general has large time constants.

If many iterations are needed the time required will be too much and too expensive. Therefore a simulation model is needed to speed up the iterative cycle, and in order to optimize the controller with respect to energy consumption the model must capture both the dynamics of the cargo and the refrigeration plant. The control optimization problem for the reefer container is to keep the cargo within certain temperature limits while using as little energy as possible. Because the efficiency of the refrigeration system is inversely proportional to the ambient temperature and the ambient temperature cycles during a day it is beneficial to apply the cooling when the ambient temperatures is at its lowest and use the cargo mass as a "storage" for cooling. This will however require that the simulator and the model are stable, fast, have adequate accuracy and are computationally light-weight enough to run on an embedded system.

There exist many different modeling and simulation tools where models composed of different components may be simulated numerically and in the following a few general purpose simulators and energy system simulators are described.

Within the field of energy system simulation the modular approach is well known and has been used in i.e. the simulation environment TRNSYS [7] for more than 35 years to simulate the behavior of a composition of system components over time, using a numerical solver. TRNSYS provides a library of common energy system components that may be combined using a custom system description language.

Another approach is used by DYMOLA [8] that implements the MODELICA [9] language which is an equation-based object-oriented modeling language. In MODELICA symbolic equations that define the dynamical behavior of a component may be entered in a non-causal way, leaving the task of ordering and reducing the final set of equations to the simulation engine before a simulation can be run.

WINDALI is a tool developed at the Department of Energy Engineering at Technical University of Denmark that is aimed specifically at simulation of refrigeration systems [10]. It uses a semi-explicit DAE solver that can handle discontinuities and models can be programmed in any language that can be compiled to a DLL that can be used by the simulation environment. Because the model runs as native machine code simulation WINDALI is very fast.

MATLAB® is a high level language for numerical computing that provides state of the art toolboxes for a wide range of engineering disciplines, including modelling and controller design. A range of solvers are available that enables simulation of ODE's, PDE's, and DAE's. Most active research in control engineering uses MATLAB® as a tool because it is easy to use and has a flexible external interface.

SIMULINK® is a tool from the MATLAB® suite that is used extensively by researchers and engineers to simulate complex systems built from blocks in a GUI. The user may define blocks consisting of many smaller elements and thereby a highly complex model can be arranged in a manageable way. SIMULINK® has a large library of pre-defined blocks that covers a wide area of applications, and a selection of explicit and implicit solvers that may be set to both variable and fixed step size. The model connections are examined prior to simulation and existing algebraic loops are attempted solved by a built-in algebraic loop solver [11].

Fault detection and estimation has been shown to work well for refrigeration systems and in [12] a complete FDI approach consisting of an extended Kalman filter and a bank of unknown input observers are described. The combination is shown to have detection and identification capability for sensor and parametric faults on a refrigeration system but it depends on an accurate model for design of the extended kalman filter and the unknown input observers.

In [13] a modular multi-rate approach where the components may be simulated by different solvers is presented. The component interconnections are managed by a time-discrete linker and scheduler that connects the inputs and outputs of the components and exchanges data at discrete time instants. The advantage of this approach is that it is possible to combine simulators from multiple domains in science and engineering that was not originally designed to work together. When coupling systems with different dynamical behavior with a discrete-time linker algebraic loops may cause instability and in [13] this problem is described and a method that guarantees stability when algebraic loops are present is proposed. The modular approach has several advantages; It is faster to build a new simulation model from a library of components than starting from scratch and it is also easier to maintain because the code is naturally divided and therefore less likely to be entangled across component models. Simulating a stiff system as a monolithic block has some drawbacks especially under changing conditions on the fast states [14], because the stiff solver must evaluate the entire system in steps small enough to achieve satisfactorily accuracy on the fast states. This is however more than adequate for the slow states and therefore many of the calculations done on them is essentially a waste of computer power. A way to lessen this problem is to convert the ordinary differential equations of the fastest states into algebraic equations [14, 15], but this can not be used on all states in the current application because too much precision is lost. Another way of reducing the computational load is by using a multi-rate simulator that divides the model in components by their dynamical speeds and thereby yield a significant increase in speed, as shown by [16] and [17], because slow components are no longer simulated at an unnecessary small step size. In [18] a modular multi-rate simulation environment for simulations of a refrigeration container was described and the theoretical increase in speed was calculated.

For the reefer container application there is a need to simulate the model in small single steps in order to incorporate an external controller, and this is possible with the modular multi-rate approach. It is attempted to quantify the impact of the multi-rate method on speed and accuracy through multiple simulation experiments on the refrigeration system using different simulator configurations on the refrigeration container model and comparing the results. Furthermore it is attempted to increase the simulation speed by using a simple solver tailored for the problem at hand instead of MATLAB®'s built-in solvers in order to enable the model to be used as an observer on an embedded system. The simple solver should be fast, reliable and easy to implement but it is not required to be as accurate or versatile as the MATLAB® solvers.

This paper presents a simulation environment for MATLAB® that provides modular multi-rate simulation of a system consisting of fast and slow dynamic components. An early version of the environment was described in [18] and some preliminary results were reported but the implications of multi-rate simulation of this system was not treated in detail. In the present effort the calculations of the speed increase is backed up by experiments and the implications of replacing MATLAB®'s numerical solvers with an explicit first order solver with a simple step-size algorithm is investigated. This leads to an improvement to the simulator that increase the simulation speed by 350% while maintaining adequate accuracy and the ability to interact with the model during simulation. The numerical stability of the new solver is investigated and it is shown that it is possible to determine a solver setup for each of the modular components a-priori that guarantees stability and a bounded local error. Simulations of the refrigeration container using different combinations of single and multi-rate, modular and monolithic simulator configurations are compared in order to identify the source and nature of decreased accuracy that arise due to the multi-rate ZOH delay between component models. Finally the simulator and model stability for long term simulations are demonstrated by simulating the model in open-loop with control signals recorded on a real system as inputs. The aim of this work is to find a simple and robust algorithm that is capable of simulating the refrigeration container model using a minimum of CPU time but with adequate accuracy for development of model based controllers and for use as a full system observer. Currently the MATLAB® `ode15s` is used to simulate the model and therefore it is attempted to find a simpler replacement, tailored for this task.

2 Methods

This Section investigates the benefits and disadvantages for multi-rate and monolithic simulation methods applied to a modular model of the Star Cool refrigeration container, with special emphasis on finding a simple solver that is suitable for implementation on an embedded platform. The simple solver should be able to simulate the model with a precision that is adequate for FDI and for the model to be used as an observer for a model based controller. Furthermore it must be able to simulate the model in short steps such that it can produce an output at regular intervals that may be used by the FDI and control algorithms on the container controller.

2.1 Refrigeration System Model

The model of the Star Cool refrigeration container is used for development, testing and validation of control and fault detection algorithms and therefore the model reflects the system properties that are important for these tasks. The salient properties are the dynamics of the refrigeration system used for control of the evaporator and the dynamics of the container walls and cargo that are relevant for control of the compressor speed. The equations are based on first principles where infinitesimal terms that has little impact on the accuracy or stability of the solution has been removed. This results in a set of mainly first order equations that due to the highly nonlinear relationship between evaporation temperature and pressure of the refrigerant have varying time constants. One good thing about a refrigeration system is that while operating within the normal limits for the system it settles at a steady state when the control inputs and ambient conditions are steady. Therefore the accuracy of the model is adequate if the equations can capture the varying rates of exponential decay towards an input dependent steady state. The system has the property that the slow states are isolated in the component that models the container walls and the cargo, and the fast states are present in five different refrigeration system component models. The refrigeration system are divided in components because it is easier to maintain code that is divided in modules with clean interfaces but it also gives the advantage that each component model can easily be substituted with another component if needed. This division also enables the use of custom solvers for each component, depending on what is better suited for the component. A schematic of the refrigeration system is shown on Figure B.1 and the details of the system model have been described in detail in [19].

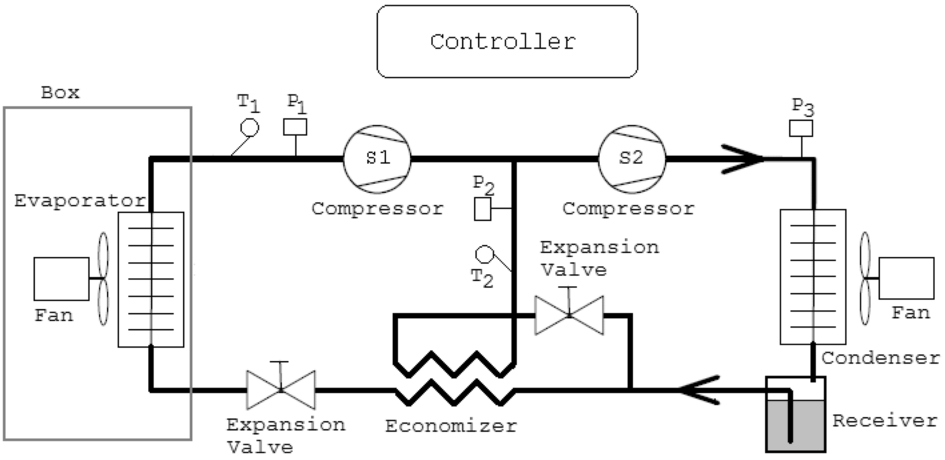


Fig. B.1: Refrigeration System for the Reefer Container

The model has 80 states of which 30 are discrete and 50 continuous, divided in three discrete and six continuous component models. The model framework is described in the following Section 2.2, and the simulation methods are described in Section 2.3.

2.2 Modelling

The model of the refrigeration system is divided into components that each represent a physical component of the system, i.e. a condenser or an evaporator. Each **component model** is described by two files; an `m` file that holds the input/output equations of the model and an `XML` file that describes the properties of the inputs and outputs of the `m` file. The **simulation model** is the overall model for the refrigeration plant and its properties are described by an `XML` file that holds a list of included component models and the connections between them. Therefore the structure of the simulation model is defined by the simulation model definition file, and from this the model loader can create a simulation object that is used by the simulator.

Component Model Syntax

Modelling of component models is basically the same as for normal ODE model functions that may be solved by MATLAB®'s built-in `ode` solvers, but additional info is needed by the model loader in order to do type-checking when connecting the inputs and outputs of the component models. Each component model is described by an `m` file containing the input/output equations, an `XML` file describing the input/output properties of the model, its execution mode, and the name of the corresponding `m` file. The syntax for writing the component model `XML` file is shown in Listing B.1.

Listing B.1: Component Model Syntax

```
 <?xml version="1.0" encoding="UTF-8"?>
<component name=["Component Name]">
<inputs>
  <input name=["Input1 Name"] type=["Input1 Type"]
    description=["Input1 Description"]/>
  .
  .
  <input name=["InputN Name"] type=["InputN Type"]
    description=["InputN Description"]/>
</inputs>

<states>
  <state name=["State1 Name"] type=["State1 Type"]
    default=["State1 Start Value"]
    description=["State1 Description"]/>
  .
  .
  <state name=["StateN Name"] type=["StateN Type"]
    default=["StateN Start Value"]
    description=["StateN Description"]/>
</states>

<connectors>
  <connector type=["type"] conid=["Conn. Name"]
  >
    <{input, state} name=["Name"] type="Physical Entity"/>
    <{input, state} name=["Name"] type="Physical Entity"/>
   </connector>
  .
  .
</connectors>

<control_inputs>
  <input name=["Input Name"] type=["Input Type"]
               description=["Input Description"]/>
  .
  .
</control_inputs>

<simulation method={"ode15s", "call"} call=[".m File Function Name"]/>
<filename>[.m File Name]</filename>
</component>
```

The `<inputs>` section contains a list of the inputs to the component model, and it is important that the inputs are listed in the same order as they occur in the input vector of the model function. Each input has a name, a type used for type-checking when inputs are connected, and a description. There must be the same number of inputs in the input list as the length of the input vector of the corresponding model function. The `<states>` section is similar to the input section except that it describes the states or outputs of the function and that a default value must be declared. The default value is used as initial value in simulations when the simulation tool is not given an initial state vector to start from. Signals may be grouped together in connectors that allows the user to connect a set of signals from one component model to a similar set on another component model in one operation.

Because this environment is used for refrigeration systems, it has a built-in connector class for refrigeration pipe interfaces but obviously, for other applications, other connections will be relevant - see Simulation Model Syntax in Subsection 2.2. On the refrigeration pipe interface three variables exist; a mass flow $\dot{m}$, a pressure p, and an enthalpy h. The model loader will return an error if each refrigeration pipe interface does not contain exactly one of each of the mentioned aforementioned types. The individual variables may be either an input or a state but the model loader will check that each of the inputs can be connected to a state of the correct type when two refrigeration pipe interfaces are connected. This saves the user the work of having to connect the variables manually, but when building the component models attention must be given to where the different states that are shared between models are located.

Simulation Model Syntax

The syntax for the simulation model XML file are shown in Listing B.2.

Listing B.2: Simulation Model Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<simulation_model name=["Simulation Model Name"]>
<component_path>[Component Library Path]
</component_path>

<components>
  <component name=["Component Model Name"]
    file=["Component Model XML File Name"]/>
  .
  .
  <component name=["Component Model Name"]
    file=["Component Model XML File Name"]/>
</components>

<connections>
  <connector name=["Connector Name"]>
    <component name=["Component Name"] conid=["Connection Name"]/>
    <component name=["Component Name"] conid=["Connection Name"]/>
  </connector>

  <connection name=["Connection Name"]>
     <component name=["Component Name"] type="Input" input=["Input Name"]/>
     <component name=["Component Name"] type="Output" output=["Output Name"]/>
  </connection>

</connections>
</simulation_model>
```

The simulation model is composed of a set of component models and their connections described by the simulation model XML file, containing a list of the included component models and a description of how the components are connected. The <components> section lists the component models used in the simulation model and it is allowed to use a component model more than once if they are given unique names. In the <connections> section all the connections in the simulation model are listed. A connector connection is established as in the <connector> sections by giving the name of the two component models and the connector on each of the components. Inputs and outputs that are not associated with connector interfaces, such as control inputs to actuators, are connected in a <connection> section by listing the two components one by one. In each of the <component> sections it must be stated whether the signal is an input or an output, and what the name of the signal is. The model loader provides type checking on connections between components and gives a precise description, with the names of the implicated components and signals, in case of an eventual error in the set of connections between components.

The model may be loaded into MATLAB® with a model loader function that loads each of the components and creates a struct containing all the information necessary for simulation. During simulation the states of the component models are kept in a single vector, denoted **X**, and therefore two matrices that maps between **X** and component model I/O are generated for each component model. The matrix $\mathbf{Z}_k$ maps from **X** to the component model x-vector $\mathbf{x}_k$ such that

$$\mathbf{X} = \mathbf{Z}_k \cdot \mathbf{x}_k \qquad \text{(B.1)}$$

$$\mathbf{x}_k = \mathbf{Z}_k^T \cdot \mathbf{X} \qquad \text{(B.2)}$$

where the index k denotes the component model number. Component model inputs $\mathbf{u}_k$ are mapped from **X** by the connection matrix $\mathbf{CM}_k$ such that

$$\mathbf{u}_k = \mathbf{CM}_k \cdot \mathbf{X} \qquad \text{(B.3)}$$

Because $\mathbf{CM}_k$ maps from state variables that reflect a physical value to inputs that take a physical value, it is a zero-one matrix and must have exactly one "1" in each row. When the model loader has created the mapping matrices they are used to check the connection integrity of the system model such that all inputs are connected to exactly one state.

2.3 Simulation

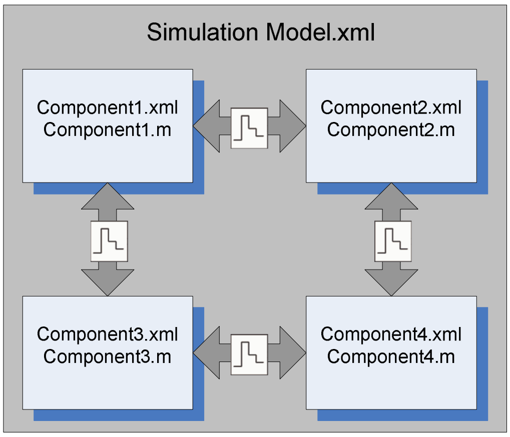
Multi Rate Simulation

Multi rate simulation of the system model is done in discrete time steps defined in the time vector given in the simulation function call. In each of the time steps the component models are simulated separately, according to the method defined in the component model `XML` file and the results are then combined into the **X** vector containing the states for all of the component models. Component models that are purely algebraic are evaluated in one operation like a normal MATLAB® function, and dynamical models are solved by one of MATLAB®'s built-in `ode` functions. Fig. (B.2) shows a possible structure for a simulation model consisting of four component models.

The discretization of the signals between the component models is equivalent to inserting a zero order hold between all models as shown on Fig. (B.2). The input to the individual component models are calculated by applying (B.2) and (B.3) and given as arguments to the appropriate simulation function such that

$$\dot{\mathbf{x}}_k(n) = f_k\left([\mathbf{t}(n-1) \quad \mathbf{t}(n)],\quad \mathbf{Z}_k \cdot \mathbf{X}(n-1),\quad \mathbf{CM}_k \cdot \mathbf{X}(n-1)\right) \qquad \text{(B.4)}$$

where f_k is the simulation function for the referenced component model and $[\mathbf{t}(n-1) \quad \mathbf{t}(n)]$ is the time interval in which to simulate. The results from the each of the



h

Fig. B.2: Example of a Simulation Model Structure

component simulations are then combined into the system state vector X by

$$\mathbf{X}(n) \quad = \quad \sum_{k=1}^{K} \mathbf{Z}_k \cdot \mathbf{x}_k(n) \tag{B.5}$$

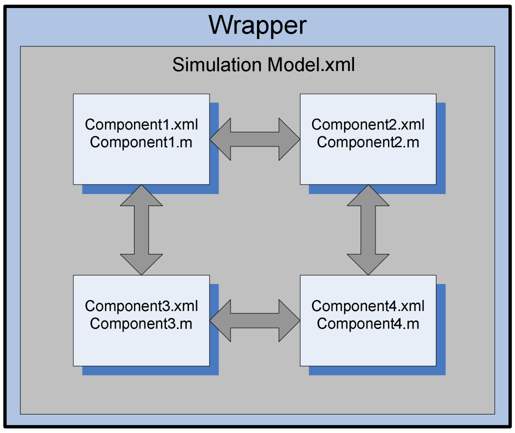
where K is the total number of components in the model. The simulation environment may then proceed and simulate the next time step with the same procedure as above.

Monolithic Simulation

In a monolithic simulation the entire collection of component models are lumped together by a wrapper function and treated as a single model that may be simulated by `ode15s` or another solver that accepts functions with a similar interface. This means that for every iteration of the solver all component functions are evaluated in order to find a derivative for the entire collection of component models.

In [18] the multi-rate simulation speed of the reefer container system was compared to the calculated speed for a monolithic simulation of the same system, showing that the modular approach was more than three times faster than the monolithic. There was no comparison to an actual monolithic simulation of the system but by introducing a monolithic wrapper this is now possible.

The monolithic wrapper encapsulates the entire model into one function as shown on the example in Figure B.3. The function `simulate _model_monolithic()` is used to simulate the model by letting `ode15s` [20] simulate the wrapper for the desired period. Therefore the exact same component functions can be simulated as a monolithic block without the need to combine them into one function or changing them in any way, giving



h

Fig. B.3: The monolithic wrapper

a lower probability for user errors. `ode15s` is as default set to use at least ten iterative steps for each time period of the supplied time vector which in this case is 1000 steps of one second, similar to the one used for the modular simulation.

In order to achieve a fair comparison to the modular simulation the monolithic solvers maximum step size is increased from the default 0.1s to 1s such that it can use multiple steps if the dynamics require it, but also run faster if possible. The wrapper will forward results from the ODE and DAE components directly because their output is the state vector gradient, but for the algebraic functions the output is simply the new state at the next sample point and therefore the gradient must be calculated and forwarded to the solver. This is simply the difference between the new and the old state because the algebraic functions of this model is designed for a sample time of one second.

Implications of using the `ode15s` solver

When calling the `ode15s` solver there is a considerable startup overhead and because the solver is called for each of the continuous components in every discrete time step the total time lost to this is large. According to [20] the `ode15s` solver relies on a Jacobian that is generated automatically when it is not supplied by the user, as is the case in this study. Because the calculation of the jacobian is requires a lot of computations the solver only generates the Jacobian when simulation is started, when the order and step size is changed or if the solution is converging too slowly. Another drawback of using the solver in short steps is that it uses a very small step size in the beginning of the simulation period and then gradually increases it. Because the simulation period is so short the solver never reaches larger step sizes and therefore it never reaches the efficiency expected of a variable time step solver, leading to a longer simulation time.

In fact, if the model is simulated for 1000 seconds as a monolithic block by `ode15s` in one go, the model can be simulated at a speed that is higher than the speed of a multi rate simulation using `ode15s` and this show that a great deal can be gained by using the solver differently, but interaction with the model during simulation is more difficult when the model is simulated in one go by `ode15s`.

Proposed numerical method

The main goal of the simulation algorithm is to lower the computational burden in order to increase simulation speed and this means that the model should be simulated using as few evaluations of the component model functions as possible and that the startup overhead for the simulation algorithm should be low. The chosen method, a variable step size forward euler (VS-FE), is the simplest possible and it requires only one evaluation of the component model function per step but it can be unstable if the selected step size is too large.

The stability region for the explicit euler method consists of the points in the region z given in (B.6) where λ is the eigenvalue of the system and h is the step size [21]. If (B.8) is observed the solution converges and therefore the maximum permissible step size can be found if the eigenvalue is known (B.10)

$$z = \lambda \cdot h \tag{B.6}$$

$$y_{n+1} = (1 + \lambda \cdot h) \cdot y_n \tag{B.7}$$

$$|1 + z| < 1 \tag{B.8}$$

$$|1 + \lambda \cdot h| < 1 \tag{B.9}$$

$$h_{max} = \frac{2}{|1 + \lambda|} \tag{B.10}$$

Although the solution converges when using the maximum permissible step size it gives a very inaccurate result because the error is dependent on the step size as shown in (B.11) and in reality the used step size should be orders of magnitude smaller than the maximum permissible step size in order to ensure a smooth solution.

For higher order numerical algorithms the local error is found by comparing the result of two different order calculations and while this requires more calls to the model function and thus a higher computational burden, it also yields a better accuracy because an accurate error estimate gives a good step size. Steps size selection algorithms normally compares the local truncation error of the solver against a given fixed tolerance and select the steps size such that the local truncation error is smaller than or equal to the tolerance. The method proposed here is an adaptation of a textbook method that is described in [22] and [21]. The local error E for the first order explicit euler method is given by

$$E = \frac{1}{2} h^2 |\ddot{y}| \tag{B.11}$$

where h is the step size and $\ddot{y}$ is the second order derivative of the state. Substituting E with the acceptable error local error tolerance TOL and isolating h yields the step size that limits how much the first order gradient is allowed to change in each step and therefore also the magnitude of the local error

$$h = \sqrt{\frac{2 \cdot TOL}{|\ddot{y}|}} \tag{B.12}$$

where TOL is the absolute tolerance for the second order derivative of the system state thus bounding the step size with respect to how fast the first order derivative changes and thereby bounding the local error.

The different types of states in the container model are numerically very different and therefore the second order derivative is also numerically very different. This means that if the step size is to be determined from the largest second order derivative given by a model with multiple states the larger states will usually also have the larger second order derivative. The consequence of this is that states that are small may have a relatively large second order derivative but it will be ignored and that can result in numerical instability for states of small magnitude. For this model the notion of an absolute error tolerance is therefore impractical and in order to address this problem the second order derivative is normalized with respect to the size of the states as shown in (B.13).

$$\ddot{\mathbf{y}}_{norm} = \frac{|\ddot{\mathbf{y}}|}{|\mathbf{y}| + 1} \tag{B.13}$$

$$h = \sqrt{\frac{2 \cdot TOL}{max(\ddot{\mathbf{y}}_{norm})}} \tag{B.14}$$

$$h = \sqrt{\frac{2 \cdot TOL \cdot (|\mathbf{y}| + 1)}{|\ddot{\mathbf{y}}|}} \tag{B.15}$$

Normalizing the second order derivative has a drawback: if a state approaches zero the normalized second order derivative will approach infinity and result in very small steps. This issue is addressed by adding one to the state vector and therefore as a state approaches zero its normalized second order derivative will approach the real second order derivative. Due to the normalization of the second order derivative TOL is now a measure of error relative to the state size for large states and it approaches a measure of absolute error as the states approach zero. For the solvers in the MATLAB® ODE suite the absolute and relative tolerances can be set independently [20], but that flexibility is not needed for this application and therefore it is left out in order to have as simple a solver as possible.

In the remaining part of this section the proposed simulation algorithm and its step size calculation method is described and analyzed, starting with the MATLAB® code for the resulting simulation algorithm that is shown in Listing B.3. The algorithm shown above uses exactly one call of the model function for every step and because the rest of the calculations are quite simple the function has a low overhead. In order to accommodate the different experiments carried out in Section 2.4 it is possible to force a fixed step size and limit the variable step size between a minimum and a maximum. Each step is started with an evaluation of the function that is being simulated in order to obtain the first order derivative. The second order derivative is then calculated from the old and the newly obtained first order derivatives and it is then normalized according to (B.13). Then the step size is calculated and limited and finally the step is taken whereafter the sequence is repeated until the simulation reaches the end time. In the next section the different experiments that are used to verify the performance of the simulation environment and the solver is described.

Listing B.3: Proposed simulation algorithm

