

# Framework for dynamic optimization of district heating systems using Optimica Compiler Toolkit

Gerald Schweiger<sup>1</sup> Håkan Runvik<sup>2</sup> Fredrik Magnusson<sup>3,2</sup> Per-Ola Larsson<sup>2</sup> Stéphane Velut<sup>2</sup>

<sup>1</sup>AEE INTEC, 8200 Gleisdorf, Austria, gerald.schweiger@aee.at

<sup>2</sup>Modelon AB, SE-223 70 Lund, Sweden, {per-ola.larsson, hakan.runvik, fredrik.magnusson, stephane.velut}@modelon.com

<sup>3</sup>Lund University, SE-221 00 Lund, Sweden, fredrik.magnusson@control.lth.se

## Abstract

Recent studies show that district heating infrastructures should play an important role in future sustainable energy systems. Tools for dynamic optimization are required to increase the efficiency of existing systems and design new ones. This paper presents a novel framework to represent, simplify, simulate and optimize district heating systems. The framework is implemented in Python and is based on Optimica Compiler Toolkit as well as Modelon's Thermal Power Library. The high-level description of optimization problems using Optimica allows flexible optimization formulations including constraints on physically relevant variables such as supply temperature, flow rate and pressures. The benefit of new algorithms for symbolic elimination in Optimica Compiler Toolkit is also investigated. The framework is applied on a test case, which is based on a planned city district located in Graz, Austria. The results demonstrate the generality of the representation as well as the accuracy of the simplification for dynamic optimization of temperature supply and pressure control. *Keywords: district heating, dynamic optimization, symbolic elimination*

## 1 Introduction

A major challenge for future energy systems is the design of systems that integrate large shares of fluctuating renewable inputs while improving the overall system efficiency. There are a number of options for increasing energy system flexibility, including the combination of different energy domains, increasing supply and demand flexibility or the integration of energy storage technologies. Previous research has shown that district heating infrastructure has the potential to play a key role in sustainable energy systems (Lund et al., 2014; Schweiger et al., 2017b). The new generation of district heating systems (called 4th generation district heating) plays an integral part of smart energy systems. Among others these systems will be characterized by intermittent operations and highly fluctuating supply temperatures. As reported by (Allegrini et al., 2015), there is much to be done to explore the full benefit of innovative district energy systems. They argue that a shift to fully dynamic

models and sophisticated control design would be supportive. Limitations of standard methods rely often on simplified models, static relationships and single-domain approaches. Therefore standard approaches are restricted and thus unsuitable for investigating many issues. The presented framework is based on the previous work of some authors (Velut et al., 2014; Runvik et al., 2015; Schweiger et al., 2017a).

The main contributions of this paper are (i) a demonstration of the capabilities of Modelon's Thermal Power Library that in version 1.14 will have out-of-the box models for dynamic thermo-hydraulic optimization of district heating systems, (ii) a demonstration of a framework for creating and manipulating district heating networks in Python as well as translating networks into executable Code for simulation and optimization and (iii) an investigation of the impact of the new algorithm for symbolic elimination available in JModelica.org and Optimica Compiler Toolkit (OCT).

## 2 Tools and languages

Three environments were used within the framework: (i) The unified network representation and the aggregation algorithms are implemented in Python; (ii) Dymola is used to simulate the complex models and (iii) JModelica.org and OCT were used to solve the dynamic optimization problem.

### 2.1 OCT and JModelica.org overview

#### 2.1.1 JModelica.org

JModelica.org (Åkesson et al., 2010) is an open-source platform developed for simulation, optimization and analysis of complex dynamical systems. It utilizes the open Modelica and FMI (Functional Mock-up Interface) standards and has a Python-based user interface. It is developed in collaboration between Modelon and several academic institutions, such as the Department of Automatic Control and the group of Numerical Analysis at the Centre for Mathematical Sciences at Lund University.

Of special interest in this project is the dynamic optimization capabilities of this tool. An extension of the

Modelica language called Optimica (Åkesson, 2008) is used for this purpose. The model dynamics of the optimization models are described using Modelica. To add the extra information necessary to describe the optimization formulation Optimica is used. This means that constraints, objective function and optimization time horizon all can be collected in one easily understandable model. More details of how optimization problems are solved in the JModelica.org toolchain are presented in the following sections.

### 2.1.2 Optimica Compiler Toolkit

Optimica Compiler Toolkit (OCT) is based on JModelica.org technology, but has several additional features (Modelon, 2016). One of these is the support for encrypted libraries, which is of special interest in relation to this project. This makes it possible to combine the usage of commercial libraries, here Modelon's Thermal Power Library, with the optimization framework. This lowers the difficulty for users to solve their own optimization problems, when predefined components and media models from the library can be used.

## 3 District Heating Network Models

A district heating network model for short-term production planning must capture the following: (i) transport delays depending on mass flows, (ii) pressure losses and (iii) heat losses.

### 3.1 General model properties

The presented framework is based on the physics-based modeling language Modelica and a high-level, large-scale dynamic optimization method available in OCT.

