

Optimisation-friendly modelling of thermodynamic properties of media

Marcus Åberg



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
ISSN 0280-5316
TFRT--6011

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2016 by Marcus Åberg. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2016

ABSTRACT

The use of dynamic modelling and simulation of thermal power systems has grown recent years and as the performance of computers have increased optimisation of such systems has become possible. When performing such optimisations, there are high numerical requirements on the model used in the system. Due to the nonlinear nature of thermodynamic properties of working fluids, extra consideration has to be made when modelling these. In this thesis such a media model has been implemented through polynomial approximation. It has been shown that such an implementation can be used in optimisation algorithms yielding reliable results. A benchmarking optimisation test case has been constructed in order to test the implementation's accuracy and performance. The physical models were implemented in Modelica used for dynamic optimisation using the JModelica.org platform. Furthermore it has also been benchmarked against existing media models designed for simulation in complex thermal power systems in order to verify the accuracy of the new model of thermodynamic properties of media.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Johan Windahl and Håkan Runvik at Modelon AB for their great advice and support. They have always been willing to help and have given me valuable input as well as helped me plan the work. I would also like to thank Fredrik Magnusson at the department of automatic control at LTH, he has come with great input along the way and his expertise on dynamic optimisation has been very valuable when analysing and figuring out new ways of taking the project forward. Lastly I would like to thank Modelon AB and all of its employees for giving me the chance to work with this thesis. It has been a great experience to get an insight into what it is to work like a simulation engineer.

CONTENTS

1	INTRODUCTION	9
1.1	Background	9
1.2	Purpose	9
1.2.1	Dymola and Modelica	10
1.2.2	JModelica.org and OCT	10
1.3	Outline	10
2	THERMODYNAMICAL EQUATIONS OF STATE	11
2.1	Basic thermodynamic properties and concepts	11
2.2	The state concept	12
2.3	Properties in the two-phase region	12
2.4	The concept of fundamental functions.	13
2.5	The IF97 standard	13
3	EVALUATION OF NUMERICAL TECHNIQUES	15
4	POLYNOMIAL FITTING THROUGH LEAST SQUARES	17
4.1	Weighted least squares	18
5	DYNAMIC OPTIMISATION PROBLEMS	19
5.1	Problem Formulation	19
5.2	Collocation method	20
6	IMPLEMENTING THE MEDIA MODEL	23
6.1	Regions	24
6.2	The smooth step function	24
6.3	Polynomial approximations	26
6.4	Accuracy of implemented media functions	30
6.4.1	Temperature	30
6.4.2	Density	31
6.4.3	Specific entropy	32
6.5	Implementing the Modelon.Media interface	33
7	OPTIMISATION BENCHMARKING CASE: START-UP OF A HEAT RECOVERY STEAM GENERATOR (HRSG)	35
7.1	Description of the HRSG model	35
7.1.1	Optimisation problem formulation	37
7.2	Initialisation	38
7.3	Optimisation	39
8	BENCHMARKING CASE: SIMULATION MODELS	45
8.1	Case 1: Heat recovery steam generator	45
8.1.1	Description of model	46
8.1.2	Result from simulation	46
8.2	Case 2: Rankine Cycle	49
8.2.1	Description of model	49
8.2.2	Result from simulation	50
9	CONCLUSION	53
9.1	Outcome	53
9.2	Future work and possible improvements	54
9.3	Experiences from the work	54

8 Contents

Bibliography 57

INTRODUCTION

The demand use of dynamic modelling and simulation of thermal power systems has grown recent years as computers have grown more powerful and are able to handle these kind of simulations making optimisation of such systems possible. These kind of calculations are very valuable when designing and evaluating new systems and can also be used in optimising controllers and analyse transient behaviours. When performing such optimisation there are high requirements on the numerical properties of the model. Due to the nonlinear nature of thermodynamic properties of working fluids, the modelling design of these properties is crucial to allow efficient optimisation and yield reliable results.

1.1 BACKGROUND

The master thesis has been written as a cooperation between Modelon AB in Lund and the The Department of Automatic Control at the technical faculty of Lund University (LTH).