```
function [T, Y, error] = FESolver(fct_handle, t, y0, opts, u, p)

N = (t(2) - t(1))/opts.FE_minstep; % Maximum number of steps.

T = zeros(1,N);                    % Time output vector
Y = zeros(length(y0),N);           % Solution output vector

T(1) = t(1);                       % Set up initial values
T_end = t(2);
Y(:,1) = y0;
h = 1;
y_dot_old = zeros(length(y0),1);   % Assume zero gradient at beginning.
n = 1;
error = 0;
while(T(n) < T_end)
  % Evaluate component function and obtain first order derivative.
  y_dot = feval(fct_handle, T(n) , Y(:,n), u, p);
  % Calculate second order derivative from previously stored first
  % order derivative and step size.
  y_dotdot = (y_dot - y_dot_old)/h;

  if(opts.FE_Stepsize == 0)
    % Normalize the second order derivative with respect to the size of the
    % states, with one added in order to avoid high values for states near
    % zero.
    norm_y_dotdot = abs(y_dotdot) ./ (abs(Y(:,n)) + 1);

    % Use the largest change in gradient to calculate the step size.
    h = sqrt(2*opts.FE_reltol / max(norm_y_dotdot));

    % Limit the stepsize between min and max step size and ensure that it
    % does not exceed the simulation end time.
    h = LimitValue(h, opts.FE_minstep, min(T_end - T(n), opts.FE_maxstep));

  else
    h = opts.FE_Stepsize; % For fixed step size.
  end

  % Update outputs
  T(n+1) = T(n) + h;
  Y(:,n+1) = Y(:,n) + y_dot * h;
  y_dot_old = y_dot;
  % Check for faults and stop simulation on error
  if(sum(isnan(y_dot)) > 0 || sum(abs(imag(y_dot))) > 0)
    disp('FESolver:␣Warning␣NaN␣or␣IMAG␣returned␣from␣simulation␣')
    error = 1;
    T(n+1) = T_end;
  end
  n = n + 1;
end