High-fidelity models of district heating networks often have high computation cost and some model properties like events or non-differentiability make them even unusable in dynamic optimization. Hence, there is a need to design simpler models, in particular regarding size and differentiability, that can be used for online optimization. There is also a need to design accurate models that can be used for dynamic simulation to validate the optimal inputs computed based on the simpler model. Models of both types will be available in Modelon's Thermal Power Library 1.14.

Pipes are the central components in district heating systems. The pipe model for simulation is implemented based on a plug-flow approach as the solution of the following one dimensional energy balance:

$$\frac{\partial T}{\partial t} + v(t) \frac{\partial T}{\partial x} + \frac{1}{S\rho c_p} q(T(x)) = 0$$

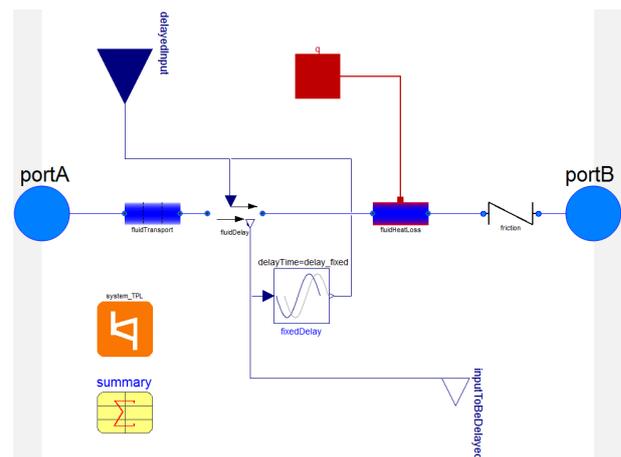
where  $v$  is the fluid velocity,  $S$  the cross-section area,  $\rho$  the fluid density,  $c_p$  the specific heat capacity of the fluid and  $q$  the heat loss to the surroundings of the pipe. The Modelica built-in operator `spatialDistribution` provides a robust method to approximate the solution of such partial differential equations when there is no heat loss, i.e.

$q = 0$  (Modelica Association, 2014). The operator keeps track of the spatial distribution via suitable sampling, interpolation and shifting of the stored distribution and it also supports flow reversal. Assuming positive flow and a heat loss  $q$  that depends linearly on  $T_{boundary} - T(x)$ , the difference between the surrounding temperature and the fluid temperature, the temperature at the pipe outlet is

$$T(x = L, t) = T_{boundary} + (T(x = 0, t - \tau) - T_{boundary}) e^{-\frac{\tau}{T_p}}$$

where  $L$  is the pipe length,  $\tau$  the time-varying transport delay and  $T_p$  a temperature decay constant. From the previous equation, it can be seen that the pipe model with heat loss can be implemented using two `spatialDistribution` operators, one to keep track of the temperature distribution inside the pipe and therefore  $T(x = 0, t - \tau)$ , and one to calculate the time-varying delay  $\tau$  that is needed to compute the impact of the heat loss given by  $e^{-\frac{\tau}{T_p}}$ .

The `spatialDistribution` operator can however not be used in the optimization framework because of insufficient differentiability of the involved equations. The pipe model for optimization (see Figure 1) contains a combination of a fixed time delay and a discretized dynamic volume. The goal is to compute the main characteristics of the pipe without having to use a model with a large number of segments which would increase model complexity. The fixed delay is dependent on the range of the mass flow for each pipe and corresponds to the minimal time delay. The dynamic volume must capture the flow-dependence of the varying transport delay. The number of segments within the dynamic volume depends on the geometry of the pipe.



**Figure 1.** Pipe model for optimization consisting (from left to right) of a dynamic volume model (`fluidTransport`), a model that captures the fixed delay (`fluidDelay`), a model that calculates the heat losses (`fluidHeatLoss`) and a model that calculates the pressure drop (`friction`).

### 3.2 Case study

The case study represents a district heating network in a planned city district in Graz/Austria that consists of one

production unit, 16 consumers and a total length of about 4200 m, see Figure 2. We assume a perfect load prediction over the entire optimization horizon. Work has also been done on non-perfect load prediction (Rantzer, 2015), but was omitted here to focus on other parts of the framework.



**Figure 2.** The scheme of the district heating system. The production unit is seen on the right side; the black/blue circles represent the 16 consumers.