Thermodynamic fluid properties of working fluids define the achievable baseline accuracy for fluid system simulation and optimisation. Developed Modelica libraries for steam tables, following the IF97 standard, have resulted in a widespread use of the Modelica technology in the energy industry. But the implementation is targeting simulation and not optimisation. Fluid properties that are used in derivative-based optimisation have higher demands on the numerical properties such as higher smoothness requirements.

1.2 PURPOSE

The aim of this thesis is to investigate whether such models can be implemented in the modelling language Modelica in other ways in order to make it possible for it to be used in optimisations. Furthermore such an implementation can perhaps also make the simulation faster. This of course at a cost of accuracy.

In this thesis only pure media will be considered. Water is the media that will be considered as it is the most frequently used working fluid in thermal power simulations. However making the implementation as general as possible will allow for easy implementation of other fluids. Furthermore transport properties will not be modelled as the implementation today already is smooth and will probably not need any modification in order to work in optimisation algorithms.

The new model should also implement the existing Modelon.Media interface so the fluid later can be used easily in existing applications for validation.

1.2.1 *Dymola and Modelica*

Dymola is a simulation tool based on the Modelica programming language. Modelica is an object-oriented programming language designed for modelling systems from several different areas of physics and integrate such systems with each other. One of the benefits of the language is that it makes use of component-based programming, making it easy to reuse models in larger, more complex systems. This property makes it very suitable for simulating thermal power systems, which often includes numerous components from different fields of physics such as electronics, thermodynamics and controllers.

1.2.2 *JModelica.org and OCT*

JModelica.org is an open-source platform used for making optimisations of Modelica models. It can be used to optimise control parameters or estimate physical parameters among other things. The commercial complement to JModelica.org is OCT (Optimica Compiler Toolkit) with extensive features.

1.3 OUTLINE

First, an investigation of the alternative methods to approach the problem will be carried out. This in order to identify a suitable numerical method to be used to approximate a pure 2-phase fluid that is intended to be used in simulation and optimisation.

The second step is to implement the pure 2-phase fluid in the modelling language Modelica and implement the Modelon.Media interface so the fluid later can be used in existing applications for validation.

The third and last step will be the evaluation and validation of the fluid implementation in JModelica.org and Dymola by looking at performance, accuracy and robustness and by benchmarking against existing Modelica fluid implementations.

THERMODYNAMICAL EQUATIONS OF STATE

When simulating thermodynamical systems, for example in simulation of thermal power systems such as those investigated in this thesis, there is need for modelling the thermodynamic properties of the different media that are used in the processes. This chapter aims at explaining these equations and how they are calculated in the Modelica IF97 implementation.

Equations of state are relations between different state variables in thermodynamics. If for example the pressure and specific enthalpy are known, the aim of the equations is to be able to calculate all other properties from these values.

2.1 BASIC THERMODYNAMIC PROPERTIES AND CONCEPTS

In this section a few basic thermodynamical concepts and properties will be explained and described in order to clarify and simplify the description of the thermodynamical equations of state.

An important concept when discussing thermodynamic properties is the internal energy U , which is the total sum of energies contained in the considered system excluding external influences such as kinetic energy due to motion of the system relative to the environment. It is the energy that is connected to the thermodynamical state of the system.

It is often convenient to use a corresponding quantity that is defined per unit of mass (kg) of the system. This quantity is referred to as specific internal energy, and is denoted with the lowercase u . All properties denoted with the word specific are defined per unit of mass of the system, such as specific volume $v = \frac{V}{m}$ etc. Another energy measurement of a thermodynamical system is the specific enthalpy, h , which is defined as the sum of the specific internal energy of the system and the energy required to make room for the system's pressure p and (specific) volume v .

$$h = u + pv \tag{2.1}$$

The specific entropy s is a property that is connected to the second law of thermodynamics [1]. It is a quantity that describes the amount of microscopic configurations of the thermodynamical system. One of the implications of the second law of thermodynamics is that an isolated system may never lose entropy.