% Only return the number of steps that was executed.
T = T(1:n);
Y = Y(:,1:n);
```

2.4 Experiments

The required accuracy for the problem at hand is given by the objectives of the model; It must be adequate to do controller experiments and for the model to be used as an observer for FDI in an embedded system. For the controller experiments it is important that the closed loop dynamical behavior of the system is accurate but because the controllers are designed to be robust to a rather large variance in the mechanical system of the containers due to wear and tear and faulty or inadequate maintenance, a max error of 2% is acceptable. The average error is the normalized average error on all states and the max error is the largest error on any of the states. The errors are calculated from the states of all components except the controller because it will be replaced by an external controller when the model is moved to the embedded system.

When used as an observer for fault detection it is important that the faults are low because a large uncertainty on the simulation result will require higher fault detection thresholds to avoid false positives and thus the ability to detect small faults will be limited or the time to detect a fault will be too long. As an observer the model will be running in open loop and therefore it is important that the long term static behavior is accurate. It has been chosen to require that the max error on the variables important for control of the system stays below 5% for a three hour open loop simulation where the control inputs are recorded from a real container.

The proposed simulation method is tested in a variety of different scenarios on order to qualify its performance and accuracy in the modes of operation where it is to be used. Furthermore a series of experiments are carried out to investigate the `ode15s` overhead and the nature of the reduction in accuracy caused by the proposed VS-FE method. Table B.1 shows a short list of the test carried out:

No.	Description	Solver
1.	Reference simulation - Fixed 1ms step monolithic	FS-FE
2.	Reference simulation - Fixed 1ms step modular	FS-FE
3.	Continuous monolithic	`ode15s`
4.	Stepwise monolithic	`ode15s`
5.	Modular ODE (previous method)	`ode15s`
6.	Modular variable step FE (new method)	VS-FE
7.	Monolithic variable step FE	VS-FE
8.	Open loop modular variable step FE	VS-FE
9.	Open loop modular ODE	`ode15s`

Table B.1: List of tests

Test 1 and 2: Reference simulations. In order to calculate the error of the different experiments it is necessary to have a reference simulation that represents the true solution and this is achieved by simulating the model with a fixed step size that is very small compared to the dynamics of the system. In this case a 1ms step size is found to be adequate by simulating at both 1ms and 2ms step size and comparing the results from which it was concluded that the error had converged. Two reference simulations are made; The first test is a monolithic simulation where the model is simulated as one large block at 1ms resolution and that represents the true solution without the impact from the ZOH of the modular simulation environment. The second reference is a modular simulation where the components are simulated independently with 1ms step size for one second at a time and the purpose is to isolate the error introduced by the ZOH between the component models.

Test 3 and 4: Stepwise and Continuous monolithic ODE.
In these two tests the model is simulated monolithically by the `ode15s` solver as a sequence of one second steps and in one continuous go in order to identify the startup overhead of the solver.

Test 5: Modular ODE (previous method).
This is the simulation method presented in [18] and it is used as a baseline for verification of the new VS-FE solver.

Test 6: Modular variable step FE (new method).
The VS-FE solver is compared to both reference simulations and to Test 5 in order to verify the performance of the simple solver in a modular environment.

Test 7: Monolithic variable step FE.
It may be a viable option to simulate the model monolithically in order to avoid any implications from the modular simulation and therefore this test is carried out to verify the speed and accuracy of this solution.

Test 8 and 9: Open loop modular VS-FE and ODE.
In order to verify the long term stability and accuracy of the simulation environment and the VS-FE solver is used to simulate the system in open loop over a three hour period with control signals recorded from a real container. The output of the model on the measurements that are important for control of the system is then tested for accuracy by comparing them to sensor measurements from the same real life data set as the control signals.

Test sequence
The simulation model controller used in this experiment is programmed to ramp the compressor speed 9 times during a 1000 second simulation in order to excite the system and create some events that challenge the solver with large gradients on the system states. The modular simulation model that was presented in [18] was not excited as

much as in this test, leading to a low error due to the small gradients on the model states. When the system is excited more strongly the state gradients will be higher yielding a larger difference to the zero-gradient of the ZOH in the modular environment, which gives a larger error.

3 Results

In this section the results of the tests carried out in Section 2.4 are listed and analyzed in order to identify viable options for simulating the model for controller experiments and for use as an observer on an embedded system. The measured variables are the simulation time, the average error and the max error. The average error is the normalized average error on all states and the max error is the largest error on any of the states. The errors are calculated from the states of all components except the controller because its states may go to zero and yield an infinite normalized error.

On Figure B.4 the main results are shown and the references for calculating the errors are the monolithic reference for the monolithic simulations and the modular reference for the modular simulations. The error tolerance for the VS-FE method has been selected such that the error is approximately the same as produced by the `ode15s` solver with standard tolerance settings [20] in order to make comparison of the results easier. From the results on Figure B.4 it can be seen that the simple VS-FE solver are able to outperform the more advanced `ode15s` solver and in the following the reason for this will be analyzed.

Closed loop tests

The results of the relevant tests compared to the monolithic reference can be seen in Table B.2, where the time column is the time used simulating the 1000 second test and the steps column has the number of steps used by the solver. For modular simulations the number of steps are the sum of steps used on the 9 components and for monolithic simulations it is the number of times the monolithic wrapper function has been called by the solver.

No	Description	Solver	Time	Mean Err.	Max Err.	Steps
2.	Modular Reference	FS-FE	1458s	0.311%	1.615%	1000000
3.	Cont. Monolithic ODE	`ode15s`	35.6s	0.059%	0.447%	11878
4.	Stepwise Monolithic ODE	`ode15s`	399s	0.007%	0.029%	126942
5.	Modular ODE	`ode15s`	37.2s	0.315%	1.654%	16562
6.	Modular VS-FE	VS-FE	11.3s	0.298%	1.499%	36288
7.	Monolithic VS-FE	VS-FE	9.89s	0.065%	0.273%	3970

Table B.2: Simulation results compared to the monolithic reference.

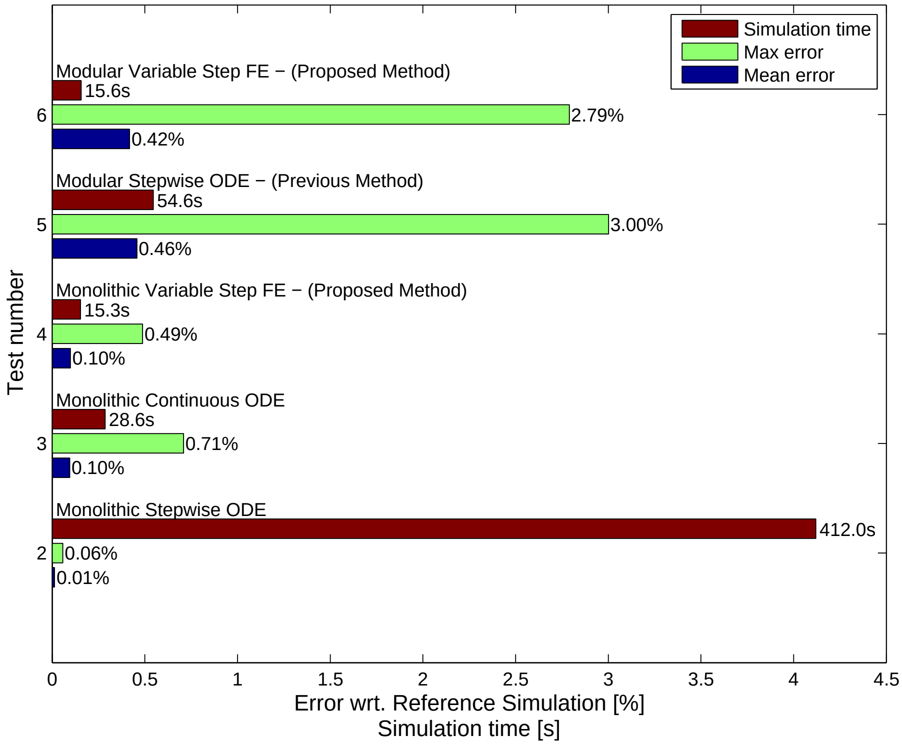


Fig. B.4: Comparison of test results for monolithic and modular methods.

The error for *Test 2 - Modular reference* shows the magnitude of the error introduced from the multi-rate simulation itself and it is a consequence of the ZOH delays between component models. Therefore this error is also large when gradients are as large as they are in this test, due to the high excitation of the model. In a controller experiment setup the ZOH delays are not critical because they are small compared to the dynamics that are important for control which according to [23] for a refrigeration system are the thermal masses of the evaporator and the condenser. Under normal operation the refrigeration system is in steady state most of the time and therefore the temporal inaccuracy during steep gradients imposed by the ZOH delays has little impact on the accuracy of long term simulations.

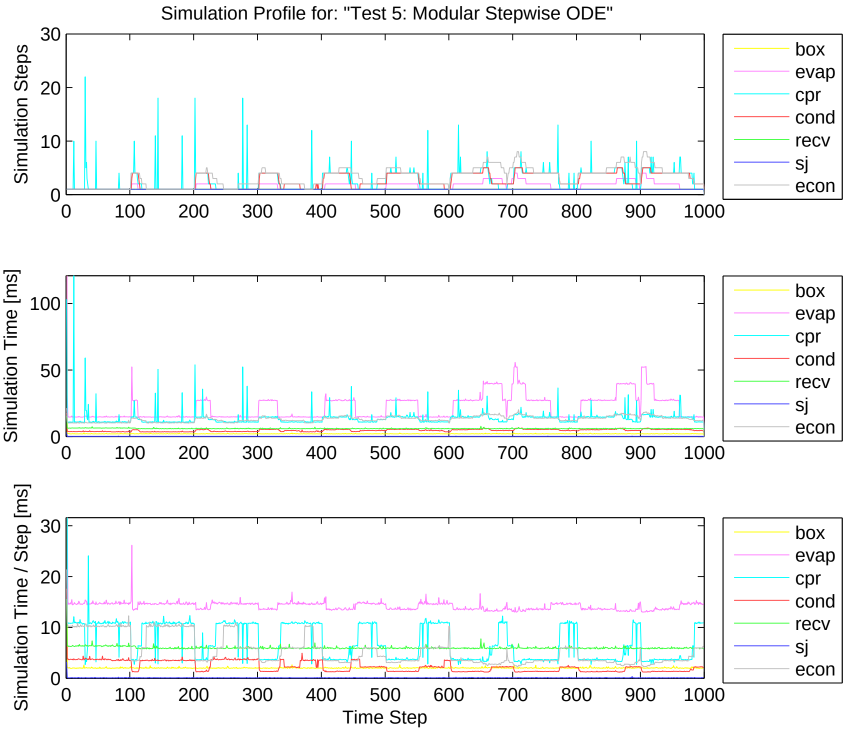


Fig. B.5: Simulation profile for the modular simulation with the `ode15s` solver.

Test *3 - Continuous Monolithic ODE* and *4 - Stepwise Monolithic ODE* illustrates the importance of using the `ode15s` solver as it is intended to be used, which is visible through the large difference in simulation time. Test 3 simulates the model in one go, that is the solver controls the simulation from start to end and can run without interruptions. In Test 4 the solver simulates one second at a time, and therefore it has to linearize the model before every step and it is only able to step 1s forward. This causes it to use more than ten times as many calls to the monolithic wrapper as is used by the solver in Test 3, resulting in a very inefficient solution of the problem.

The results for the old method [18] running on this test sequence is shown in *Test 5 - Modular ODE* and it can be seen that the error is a bit higher than the error for Test 2 which indicates that the `ode15s` does a good job. The simulation profile that show the number of steps and time consumed for each of the components in this test are shown on Figure B.5.

The simple solver show an unexpected result in *Test 6 - Modular VS-FE*, it has a shorter simulation time than the old method but it still manages to produce a smaller error than the modular reference simulation. This is because the solvers crude first order method causes it to settle faster than the analytical solution and that cancels out some of the error caused by the ZOH in the modular environment. This means that a good comparison of Tests 5 and 6 is is difficult when using the monolithic reference and therefore a comparison to the modular reference is also carried out. The simulation profile for this test are shown on Figure B.6 and it can be seen that the step size changes more often than for the `ode15s` solver but for both tests it is easy to identify the ramps that starts every 100 seconds and are getting larger and thus longer as the test progresses. This show that the step size selection strategy of the VS-FE solver is adequate for this system although it uses more steps for the model of the economizer than `ode15s`.

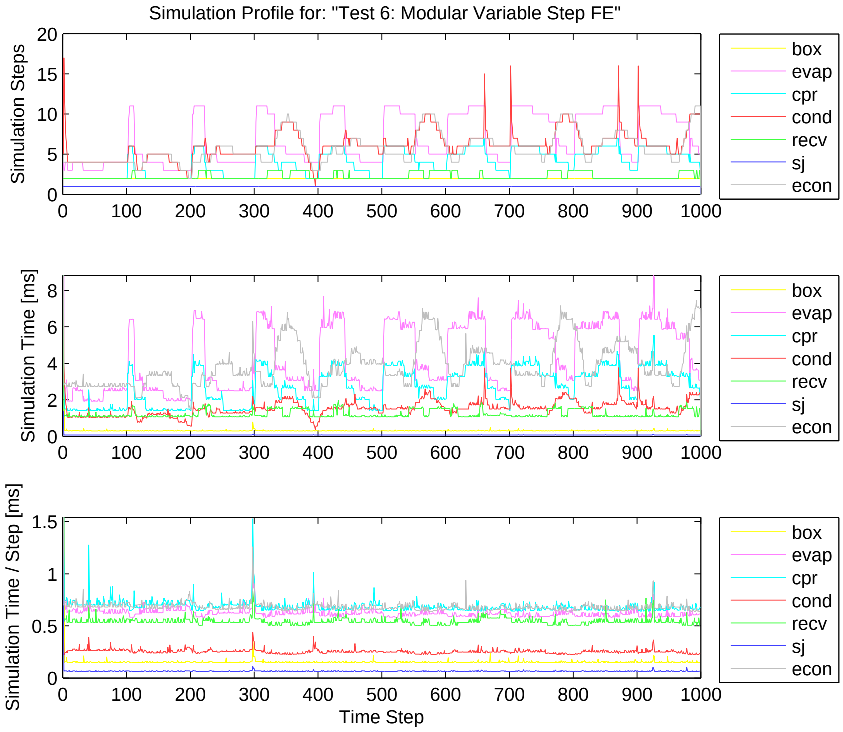


Fig. B.6: Simulation profile for the modular simulation with the VS-FE solver.

From *Test 7 - Monolithic VS-FE* it can be seen that the VS-FE solver handles the monolithic model quite well. It finishes faster than the continuous monolithic simulation in test 3 with an error that is roughly the same and it is just as fast as the modular solution which means that using the monolithic simulation method may be on option when there is a need for high accuracy.

Because of the large error from the modular simulators ZOH it is difficult to verify the performance of test 5 and 6 and they are therefore compared to the modular reference, with the results shown in Table B.3. from the results it is clear that the VS-FE solver are capable of simulating this system with the same accuracy as the `ode15s` solver 3.5 times faster, even though it needs twice the amount of steps to do it. The `ode15s` solver uses approximately 3.75ms per step due to the solvers startup overhead and because the VS-FE solver uses only 0.455ms per step and has almost no startup overhead its speed advantage increases if the error tolerance of the two solvers are increased. Therefore the VS-FE solver is a better choice for a solver in the modular configuration.

No	Description	Solver	Time	Mean Err.	Max Err.	Steps
5.	Modular ODE	`ode15s`	54.6s	0.021%	0.213%	15247
6.	Modular VS-FE	VS-FE	15.6s	0.032%	0.190%	34072

Table B.3: Simulation results compared to the modular reference.

Open loop tests

The test sequence in Tests 1 to 7 are designed to show dynamical errors that are relevant when doing controller experiments but they do not verify the long term stability of the simulation environment and the model. Long term stability and accuracy are two very important parameters for a simulation model and this is verified with an open loop test that simulates the system model without feedback control for three hours. The initial state for the model is set to match the initial state for a real container and the model is then simulated with the sampled control signals from the refrigeration container. Five measurements that are characteristic for the system are then compared to the result of the simulation in order to verify that the model is stable for longer runs and that the VS-FE solver does not compromise the accuracy of the solution. In Table B.4 the results of these two tests are shown. Tret and Tsup are the return and supply air temperatures for the container, respectively and they are important for the temperature control and estimation of actual cooling capacity. The suction temperature Tsuc is the temperature of the refrigerant vapor going from the evaporator to the compressor and combined with the suction pressure Psuc it forms the control signal for control of the evaporator. The last signal Pdis is the discharge pressure of the compressor. With feedback from these

No	Solver	Time	Tret	Tsup	Tsuc	Pdis	Psuc
8.	Modular ODE	622s	0.541%	0.957%	0.790%	2.183%	1.369%
9.	Modular VS-FE	97.6s	0.545%	0.954%	0.795%	2.183%	1.369%

Table B.4: Open loop simulation results.

five signals it is possible to control the refrigeration system. As it can be seen, the error of the two simulations are almost identical but the VS-FE solver is six times faster than the previous method that use the `ode15s` solver and therefore the obvious choice for a solver in an observer is the VS-FE method.

4 Conclusion

A modular simulation environment was presented with a dynamical model of a refrigeration container. Different options for simulating the model for controller experiments or as a full system observer to be used in the embedded controller for the refrigeration container was tested and analyzed. It was demonstrated that the variable step forward euler modular simulation approach may be used to simulate vapor compression cycles without the loss of accuracy but with a significant increase in speed. The impact of simulating a model of a refrigeration system with the multi-rate method has been analyzed with respect to dynamical and static errors and it was shown that the multi-rate method combined with the variable step forward euler solver was the best option for an embedded observer. A comparison between monolithic and multi-rate simulations on the same model was done and showed that the variable step forward euler solver was faster than the `ode15s` solver in both scenarios without increasing the error. From the tests it can be concluded that it is very important to use a solver with low overhead in a multi-rate environment because simulation times are short and therefore a an advanced solver never reaches its true potential. Design and simulation of reefer systems based on a modular simulation concept have shown very promising results. Simulation speed has been increased by up to 350% and it has been shown that the trade-off - inaccuracy - has no real impact on results. The simulation environment and the model have been verified to be more accurate than necessary for control experiments and therefore a decrease in simulator accuracy is acceptable when modular simulation is used.

References

[1] D. S. C. Limited, “Reefer shipping market annual report,” Drewry House, 213 Marsh Wall, London, E14 9FJ,England, 2009. [Online]. Available: http://www.drewry.co.uk/get_file.php?id=1161

[2] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer, 2003.

[3] “Container haandbook, cargo loss prevention information from german marine insurers.” [Online]. Available: http://www.containerhandbuch.de/chb_e/wild/index.html?/chb_e/wild/wild_08_01_02.html

[4] MCI, “Maersk container industry, starcool refrigerated container specifications,” 2013. [Online]. Available: http://www.maerskbox.com/

[5] L. S. Larsen, C. Thybo, and H. Rasmussen, “Intelligent control - optimizing the operation of refrigeration systems under daily variations in ambient temperature,” in *Danske Køledage, March 15th-16th, 2007. Proceedings of Danske Køledage 2007.*, 2007.

[6] J. Cai, J. Stoustrup, and J. Jorgensen, “Preventing refrigerated foodstuffs in supermarkets from being discarded on hot days by mpc,” in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, 2008. [Online]. Available: http://www.control.aau.dk/~jakob/selPubl/papers2008/ifacwc_2008_3.pdf. doi:10.3182/20080706-5-KR-1001.01880

[7] “Trnsys.” [Online]. Available: http://sel.me.wisc.edu/trnsys/

[8] “Dymola.” [Online]. Available: http://www.dynasim.se/index.htm

[9] P. Fritzson and P. Bunus, “Modelica – a general object-oriented language for continuous and discrete-event system modeling,” in *IN PROCEEDINGS OF THE 35TH ANNUAL SIMULATION SYMPOSIUM*, 2002, pp. 14–18.

[10] M. Skovrup, “Windali - an open-structured component modeling and simulation program based on standard programming languages,” in *Proceedings of SIMS Conference. Lyngby*, 2000, pp. 157–172.

[11] “Matlab® Simulink®.” [Online]. Available: http://www.mathworks.com/products/simulink/

[12] Z. Yang, K. Rasmussen, A. Kieu, and R. Izadi-Zamanabadi, “Fault detection and isolation for a supermarket refrigeration system - part two,” *Proceedings of the 18th IFAC World Congress, 2011*, pp. 4238–4243, 2011, iFAC Workshop Series 2011.

[13] R. Kübler and W. Schiehlen, “Modular simulation in multibody systems,” *Multibody System Dynamics*, vol. 4, pp. 107–127, 2000. doi:10.1023/A:1009810318420

[14] C. Rosen, D. Vrecko, K. Gernaey, and U. Jeppsson, “Implementing adm1 for benchmark simulations in matlab/simulink,” *Journal of Water science and technology*, vol. 54, no. 4, pp. 11–19, 2006. doi:10.2166/wst.2006.521

[15] Guo-liang Ding, “Recent developments in simulation techniques for vapour-compression refrigeration systems,” *International Journal of Refrigeration*, vol. 30, no. 7, pp. 1119–1133, 2007. doi:10.1016/j.ijrefrig.2007.02.001

[16] R. M. Howe, “Accuracy and stability tradeoffs in multirate simulation,” in *Proceedings of SPIE*, vol. 4367, 2001, pp. 113–126. doi:10.1117/12.440011

[17] J. G. Pearce, R. E. Crosbie, J. J. Zenor, R. Bednar, D. Word, and N. G. Hingorani, “Developments and applications of multi-rate simulation.” in *UKSim*, D. Al-Dabass, Ed. IEEE, 2009, pp. 129–133. [Online]. Available: http://dblp.uni-trier.de/db/conf/uksim/uksim2009.html#PearceCZBWH09. doi:10.1109/UKSIM.2009.23

[18] K. Sørensen and J. Stoustrup, “Modular modelling and simulation approach - applied to refrigeration systems,” in *Proceedings of IEEE International Conference on Control Applications*, San Antonio, Texas, USA, September 2008, pp. 983 – 988. doi:10.1109/CCA.2008.4629691

[19] K. Sørensen, M. Skovrup, L. Jessen, and J. Stoustrup, “Modular modelling of a reefer container,” *Submitted for publication*, 2013.

[20] L. F. Shampine and M. W. Reichelt, “The matlab ode suite,” *Journal on Scientific Computing*, vol. 18, pp. 1–22, 1997. doi:10.1137/S1064827594276424

[21] J. C. Butcher, *Numerical methods for ordinary differential equations*. J. Wiley, 2003. [Online]. Available: http://www.worldcat.org/isbn/9780471967583

[22] E. Kreyszig, *Advanced Engineering Mathematics*, 9th ed. Wiley, Nov. 2005. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471488852

[23] B. Rasmussen, A. Musser, and A. Alleyne, “Model-driven system identification of transcritical vapor compression systems,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 444–451, 2005. doi:10.1109/TCST.2004.839572

Paper C

Modular Modeling of a Refrigerated Container

Kresten K. Sørensen, Lars M. Jessen, Morten Juel Skovrup, Jakob Stoustrup

Submitted to: *International Journal of Refrigeration*, November 2013.

The layout has been revised.

Abstract

This paper presents the modeling of a refrigeration container for use as a reference for energy-optimizing controller design. The model is based on first principles in order to conserve mass and energy, but various assumptions are used to simplify the equations, resulting in a unified model for the Star Cool refrigeration unit, the container and the cargo. Comparisons between simulations using the model and measurements from a real container show an average error of less than $\pm 1K$ *on the states important for control.*

Nomenclature

Latin symbols

M	Mass (kg)
$\dot{m}$	Mass flow (kg/s)
Q	Energy flow (W)
V	Volume (m^3)
v	Specific Volume (m^3/kg)
h	Specific enthalpy (J/kg)
p	Pressure (Bar)
T	Temperature (C°)
X	Refrigerant Quality (kg/kg)
c	Specific heat (J/(Kg K))
UA	Overall heat transfer coefficient (W/K)
s	Second

Greek symbols

Δ	Property variation
γ	Heat capacity ratio
K	Expansion device characteristic constant
λ	Pressure drop ratio (Bar s/kg)
ω	Compressor Speed (Hz)
σ	Control volume boundary
ρ	Density (kg/m^3)

Subscripts

a	Air
m	Metal
amb	Ambient
c	Cargo
Cp	Constant pressure
Cv	Constant volume
f	Floor
C	Clearance
i	Control volume, i
l	Liquid
v	Vapor
r	Refrigerant
sc	Sub-cooled
aa	Ambient to air
af	Ambient to floor
ca	Cargo to air
ret	Return air from cargo
sup	Supply air to cargo
amv	Air to metal around vapor volume
aml	Air to metal around liquid volume
fa	Floor to air

1 Introduction

Modern control theory is a discipline in which controllers are designed from mathematical models of the dynamical systems that they are intended to control. The performance of the controller relies on good system models, and therefore, the modeling of the system should receive as much attention as designing the controller itself. Another important application of system models is simulations, in which a computer is used to simulate the behavior of a real system. Numerical simulations are extensively used for experiments within the field of control engineering, and with the increasing power of computers, faster than real time simulations of very large dynamical systems have become possible using ordinary desktop PCs. The model presented in this paper is constructed for the modular simulation environment presented in [1, 2].