Optimica files that extend the optimization models are used to describe the optimization problems. The dynamic optimization problem used for all optimization cases has the general form

$$\begin{aligned} \min. \quad & \int_{t_0}^{t_f} (\alpha T_{\text{prod}} + \beta dp_{\text{prod}} + \gamma \dot{Q}_{\text{prod}}^2 + \delta \dot{d}p_{\text{prod}}^2) dt, \\ \text{s.t.} \quad & \text{model dynamics,} \\ & m_{\text{Prod}}(t) \leq m_{\text{Prod}}^U \quad \forall t \in [t_0, t_f], \\ & T_{\text{Customer}}^L \leq T_{\text{Customer}}(t) \quad \forall t \in [t_0, t_f], \\ & dp_{\text{Customer}}^L \leq dp_{\text{Customer}}(t) \quad \forall t \in [t_0, t_f], \end{aligned}$$

where  $T_{\text{prod}}$  is the supply temperature,  $dp_{\text{prod}}$  the differential pressure at the production unit,  $Q_{\text{prod}}$  the load and  $\alpha, \beta, \gamma$  as well as  $\delta$  are weights. The load derivative  $\dot{Q}_{\text{prod}}$  and the pressure derivative  $\dot{d}p_{\text{prod}}$  are the degrees of freedom in the optimization formulation. These are squared in the cost function to penalize fast control signal changes.  $m_{\text{Prod}}^U$  is the upper limit of the mass flow at the production unit and it was set to 65 kg/sec; it is representing the pump limitations.  $T_{\text{Customer}}^L$  is the lower limit of the supply temperature for all customers and it was set to 60 deg.C. The lower limit of the differential pressure for all customers ( $dp_{\text{Customer}}^L$ ) was set to 0.5 bar. Minimum

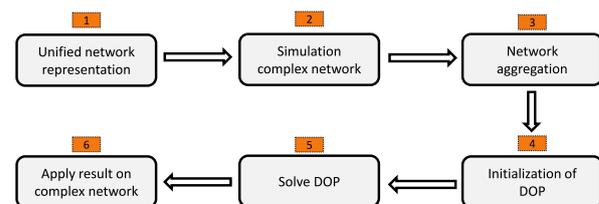
supply water temperatures and pressure differences for all customers were introduced based on real network limits to satisfy the customers' demand. The minimization of supply water temperature and pressure difference for the production unit mimics the situation in a real plant where low temperatures and differential pressures are desirable in order to reduce heating and pumping costs. The weights are chosen such that a low temperature is given a higher priority than pressure minimization, as this is the relatively larger cost in reality. The optimization constraints are inequality constraints defined using the `min` and `max` variable attributes.

## 4 Framework

### 4.1 Overview

A schematic view of the framework is presented in Figure 3 and each step is explained below.

- Step 1: The network is created using a unified network representation that includes data of the network, demand and boundary conditions.
- Step 2: The unified network representation is translated into executable Modelica code, including graphics annotations (can be read by any Modelica authoring tool). A dynamic simulation with fixed nominal control signals is performed, to get a nominal operation conditions where the aggregation will be done (Loewen, 2001).
- Step 3: The original network is aggregated to a size suitable for optimization. The aggregation depth is flexible and certain consumers can be excluded from the aggregation.
- Step 4: The aggregated network is simulated to get initial trajectories for the dynamic optimization.
- Step 5: The dynamic optimization problem is solved.
- Step 6: The optimal trajectories are applied to the original network.



**Figure 3.** Schematic view of the framework showing the different steps for network creation, simulation and optimization.

## 4.2 Modeling

### 4.2.1 District heating network representation

The core of the framework is a unified network representation implemented in Python. In general, the representation is applicable for all kinds of network-based energy systems including district heating systems, gas systems and power systems. Typically such systems consist of edges and nodes. Edges could be transmission or distribution lines, gas pipes or pipes within a district heating system. Nodes could represent consumers and producers in any energy domain, storage or hybrid technologies. Such a unified network representation is required for two reasons. Firstly, an automatically generated simulation or optimization network model based on a unified network representation reduces the effort for modeling as well as the liability for errors. Secondly, several steps require detailed information of the network topology and other steps change the topology of the network. The unified network representation consists of three central modules: network representation, aggregation of the network and translation into executable simulation/optimization code. The first two modules are independent of the actual simulation and optimization language. The library is implemented in the Python module `networkX` (Hagberg et al., 2008) that is suitable for the creation and manipulation of complex networks. The network representation is in this paper combined with models from Modelon’s Thermal Power Library suitable for dynamic simulation and optimization of district heating systems.

### 4.2.2 Aggregation method

Dynamic thermo-hydraulic optimization of large-scale district heating systems is very complex and numerically challenging. Several concepts approach the problem by simplifying (some) models (Olsthoorn et al., 2016; Orehounig et al., 2015), others by simplifying the network topology using aggregation methods (Larsen et al., 2004; Grosswindhager et al., 2012). Two methods have been developed in Denmark and Germany (Larsen et al., 2004); they are called “the Danish” and “the German” method. The idea behind the aggregation is (i) to change the tree structure of a network into a line structure and (ii) to remove short branches. The German method can handle network topologies with loops as well; this was the reason why we implemented the German method in our framework. Both methods were originally defined for steady state operation. The methods have different starting points: The German method conserves volume, mass flow and temperatures in all nodes. Thus, heat losses from the original and the aggregated networks are not exactly the same. The Danish method conserves heat losses. Thus, the node temperatures of the original and the aggregated networks are not exactly the same. Previous works on aggregation methods show that networks can be aggregated up to a very high level even in dynamic operations without losing significant accuracy (Loewen,

2001; Larsen et al., 2004).