2.2 THE STATE CONCEPT

The thermodynamical state of a closed system is decided by its thermodynamic properties such as temperature, density, pressure etc. It is known that most of these properties are not independent from the others, and hence most properties can be calculated from the values of others. The properties used for calculating the others are called *state variables* as they can be used to determine all properties of the system considered. These variables consequently have to be independent of each other. The substance considered in this thesis is pure water, which state can be described by two independent properties. These two could for example be chosen as pressure and temperature, pressure and specific enthalpy or density and temperature [1].

2.3 PROPERTIES IN THE TWO-PHASE REGION

When a liquid changes phase to vapour and vice versa, the media exists as a mixture between saturated liquid and saturated vapour. Saturated in this case means that the liquid cannot raise its enthalpy at a fixed pressure without changing phase and vice versa for the vapour lowering its enthalpy. To raise the enthalpy some of the fluid must change its phase. This region is referred to as the two-phase region in this thesis as the media exists in two phases at the same time [2].

To analyse such a mixture properly, the proportions of vapour and liquid must be known. To do this the mass fraction x is defined as the ratio between the mass of the saturated vapour (m_v) and the total mass of the mixture (m_{tot}).

$$x = \frac{m_v}{m_{tot}} \quad (2.2)$$

In the two-phase region only the amount of saturated liquid and vapour changes in the mixture and not its properties, hence the saturated mixture can be treated as a combination of two subsystems. Since the masses of the two different subsystems usually are not known, it is convenient to assume that the two phases are well-mixed and treat them as a homogeneous mixture. Then the thermodynamic properties of the mixture will simply be the average properties. If we consider an isolated system with this saturated mixture the average specific volume of the media v_{avg} can be calculated via the specific volumes of the saturated liquid v_l and vapour v_v . The total volume of the system is V_{tot} and is the sum of the volumes occupied by vapour V_v and liquid V_l .

$$V_{tot} = V_v + V_l \quad (2.3)$$

$$\rightarrow m_{tot}v_{avg} = m_vv_v + m_lv_l \quad (2.4)$$

$$\rightarrow v_{avg} = \frac{m_vv_v + (m_{tot} - m_v)v_l}{m_{tot}} \quad (2.5)$$

$$\rightarrow v_{avg} = xv_v + (1 - x)v_l \quad (2.6)$$

$$(2.7)$$

In this way the mass fraction can be calculated as

$$x = \frac{v_{avg} - v_l}{v_v - v_l} \quad (2.8)$$

In a similar manner the mass fraction can be calculated from the enthalpy as

$$x = \frac{h_{avg} - h_l}{h_v - h_l} \quad (2.9)$$

These relations are used when for example density is to be calculated in the two phase region with the pressure p and specific enthalpy h as input. From p the saturated properties h_v , h_l , v_v and v_l are calculated. From h_v and h_l the mass fraction x can be determined and from x , v_v and v_l the average specific volume v_{avg} can be calculated. The average density is the inverse of this value, $\rho_{avg} = 1/v_{avg}$.

2.4 THE CONCEPT OF FUNDAMENTAL FUNCTIONS.

The idea of the fundamental functions is that they provide a complete description of the thermodynamical state of the considered media. A fundamental function is a function that has as input two independent state variables, and by differentiation and other operations, all other properties of the state of the media can be calculated. When pure substances with only two independent state variables are to be described (as in the case of this thesis, where pure water is considered) one of these forms can be used for this purpose [1].

$$u = u(s, v) \quad (2.10)$$

$$h = h(s, p) \quad (2.11)$$

$$\phi = \phi(T, v) \quad (2.12)$$

$$g = g(T, p) \quad (2.13)$$

Here, ϕ is the Helmholtz function and g the Gibbs function defined as

$$\phi = u - Ts \quad (2.14)$$

$$g = h - Ts \quad (2.15)$$

These two functions have the same unit as u and h (J/kg), and can thus be considered energies. Therefore they are often referred to as the Gibbs and Helmholtz free energies. From these functions all other properties can be determined by combination and differentiation. For example, the specific internal energy u can be calculated by the definition of the Helmholtz energy as $u = \phi + Ts$. From this relation the isochoric heat capacity c_v can be determined by differentiating the expression for u , $c_v = (\partial u / \partial T)_{v,}$ and like this one could go on for all properties. For a full table on how to relate the different thermodynamic properties from the fundamental functions, see [3].