A refrigeration container is an insulated container with an integrated refrigeration unit that is used for the international transport of different types of chilled or frozen foods. Good controller performance is essential for these systems if both high food quality and energy efficiency are to be achieved [3]; consequently, a good model is required. The subject of refrigeration plant modeling is well covered for industrial and HVAC systems, but the modeling of refrigeration containers is only covered in a few works [4, 5, 6], which primarily focus on product quality control of chilled foods. Chilled foods generally ripen in the container and that must be controlled if the cargo is to survive the trip, and therefore, the focus of these models is the cargo dynamics and not the refrigeration plant itself. It has been shown by [7, 3] that a significant amount of energy can be saved by adapting to daily variations in ambient temperature using a Model Predictive Controller (MPC). To enable simulation experiments using this technology, a detailed model of the refrigeration plant and container is needed.

The objective of the present was to develop a model that captures the dynamics required to test and design an energy-optimizing controller for a refrigeration container. The requirements for such a model are that the static and dynamic behaviors should sufficiently match those of the real system to be used for closed-loop control experiments and that the simulations should run at least one order of magnitude faster than experiments on the real system. Another important requirement is that it must be possible to change the model configuration without having to rewrite or reorganize the entire set of equations for the model by hand. Therefore, a modular approach is selected, in which the model of the refrigeration system is composed of a set of interchangeable component models that are based on a mix of first principles and assumptions. First principles are substituted by assumptions where it has a positive effect on simulation speed and a small effect on model accuracy.

We present a simulation model of a refrigeration container that focuses on the dynamics that are important from a control perspective. The presented model is nonlinear to properly reflect the static performance of the system under different operating conditions, and it is primarily used as a tool for testing and validating controllers; however, linear models derived from the nonlinear simulation model are also used for controller design. The container modeled in this paper is constructed by Maersk Container Industry (MCI) and is equipped with a Star Cool refrigeration unit. This unit possesses some unique properties that make it an obvious target for advanced control. The compressor is controlled by a Variable Frequency Drive (VFD) that allows the compressor speed to be adjusted from 20Hz to 110Hz and electronic expansion valves that can be controlled from 0% to 100%. The fact that the control inputs can be controlled independently means that a certain cooling power can be achieved with a range of different control set-point combinations. However, not all combinations of set-points are equally efficient, and therefore, it is important that the model is able to reflect the overall efficiency of the system such that an energy-optimizing controller can be designed from the model.

2 Modeling

The container consists of a large insulated box with a refrigeration unit at one end and a loading door at the other. The refrigeration unit cools and circulates cold air around the cargo, as depicted in Figure C.1. The cold air is injected into the floor of the container by the evaporator fans and travels below the cargo to the far end of the container. The air heats up as it rises between the cargo and along the walls to the top of the container, where the air flows back to the refrigeration unit. A schematic of the refrigeration system is presented in Figure C.2.

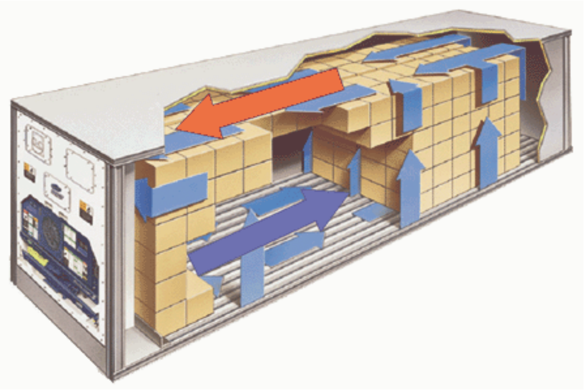


Fig. C.1: Airflow in the Refrigeration Container

The refrigeration system consists of a two-stage piston compressor, a condenser, an evaporator, two expansion valves, three fans and an economizer. The economizer is a counter-flow plate heat exchanger used to increase the capacity of the system when there are high temperature differences between the cold and hot sides. The expansion valves are electromagnetically pulsed on/off valves, and the fans may run at half or full speed or be turned off completely. There are three major pressure levels in this setup: p_1 is the evaporator pressure, p_2 is the intermediate pressure between the two compressor stages, and p_3 is the condenser pressure.

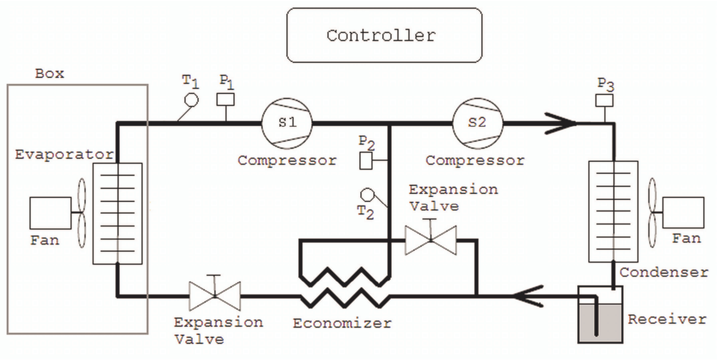


Fig. C.2: Refrigeration System for a Refrigeration Container

The objective of the model of the refrigeration container is to capture the dynamics that are important to the temperature inside the box to create a model-based controller for the container. Additionally, for future studies, it is desired to be able to track the movement of the refrigerant charge in the system to enable experiments aiming to reduce the overall refrigerant charge needed.

According to [8], the salient dynamics of a refrigeration system are the thermal time constants of the metal surfaces in the heat exchangers. Furthermore, according to [8], some components have dynamics that are so fast compared to the dominant dynamics that they may be replaced with algebraic equations, thereby reducing the model order while preserving the physical behavior of the model. The components that may be modeled algebraically are the expansion valves and the compressor [9], while the remainder of the components are modeled using a mix of differential and algebraic equations. The resulting model is a dynamical model of the thermal masses in the heat exchangers, the cargo and the metal in the cargo hold with a model of the refrigeration cycle that includes the mass balances for the refrigerant in each control volume but with steady-state equations for the momentum and energy balances [10]. The disadvantage of this approach is that some dynamics are missing, but the advantage is that the resulting system is less stiff, which results in a higher simulation speed [11].

The model is composed of a set of component models, where each model corresponds to a physical component in the refrigeration system. The state in each control volume is given by pressure and enthalpy. Pressure is used to balance the exchange of refrigerant flow between component models, and therefore, two connected components must have access to the pressure of the other component. Refrigerant conversion calculations are performed using the *WinDali* [12] refrigeration equation toolbox for MATLAB®.

A typical set of state equations for a control volume containing refrigerant would then be

$$\frac{dM}{dt} = \dot{m}_{in} - \dot{m}_{out} \tag{C.1}$$

$$h_{out} = h_{in} + \frac{Q_{in}}{\dot{m}_{in}} \tag{C.2}$$

Equation (C.1) is the mass balance, where the change in mass $\frac{dM}{dt}$ is the net mass flow into the volume given by the difference between $\dot{m}_{in}$ and $\dot{m}_{out}$. Equation (C.2) is the energy balance.

In most cases, the interface between the component models is the pipes carrying refrigerant, and such an interface has three states: pressure, mass flow and specific enthalpy. Zero-order-holds are inserted between the component models by the modular simulation environment [1, 2], and therefore, changes in the states in one component are not effective in the other components until the next step in the simulation.

This means that the dynamics of the states on the interfaces should be slow compared to the sampling time of the simulation environment to avoid simulation instability and imprecise results.

The refrigeration system has some very fast, pulsating, pressure dynamics around the compressor, which are not interesting from a control perspective. The primary objective of the control algorithm is to keep the supply air temperature steady; therefore, fast dynamics that have little influence on the supply air temperature can be neglected. To avoid modeling fast pressure transients, it was decided to place the pressure equations in the components with the slowest dynamics. Therefore, at the interface between the evaporator and the compressor, the pressure is modeled by the evaporator and the mass flow is modeled by the compressor.

Two pipe junction models are required to model the joining and splitting of refrigerant flows that occur between the compressor stages and after the receiver, respectively.

2.1 Pipe Joining Junction

The pipe joining junction model is located between the two compressor stages and the economizer and joins the mass flows from the first compressor stage and the economizer into one mass flow that is fed to the second compressor stage. This model includes the internal volume of the compressor and has two states, refrigerant mass M and output enthalpy h_{out}, and five inputs, which are the mass flows on all three interfaces $\dot{m}_{in1}$, $\dot{m}_{in2}$, and $\dot{m}_{out}$ and the input enthalpy for both refrigerant inputs h_{in1} and h_{in2}. The governing equations of the model are

$$\frac{dM}{dt} = \dot{m}_{in1} + \dot{m}_{in2} - \dot{m}_{out} \tag{C.3}$$

$$h_{out} = \frac{h_{in1} \cdot \dot{m}_{in1} + h_{in2} \cdot \dot{m}_{in2}}{\dot{m}_{in1} + \dot{m}_{in2}} \tag{C.4}$$

The change in internal mass is modeled by Equation (C.3) as the net sum of the mass flow on the three interfaces. Equation (C.4) is the energy balance, where it is assumed that no heat transfer to the surroundings occurs.

2.2 Pipe Splitting Junction

The splitting junction is located between the receiver and the economizer, and the model is very simple because it is only a division of one mass flow into two. The pressure and enthalpy on the outputs are equal to the input pressure and enthalpy, and the mass flows on the inputs are equal to the sum of the mass flows on the two outputs. Because there are no slow internal dynamics, this model is implemented as an algebraic function. The equations for this component are as follows:

$$\dot{m}_{in} = \dot{m}_{out1} + \dot{m}_{out2} \tag{C.5}$$

$$p_{out1} = p_{in} \tag{C.6}$$

$$p_{out2} = p_{in} \tag{C.7}$$

$$h_{out1} = h_{in} \tag{C.8}$$

$$h_{out2} = h_{in} \tag{C.9}$$

2.3 Compressor

The compressor has two almost identical stages, in which the only difference is that the displacement volume of the first stage is twice as large as that of the second stage; therefore, only the equations for one stage are shown here. The compressor stages are modeled by two algebraic functions that give the mass flow $\dot{m}$ and output enthalpy h_{out} as a function of the input and output pressures p_{in} and p_{out}, the input enthalpy h_{in} and the speed of the compressor ω. The physical behavior in the model includes adiabatic compression, clearance volume, and valve pressure loss. The dynamics caused by the piston strokes are decoupled from the slower evaporator dynamics because a single piston stroke has a very small impact on the evaporator temperature. Therefore, the dynamics from the pulsating pressure caused by the piston strokes can be neglected and substituted with an algebraic model in which the mass flows on the input and output are equal. The model calculates the mass displaced in a single stroke from the density of the incoming refrigerant vapor and the clearance volume, and then the mass is scaled with the compressor speed. The pressures p_1 and p_2 are the actual input and output pressures at the piston and are calculated from the valve loss as

$$p_1 = p_{in} - kl_1 \cdot \omega \tag{C.10}$$

$$p_2 = p_{out} + kl_2 \cdot \omega \tag{C.11}$$

where ω is the compressor speed and kl is the valve loss constant. The subscript 1 is used for variables calculated on the basis of p_{in}, and the subscript 2 is used for variables calculated on the basis of p_{out}. Assuming an adiabatic compression, the mass flow and output enthalpy can be calculated as

$$\gamma = C_{cp}/C_{cv} \tag{C.12}$$

$$v_2 = \left(\frac{p_2}{p_1}\right)^{\frac{-1}{\gamma}} \cdot v_1 \tag{C.13}$$

$$\dot{m} = \left(\frac{V_1}{v_1} - \frac{V_C}{v_2}\right) \cdot \frac{\omega}{2} \tag{C.14}$$

$$T_{out} = T_{in} \cdot \left(\frac{p_{out}}{p_{in}}\right)^{\frac{\gamma-1}{\gamma}} \tag{C.15}$$

$$h_{out} = HTP(T_{out}, \quad p_{out}) \tag{C.16}$$

Equation (C.12) yields the ratio γ between the specific heat of the refrigerants at constant pressure C_{cp} and constant volume C_{cv}. In Equation (C.13), the specific volume of the refrigerant at the discharge pressure p_2 is calculated, assuming adiabatic compression, from the specific volume v_1 and pressure of the input refrigerant p_1. In Equation (C.14), V_1 and v_1 are the internal volume of the cylinder and the specific volume of the refrigerant before the stroke, respectively; V_C is the clearance volume and thus the internal cylinder volume after the stroke; and v_2 is the specific volume of the refrigerant after the stroke. The mass displaced in a single stroke is the difference between the mass of the gas in the cylinder with the piston in either extremity, and this is multiplied by the compressor speed to obtain the mass flow. The output temperature T_{out} is given by Equation (C.15) and is used to find the output enthalpy in Equation (C.16) by table lookup [13].

2.4 Condenser

The condenser models the discharge pressure of the refrigeration system, and the accuracy of this state is important because the energy consumption of the compressor and the mass flow through the expansion valves are dependent on the discharge pressure. In [8], it is shown that the dominant dynamics in the condenser are determined by the temperature of the metal, and therefore, a simple model with a single refrigerant control volume is used.

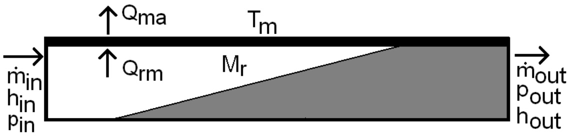


Fig. C.3: Schematic of the Condenser model

The state equations are given by Equations (C.17) to (C.21)

$$h_{out} = h_{in} - \frac{Q_{rm}}{\dot{m}_{in}} \tag{C.17}$$

$$\frac{dM_r}{dt} = \dot{m}_{in} - \dot{m}_{out} \tag{C.18}$$

$$\frac{dT_m}{dt} = \frac{Q_{rm} - Q_{ma}}{M_m \cdot Cp_m} \tag{C.19}$$

The energy balance is given by Equation (C.17), and the mass balance is given by Equation (C.18). The internal energy in the metal was modeled by its temperature T_m, as shown in Equation (C.19), which is the energy balance for the metal divided by its heat capacity Cp_m to give the temperature derivative.

The pressure drop is assumed to be a linear function of the mass flow, and Equation (C.20) models the discharge pressure as the pressure on the output of the condenser, or the receiver pressure, plus an offset that is the product of the mass flow and the pressure drop constant λ determined experimentally. The high-side pressure is thus not calculated by the condenser but rather in the next component model, the receiver. The reason for this is that the receiver has a larger internal volume, and the condenser is therefore modeled as a "refrigerant cooler" that follows the pressure of the receiver. This does not mean that the condenser has no influence on the high-side pressure because the receiver pressure depends on the output enthalpy from the condenser. In general, the pressure is given by the state of the refrigerant in the control volume, but for this model, the pressure is given by the receiver, and therefore, the state of the refrigerant must follow the pressure. This is achieved in Equation (C.21), where the mass flow out of the condenser ensures that the volume of the refrigerant converges towards the internal volume of the condenser.

$$p_{in} = p_{out} - \lambda \cdot \dot{m}_{in} \tag{C.20}$$

$$\dot{m}_{out} = \dot{m}_{in} + \frac{M_r - \frac{V_i}{v}}{1s} \tag{C.21}$$

The two energy flows are defined by

$$Q_{rm} = UA_{rm} \cdot (T_r - T_m) \tag{C.22}$$

$$Q_{ma} = UA_{ma} \cdot (T_m - T_a) \cdot U_{fan} \tag{C.23}$$

where Q_{rm} is the flow from refrigerant to metal and Q_{ma} is the flow from metal to air. T_m is the temperature of the metal, and T_a is the temperature of the air that is blown over the condenser coil by the fan. T_r is the refrigerant temperature at the output of the condenser calculated from p_{out} and h_{out}, under the assumption that the heat transfer is independent of the refrigerant mass flow. It is assumed that the energy flow from the metal to the air is linearly dependent on the airflow, and therefore, the energy flow is multiplied by the speed of the condenser fan, which is denoted U_{fan}. The fan speed can have the values of 0.05, 1 and 2, which correspond to stopped, low speed and high speed, respectively. When the fan is stopped, some natural convection still exists, and this is modeled by assuming a fan speed of 0.05 when the fan is off. The value was found to be the best match for the data used to verify the model, but the amount of convection when the fan is off is expected to vary depending on the wind conditions.

The heat transfer coefficients UA_{rm} and UA_{ma} determine the steady-state condensing pressure, and they were found in a steady-state experiment on a special test unit equipped with extra temperature and pressure sensors. The mass flow of refrigerant was calculated from the compressor speed and suction pressure, using manufacturer data for the compressor. This was used to calculate the heat flux from the refrigerant to the metal, which in steady state is equal to the heat flux from the metal to the air. Using the air, metal and refrigerant temperatures, the heat transfer coefficients were then calculated to be 650 J/K for UA_{ma} and 1500 J/K for UA_{rm}. The same method was applied when the UA values for the heat transfers between metal, air and refrigerant were found for the evaporator.

2.5 Receiver

The receiver is a buffer tank for excess refrigerant; the refrigerant is led from the condenser into the top of the receiver, and liquid refrigerant to the expansion valves is taken from the bottom. The receiver is generally either neglected in dynamical models of refrigeration plants or not existent in the modeled plant, but the receiver is, however, not entirely without influence on the refrigeration systems dynamics, particularly during rapid pressure changes [14]. The reason for this is that the liquid in the receiver acts as a buffer and has a dampening effect on pressure transients from the condenser and refrigerant temperature transients on the input of the expansion valves. The liquid in the receiver that flows to the expansion valves may be either sub-cooled or at the boiling point, and if the liquid starts to boil, it will turn into a two-phase mixture of liquid and vapor with a quality that depends on the size of the pressure drop. When the condenser fan is turned on, the pressure in the receiver can drop rapidly, and if the temperature of the liquid in the receiver is close to the boiling point, it will begin to boil until the refrigerant temperature drops below the boiling point or the pressure increases above the bubble point. The liquid boiling into vapor counteracts the pressure drop, and the hybrid [15] behavior caused by this is included in the model.

The receiver has two control volumes, one with vapor and one with liquid, and each of these volumes has four possible sets of equations that are determined by the quality of the refrigerant entering the control volumes and the quality of the refrigerant in the control volume itself. The input refrigerant is divided between the liquid and vapor control volumes when the refrigerant entering the receiver has a quality above zero and to the liquid control volume only when the quality of the input refrigerant is less than or equal to zero. Figure C.4 presents a schematic of the mass and heat flows modeled in the receiver.

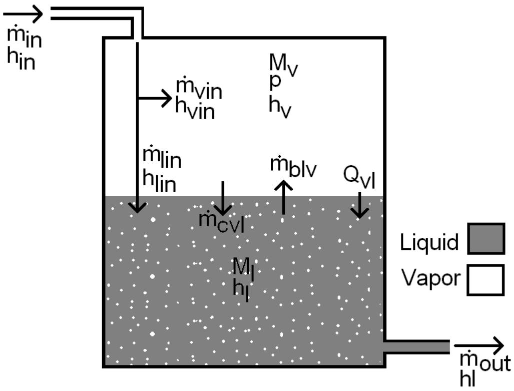


Fig. C.4: Schematic of the Receiver

The inputs of the model are the input mass flow $\dot{m}_{in}$, the input enthalpy of the refrigerant coming from the condenser h_{in} and the mass flow of the refrigerant leaving the receiver $\dot{m}_{out}$. The states of the model are the pressure p, the enthalpy of the liquid control volume h_l, the enthalpy of the vapor control volume h_v, the mass of refrigerant in the liquid volume m_l and the mass of refrigerant in the vapor volume m_v.

The hybrid behaviour related to the input refrigerant is given by Equations (C.24) to (C.27).

$$\dot{m}_{lin} = \dot{m}_{in} \cdot (1 - X_{in})\big|_{X_{in}=[0,1]} \tag{C.24}$$

$$\dot{m}_{vin} = \dot{m}_{in} \cdot X_{in}\big|_{X_{in}=[0,1]} \tag{C.25}$$

Equations (C.24) and (C.25) show the mass flows into the two control volumes of the receiver, which are the input mass flow split by the void fraction of the refrigerant.

The enthalpy of the refrigerant that flows from the condenser to the two receiver control volumes are given by Equations (C.26) and (C.27):

$$h_{vin} = HDewP(p_{in}) \tag{C.26}$$

$$h_{lin} = \begin{cases} HBubP(p_{in}) & \text{if } X_{in} > 0; \\ h_{in} & \text{otherwise.} \end{cases} \tag{C.27}$$

The vapor entering the vapor volume, h_{vin}, is modeled as being at the dew point, under the assumption that the condenser always manages to condense at least a fraction of the refrigerant. The mass flow into the liquid control volume, m_{lin}, may be either at the bubble point or sub-cooled depending on the performance of the condenser.

The mass flow from the liquid volume to the vapor volume, m_{blv}, is positive when the liquid is boiling and otherwise zero, as given by:

$$\dot{m}_{blv} = \begin{cases} X_l \cdot M_l & \text{if } h_l > HBubP(p); \\ 0 & \text{otherwise.} \end{cases} \tag{C.28}$$