### 4.2.3 Generation of Modelica models for simulation and optimization

The network representations are translated into Modelica models using Python functions. Based on the information in the network, corresponding Modelica code is generated, complete with annotations to enable visual inspection of the resulting model. In Figure 4, a generated network for the Graz network is visualized in the diagram view in Dymola. This method allows for the creation of complete models for simulation or optimization with components, connect statements and parameter values defined by the network model. Apart from the actual network, the generated Modelica models intended for optimization also contain input and output connectors, to handle the control signals and delay modeling in the optimization setup.

The components of the Modelica models which are used for optimization in this project come from Modelon’s Thermal Power Library 1.14.

Both the complete district heating network described in Section 3.2 and aggregated versions of this are translated into Modelica models. The complete models are used for simulation, while the aggregated models are used for optimization and for creating initial trajectories for optimization. Different aggregation levels are evaluated in optimization, as explained in Section 5.1.

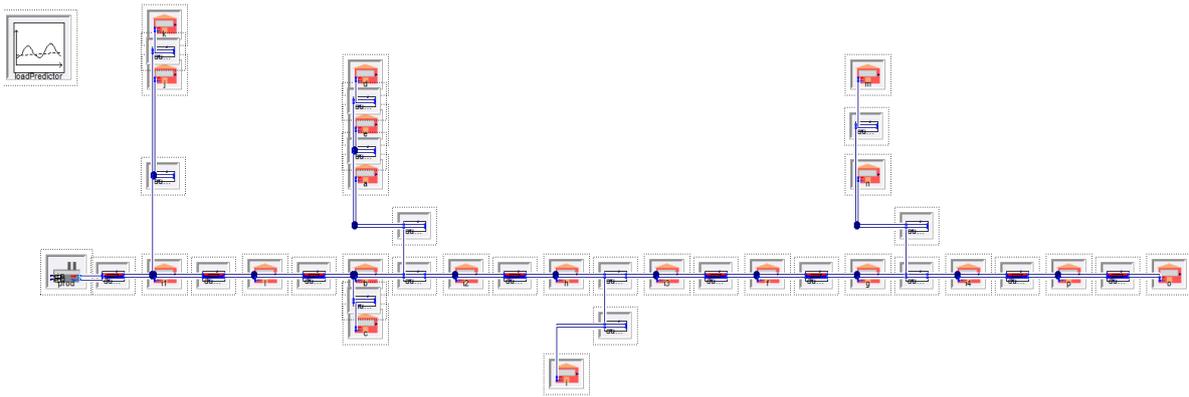
## 4.3 Optimization

The OCT toolchain that is used to solve the dynamic optimization problems starts by transferring the generated Modelica and Optimica code to CasADi Interface (Lennernäs, 2013), which has a flattened and symbolic representation of the model and optimization problem based on CasADi (Andersson, 2013). This representation is then propagated to the dynamic optimization algorithm implemented in JModelica.org (Magnusson and Åkesson, 2015). This algorithm implements direct collocation (Biegler, 2010) to transcribe the problem into a nonlinear program (NLP), which is then solved by IPOPT (Wächter and Biegler, 2006). CasADi is used to compute first- and second-order sparse derivatives using algorithmic differentiation (Griewank and Walther, 2008).

The dynamic optimization framework has recently been extended to treat delay differential-algebraic equations where the delay is fixed a priori, which is needed for the pipe models discussed in Section 3.1. Methods based on direct, local collocation are well-suited for handling such models (Betts et al., 2016).

### 4.3.1 Symbolic elimination

Before the model is transferred to CasADi Interface, the OCT compiler performs alias elimination, variability propagation and index reduction. The flattened, fully implicit differential-algebraic equation (DAE) is then transferred to CasADi Interface and later exposed to the direct



**Figure 4.** Modelica model of Graz network generated in Python from network description. The customers are represented by the orange red house icons, the pipe models are red and blue and the production unit is the gray model located furthest to the left.

collocation.

This approach leads to very large and sparse NLPs because of the multitude of algebraic variables in the network model. There has been recent work carried out (Magnusson and Åkesson, 2016) to address this problem in general by applying a block-lower triangular (BLT) transformation of the DAE to identify algebraic variables that only depend affinely on the corresponding block variables. This allows symbolic elimination of such variables by forward substitution. Further variables can be eliminated by applying tearing (Meijer, 2011; Baharev et al., 2016) to handle nonlinear dependencies. The majority of algebraic variables are thus eliminated prior to discretization by direct collocation, drastically reducing the size of the NLP. However, although the number of variables and equations are reduced, the resulting NLP Jacobian and Hessian tend to become more dense as a result, potentially crippling the performance of the sparse numerical linear algebra. A novel heuristic, similar to local minimum fill-in (Duff et al., 1986), is used to identify algebraic variables that should not be eliminated in order to preserve the sparsity of the NLP, typically leading to faster solution times.