It is assumed that the mass flow between the two control volumes is instant and therefore equal to the product of the fraction of vapor or liquid and the mass of refrigerant in the control volume. The following equation models the mass flow from the vapor volume to the liquid volume, given by condensation in the vapor volume.

$$\dot{m}_{cvl} = \begin{cases} (1 - X_v) \cdot M_v & \text{if } h_v < HDewP(p); \\ 0 & \text{otherwise.} \end{cases} \tag{C.29}$$

The two control volumes are thermally connected through the surface of the liquid volume and through the metal housing of the receiver. The resulting heat flow is given by Equation (C.30):

$$Q_{vl} = UA \cdot (T_v - T_l) \tag{C.30}$$

The equations for mass and energy balances are given by Equations (C.31) to (C.34):

$$h_l = h_{lin} + \frac{Q_{vl}}{\dot{m}_{lin}} \tag{C.31}$$

$$h_v = h_{vin} - \frac{Q_{vl}}{\dot{m}_{vin}} \tag{C.32}$$

$$\frac{dM_l}{dt} = \dot{m}_{lin} + \dot{m}_{vl} - \dot{m}_{out} \tag{C.33}$$

$$\frac{dM_v}{dt} = \dot{m}_{vin} - \dot{m}_{vl} \tag{C.34}$$

where $\dot{m}_{vl}$ is the flow of refrigerant from the vapor volume to the liquid volume given by the difference between $\dot{m}_{cvl}$ and $\dot{m}_{blv}$

$$\dot{m}_{vl} = \dot{m}_{cvl} - \dot{m}_{blv} \tag{C.35}$$

2.6 Expansion Valve

It is important for the mass flow given by the expansion valve model to be accurate because the mass flow is one of the determining factors for the cooling capacity of the system. According to [16], the expansion process can be considered adiabatic due to the very small internal volume of the expansion valve. Therefore, the expansion valve model is purely algebraic and modeled as a continuous valve that gives the average mass flow of the electromagnetically pulsed on/off valves used in the container. The continuous model was chosen because the pressure transients on the suction side of the evaporator, caused by the on/off action of the valves, are small when the evaporator is adequately filled with refrigerant. According to [17], the mass flow through an expansion valve may be calculated as shown by Equation (C.36)

$$\dot{m} = C \cdot A \cdot \sqrt{\rho \cdot \Delta P} \tag{C.36}$$

where C is the discharge coefficient, A is the cross-sectional area, ρ is the density of refrigerant entering the valve, and ΔP is the pressure difference over the device. The parameters C and A were unavailable for the expansion valves on the refrigeration container, and therefore, they were combined into the expansion device characteristic constant K, as shown in [16]. The resulting calculation of the mass flow is shown in Equation (C.37), where the density is calculated from the specific volume of the refrigerant on the input of the valve. The mass flow is multiplied by T_{on}, which is the fraction of time where the valve is on. The values of K were determined for both valves in the system through measurements of the refrigerant flow and are $1.18 \cdot 10^{-5}$ for the evaporator expansion valve and $3.37 \cdot 10^{-6}$ for the economizer expansion valve.

$$\dot{m} = T_{on} \cdot K \cdot \sqrt{\frac{1}{v_{in}} \cdot (p_{in} - p_{out})} \tag{C.37}$$

$$h_{out} = h_{in} \tag{C.38}$$

In the implementation of the model, the expansion valves are included in the models of the evaporator and the economizer rather than being standalone models.

2.7 Economizer

The economizer cools the refrigerant going from the receiver to the evaporator to a temperature that is close to the evaporation temperature at the intermediate pressure.

This adds some additional cooling potential to the refrigerant going into the evaporator in the sense that more cooling can be achieved at the same mass flow into the evaporator and at the same pressure difference from the evaporator to the condenser. Because the extra cooling potential is achieved by evaporating over a smaller pressure difference, less work is required by the compressor to reach a certain cooling capacity, and thus, the efficiency and capacity of the overall system are increased. The most important aspect to be reflected by the economizer model is the heat transfer from the hot side to the cold side because the heat transfer has a large impact on the enthalpy of the refrigerant entering the evaporator. The pressure dynamics in the cold side of the economizer are less important because they have a slower and more indirect effect on the remainder of the system. Figure C.5 presents a schematic of the economizer.

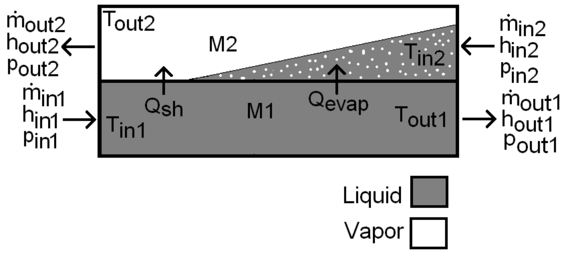


Fig. C.5: Schematic of the Economizer

The hot side of the economizer is filled with liquid refrigerant that runs from the receiver to the evaporator expansion valve and is modeled as a single region with heat transfer directly to the cold side. The equations related to the hot side of the economizer are denoted with the subscript "1". The cold side is also modeled as a single control volume, and the equations related to this control volume are denoted with the subscript "2". The dynamics caused by the metal of the economizer is not included in the model because its heat capacity is small compared to the energy flow, which results in a small impact on the overall dynamics of the economizer model.

The energy flow from the hot side to the cold side is calculated as a logarithmic mean temperature difference

$$Q = UA \cdot \frac{\Delta T1 - \Delta T2}{ln(\frac{\Delta T1}{\Delta T2})} \tag{C.39}$$

The overall heat transfer coefficient UA was obtained from the rated conditions given in the data sheet of the economizer [18] and is 333 W/K. The temperature differences $\Delta T1$ and $\Delta T2$ used in the log mean temperature difference are given by:

$$\Delta T1 = T_{in1} - T_{out2} \tag{C.40}$$

$$\Delta T2 = T_{out1} - T_{in2} \tag{C.41}$$

The hot side of the economizer is a cooler for liquid refrigerant, and the control volume is modeled by Equations (C.42) to (C.45).

$$h_{out1} = h_{in1} - \frac{Q}{\dot{m}_{in1}} \tag{C.42}$$

$$p_{out1} = p_{in1} + \Delta p \tag{C.43}$$

$$\frac{dM_1}{dt} = \dot{m}_{in1} - \dot{m}_{out1} \tag{C.44}$$

$$\dot{m}_{in1} = \dot{m}_{out1} + \frac{\frac{V_1}{v_1} - M_1}{1s} \tag{C.45}$$

Equation (C.42) models the energy balance, and Equation (C.44) models the mass balance. The output pressure is given by Equation (C.43), where the output pressure is modelled as the input pressure plus the fixed pressure drop Δp. Equation (C.45) models the mass flow into the sub-cooling control volume using the same principle as that used for the condenser in subsection 2.4. The specific volume of the refrigerant v_1 is calculated from h_{out1} and p_{out1}.

The state equations for the cold side of the economizer are given by Equations (C.46) to (C.49)

$$h_{out2} = h_{in2} + \frac{Q}{\dot{m}_{in2}} \tag{C.46}$$

$$\frac{dM_2}{dt} = \dot{m}_{in2} - \dot{m}_{out2} \tag{C.47}$$

$$p_{in2} = p_{out2} - \Delta p \tag{C.48}$$

$$\dot{m}_{out2} = \dot{m}_{in2} + \frac{M_2 - \frac{V_2}{v_2}}{1s} \tag{C.49}$$

using the same principles as the hot side for all states. The specific volume of the refrigerant in the control volume, v_2, is calculated from p_{out2} under the assumption that the refrigerant has a fixed quality X_2 such that $v_2 = V(p_{out2}, X_2)$. Equation (C.49) models the mass flow out of the control volume using the same principle as that used for the condenser in subsection 2.4.

2.8 Evaporator

The superheat is the most difficult state to control, and to enable the design of a good controller, the evaporator model must capture the superheat dynamics well. The evaporator is modeled with two refrigerant volumes, the first volume with a mixture of liquid and vapor and the second volume with vapor only, divided by a moving boundary σ, which is a principle that is successfully used in many works [19], [20], [21]. The heat transfer coefficient from metal to liquid is different from the heat transfer coefficient from metal to vapor, and therefore, the metal of the evaporator is also lumped into two control volumes divided by the moving boundary σ, where T_{ml} is the temperature of the metal around the mixed refrigerant volume and T_{mv} is the temperature of the metal around the vapor volume.

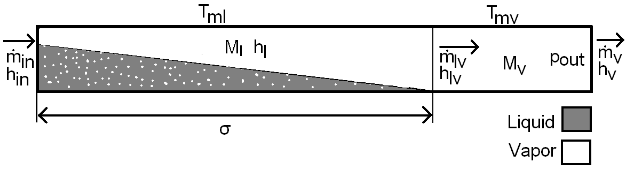


Fig. C.6: Schematic of the Evaporator

The calculation of the boundary location σ is based on the assumption that the volume containing mixture has a constant average quality of X_e, as in Equation (C.50)

$$\sigma = \frac{M_l \cdot v_1}{V_i} \tag{C.50}$$

where v_1 is the specific volume of the refrigerant in the first control volume, as calculated from the pressure p_{out} and the quality X_e that has a value of 0.1. This value was found to provide the best match to the pressure dynamics of the evaporator in the verification data. V_i is the total internal volume of the evaporator. The energy flows are given by

$$Q_{fan} = (155 \cdot U_{fan}{}^2 + 40 \cdot U_{fan}{}^3) \cdot 0.2 \tag{C.51}$$

$$T_{retfan} = T_{ret} + \frac{Q_{fan}}{\dot{m}_{air} \cdot Cp_{air}} \tag{C.52}$$

$$Q_{amv} = Cp_{air} \cdot \dot{m}_{air} \cdot (T_{retfan} - T_{mv}) \tag{C.53}$$

$$T_{retsh} = T_{retfan} - \frac{Q_{amv}}{\dot{m}_{air} \cdot Cp_{air}} \tag{C.54}$$

$$Q_{aml} = Cp_{air} \cdot \dot{m}_{air} \cdot (T_{retsh} - T_{ml}) \tag{C.55}$$

$$Q_{mvml} = UA_3 \cdot (T_{mv} - T_{ml}) \tag{C.56}$$

$$Q_{ml} = UA_1 \cdot (T_{ml} - T_l) \cdot \sigma \tag{C.57}$$

$$Q_{mv} = UA_2 \cdot (T_{mv} - T_v) \cdot (1 - \sigma) \tag{C.58}$$

Equation (C.57) is the energy flow from the evaporator metal to the liquid refrigerant control volume, where T_{ml} is the temperature of the metal surrounding the volume and T_l is the saturated evaporation temperature of the liquid refrigerant. The energy flow from the metal surrounding the vapor volume is given by Equation (C.58), where T_{mv} is the temperature of the metal and T_v is the temperature of the refrigerant leaving the evaporator calculated from the output pressure and enthalpy. The heat transfer coefficients UA_1 and UA_2 were found to be 3510 J/K and 1930 J/K, respectively, through the steady-state experiment described in Section 2.4. The value of UA_3 was set to 50 J/K. Because the salient dynamics of the refrigeration system are closely related to the metal dynamics [8], it is assumed that the air that passes through the evaporator is cooled to the metal temperature, and therefore, the energy flows from air to metal given in Equations (C.53) and (C.55) yield the energy that is required to reach the metal temperature. The evaporator is a counter-flow heat exchanger, and therefore, the air passes over the super heating section of the evaporator before the evaporation section. Therefore, to calculate the energy flow for each of the sections, the air temperatures at the input and output of the sections must be available. When the air enters the refrigeration unit, it passes over the fans before it enters the evaporator, and therefore, the input air temperature on the evaporator is slightly higher than T_{ret} due to the energy added by the fan given in Equation (C.51). The fan motor has a heat loss of 20%, and its speed is given by U_{fan}, which is an integer between 0 and 2, corresponding to off, low speed and high speed, respectively. Equation (C.52) calculates the air temperature after the fan T_{retfan} as the return temperature from the container, T_{ret}, plus the heat added by the fan. In Equation (C.53), the energy required to decrease the airflow temperature from T_{retfan} to T_{mv}, the temperature of the metal around the vapor volume, is calculated. Q_{amv} is then used to calculate the temperature drop of the air over the super heating part of the evaporator in Equation (C.54), and the resulting air temperature, T_{retsh}, is then used to calculate Q_{aml}, which is the energy required to decrease the airflow temperature from T_{retsh} to the metal temperature around the liquid volume, T_{ml}. The metals around the two control volumes are divided by the artificial moving boundary σ but are still physically connected, and therefore, there is also an energy flow between the two volumes, which is modeled by Equation (C.56).

The energy flow from refrigerant to the metal is modeled by simple heat transfers in Equations (C.57) and (C.58), where the energy flow is the product of a heat transfer constant and the temperature difference between metal and refrigerant in the control volume. The sizes of UA_1, UA_2 and UA_3 were determined by simulation experiments on the evaporator model over a range of working points that spans the normal area of operation for the refrigeration system.

The airflow over the evaporator is driven by a fan that has rotational inertia, and the air has some inertia itself; therefore, the airflow does not instantly change when the fan speed is changed. This behavior is modeled by Equations (C.59) and (C.60):

$$\bar{\dot{m}}_{air} = \frac{{U_{fan}}^2 \cdot 3400.5 + {U_{fan}}^3 \cdot -1103.5}{3600 \cdot \rho_{air}} \tag{C.59}$$

$$\frac{\Delta \dot{m}_{air}}{\Delta t} = \frac{\bar{\dot{m}}_{air} - \dot{m}_{air}}{10s} \tag{C.60}$$

In Equation (C.59), the instant mass flow of the air, $\bar{\dot{m}}_{air}$, is calculated by a polynomial fit that yields the correct mass flow at the discrete speeds of the fan. The instant mass flow is then used in Equation (C.60) as a reference for the actual airflow, $\dot{m}_{air}$, in a first-order difference with a time constant of 10 seconds.

The remaining state equations are given by Equations (C.61) to (C.68)

$$\frac{dT_{ml}}{dt} = \frac{Q_{aml} - Q_{ml} + Q_{mvml}}{M_m \cdot Cp_m \cdot \sigma} \tag{C.61}$$

$$\frac{dT_{mv}}{dt} = \frac{Q_{amv} - Q_{mv} - Q_{mvml}}{M_m \cdot Cp_m \cdot (1-\sigma)} \tag{C.62}$$

$$p_{out} = PHV\left(h_v, \frac{V_i - V_l}{M_v}\right) \tag{C.63}$$

$$h_l = h_{in} + \frac{Q_{ml}}{\dot{m}_{in}} \tag{C.64}$$

$$h_v = h_{lv} + \frac{Q_{mv}}{\dot{m}_{lv}} \tag{C.65}$$

$$\frac{dM_l}{dt} = \dot{m}_{in} - \dot{m}_{lv} \tag{C.66}$$

$$\frac{dM_v}{dt} = \dot{m}_{lv} - \dot{m}_{out} \tag{C.67}$$

$$T_{sup} = T_{retfan} + \frac{Q_{aml} + Q_{amv}}{Cp_{air} \cdot \dot{m}_{air}} \tag{C.68}$$

Equations (C.61) and (C.62) model the change in metal temperature as the net flow of energy into the control volume divided by the specific heat Cp_m and mass M_m of the metal. The amount of metal in each of the control volumes depends on σ, and therefore, the mass is divided by the boundary.

The pressure in the evaporator, p_{out}, is calculated by table lookup and given by Equation (C.63). It is assumed that the pressure is the same in both control volumes and calculated from the state of the refrigerant in the vapor volume using the specific enthalpy and the density of the vapor. The density is calculated from the mass of the vapor and the volume of the vapor, which is the total volume of the evaporator minus the volume of the liquid refrigerant. M_v and v_v are the mass and specific volume for the vapor volume, respectively, and V_l is the volume occupied by the liquid in the evaporator. The energy balances for the liquid and vapor control volumes are given by Equations (C.64) and (C.65), respectively. The enthalpy of the refrigerant moving from the liquid to the vapor volume, h_{lv}, is the dew point enthalpy at the pressure in the evaporator, p_{out}. The mass flow between the two volumes is given by

$$\dot{m}_{lv} = \frac{Q_{ml}}{h_{dew} - h_{in}} \tag{C.69}$$

which is the amount of refrigerant that can be evaporated by Q_{ml}. The mass balances for the two control volumes are given in Equations (C.66) and (C.67) as the net sum of the mass flows in and out of the respective volumes. The final state equation of the evaporator is for the supply air to the cargo hold of the container, T_{sup}, which is modeled as the instantaneous temperature drop from the temperature after the fan T_{retfan} calculated from the cooling energy flows Q_{aml} and Q_{amv}, the air mass flow $\dot{m}_{air}$ and the specific heat of air Cp_{air}. The air flow over the evaporator is high compared to the mass of the air around the evaporator, and therefore, the dynamics of the air around the evaporator is neglected.

2.9 Box

The box is the main thermodynamic capacitance in the container due to the large masses of the aluminum T-floor and the cargo. The cargo and the T-floor have a large surface area; therefore, the temperature of the air inside the container has a strong coupling to the temperature of these two control volumes. The model of the box consists of the three main thermal capacities: the temperatures of the cargo, the T-floor and the air; their state equations are shown in Equations (C.70) to (C.72).

$$\frac{dT_{air}}{dt} = \frac{Q_{ca} + Q_{aa} + Q_{fa} + Q_{fan} - Q_{cool}}{M_{air} \cdot Cp_{air}} \tag{C.70}$$

$$\frac{dT_{floor}}{dt} = \frac{Q_{af} - Q_{fa}}{M_{floor} \cdot Cp_{floor}} \tag{C.71}$$

$$\frac{dT_{cargo}}{dt} = \frac{-Q_{ca}}{M_{cargo} \cdot Cp_{cargo}} \tag{C.72}$$

The change in air temperature is modeled by Equation (C.70), where Q_{ca} is the energy flow from the cargo to the air, Q_{aa} is the energy flow through the walls from the outside to air, Q_{fa} is the energy flow from the floor to the air, Q_{fan} is the energy from inertia added to the air by the fan that is converted to heat as the air slows down, and Q_{cool} is the cooling provided by the evaporator. Q_{cool} is calculated from the mass flow of the air, the specific heat of air and the temperature difference between the return and supply temperature:

$$Q_{cool} = Cp_{air} \cdot \dot{m}_{air} \cdot (T_{ret} - T_{sup}) \tag{C.73}$$

$$Q_{aa} = (T_{amb} - T_{air}) \cdot UA_{amb} \cdot 0.81 \tag{C.74}$$

$$Q_{af} = (T_{amb} - T_{floor}) \cdot UA_{amb} \cdot 0.19 \tag{C.75}$$

$$Q_{ca} = (T_{cargo} - T_{air}) \cdot UA_{cargo} \tag{C.76}$$

$$Q_{fa} = (T_{floor} - T_{air}) \cdot UA_{floor} \tag{C.77}$$

$$Q_{fan} = (155 \cdot {U_{fan}}^2 + 40 \cdot {U_{fan}}^3) \cdot 0.8 \tag{C.78}$$

The energy flows between control volumes are all modeled as standard heat transfers with a temperature difference multiplied by a heat transfer coefficient UA_*. The heat transfer coefficients UA_{cargo} and UA_{floor} were determined through experiments on a container with cargo in which the cooling capacity was increased, and the heat transfer coefficients were calculated from the dynamical response of the floor and cargo temperatures to the change in air temperature. The determined values were 500 J/K for UA_{cargo} and 1100 J/K for UA_{floor}. The heat transfer coefficient for the walls and floor of the container is UA_{amb} and specified by the manufacturer to be 43 J/K. The area of the floor corresponds to 0.19 times the total surface area of the container, and therefore, Equations (C.75) and (C.74) are multiplied by 0.19 and 0.81, respectively.

3 Results

The performance of the model is evaluated by running the model in open-loop with control inputs sampled from a real container running with closed-loop control. Measurements from the container are then compared with the corresponding states from the model, and the error distribution for each of the measured variables is presented. The test sequence inputs were obtained from a container loaded with pork that was stepped through different cooling capacities at a set point of -20°C. The sampled variables that are used as model inputs are all the control inputs plus the ambient temperature. The initial conditions of the model was set to reflect the initial conditions of the sampled data from the real container. The simulation period is three hours, during which the container is running in both start/stop and continuous modes at different cooling powers. The fact that the model is able to run in open loop for such a long period without diverging outside the stable operational envelope of the model is due to the slow dynamics and a certain degree of self-stabilization that exists in the refrigeration system. The error distributions for the measured variables are shown in the histogram in Figure C.7, the results from the three-hour test run can be seen in Figure C.8, and a closeup of the first 30 minutes can be seen in Figure C.9.

In the first subplot of Figures C.8 and C.9, the control signals applied to the model are shown. *Fcpr* is the compressor speed, *Vexp* is the evaporator expansion valve opening, *Veco* is the economizer expansion valve opening, *Fcond* is the condenser fan speed, and *Fevap* is the evaporator fan speed.

The measurements that are available for verification of the model cannot be used to directly verify that certain component models, such as the compressor and expansion valves, have the correct mass flow. There is no significant drift on the suction pressure, which indicates that the mass flow of the evaporator expansion valve and the first stage of the compressor are the same, but it does not provide any indication that the mass flow is correct. To obtain indication that the mass flow is correct, the cooling capacities of the real unit and the model were calculated from the supply and return temperatures and the airflow. If the two cooling capacities match and the energy balance of the model holds, then the mass flow of the compressor and the expansion valve models matches the real system. The cooling capacity of the model is accurate when the unit is running

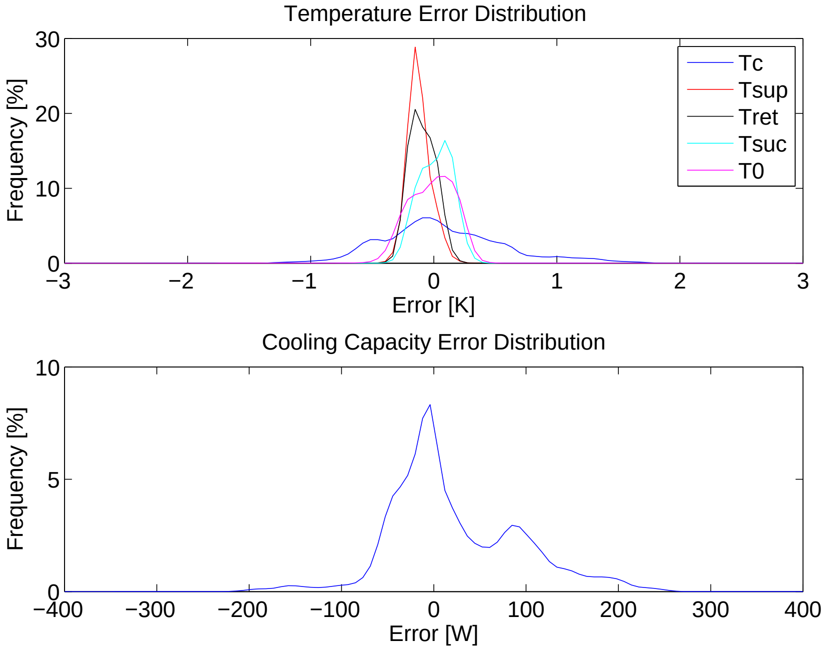


Fig. C.7: Model error distribution

continuously, but it is too large in the initial start/stop phase of the test, which could be due to a compressor model that removes too much refrigerant from the evaporator at low speed or due to too much sub-cooling on the refrigerant from the economizer model. When the compressor is running continuously, the cooling capacity shows a very good match to the model data throughout the simulation, indicating that the component models are able to accurately reflect the static behavior of the real system.