In Modelica tools it is common to “eliminate” *all* algebraic variables by embedding Newton iterations in the right-hand side of an explicit ordinary differential equation, which is the foundation of FMI. In the spirit of simultaneous discretization, this approach is not used in OCT to avoid the long evaluation times that may result from solving implicit equations in each iteration and also the increased problem density resulting from elimination.

As demonstrated in (Magnusson and Åkesson, 2016), and as we will also see is the case in this work, the symbolic elimination not only reduces the solution time, but also improves convergence robustness, that is, probability of successfully solving an optimization problem in a timely manner.

## 5 Results

The production planning formulation described in Section 3.2 was solved for different optimization and model se-

tups. The goal is to understand the impact of the aggregation level on the production plans and of the symbolic elimination on the convergence and robustness of the optimization problem.

### 5.1 Optimization setups

The optimization problem was solved for three different aggregation levels resulting in two, five and seven customers. Very little difference in the optimal trajectories could be observed (data not shown). This indicates that aggregating the network to just two customers is sufficient to describe the current network with good accuracy.

The convergence of the optimization algorithm was also analyzed in detail for each aggregation level, to investigate the scalability of the current approach. All optimization cases were run on a laptop with 8 GB RAM and four 2.6 GHz CPUs, with convergence results and optimization model statistics displayed in Tables 1 and 2. The results show that the main benefit of the elimination occurs for larger network models, when both the time per iteration and the number of iterations is significantly reduced, resulting in a much better overall performance. The total time for running the entire script is reduced by more than a factor 2 and the optimization convergence is also significantly more robust, as indicated by the number of iterations and by manual inspection of the output from IPOPT. For fewer customers, the comparison between the two methods give less clear results. The overall time for running the script is approximately the same, as is the robustness of the convergence. The reason for the similar performance is that the time gained by eliminating variables is lost from the extra time needed to perform the elimination.

### 5.2 Optimal trajectories

The optimized trajectories are studied for the aggregated system with five customers. The network aggregation is displayed in Figure 5, and the trajectories are displayed in Figure 6. It is clear from the temperature plot that the heat loss and transport delay are correctly captured in the optimization model: the customer at the network periphery

**Table 1.** Optimization model statistics.

Nbr of customers	2	5	7
Nbr of states	12	22	30
Nbr of algebraics	136	298	413
Nbr of algebraics a.e	17	45	66
Nbr of variables in NLP	61354	130412	150159
Nbr of variables a.e	18395	39079	45712

**Table 2.** Optimization convergence results.

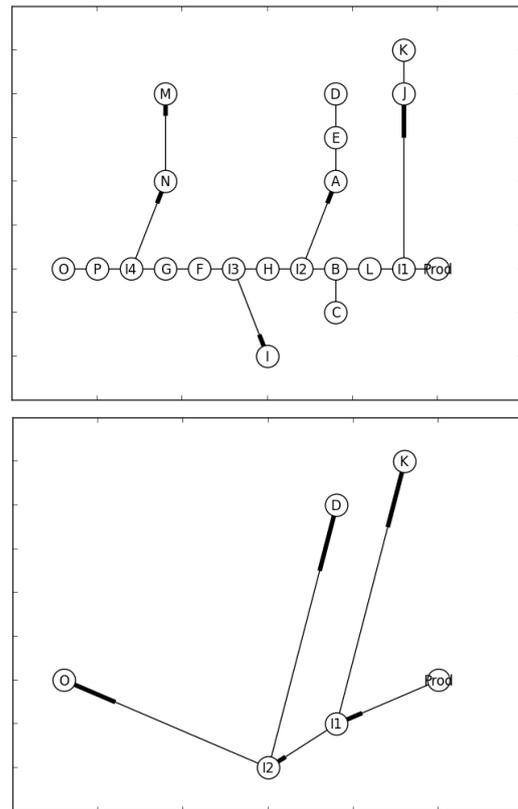
Nbr of customers	2	5	7
Without symbolic elimination			
Nbr of iterations	29	42	88
IPOPT CPU time [s]	13	163	1103
NLP function eval time [s]	24	60	161
IPOPT total time [s]	103	422	1516
Script total time [s]	120	448	1548
With symbolic elimination			
Nbr of customers	2	5	7
Nbr of iterations	31	43	44
IPOPT CPU time [s]	7	31	60
NLP function eval time [s]	25	68	71
IPOPT total time [s]	75	240	314
Script total time [s]	121	431	685

received a slightly colder water and with some delay. It is also visible that the optimization minimizes pump cost, i.e. discharge pressure at the producer, while respecting the differential pressure constraint across the customers' valve: the customer O, furthest away from the producer has its dp constraint active most of the time. During high load, the distribution pump of the producer is at its maximum capacity and the mass rate saturates. As a consequence the supply temperature is increased to fulfill the heat demand of all customers. The temperature increase is done in advance to compensate for the mass flow dependent delays in the network.

Another interesting phenomenon can be seen when customer O, far away from the producer is operating at maximum valve opening, at about  $t=2.5h$  and  $9.5h$ . Figure 7 displays this phenomenon around the first load peak. It shows that the increase in the supply temperature at the producer propagates with the flow in the network and results in valve closing at the customers close to the producer, in the figure illustrated with customer I1, which is closest to the producer. This shifts the mass flow rate from the close customers to customer O that gets its higher load fulfilled. The increase in the producer's supply temperature propagates quicker than the speed of the hot water.

### 5.3 Verification in simulation

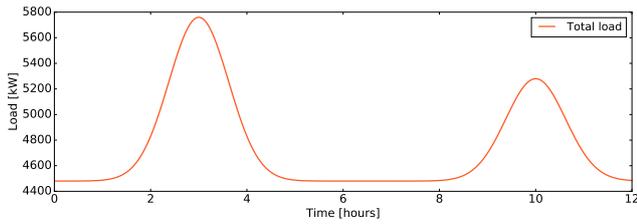
The previous section demonstrates that the optimization method is able to generate optimal trajectories for temperature and pressure that fulfill the operational constraints from the customers and the distribution network. The net-


**Figure 5.** Complete and aggregated network models.

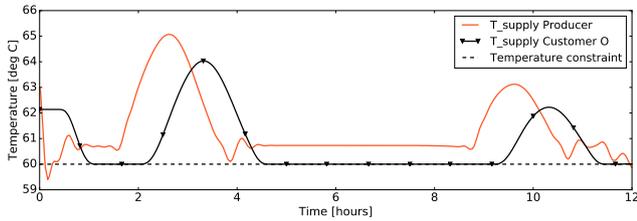
work model used for optimization differs however from the original one as it has been simplified by the aggregation method described in Section 4.2.2. The idea is now to validate the optimization results and the aggregation method by applying the optimal trajectories on the complex model with 16 customers. As the pressure profile would not be applied in reality, the supply pressure at the production unit is instead manipulated by a controller that maintains a minimum pressure difference over all customers. Only the supply temperature trajectory is applied to the complex network model. The results are shown in Figure 8. The supply temperature at the customer furthest away from the plant is very similar when the optimization and simulation results are compared. The mass flowrate computed by the differential controller is also very similar to the optimized trajectory. Some differences in the differential pressure can be seen as the optimization does not always operate at the minimum value but sometimes at a higher level to minimize the overall cost. The results indicate in general that the aggregation to two customers is good enough for this 16 customers network model.

## 6 Discussion

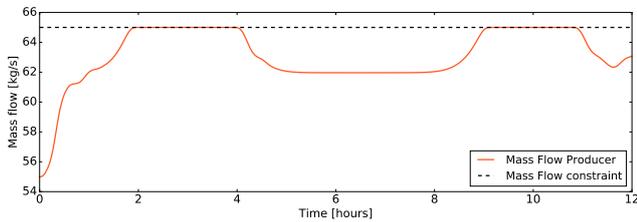
This paper presents new features of Modelon's Thermal Power Library 1.14 in the field of dynamic optimization of district heating systems as well as the impact of the new algorithm for symbolic elimination available in Optimica Compiler Toolkit. The new features together with



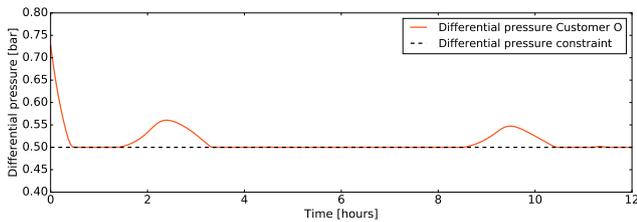
Total customer load



Production unit and Customer O feed water temperature and temperature constraint



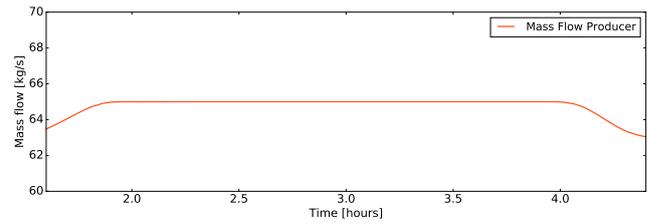
Production unit mass flow and mass flow constraint



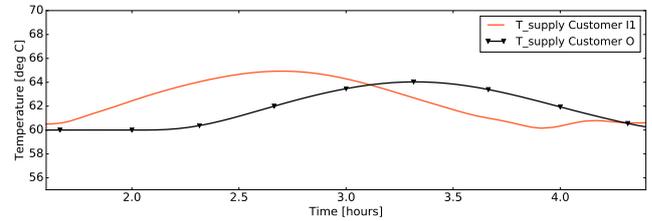
Customer O differential pressure and differential pressure constraint