The saturated discharge temperature of the condenser matches the overall trends, but the dynamics are inaccurate due to the simple single control volume model that was used. From the start/stop phases of the test, it can be observed that the pressure increases rapidly just after the start of the compressor and the slowly drops again. This behavior is caused by the inertia of the liquid in the condenser, which is not modeled, and therefore, this dynamic behavior is missing from the model. The dynamic response from changes in Tamb to changes in Tc is too weak, and this could be due to the lumped model. Due to the proximity of the condenser air intake and discharge, some amount of recirculation that is highly sensitive to the air flow around the container occurs. The container was located in a yard with other running containers when the data used for verification of the model were recorded, and this could have also caused disturbances in

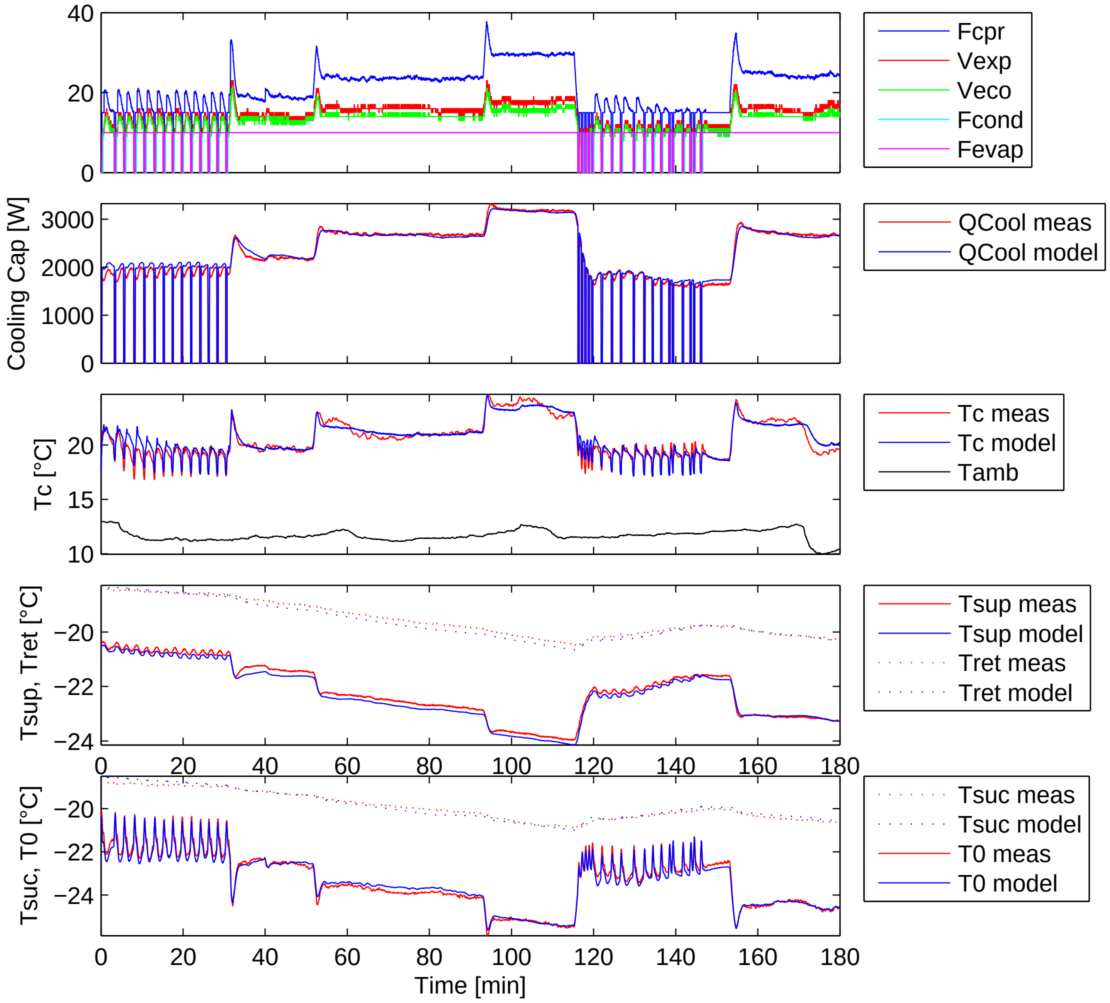


Fig. C.8: Model verification results

the airflow around the refrigeration unit end of the container that impacts the recirculation of the condenser air.

The return temperature has very slow dynamics due to the large thermal inertia of the cargo, but the model is within $\pm 0.25K$ of the measured return temperature. Because the cargo heat transfer coefficient and thermal inertia change depending on what is shipped in the container, it is only important to have an accurate model of the

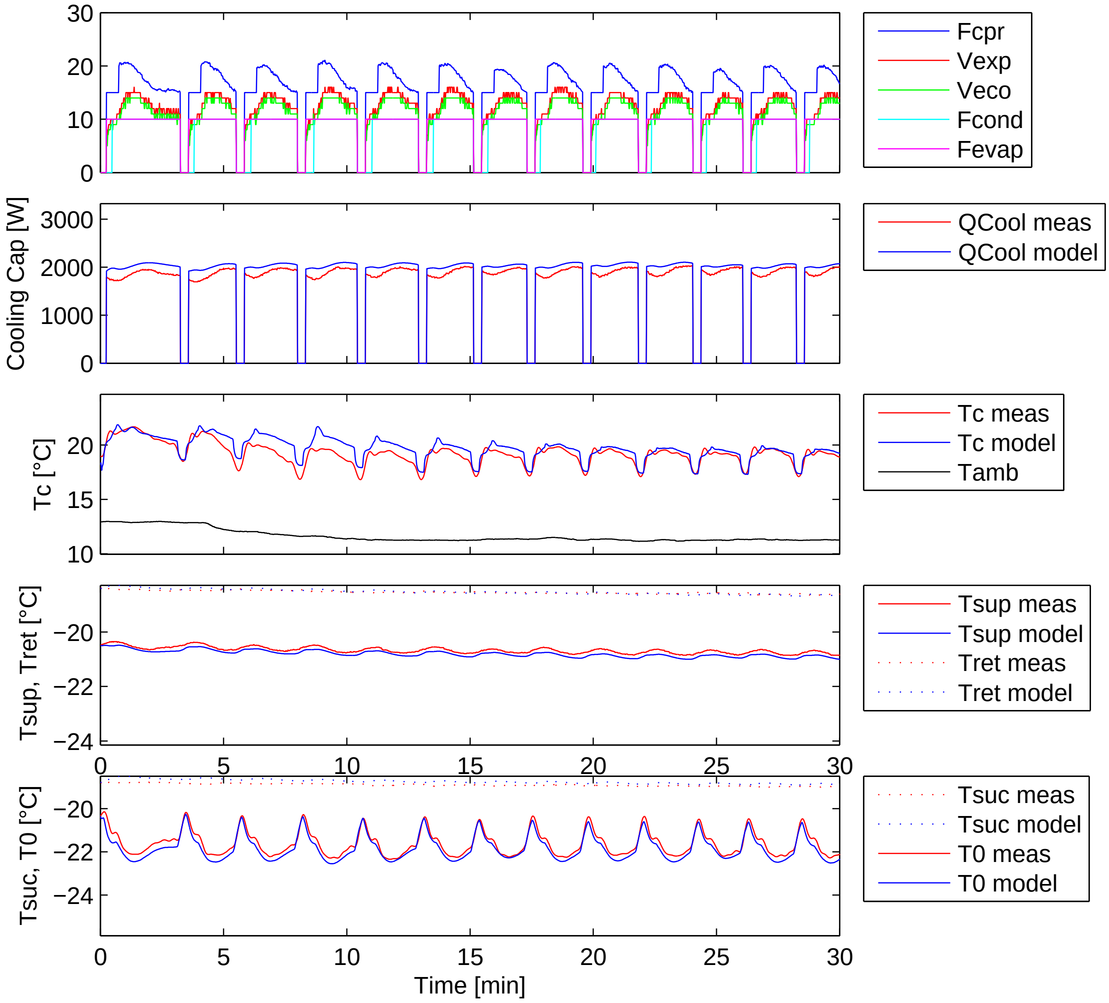


Fig. C.9: Close-up of the initial start/stop phase

cargo for verification of the model because the return temperature has a large impact on the evaporator states. The supply temperature is too low during most of the simulation, but this is due to Tret being too low as well. The accuracy of the cooling capacity shows that the difference between Tret and Tsup and the dynamics of Tsup are accurate, which is an indicator of model accuracy.

The suction temperature is within ±0.4 K, and the suction pressure is within ±0.8

K, which shows that the model of the evaporator is a good match to the real system both statically and dynamically and that there is some accumulation of errors over time. This is however in open loop where small inaccuracies can accumulate over time, and when the loop is closed in controller experiments, this buildup disappears because the controller compensates for model inaccuracy. In the beginning of the test, there is a mismatch because the initial conditions of the model were not perfectly established, but the model quickly stabilized. A close-up of the beginning of the test is shown in Figure C.9, where the match is good because errors have not yet accumulated. The performance of the model on the temperatures on the evaporator is very accurate in the first 110 minutes, but later in the test, sequence errors begin to accumulate, resulting in an inaccurate suction pressure two hours into the test sequence.

From the temperature error distribution shown in Figure C.7, it can be observed that all temperatures except the saturated condenser pressure are within ±0.8 K, demonstrating that the model matches the real system well on these parameters. The model is intended to be used to test controllers that are being developed for the container, as a simulation tool and as the basis for generating linear models that can be used to develop controllers. These controllers must be robust to sensor inaccuracies that can be an order of magnitude larger than the model inaccuracy, and therefore, it is concluded that the model accuracy is adequate for the measurements on the evaporator and the cargo hold. To design a good controller for the condenser, a better model with more accurate dynamics and a better response to changes in ambient temperature is required, but because the salient dynamics are present in the current model, it would also be possible to use it for controller design.

4 Conclusion and Future Work

4.1 Conclusion

In this study, a dynamical simulation model of a refrigeration container and its refrigeration unit has been developed, with the objective of being sufficiently accurate for use in controller design and simulation experiments. The results demonstrated that adequate performance can be reached with a model in which small terms and fast dynamics are neglected and assumptions are used rather than first principles, which corresponds to a reduction in the model order. The results verify that the dynamics of a container refrigeration system can be modeled by the thermal masses in the heat exchangers, the metal in the cargo hold and the cargo, with a model of the refrigeration cycle that includes normal mass balances but with steady-state momentum and energy balances. The model exhibits good matching of both static and dynamical behaviors, and therefore, the model can be used for controller development where little or no tuning is necessary when moving from the model to the real system.

4.2 Future Work

Because the model is to be used in the development of controllers for use on a real system, the model also has to reflect the important aspects of this system. One aspect that has been ignored in this work is ice buildup on the evaporator. Ice degrades the airflow over the evaporator and the heat transfer coefficient from the evaporator metal to the air and has an impact on efficiency that cannot be neglected. The dynamical response of the discharge pressure to the ambient temperature must be improved to be able to develop a controller that has proper disturbance rejection of sudden changes in the ambient temperature. Another useful improvement would be better robustness of the equations on which the model is based, particularly with respect to the model's ability to handle both liquid and vapor refrigerants in all control volumes.

5 Acknowledgment

The authors gratefully acknowledge the assistance from Kim Madsen from MCI for counseling on modeling of the refrigeration container components and MCI for providing a container with cargo for testing.

References

[1] K. Sørensen and J. Stoustrup, "Modular modelling and simulation approach - applied to refrigeration systems," in *Proceedings of IEEE International Conference on*

Control Applications, San Antonio, Texas, USA, September 2008, pp. 983 – 988. doi:10.1109/CCA.2008.4629691

[2] K. Sørensen, D. Jens, and J. Stoustrup, "Modular simulation of reefer container dynamics," *Published online before print in Simulation*, December 2013. doi:10.1177/0037549713515542

[3] J. Cai, J. Stoustrup, and J. Jorgensen, "Preventing refrigerated foodstuffs in supermarkets from being discarded on hot days by mpc," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, 2008. [Online]. Available: http://www.control.aau.dk/~jakob/selPubl/papers2008/ifacwc_2008_3.pdf. doi:10.3182/20080706-5-KR-1001.01880

[4] R. G. M. van der Sman and G. J. C. Verdijck, "Model predictions and control of conditions in a ca-reefer container," in *ISHS Acta Horticulturae 600: VIII International Controlled Atmosphere Research Conference*, Agrotechnological Research Institute ATO, P.O. box 17, 6700 AA Wageningen, the Netherlands, 2003, pp. 163–171. [Online]. Available: http://www.actahort.org/members/showpdf?booknrarnr=600_20

[5] G. J. C. Verdijck, H. A. Preisig, and G. van Straten, "Direct product quality control for energy efficient climate controlled transport of agro-material," *Journal of Process Control*, vol. 15, no. 2, pp. 235 – 246, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152404000514. doi:10.1016/j.jprocont.2004.04.004

[6] L. Filina and S. Filin., "An analysis of influence of lack of the electricity supply to reefer containers serviced at sea ports on storing conditions of cargoes contained in them," in *Polish Maritime Research*, vol. 15, 2009, pp. 96–102. doi:10.2478/v10012-008-0002-z

[7] L. S. Larsen, C. Thybo, and H. Rasmussen, "Intelligent control - optimizing the operation of refrigeration systems under daily variations in ambient temperature," in *Danske Køledage, March 15th-16th, 2007. Proceedings of Danske Køledage 2007.*, 2007.

[8] B. Rasmussen, A. Musser, and A. Alleyne, "Model-driven system identification of transcritical vapor compression systems," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 444–451, 2005. doi:10.1109/TCST.2004.839572

[9] Guo-liang Ding, "Recent developments in simulation techniques for vapour-compression refrigeration systems," *International Journal of Refrigeration*, vol. 30, no. 7, pp. 1119–1133, 2007. doi:10.1016/j.ijrefrig.2007.02.001

[10] R. Zhou, T. Zhang, J. Catano, J. T. Wen, G. J. Michna, Y. Peles, and M. K. Jensen, “The steady-state modeling and optimization of a refrigeration system for high heat flux removal,” *Applied Thermal Engineering*, vol. 30, no. 16, pp. 2347 – 2356, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S135943111000219X. doi:10.1016/j.applthermaleng.2010.05.023

[11] L. F. Shampine and S. Thompson, “Stiff systems,” vol. 2, no. 3, p. 2855, 2007.

[12] M. Skovrup, “Windali - an open-structured component modeling and simulation program based on standard programming languages,” in *Proceedings of SIMS Conference. Lyngby*, 2000, pp. 157–172.

[13] M. J. Skovrup, “Refrigerant equations library,” Department of Mechanical Engineering, Section of Energy Engineering, Technical University of Denmark, Nils Koppels Allé, Building 402 DK-2800 Lyngby, DENMARK. [Online]. Available: http://www.et.web.mek.dtu.dk/WinDali/Files/RefEqns3.10.ZIP

[14] K. Prölss, G. Schmitz, D. Limperich, and M. Braun, “Influence of refrigerant charge variation on the performance of an automotive refrigeration system,” in *International Refrigeration and Air Conditioning Conference, Paper 843*, 2006. [Online]. Available: http://docs.lib.purdue.edu/iracc/843

[15] T. Henzinger, “The theory of hybrid automata,” in *Logic in Computer Science, 1996. LICS ’96. Proceedings., Eleventh Annual IEEE Symposium on*, 1996, pp. 278–292. doi:10.1109/LICS.1996.561342

[16] R. Koury, L. Machado, and K. Ismail, “Numerical simulation of a variable speed refrigeration system,” *International Journal of Refrigeration*, vol. 24, no. 2, pp. 192 – 200, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700700000141. doi:10.1016/S0140-7007(00)00014-1

[17] H. Li, J. E. Braun, and B. Shen, “Modeling adjustable throat-area expansion valves.” International Refrigeration and Air Conditioning Conference, 2004, p. Paper 705. [Online]. Available: http://docs.lib.purdue.edu/iracc/705/

[18] Danfoss, “Brazed plate heat exchanger datasheet,” 2014. [Online]. Available: http://www.ra.danfoss.com/TechnicalInfo/Literature/Manuals/24/DKQBPD000B102.pdf

[19] M. Willatzen, N. Pettit, and L. Ploug-Sørensen, “A general dynamic simulation model for evaporators and condensers in refrigeration. part i: moving-boundary formulation of two-phase flows with heat exchange,” *International Journal of Refrigeration*, vol. 21, no. 5, pp. 398 – 403, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000911. doi:10.1016/S0140-7007(97)00091-1

[20] N. Pettit, M. Willatzen, and L. Ploug-Sørensen, "A general dynamic simulation model for evaporators and condensers in refrigeration. part ii: simulation and control of an evaporator," *International Journal of Refrigeration*, vol. 21, no. 5, pp. 404 – 414, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700797000923. doi:10.1016/S0140-7007(97)00092-3

[21] H. Rasmussen, C. Thybo, and L. Larsen, *Nonlinear Superheat and Evaporation Temperature Control of a Refrigeration Plant.* Pergamon Press, 2006, pp. 252–254. doi:10.3182/20061002-4-BG-4905.00043

Paper D

Adaptive MPC for a Refrigerated Container

Kresten K. Sørensen and Jakob Stoustrup

Submitted to: *Control Engineering Practice*, November 2013.

The layout has been revised.

Abstract

In this work, the potential energy savings from adapting to daily ambient temperature differences for frozen cargo in reefer containers are studied using a model of the Star Cool reefer. The objective is to create a controller that can be implemented on an embedded system, and a range of methods are used to reduce the computational load. A combination of MPC and traditional control is used, and the accuracy of the MPC is enhanced with an on-line update of the model parameters. The simulation experiments show that potential energy savings of up to 21% are achieved when the MPC is allowed to control both the cooling capacity and the ventilation of the cargo area. The largest cost reduction is achieved through a reduced ventilation rate.

Nomenclature

Latin symbols	
M	Mass (kg)
Q	Energy flow (W)
V	Fan speed
T	Temperature (C°)
Cp	Specific heat (J/(Kg K))
C	Heat capacity (J/K))
α	Heat transfer coefficient (W/K)
P	Power (W)
Subscripts	
air	Air in the cargo hold
amb	Ambient
floor	Floor of the cargo hold
cool	Cooling capacity
fan	Evaporator fan
cargo	Cargo inside the cargo hold
sup	Supply air to cargo hold
ret	Return air from cargo hold
box	Walls of the cargo hold
s	Slack
c	Compressor
min	Minimum
max	Maximum

1 Introduction

A large amount of perishable cargo is currently transported by sea in reefer containers. By mid-2008, the fleet consisted of 4500 reefer vessels with a combined reefer capacity of 11.4 million TEU[1], and the capacity is predicted to increase by 69% by 2013; see [1]. These reefer containers are powered by electricity with a consumption of up to 6kW per TEU, depending on the ambient temperature and the temperature inside the container itself. With an average consumption of 3.6 kW per TEU, this yields a combined consumption of 41 GW, which is, on average, 8.9 MW per ship; see [2] and [3]. Previously, this energy consumption was considered to be insignificant with respect to the large amount of energy used to propel the ship, but due to increasing oil prices, greater competition in the shipping market and the environmental impact of shipping, reducing the energy consumption of reefer containers has become of interest.

The control solutions currently employed are based on classical control theory, in which the individual components are controlled by separate controllers, with a limited amount of controller interconnections and gain scheduling. The objective of these controllers is to maintain the temperature inside the container close to a setpoint at ambient temperatures between -20C° and +37C° (hot side). The setpoint is in the range of -29C° to +25C° (cold side). Due to the large range of operation on the hot and cold sides and the non-linearities in the refrigeration system, the controllers used must be conservative to provide stability over the entire area of operation.

Fruits and vegetables are generally quite sensitive to atmosphere and temperature variations, which means that the cargo temperature and air composition in the container must be kept within certain limits. This reduces the potential for control optimizations with respect to energy consumption by using the thermal inertia of the cargo as a buffer. It has been shown that the cost of operating a refrigeration system may be decreased by using thermal inertias in the system as a buffer to offset cooling to periods where the cost is low. In [4], the cost of a running household heat pump is lowered by using an MPC that is fed the forecasted cost of electricity. Exploiting ambient conditions to lower the energy consumption of building HVAC systems while respecting occupant comfort constraints is demonstrated in [5, 6, 7, 8] and is shown to significantly reduce energy consumption while ensuring good occupant comfort. A learning-based approach for the pre-cooling of foodstuffs to avoid saturation of the refrigeration system on hot days is demonstrated by [9] using an MPC that is updated on-line with the learning-based constraints and predicted future load. In [10], an MPC is used to control the product quality of chilled cargo in refrigerated containers with the primary focus on modelling and controlling the cargo quality, resulting in a reduction of mass loss in the cargo due to evaporation and lower energy consumption due to a reduced ventilation rate.

An existing energy saving control strategy that is in use is QUEST (see [11]), which is a control scheme for fruits and vegetables that allows for larger supply air temperature

[1] Twenty Foot Equivalent. The equivalent of a twenty foot reefer container.

variances and lower fan speeds based on a predefined set of rules. This can be allowed because research has shown that there is no degradation in produce quality if the supply air temperature is varied around the setpoint due to the thermal insulation of the produce packaging and the slow metabolic rates of the produce; see [12]. The rules for ventilation rate and temperature variation have been determined by testing a wide range of different types of fruits and vegetables and ensuring that the cargo quality remains unaffected by the variations. There are various rules for different product classes, and they have been designed dependent of the temperature setpoint to make it easy to operate. This is important because it is infeasible to educate loading crews all over the world in complex set up procedures. For reefer containers without a VSD (variable speed drive) on the compressor, the energy savings from using QUEST can be as high as 53% because the refrigeration system is very inefficient at part load, and for containers with a VSD on the compressor, the energy savings are smaller but still significant.

For frozen goods, the rules that must be followed to preserve cargo quality are more lenient than those for chilled goods, and therefore, the cargo thermal inertia can be used to offset cooling from the periods where the system is less efficient to periods where the system is more efficient. The ambient temperature has a direct influence on the condensation pressure and thus also on the system efficiency, which leads to the possibility of moving some cooling from the day where the ambient temperature is high to the night where the ambient temperature is lower. Another approach for reducing energy consumption is to reduce the amount of ventilation inside the cargo hold because the power consumed by the fans has a dual impact on the cost of running the system. The fans consume energy, which is added to the cost of running the system, but the kinetic energy that the air receives from the fans is eventually converted to heat inside the container, which means that it must be removed by the refrigeration system. Therefore, investigating the potential of optimizing how the fans are used is of interest.

Model predictive control (MPC) was introduced in the petro-chemical industry to control difficult processes with long delays and unknown states, but today, MPC is used in a wide range of applications, such as power plant control and the automotive industry. MPC is used for optimizing control of processes with respect to known future demands or known future changes in external conditions while remaining within a given set of constraints. The performance of MPC is dependent on the quality of the model on which it is based because it is used to predict the behaviour of the system over the prediction horizon. For systems where the model dynamics may change, either a non-linear or an adaptive linear approach must be used to retain the performance and to avoid violating the imposed constraints.

A refrigeration system has several degrees of freedom, meaning that the same cooling capacity can be obtained with different actuator setpoints. [13] showed that selecting the correct setpoints can have a high impact on the efficiency of the refrigeration system, and therefore, any controller aiming to save energy should focus on selecting the correct setpoints.

In this study, the potential for energy savings by adapting to daily ambient temperature differences is studied for frozen cargo in refrigerated containers. The observation from [10] that adequate cooling may be achieved at a lower ventilation rate is used to formulate control laws for an MPC that ensures optimal utilization of the fans when they are running. Cargo parameters such as thermal inertia and heat transfer coefficient are estimated and used as the basis for the MPC model, resulting in flexibility towards changes in these parameters that does not exist in QUEST. Furthermore, the MPC is set up to exploit daily variations in ambient temperature by cooling more when the ambient temperature is low and when the efficiency of the refrigeration system is higher. This extra cooling is "stored" in the cargo thermal inertia, allowing for a smaller cooling effort during the day when the ambient temperature is high and the refrigeration system efficiency is low. The future ambient temperature is predicted from measurements from the last 24 hours by an oscillator and a simple phase-locked loop and fed to the MPC.

In this paper, we present an adaptive MPC controller that utilizes the same principles as QUEST but with the added benefit of adapting to cargo parameters and daily cycles in ambient temperature for increased energy efficiency. The potential energy savings at different ambient temperatures and fan control methods are investigated and presented. The computational load is reduced due to varying step sizes in the prediction horizon and linearizing local controllers, which enables the use of a reduced linear model for the MPC.

In the following, the methods used in this paper are described, starting with a short introduction to the refrigeration system in Section 2.1. Then, a description of the parameter and state estimators used by the controller follows in Section 2.2. Finally, the controller itself is described in Section 2.3.

2 Methods

2.1 Refrigeration System Simulation Model

The refrigeration container used in this paper is constructed by *Maersk Container Industry* and is equipped with a *Star Cool* refrigeration unit; see [14]. A refrigeration container is an insulated box with a door for loading cargo at one end and a refrigeration system at the other end, as shown in Figure D.1.

The cargo is kept cold by air that is circulated from the evaporator towards the door end of the container through a T-profile floor, which allows air to enter small gaps between the produce. The air is heated by the produce or the walls and rises to the ceiling of the box, where the hot air flows back to the evaporator. Natural convection is not sufficient to ensure an even distribution of air in the box, and therefore, the air flow is driven by fans located above the evaporator. The energy from these fans ends up as heat in the box and has to be removed by the refrigeration system. Therefore, it is desirable to run the fans as little as possible. It is however necessary to start the fans