**Figure 6.** Optimal trajectories for an aggregated network model with five customers.

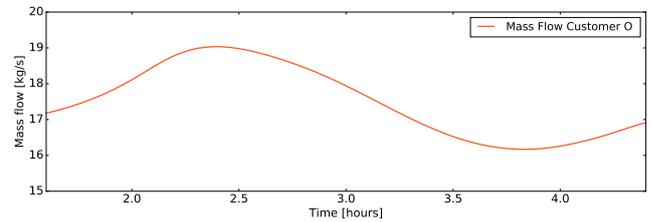
a unified representation of network-based energy systems make it possible to analyze, simulate and optimize small and larger district heating or cooling systems. It is also possible to include physical constraints based on operational limitations into the optimization formulation. Based on the results it can be concluded that the aggregation method achieves accurate results at an aggregation depth of about 90 %. Furthermore it can be concluded that the main benefit of the elimination occurs for larger models where the computation time could be reduced by more than a factor 2. Enabling the elimination yields an overall computation time for seven remaining customers of about 11 minutes and a solution time of about 5 minutes. In the context of model predictive control the solution time is sufficiently low for a real-time application and it could be further reduced by initializing the optimization with the results of the latest iteration. In an offline opti-



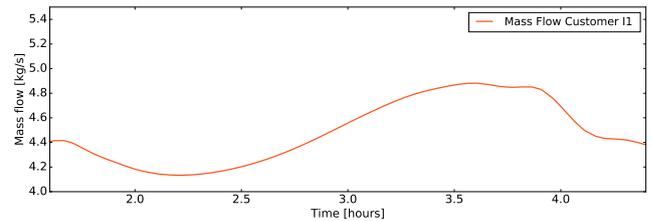
Total supply mass flow



Supply temperatures for closest and furthest Customer



Customer O mass flow



Customer I1 mass flow

**Figure 7.** Heat to customer O based on mass flow change for remaining customers.

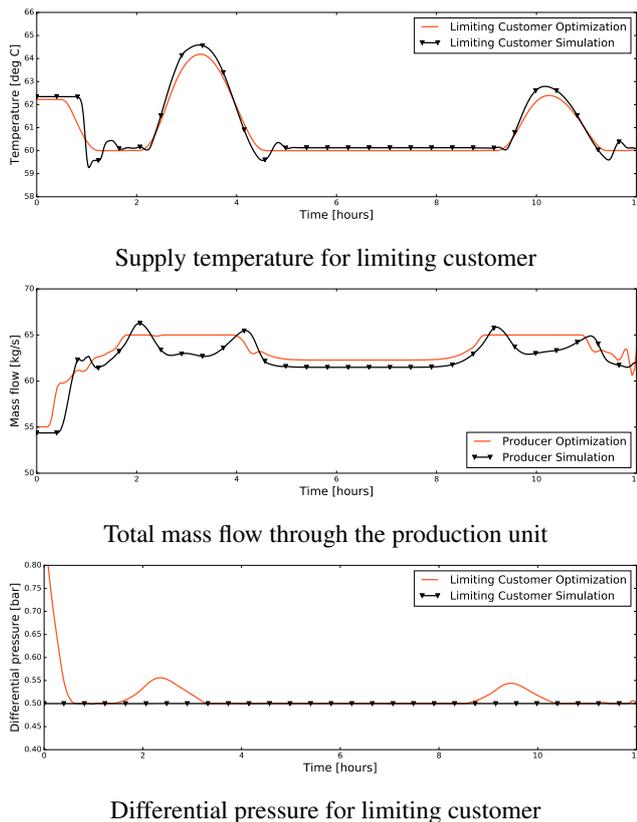
mization context, the overall computation time could also be reduced by using the optimization results for lower aggregation levels as initial guesses for the optimization of more complex networks. The next stage of our research will include scale-up studies and the integration of the unit commitment problem in the overall framework.

## 7 Acknowledgements

Fredrik Magnusson acknowledges support from the LCCC Linnaeus Center and eLLIIT Excellence Center at Lund University. Gerald Schweiger acknowledges the Austrian Federal Ministry of Science, Research and Economics for funding the project “FlexEnergySys (848346)”. Modelon AB acknowledges support from PiiA – Processindustriell IT och Automation.

## References

Johan Åkesson. Optimica—an extension of Modelica supporting dynamic optimization. In *Proceedings of the 6th Interna-*



**Figure 8.** Comparison between optimization results for two customers and simulation results for the complete network with optimal inputs.

*tional Modelica Conference*, pages 57–66, 2008.

Johan Åkesson, Karl-Erik Årzén, Magnus Gäfvert, Tove Bergdahl, and Hubertus Tummescheit. Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering*, 34:1737–1749, 2010.

Jonas Allegrini, Kristina Orehounig, Georgios Mavromatidis, Florian Ruesch, Viktor Dorer, and Ralph Evins. A review of modelling approaches and tools for the simulation of district-scale energy systems. *Renewable and Sustainable Energy Reviews*, 52:1391–1404, 2015. URL <http://dx.doi.org/10.1016/j.rser.2015.07.123>.

Joel Andersson. *A General-Purpose Software Framework for Dynamic Optimization*. Ph.D. thesis, Arenberg Doctoral School, KU Leuven, Belgium, 2013.