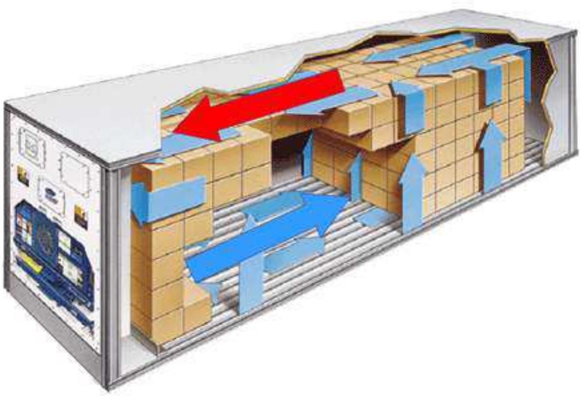


Fig. D.1: Airflow in the Refrigeration Container

at regular intervals to be able to measure the air temperature in the box itself because no air temperature sensors are placed here and to prevent local hot-pockets of air from forming and spoiling the produce.

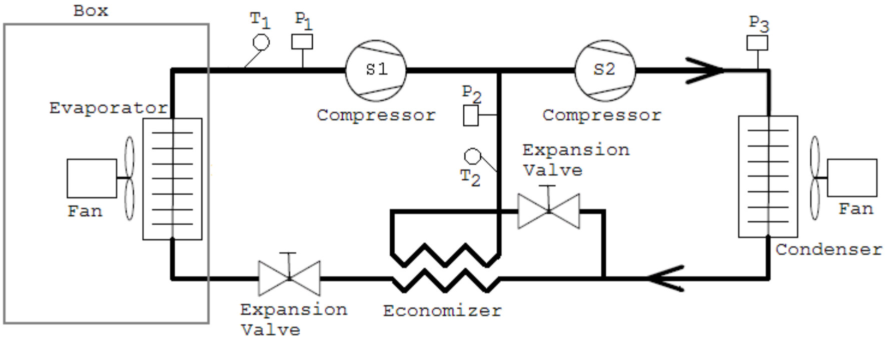


Fig. D.2: Refrigeration System Schematic

A schematic of the refrigeration system used in this study is shown in Figure D.2. This refrigeration system is a two-stage cycle that uses an economizer to increase the efficiency of the system under high temperature differences between the cold and hot sides.

The compressor has high- and low-pressure stages, which are shown as two single-stage compressors in the figure. The compressor is equipped with a VSD, and the fans may be either stopped, at half speed or at full speed. The expansion valves are pulse modulated, that is, they can be either closed or fully open. These expansion valves are controlled by a PWM signal with a period of six seconds.

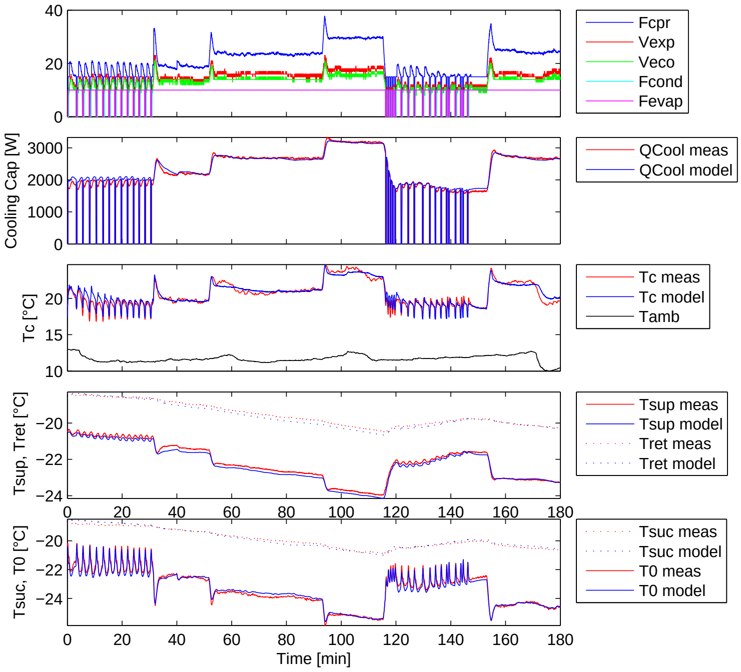


Fig. D.3: Simulation model verification results

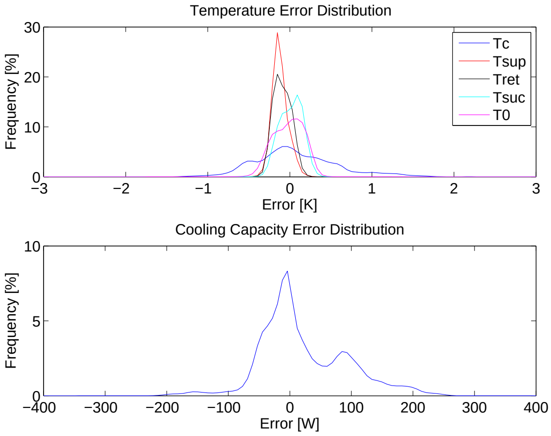


Fig. D.4: Simulation model error distribution

The reefer container has been modelled in detail [15], and a simulation model that accurately reflects the refrigeration system and cargo dynamics at a one second resolution is available. The comparison of the different controllers must be performed under comparable ambient conditions to minimize the uncertainties in the results, and therefore, the simulation model is used. This model has 81 states and models the flow, energy and mass of the refrigerant in the components shown in the system schematic in Figure D.2. The simulation model has been verified against a refrigerated container packed with 20,000 kg of frozen pork meat, and the results of the verification are shown in Figure D.3. The verification consists of a simulation model running in open loop for three hours using control inputs that were recorded from a real container during a series of capacity steps. The output of the model in terms of the variables significant for control is then compared to the recorded measurements of the same variables from the real system. Qcool is the cooling capacity of the system, Tc is the saturated discharge temperature at the outlet of the compressor, Tsup is the supply air temperature, Tret is the return air temperature, Tsuc is the refrigerant temperature after the evaporator, and T0 is the saturated suction temperature at the inlet of the compressor. Figure D.4 shows the error distribution of the test results presented in Figure D.3, from which it can be observed that the model is a good match for the real system. The model verification is performed at the same temperature setpoint that the controller in this study is tested at.

2.2 Parameter and State Estimation

Model predictive control requires an accurate model of the system that is to be controlled and a prediction of the trajectory of external conditions relevant to the objective of the controller. In this subsection, the methods that were used to predict the future ambient temperature are described. Furthermore, an estimator for the temperature, heat transfer coefficient and heat capacity of the cargo is described.

Ambient Temperature Prediction

The ambient temperature must be predicted 24 hours into the future as a reference to the MPC to be able to exploit its daily cycles, and an observer is constructed for this purpose. Because the temperature is an oscillation with a period of 24 hours, an oscillator is used and synchronized by a phase-locked loop (PLL). For simplicity, the prediction is only based on measurements from the past 24 hours, even if more data are available. In Figure D.5, the predictions for the first 100 hours of a container's journey from a Danish port are shown, with five hours between each of the predictions. In the implementation of the MPC, a new estimate for the ambient temperature is calculated for each iteration of the MPC.

It can be observed that for the first 24 hours, where the measurement set is incomplete, the prediction is unreliable, but after this period, the PLL is locked and the

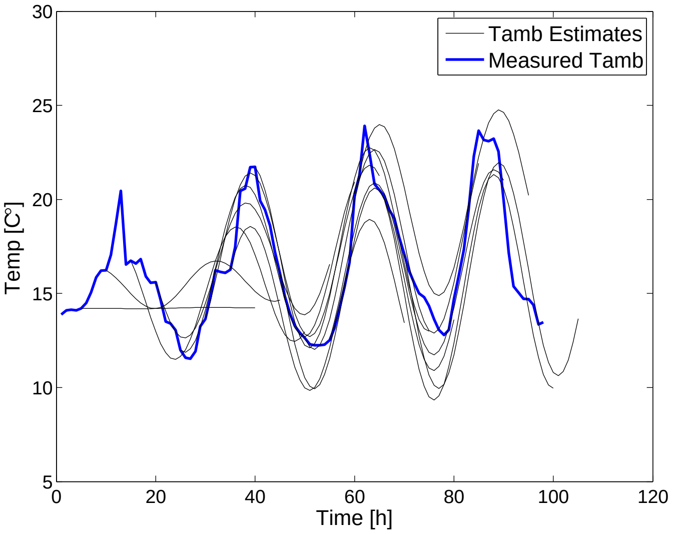


Fig. D.5: Prediction of future ambient temperature

prediction improves. Due to a non-sinusoidal oscillation of the measured ambient temperature, the amplitude estimate can be off, but as shown later, the most important factor is that the phase is correct, and therefore, this predictor is adequate for the MPC.

Cargo State and Parameter Estimation

The quality of the solution to the optimization problem in the MPC is dependent on the accuracy of the linear model used. Although the properties of the refrigeration system are well defined, the properties of the cargo are very uncertain because reefer containers are used to transport a wide range of different goods. The largest thermal mass is generally the cargo, and therefore, it is also the most interesting property with respect to exploiting the daily variances in ambient temperature. If the cargo heat capacity is not known, the lowest value must be used to ensure that the constraints are not violated because the lowest heat capacity also provides the fastest dynamics. Using the lowest possible heat capacity will limit the degree to which the variations in COP can be exploited. To resolve this issue, a combined parameter estimator and unknown input observer is introduced, and the estimates are then used to update the MPC on-line. The observer is based on the model of the cargo hold and cargo of the reefer containers, which is presented in the simulation model described in Section 2.1. It has been modified to simulate the unknown states and estimate and update the parameters for the cargo when conditions allow. Knowledge of parameter and state constraints are incorporated and used to ignore corrections that are outliers and to select sensible starting conditions.

The unknown states and parameters that must be estimated are shown in the following table:

Description	Unit
Cargo heat capacity, C_{cargo}	J/K
Cargo heat transfer coefficient, α_{cargo}	W/K
Cargo temperature, T_{cargo}	°C
Aluminium T-floor temperature, T_{floor}	°C

The heat capacity is the amount of energy required to increase the cargo temperature by one Kelvin, and the total heat transfer is the surface area of the cargo multiplied by the heat transfer coefficient. The estimation has to be performed on-line and must be based on the available measurements and actuator signals, which are the air return temperature T_{ret}, the air supply temperature T_{sup}, the ambient temperature T_{amb} and the fan speed V_{fan}.

The state equations for the model of the cargo and cargo hold are given by Equations (D.1) to (D.3)

$$\dot{T}_{air} = \frac{Q_{Cargo\to air} + Q_{amb\to air} + Q_{fan} + Q_{floor\to air} - Q_{cool}}{M_{air} \cdot Cp_{air}} \tag{D.1}$$

$$\dot{T}_{cargo} = \frac{-Q_{cargo\to air}}{C_{cargo}} \tag{D.2}$$

$$\dot{T}_{floor} = \frac{Q_{amb\to floor} - Q_{floor\to air}}{M_{floor} \cdot Cp_{floor}} \tag{D.3}$$

The change in air temperature is given by Equation (D.1) as the sum of all energy flows going to the control volume divided by the heat capacity of the air. This equation is essential for the estimator because the amount of energy going from the cargo to the air, Q_{cargo}, can be derived from this equation. The average of the measured supply temperature T_{sup} and return air temperature T_{ret} is assumed to be equal to T_{air}. In Equation (D.2), the change of the unknown state T_{cargo} is given as the energy going from the cargo to the air divided by the estimated heat capacity of the cargo. The heat capacity is used because it is not possible to individually estimate the mass and specific heat capacity of the cargo from the available measurements. The final state equation (D.3) provides the change in temperature in the aluminium floor of the container, and this has been included in the model because it has a very strong thermal coupling to the air and therefore also significantly slows the dynamics of the air temperature. The heat transfers in the above equations are given by Equations (D.4) to (D.7):

$$
\begin{aligned}
Q_{amb\to air} &= (T_{amb} - T_{air}) \cdot (1-\beta) \cdot \alpha_{box} && \text{(D.4)}\\
Q_{amb\to floor} &= (T_{amb} - T_{floor}) \cdot \beta \cdot \alpha_{box} && \text{(D.5)}\\
Q_{floor\to air} &= (T_{floor} - T_{air}) \cdot \alpha_{floor} && \text{(D.6)}\\
BQ_{cargo\to air} &= (T_{cargo} - T_{air}) \cdot \alpha_{cargo} && \text{(D.7)}\\
Q_{cool} &= f\,(T_{sup}, T_{ret}, V_{fan}) && \text{(D.8)}
\end{aligned}
$$

$Q_{amb\to air}$ is the heat transfer from the surroundings to the air in the container through the walls, roof and ends of the cargo hold given by the temperature difference multiplied by the fraction of total surface area represented by the walls, roof and ends and the heat transfer coefficient α_{box}, which is 43 $\frac{W}{K}$ [16]. The floor of the container also receives some heat from the outside, as given by Equation (D.5) as the temperature difference multiplied by the heat transfer coefficient α_{box} and the fraction of total surface area represented by the floor β, which is 0.190. The heat transfer from the floor to the air is given by Equation (D.6), where the heat transfer coefficient α_{floor} has been estimated to be 5000 $\frac{W}{K}$ in a step response experiment. The energy going from the cargo to the air, $Q_{cargo\to air}$, is given by Equation (D.7), and it includes two unknowns: the cargo temperature T_{cargo} and the heat transfer coefficient from the cargo to the air α_{cargo}. T_{air} is the mean of the supply and return temperature for the air entering and leaving the cargo hold, which is measured and thus very reliable. The cooling capacity delivered by the refrigeration system is given by Equation (D.8), which is the calculated cooling capacity from the temperature difference between the return and supply air and the air flow calculated from the fan speed V_{fan}.

There is an uncertainty in the heat influx through the container wall that depends on the air flow over the outside surface, rain and direct exposure to the sun. From Equation (D.1), it can be observed that this will lead to an uncertainty in the estimation of $Q_{cargo\to air}$ because Q_{fan}, $Q_{floor\to air}$ and Q_{cool} are known or can be measured.

The easiest state to estimate is the floor temperature because it will reach steady state equilibrium between the ambient and air temperatures, which are both measured; therefore, if the floor temperature is estimated using Equations (D.5), (D.6) and (D.3), the estimated floor temperature will over time track the actual floor temperature.

The cargo temperature may be estimated using the same method as for the floor because it converges towards the air temperature, but this estimate includes the uncertainty of the parameters α_{cargo} and C_{cargo}. If Equation (D.7) is inserted into Equation (D.2) and rearranged

$$\dot{T}_{cargo} = (T_{air} - T_{cargo}) \cdot \frac{\alpha_{cargo}}{C_{cargo}} \qquad \text{(D.9)}$$

it is clear that the change in cargo temperature is a first-order filter on the temperature difference between the air and the cargo, with a time constant that is given by the two unknown parameters. Therefore, the cargo temperature estimate will converge towards the mean air temperature, but it will only be accurate if the estimates of α_{cargo} and C_{cargo} are also accurate. It is assumed that the cargo is lumped and has a uniform temperature, which is a simplification, but the model is able to reflect the salient dynamics of the cargo on the time scale required by the controller.

A controller that uses the thermal inertia of the cargo to offset cooling to more efficient conditions will cool the cargo in pulses, which in this case have a duration of several hours, where the cargo temperature is decreased towards the lower temperature constraint. This excitation of the cargo dynamics is exploited to estimate the two unknown cargo parameters α_{cargo} and C_{cargo}.

It is assumed that the system is linear and time invariant (LTI), and therefore, it can be assumed that the heat transfer constant and the heat capacity of the cargo are constant over time. This can be used to estimate α_{cargo} by combining Equation (D.7) and Equation (D.10) and setting up two equations with two unknowns, which are solved for α_{cargo}.

$$\begin{aligned} Q_{cargo\rightarrow air-calc} &= Q_{cool} - Q_{amb\rightarrow air} - Q_{floor\rightarrow air} - Q_{fan} && \text{(D.10)} \\ Q_{cargo\rightarrow air-calc-t1} &= (T_{cargo-t1} - T_{air-t1}) \cdot \alpha_{cargo} && \text{(D.11)} \\ Q_{cargo\rightarrow air-calc-t2} &= (T_{cargo-t1} + T_{cargo-delta} - T_{air-t2}) \cdot \alpha_{cargo} && \text{(D.12)} \\ \alpha_{cargo} &= \frac{Q_{cargo\rightarrow air-calc-t2} - Q_{cargo\rightarrow air-calc-t1}}{T_{air-t1} - T_{air-t2} + T_{cargo-delta}} && \text{(D.13)} \end{aligned}$$

The measurements of $Q_{cargo\rightarrow air-calc}$, T_{cargo} and T_{air} in Equations (D.11) and (D.12) must be taken at two different times that have a significant difference in $Q_{cargo\rightarrow air-calc}$ to produce a reliable estimate of α_{cargo}. $T_{cargo-delta}$ is the estimated change in cargo temperature between the two sample points, and because the cargo dynamics are significantly slower than the dynamics of the air, this change can be obtained from the estimated cargo temperature given by Equation (D.2). To prevent inaccurate estimates due to disturbances from the controller, it is required that the slope of T_{sup} is smaller than $0.01 K/s$ before the first set of measurements are acquired because this means that the cooling capacity is stable. This also ensures that fast higher-order dynamics have settled and do not interfere with the accuracy of the estimate. The first measurement is taken at time = t1, and after 15 minutes, the second set of measurements are acquired, at time = t2. If the energy flows to the cargo, given by Equations (D.11) and (D.12), differs by more than 100 W, an update of α_{cargo} is performed.

The heat capacity of the cargo is more difficult to estimate because the estimate has to be inferred from the rate of change in air temperature measurements and the rate of change in the calculated energy flow from the cargo to the air. At a temperature setpoint of -20°C, the refrigeration system is able to cool the cargo with up to 4 kW, which over a period of one hour is sufficient to cool the cargo used in this study by 0.686

K. This means that the slopes involved are very small and that the result is sensitive to noise and disturbances. For α_{cargo}, this may be resolved by determining the slopes over a period, thereby reducing the impact from noise and disturbances.

The slope of the difference between the cargo temperature and the mean air temperature is given by Equation (D.14), where $\dot{Q}_{cargo\rightarrow air-calc}$ is the derivative of the energy going from the cargo to the air.

$$\dot{T}_{cargo\rightarrow air-diff} = \frac{\dot{Q}_{cargo\rightarrow air-calc}}{\alpha_{cargo}} \tag{D.14}$$

$$\dot{T}_{cargo} = \dot{T}_{cargo\rightarrow air-diff} + \dot{T}_{air-mean} \tag{D.15}$$

$$C_{cargo} = \frac{Q_{cargo\rightarrow air-calc}}{\dot{T}_{cargo}} \tag{D.16}$$

It is then possible to calculate the derivative of the cargo temperature as in Equation (D.15), where $\dot{T}_{air-mean}$ is the derivative of the measured mean air temperature in the container. To acquire accurate derivatives, the acquisition period should be as long as possible. The estimator requires 15 minutes where the cooling capacity of the refrigeration system is constant and large enough to ensure that at least 100 W of cooling is applied to the cargo before an update of the heat capacity of the cargo is calculated, as shown in Equation (D.16). In Figure D.6, the result of running the estimator on data from a real reefer container packed with 20,000 kg of bacon is shown.

On the top axes, the measured and estimated cargo temperature are shown. The estimated heat capacity of the cargo has been converted to mass in metric tons, using the specific heat capacity for bacon, which is $1050\frac{J}{kg\cdot K}$ [17], and should therefore level out at approximately 20. The data were obtained from a reefer that was located in an open field and therefore subject to weather disturbances.

The controller was an early attempt at offsetting some cooling from day to night using MPC but without the estimator for the cargo parameters. The cooling that is applied is mainly in pulses of 10 minutes, which is unfortunately a poor basis for estimating the cargo heat capacity because the pulses are too short to reliably estimate the change in cargo temperature through the air temperature. Therefore, the estimate of the cargo heat capacity is updated only 15 times. The heat transfer coefficient for this cargo has been estimated to be $550\frac{W}{K}$ from the measured cargo and air temperature and the calculated power to the cargo.

In Figure D.7, the result of the parameter estimation algorithm running on results from the simulation model with an α_{cargo} of $550\frac{W}{K}$ and a 20,000 kg cargo mass with a specific heat capacity of $1050\frac{J}{kg\cdot K}$ is shown. Measurement noise has been added to the simulation results. It is expected that α_{cargo} converges towards 550 $\frac{W}{K}$ and that M_{cargo} converges towards 20 t. The estimates quickly converge towards the true values immediately after start up, and within 24 hours, the estimates are close to the correct values. In this test, the starting values have been set to demonstrate convergence,

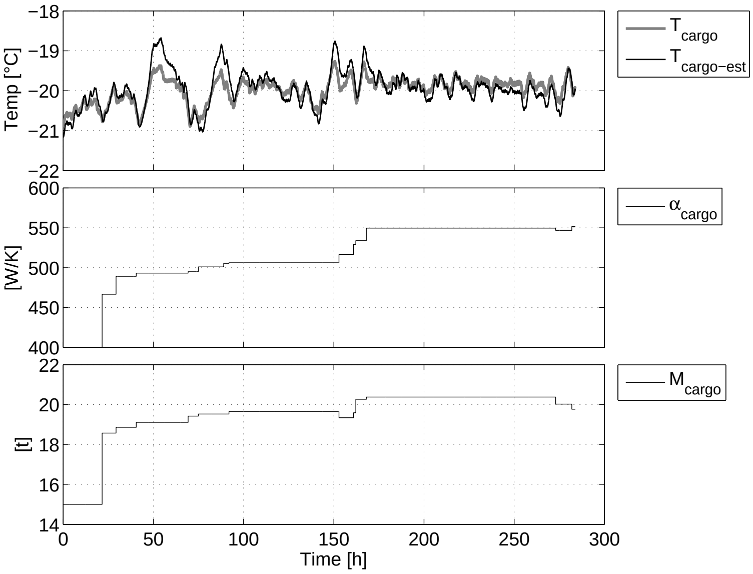


Fig. D.6: Cargo parameter estimator on reefer container measurements

but it is often possible to deduce the type of cargo from the temperature setpoint and subsequently select better start values that will enable faster estimator convergence.

2.3 Controller Setup

A non-linear simulation model has been developed as described in Section 2.1, but this model has 81 states and a reduced model is desired because the controller eventually is to be used on an embedded hardware platform with limited resources. The objective of this study is to exploit the daily cycles in ambient temperature and COP. To exploit daily variations in ambient temperature, the MPC prediction horizon must be at least 24 hours, and if it is to control the fast dynamics directly, the resolution must be high. This leads to a high computational load due to the many steps in the prediction horizon, and an alternative must be found. A large part of the simulation model dynamics are considerably faster than the ones relevant to the long-term objective, and therefore, a reduced model containing only the slow states is derived for the MPC. The proposed set up is shown in Figure D.8.

Because the MPC only handles the slow dynamics, a classical controller is inserted between the MPC and the plant (reefer) to perform closed loop control of the fast dynamics while accepting setpoints from the MPC; see [18, 19]. This has several benefits. First, the model for the MPC can be heavily reduced because only the states directly

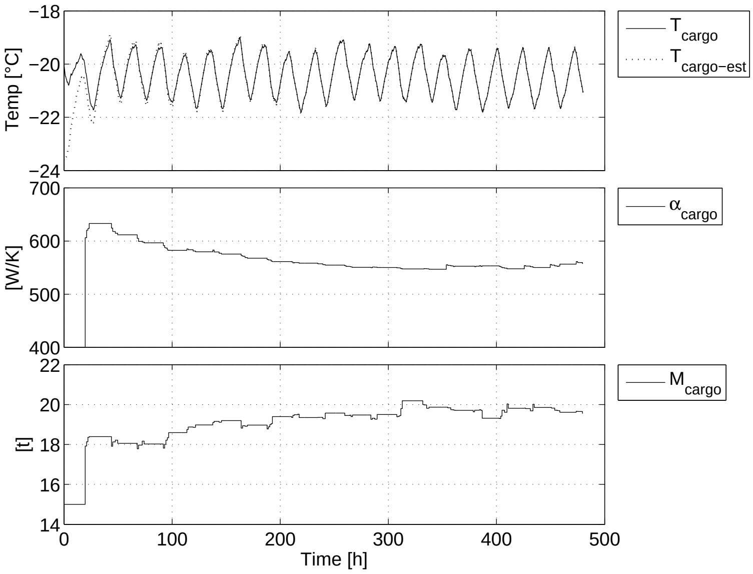


Fig. D.7: Cargo parameter estimator on simulation model

relevant to the objective are required. Second, the resolution of the prediction horizon can be reduced because the model for the MPC only has slow dynamics. Third, it is possible to use linear MPC because of the linearizing effect of the classical controller. This is of course only obtainable if the interface between the MPC and the classical controller can be chosen such that the level of abstraction created by the classical controller is high enough to mask the non-linear dynamics while still being able to effectively control all the inputs relevant to the MPC objective.

Interface from MPC to Linearizing Controller

According to [20], the dominant dynamics of a refrigeration system, with respect to control applications, are the thermal constants of the metal surfaces in the heat exchangers and the refrigerant mass time constants. The largest thermal mass in the refrigeration system itself is the evaporator, which has a mass of 23 kg, yielding a heat capacity of 20.7 kJ/K, but the T-floor has a heat capacity of 2.7 MJ/K, and a typical cargo of frozen meat has a heat capacity of 100 MJ/K. This gives a separation of dynamical speed that is several orders of magnitude between the cargo and the refrigeration system, but the temperature of the air is problematic. The heat capacity of the air in the box at -20 C° is 95.36 kJ/K, and this is not far from that of the evaporator, but because of the thermal coupling to the T-floor and cargo, the actual dynamics of the air temperature

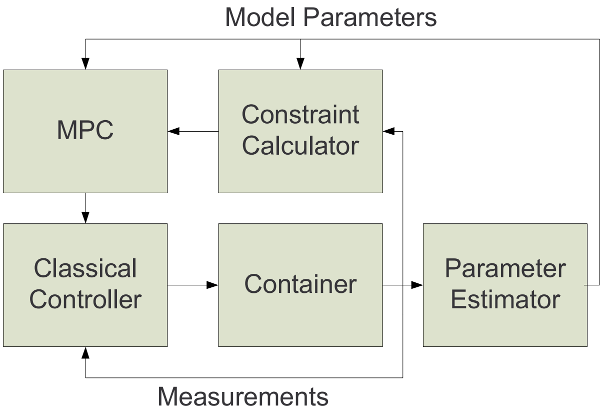


Fig. D.8: Overall Controller Block Diagram

are considerably slower. The important factor is the response from the supply temperature T_{sup} to the return temperature T_{ret}, and because the air has to flow over both the T-floor and the cargo, the response is slowed considerably. The T-floor alone is sufficient to leave a comfortable gap in dynamical speeds between the dynamics that must be controlled by the MPC and the non-linear refrigeration system dynamics.

The cooling capacity Q_{cool} is chosen as the reference from the MPC to the classical controller because it has a direct and nearly linear effect on the cargo temperature. The proposed controller set up is shown in Figure D.9.

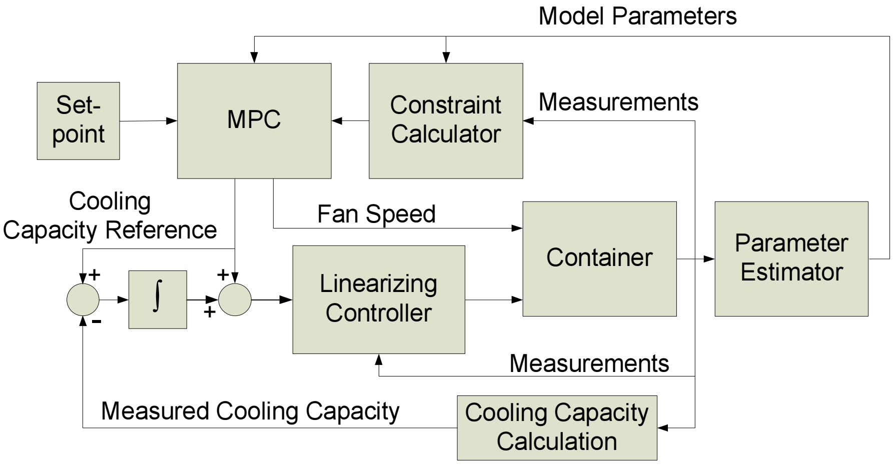


Fig. D.9: Controller Block Diagram

The cooling capacity reference from the MPC is discrete, that is, it changes instantly and remains constant until the next update from the MPC. It is infeasible for the refrigeration system to follow such a reference, and therefore, an integrator on the difference between the actual and requested cooling capacity is added to remove the error from lag in the refrigeration system.