Ali Baharev, Hermann Schichl, and Arnold Neumaier. Decomposition methods for solving sparse nonlinear systems of equations. Submitted for publication. Available online: [http://reliablecomputing.eu/baharev\\_tearing\\_survey.pdf](http://reliablecomputing.eu/baharev_tearing_survey.pdf), 2016.

John T. Betts, Stephen L. Campbell, and Karmethia C. Thompson. Solving optimal control problems with control delays using direct transcription. *Applied Numerical Mathematics*, 108:185–203, 2016.

Lorenz T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. MOS-SIAM, Philadelphia, PA, 2010.

Iain S. Duff, Albert. Erisman, and John K. Reid. *Direct Methods for Sparse Matrices*. Clarendon Press, Oxford, United Kingdom, 1986.

Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, PA, 2nd edition, 2008.

Stefan Grosswindhager, Andreas Voigt, Martin Kozek, and A Varying-coefficient Models. Predictive Control of District Heating Network using Fuzzy DMC. In *International Conference on Modelling, Identification and Control*, 2012.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, (SciPy):11–15, 2008.

Helge V Larsen, Benny Bøhm, and Michael Wigbels. A comparison of aggregated models for simulation and operational optimisation of district heating networks. *Energy Conversion and Management*, 45:1119–1139, 2004.

Björn Lennernäs. A CasADi based toolchain for JModelica.org. M.Sc. thesis, Department of Automatic Control, Lund University, Sweden, 2013.

Achim Loewen. *Entwicklung eines Verfahrens zur Aggregation komplexer Fernwärmenetze*. Ph.D. thesis, Fraunhofer UMSICHT, Germany, 2001.

Henrik Lund, Sven Werner, Robin Wiltshire, Svend Svendsen, Jan Eric Thorsen, Frede Hvelplund, and Brian Vad Mathiesen. 4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems. *Energy*, 68:1–11, 2014. URL <http://www.sciencedirect.com/science/article/pii/S0360544214002369>.

Fredrik Magnusson and Johan Åkesson. Dynamic optimization in JModelica.org. *Processes*, 3(2):471–496, 2015.

Fredrik Magnusson and Johan Åkesson. Symbolic elimination in dynamic optimization based on block-triangular ordering. *Optimization Methods and Software*, 2016. Accepted for publication.

Patrik Meijer. Tearing differential algebraic equations. M.Sc. thesis, Centre for Mathematical Sciences, Lund University, Sweden, 2011.

Modelica Association. Modelica® - A Unified Object-Oriented Language for Systems Modeling Language Specification Version 3.3 Revision 1. 2014. URL <https://www.modelica.org/documents/ModelicaSpec33Revision1.pdf>.

Modelon. OPTIMICA Compiler Toolkit, 2016. URL <http://www.modelon.com/products/optimica-compiler-toolkit/>.

- Dave Olsthoorn, Fariborz Haghghat, and Parham A Mirzaei. Integration of storage and renewable energy into district heating systems : A review of modelling and optimization. *Solar Energy*, 136:49–64, 2016. URL <http://dx.doi.org/10.1016/j.solener.2016.06.054>.
- Kristina Orehounig, Ralph Evins, and Viktor Dorer. Integration of decentralized energy systems in neighbourhoods using the energy hub approach. *Applied Energy*, 154:277–289, 2015. URL <http://dx.doi.org/10.1016/j.apenergy.2015.04.114>.
- Jonatan Rantzer. Robust production planning for district heating networks. M.Sc. thesis, Centre for Mathematical Sciences, Lund University, Sweden, 2015.
- Håkan Runvik, Per-Ola Larsson, Stéphane Velut, Jonas Funquist, Markus Bohlin, Andreas Nilsson, and Sara Modarrez Razavi. Production Planning for Distributed District Heating Networks with JModelica.org. In *11th International Modelica Conference*, pages 217–223, 2015.
- Gerald Schweiger, Per-Ola Larsson, Fredrik Magnusson, Patrick Lauenburg, and Stéphane Velut. District heating and cooling systems – framework for modelica-based simulation and dynamic optimization. *Energy*, 2017a. ISSN 0360-5442. doi:<https://doi.org/10.1016/j.energy.2017.05.115>. URL <http://www.sciencedirect.com/science/article/pii/S0360544217308691>.
- Gerald Schweiger, Jonatan Rantzer, Karin Ericsson, and Patrick Lauenburg. The potential of power-to-heat in swedish district heating systems. *Energy*, 2017b. ISSN 0360-5442. doi:<http://dx.doi.org/10.1016/j.energy.2017.02.075>. URL <http://www.sciencedirect.com/science/article/pii/S0360544217302499>.
- Stéphane Velut, Per-Ola Larsson, Johan Windahl, Linn Saarinen, and Katarina Boman. Short-term production planning for district heating networks with JModelica.org. In *Proceedings of the 10th International Modelica Conference*, pages 959–968, 2014.
- Andreas Wächter and Lorenz T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.