Linearizing Controller

The linearizing controller is a non-linear feed forward that is based on the model, with a traditional PI controller to correct for inaccuracies, and this ensures that the system reaches the capacity requested by the MPC quickly. The actuators controlled by this controller are the condenser fan, the compressor and the expansion valves for the evaporator and the economizer. The compressor has a minimum on-time of 30 s that must be observed, and therefore, the rules for this are also built into the linearizing controller. The linearizing controller was tested on the reefer container packed with bacon, which is the basis of this study. The test results are shown in Figure D.3, where the reefer is taken through a series of cooling capacity request steps that the reefer must then follow.

Model Predictive Controller

The MPC is constructed using Yalmip (see [21]), with the objective of reducing the amount of energy consumed by exploiting daily variations in temperature; therefore, the prediction horizon should be at least 24 hours. The maximum step size is limited by the fastest dynamics that must be controlled, in this case, the air temperature. A step size of ten minutes is required to have adequate control of the air temperature, but this leads to a prediction horizon of 144 steps, which is estimated to be too computationally heavy for the embedded hardware. It is therefore chosen to solve this problem by dividing the prediction horizon (see [22]) into two sections with different step sizes; the first one with six ten-minute steps and after those, another 23 steps of one hour each, as shown in Figure D.10.

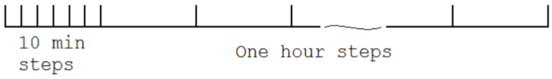


Fig. D.10: Prediction Horizon

The MPC runs once an hour, and the first six ten-minute steps are then implemented one-by-one, after which the process starts over again. The advantage of this approach is that fine-grained control can be achieved while the total number of steps and iteration frequency remain low. For this system, fine-grained control is an advantage because the most efficient point of operation in some instances delivers more cooling than required to maintain the temperature. Therefore, the best option is to run at the optimal cooling capacity for shorter time, and by dividing the first hour into six smaller steps, the effective minimum capacity that can be delivered per hour is reduced to a level that is more suitable for this application.

Cost Calculation and Constraint Setup

The evaporator fan circulates the air that moves energy from the box to the evaporator; therefore, it is required to run while the compressor is turned on. The power consumption of the fan has an impact on the cost of running the system in two ways: there is the direct power driving the fan and the heat generated by the fan that must be removed again by the refrigeration system. The fan speed is controlled in discrete steps; thus, it is important to model this behaviour in the actuator constraints of the controller. This is performed by declaring the fan speed as a binary variable, which transforms the problem to a mixed-integer program. The resulting constraint is shown in Equation (D.19).

The fan speed variable $\mathrm{V}_{\mathrm{fan}}$ is 1 when the fan is turned on, and $\mathrm{Q}_{\mathrm{min}}$ and $\mathrm{Q}_{\mathrm{max}}$ are the constraints on the cooling capacity, which are calculated based on the current operating point. From Equation (D.19), it is obvious that $\mathrm{Q}_{\mathrm{cool}}$ is required to be zero when the fan is turned off, and it is constrained by $\mathrm{Q}_{\mathrm{min}}$ and $\mathrm{Q}_{\mathrm{max}}$ when the fan is turned on.

The maximal cooling capacity of the refrigeration system is dependent on suction pressure and thus the temperature in the box, and therefore, $\mathrm{Q}_{\mathrm{max}}$ and $\mathrm{Q}_{\mathrm{min}}$ from Equation D.19 must be calculated from the box temperature. This is performed using manufacturer data given as polynomials according to [23].

In many real systems, we encounter a non-linear cost on a control input, typically due to decreasing efficiency as the speed of an actuator increases. This is also the case for this system, but only to a limited degree for the compressor. The largest change in COP for the refrigeration system is dependent on the ambient temperature because it has a strong coupling to the discharge pressure, which is a determining factor on the efficiency of the compressor. In Figure D.11, the COP of the system is shown for a fixed setpoint of -20°C and varying cooling capacity and ambient temperature is shown.

The dotted line is the COP of the system when the compressor is running in PWM mode, where the compressor is stopped and started with a duty cycle that matches the required cooling capacity. It is necessary to perform this when the cooling capacity required to maintain the setpoint in the container is lower than that provided by the refrigeration system at the lowest possible compressor speed. The reason for the sharp decline of COP in PWM mode is that the evaporator fan continues running while the compressor is off, and therefore, the contribution from the fan compared to the cooling capacity is increased as the duty cycle approaches zero. The expression for the COP in

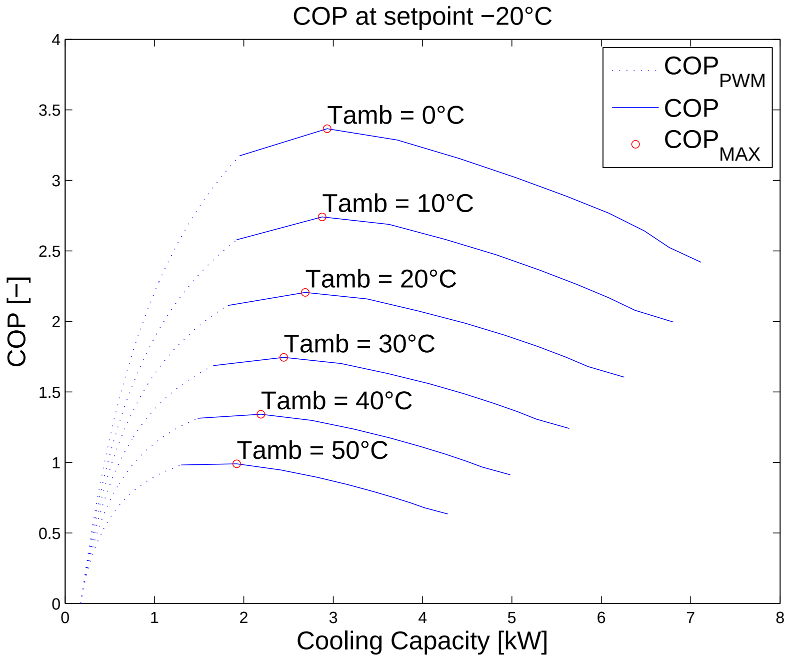


Fig. D.11: The COP of the refrigeration system at varying ambient temperatures

PWM mode is given in Equation (D.17)

$$COP_{PWM} = \frac{D \cdot Q_{cool-min} - Q_{fan}}{D \cdot P_{cpr-min} + Q_{fan}} \tag{D.17}$$

where D is the duty cycle, $Q_{cool-min}$ is the cooling capacity at minimum compressor speed, $P_{cpr-min}$ is the consumed compressor power at minimum compressor speed, and Q_{fan} is the power consumed by the fan. The ambient temperature has a substantial impact on both the level and the shape of the COP curve, and to effectively exploit this property, the objective of the MPC must accurately reflect the cost over the length of the prediction horizon. This requires knowledge of the future ambient temperature, which is not available; therefore, the predictor outlined in Section 2.2 is used. The predicted ambient temperature is used to select the appropriate COP curve, which must be converted to an affine cost that reflects the shape of the COP curve. The COP curve shown here has 9 points because it is generated by simulating the model, as described in Section 2.1, to steady state at fixed compressor speeds from 20 Hz to 110 Hz in increments of 10 Hz. For the MPC, a set of COP curves is generated at higher resolution for the ambient temperature to ensure optimal conditions for the cost optimization.

A linear solver is used, and therefore, the COP data are converted to a series of linear segments in the form $y = ax + b$ such that they may be used in an epigraph representation of the cost of running the compressor in the objective function. The objective and constraints are listed in Equations (D.18) to (D.27):

Objective:

$$P_c(k) + V_{fan}(k) \cdot 195 + T_s(k) \cdot 10^4 \tag{D.18}$$

Constraints:

$$V_{fan}(k) \cdot Q_{min}(k) <= Q_{cool}(k) <= V_{fan}(k) \cdot Q_{max}(k) \tag{D.19}$$
$$Q_{cool}(k) \cdot a_1(k) + b_1(k) <= P_c(k) \tag{D.20}$$
$$Q_{cool}(k) \cdot a_2(k) + b_2(k) <= P_c(k) \tag{D.21}$$
$$Q_{cool}(k) \cdot a_3(k) + b_3(k) <= P_c(k) \tag{D.22}$$
$$Q_{cool}(k) \cdot a_4(k) + b_4(k) <= P_c(k) \tag{D.23}$$
$$V_{fan-min} <= V_{fan}(k) <= 1 \tag{D.24}$$
$$0 <= T_s(k), \tag{D.25}$$
$$T_{cargo-min} - T_s(k) <= T_{cargo}(k) <= T_{cargo-max} + T_s(k), \tag{D.26}$$
$$T_{air}(k) < T_{air-max} + T_s(k) \tag{D.27}$$

The objective function shown in Equation (D.18) reflects the power used by the container, expressed by the first and second terms, where P_c is the power used by the compressor and condenser fan and $V_{fan}(k) \cdot 195$ is the power consumed by the evaporator fans. The constraints given in Equations (D.20) to (D.23) are the linear approximation of the convex COP, where the parameters $a_n(k)$ and $b_n(k)$ are derived from the COP curve that matches the predicted ambient temperature at the time of the solution point, which results in a cost for the compressor that forms a surface with cooling capacity on one axis and time on the other.

The third term of the objective, the slack variable T_s, is the cost of violating the constraints for cargo and air temperatures, which are defined in Equation (D.26) and (D.27). It ensures that the controller will keep running and produce solutions in the event that one of the temperature constraints is violated. The cost of violating the constraint is very high, and therefore, the controller will prioritize getting back inside the constraints over all other objectives, and this is the desired behaviour.

It is not expected that the constraints will be violated under normal operation, but a violation could occur if the ambient conditions of the reefer change suddenly. Because the MPC is recalculated only once every hour, the applied cooling might be incorrect for up to an hour; however, for frozen cargo, this is not critical. When reefers are unloaded from the ship, they are unpowered and may remain so for up to four hours while being transported by truck. For chilled cargo, it is critical not to cause chill injuries by injecting air that is too cold into the cargo hold, and if the weather suddenly changes from sunshine to rain, this situation might occur. Therefore, this controller would need to be supervised and the MPC recalculated ahead of time if the constraints were violated to ensure adequate disturbance rejection.

The temperature constraint that has been selected for the cargo is $T_{set}\pm0.25$ K, which ensures that the variation in cargo temperature remains small, thereby reducing the risk of damaging the cargo. The model used for the controller lumps the entire cargo into one large volume, but in reality, the temperature distribution inside the container is non-uniform; see [10]. Therefore, the air temperature has been constrained to $T_{set}+2$ K because this will ensure that the air is cooled and circulated regularly, which prevents the formation of local hot-spots.

3 Results

The objective of the experiments conducted in this work is to identify potential reductions in energy consumption by introducing modern control methods, and two different scenarios are investigated at three different ambient temperatures in an attempt to map the energy savings potential. The traditional approach for controlling the evaporator fan is that it must always be running because the measurement of the cargo temperature is performed indirectly through the return air temperature, and this measurement becomes invalid when the fans are turned off. However, with the cargo estimator, it is possible to turn off the fans and use the estimate of the air and cargo temperature instead, which results in a substantial reduction in consumed energy. It is however interesting to know the fraction of the reduction that comes from cooling storage in the cargo and how much that comes from savings on the fans. Therefore, the first scenario uses the MPC with the fans forced to be always on, and the second scenario allows the MPC to control both the fans and the cooling capacity. The references used in the experiments are simulations using the same linearizing controller as for the MPC but with a traditional PI controller for generation of the cooling capacity reference. In Figure D.12, a section of the simulation results for the reference and the two test scenarios are shown. The two panels on top are the reference simulation, the two panels in the middle are the scenario where the fans are always on, and the two panels in the bottom are the scenario where the controller is allowed to turn off the fans. In the reference simulation, the fans are running continuously and the controller keeps the cargo and air temperature close to the setpoint. The compressor is running in PWM mode, and it can be observed that the duty cycle and cooling request are increased at high ambient temperatures to compensate for the higher influx of heat into the cargo hold.

For the scenario where the fans are always on, it can be observed that the MPC uses the cargo's thermal inertia, allowing the air and cargo temperature to rise to its upper constraint during the period where the ambient temperature is at its highest. The compressor is running in PWM most of the time because there is nothing to gain

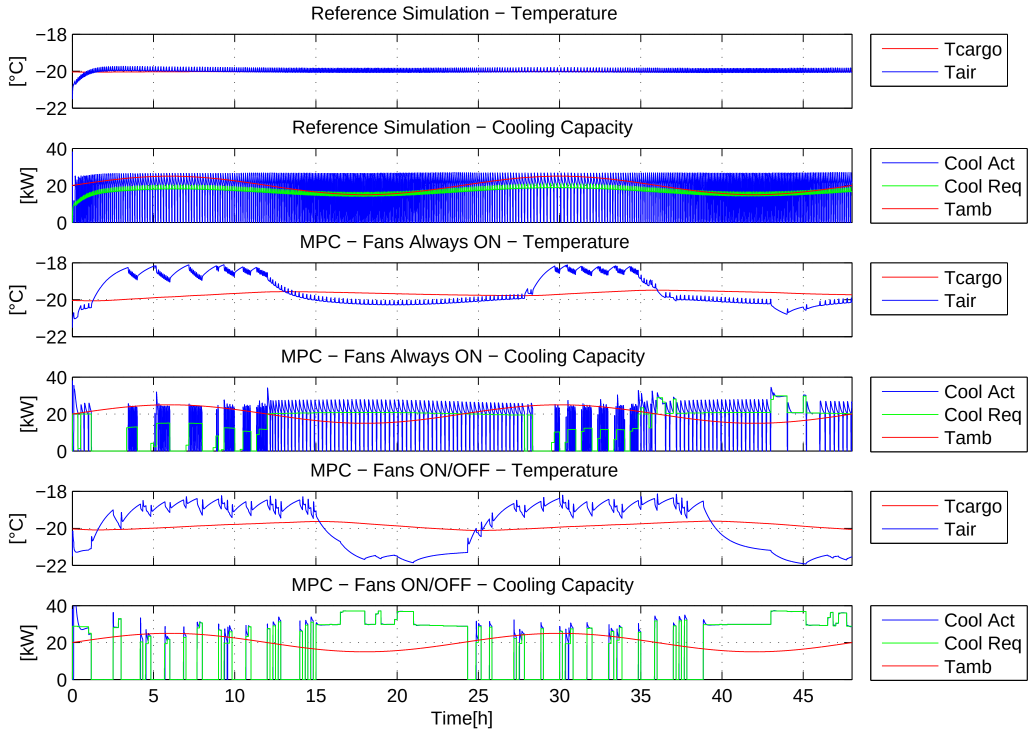


Fig. D.12: Test results for the reference and the two test scenarios at 20°C ambient temperature.

by running the compressor faster, at a lower efficiency, if it is not possible to turn off the fans for a longer period afterwards.

In the last scenario, the fans are turned off when the compressor is not running, and the compressor is now running at a higher capacity, which allows the compressor and fans to be turned off for longer periods, thus saving power. During the periods of high ambient temperature, the compressor is only turned on to keep the air temperature below the upper limit while the cargo temperature is slowly increasing, and this shows that the controller behaves as intended.

The power savings found in the six tests are listed in Table D.1.

From the results, it is clear that the potential power savings are highly dependent on the operating conditions, and the reason for this is found in the COP curves in Figure D.11 and in the way the refrigeration system was designed. During normal operation of the container, when the cargo is at its setpoint, the system is typically running at a fraction of the cooling capacity that is available because the system is designed to be able to cool a hot cargo within a reasonable amount of time. However, the system is also most efficient when it is running at low capacity because the losses in the system are smaller at low capacity. The COP curve for the compressor alone is monotonically decreasing as the speed increases, but when the power from the fans is added, the COP curves shown in Figure D.11 with a maximum in the lower capacity range emerge.

Ambient Temperature	Fans Always ON	Fans ON/OFF
10 ± 5°C	2.53%	21.9%
20 ± 5°C	3.07%	11.1%
30 ± 5°C	2.70%	3.96%

Table D.1: Results from the different test scenarios.

This means that when the amount of cooling required to maintain the setpoint in the cargo hold matches the most efficient capacity, the potential energy reduction from being able to turn off the fans is low. This is reflected by the results that show that the energy savings decrease as the ambient temperature increases. When the most efficient mode of operation is running the fans continuously, the cargo may still be used as a cooling storage, and therefore, there will always be a gain from using this control strategy. In the present experiments, potential savings between 2.5% and 3% are possible with a 10 K variation in ambient temperature. With a consumption of 8.9 MW for all the containers on a ship, a 2.5% reduction yields 403.7 GJ over a 21 day trip, which is approximately 10000 kg of heavy fuel oil. The six tests presented here show that there is a large difference in potential savings depending on the cooling demand. Therefore, further tests should be conducted to test the entire range of operation of the container. It is expected that the largest savings will be found where the difference between the temperature setpoint and ambient temperature is low because, in this case, the majority of the heat input to the cargo hold is from the evaporator fans. This means that switching off the fans yields a large decrease in the required cooling capacity and results in relatively large savings on both the fan and the compressor. Therefore, it is expected that it is possible to save a larger percentage of the power for shipments of fruits and vegetables that run at a higher setpoint.

4 Conclusion

In this study, a model predictive controller that reduced the power consumption of a refrigerated container by turning off the cargo hold fans when they were not needed and by using the cargo thermal inertia to store cooling was presented. Simulation experiments were performed using a detailed model of the refrigeration system at three different ambient temperatures, with savings in power consumption found to be up to 21% depending on the ambient temperature. It is expected that larger savings are possible for cargoes that requires a higher setpoint. The largest savings were found to be possible only if the controller could control the fans and turn them off when they were not needed. An estimator for cargo temperature, heat capacity, and heat transfer value was developed and used to update the MPC on-line, thereby enabling optimal exploitation of the available thermal inertia while keeping the cargo temperature within its constraints. The controller was divided into two layers, with the MPC on top providing a cooling power request to a linearizing controller, which handled the faster non-linear dynamics of the refrigeration system. A prediction horizon with varying step sizes led to a reduction in the computational demand from the MPC.

References

[1] Marinelink, "World trade in perishable cargo to grow," 2013. [Online]. Available: http://www.marinelink.com/news/article/world-trade-in-perishable-cargo-to-grow/328311.aspx

[2] Container-Handbook, "8.1.2 actual power consumption," 2009. [Online]. Available: http://www.containerhandbuch.de/chb_e/wild/index.html?/chb_e/wild/wild_08_01_02.html

[3] R. G. M. van der Sman and G. J. C. Verdijck, "Model predictions and control of conditions in a ca-reefer container," in *ISHS Acta Horticulturae 600: VIII International Controlled Atmosphere Research Conference*, Agrotechnological Research Institute ATO, P.O. box 17, 6700 AA Wageningen, the Netherlands, 2003, pp. 163–171. [Online]. Available: http://www.actahort.org/members/showpdf?booknrarnr=600_20

[4] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jorgensen, "Economic model predictive control for building climate control in a smart grid," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, IEEE. http://orbit.dtu.dk/fedora/objects/orbit:74519/datastreams/file_6543920/content, 2012, pp. 1–6.

[5] Y. Ma and F. Borrelli, "Fast stochastic predictive control for building temperature regulation," in *American Control Conference (ACC), 2012*, June 2012, pp. 3075–3080. doi:10.1109/ACC.2012.6315347

[6] F. Oldewurtel, A. Parisio, C. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," in *American Control Conference (ACC), 2010*, June 2010, pp. 5100–5105. doi:10.1109/ACC.2010.5530680

[7] J. E. Braun, "Reducing energy costs and peak electrical demand through optimal control of building thermal mass," *ASHRAE Transactions*, pp. 264–273, 1990.

[8] J. E. Braun and R. W. H. Laboratories, "Load control using building thermal mass," in *Transactions of the ASME Vol. 125*, 2003, pp. 292–301. doi:10.1115/1.1592184

[9] K. Vinther, H. Rasmussen, R. Izadi-Zamanabadi, J. Stoustrup, and A. Alleyne, "A learning based precool algorithm for utilization of foodstuff as thermal energy storage," in *Control Applications (CCA), 2013 IEEE International Conference on*, Aug 2013, pp. 314–321. doi:10.1109/CCA.2013.6662777

[10] G. J. C. Verdijck, "Model-based product quality control applied to climate controlled processing of agro-material," 2003. [Online]. Available: http://alexandria.tue.nl/extra2/200310443.pdf

[11] Wageningen, "Agrotechnology & food sciences group, quest-leaflet," 2015. [Online]. Available: https://www.wageningenur.nl/upload_mm/d/6/b/2b3cd77e-4c34-4a62-95a9-1142267427de_Questleaflet2008.pdf

[12] J. d. Boogaard, G.J.P.M. van den; Kramer, "Quest: Quality and energy in storage and transport of agro-materials, final public report," Wageningen Agrotechnology & Food Sciences Group, (Rapport / AFSG 657), 2006.

[13] A. Jakobsen, B. Rasmussen, and M. Skovrup, "Development of energy optimal capacity control in refrigeration systems," in *Proceedings of 2000 International Refrigeration Conference*. Purdue University, Indiana, USA: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1498&context=iracc, 2000, pp. 329–336.

[14] MCI, "Maersk container industry, starcool refrigerated container specifications," 2013. [Online]. Available: http://www.maerskbox.com/

[15] K. Sørensen, M. Skovrup, L. Jessen, and J. Stoustrup, "Modular modelling of a reefer container," *Accepted - In press, International Journal of Refrigeration*, 2015. [Online]. Available: http://authors.elsevier.com/sd/article/S0140700715000766. doi:10.1016/j.ijrefrig.2015.03.017

[16] "Star cool reefer specification," 2015. [Online]. Available: http://ipaper.ipapercms.dk/MCI/Brochures/MarkQ/

[17] "Specific heats of common food and foodstuff," 2015. [Online]. Available: http://www.engineeringtoolbox.com/specific-heat-capacity-food-d_295.html

[18] S. Engell, "Feedback control for optimal process operation," *Journal of Process Control*, vol. 17, no. 3, pp. 203 – 219, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152406001326. doi:10.1016/j.jprocont.2006.10.011

[19] L. F. S. Larsen, "Model based control of refrigeration systems," 2005. [Online]. Available: http://www.control.aau.dk/~jakob/phdStudents/lfslThesis.pdf

[20] B. Rasmussen, A. Musser, and A. Alleyne, "Model-driven system identification of transcritical vapor compression systems," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 444–451, 2005. doi:10.1109/TCST.2004.839572

[21] J. Löfberg, "Yalmip is a modelling language for defining and solving advanced optimization problems," 2009. [Online]. Available: http://users.isy.liu.se/johanl/yalmip/

[22] U. Halldorsson, M. Fikar, and H. Unbehauen, "Nonlinear predictive control with multirate optimisation step lengths," *IEE Proceedings - Control Theory and Applications*, vol. 152, no. 3, pp. 273–285, 2005. doi:10.1049/ip-cta:20041310

[23] BS, "British standard/en 12900, refrigerant compressors. rating conditions, tolerances and presentation of manufacturer's performance data," ISBN: 0 580 32694 2, 2005.