2.5 THE IF97 STANDARD

The existing media implementation for water in Modelica today is based upon steam tables following the standard from IAPWS (International Association for the Properties of Water and Steam) Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, short IF97. For complete documentation of this standard, see

[3]

The IF97 consists of a set of equations defined over 5 different regions. In these different regions, different fundamental functions (see section 2.4) are used to calculate the thermodynamic properties. These are the Gibbs free energy $g(T, p)$ (three of the five regions), the Helmholtz free energy $\phi(T, v)$ and a saturation-pressure equation $p_s(T)$ (both over one region each). The standard also provides inverse functions, which allows for calculating the temperature, pressure and density if these quantities are not known.

To give an idea how these functions work, the fundamental equation for region 1 in the standard is stated, the Gibbs free energy. The equation is given in dimensionless form $\gamma(\tau, \pi) = g(T, p)/RT$, where R is the specific gas constant for ordinary water, $\tau = T/T^*$ is a dimensionless temperature and $\pi = p/p^*$ is a dimensionless pressure.

$$\frac{g(T, p)}{RT} = \gamma(\tau, \pi) = \sum_{i=1}^{34} n_i (7.1 - \pi)^{I_i} (\tau - 1.222)^{J_i} \quad (2.16)$$

Here I_i, J_i, n_i all are coefficients used for defining the steam table. From this equation all thermodynamic properties can be calculated. Some of these properties are listed in table 1.

Property	Relation	IF97
Specific volume	$v = (\partial g / \partial p)_T$	$v(\tau, \pi) \frac{p}{RT} = \pi (\partial \gamma / \partial \pi)_\tau$
Specific internal energy	$u = g - T(\partial g / \partial T)_p - p(\partial g / \partial p)_T$	$\frac{u(\tau, \pi)}{RT} = \tau (\partial \gamma / \partial \tau)_\pi - \pi (\partial \gamma / \partial \pi)_\tau$
Specific entropy	$s = -(\partial g / \partial T)_p$	$\frac{s(\tau, \pi)}{R} = \tau (\partial \gamma / \partial \tau)_\pi - \gamma$
Specific enthalpy	$h = g - T(\partial g / \partial T)_p$	$\frac{h(\tau, \pi)}{RT} = \tau (\partial \gamma / \partial \tau)_\pi$

Table 1: Some thermodynamic properties and their relation to the Gibbs free energy along with the IF97 formulation.

The current implementation of the IF97 standard in Modelica makes use of a "state" formulation. This means that every time a media function of the media model is called upon, the whole state of the media is calculated. This data is stored in a state variable, which is used as input in the media functions. This has the advantage that practically all media properties of the considered state is calculated at once, making new media calls computationally very cheap as the property already has been calculated. This is very beneficial in simulation of large systems where many media functions are called upon and where many components of the model share the same media state.

EVALUATION OF NUMERICAL TECHNIQUES

The main issue with existing two-phase media models already implemented in Modica when used in optimisation algorithms is the discontinuities at the phase borders. These are as of today modelled with if-statements, deciding which working region the fluid is in. These statements triggers events in the solver. An event is something that changes the properties of the system (i.e. the equations) and the solver therefore has to re-initiate the system in order to handle the discontinuity. When such models are used in optimisation algorithms, they can cause the solver to converge slowly or not at all. The theoretical requirement for convergence in considered optimisation algorithms in this thesis are that functions called in the model must be twice continuously differentiable (C^2) [4]. Therefore, only models fulfilling this requirement will be considered.

The main ideas for approximation of the model are polynomial approximation through regression defined over regions, and the use of piecewise defined polynomials (known as splines).

Polynomial approximation over different regions has the main advantage that it is smooth over the defined region, the main challenge here is to find a way to accurately and smoothly connect the different regions. In a first implementation step functions is used to make a smooth transition between the regions. These can be defined so that the functions are twice continuously differentiable and the smoothness requirement hence is fulfilled. Furthermore there has been successful attempts implementing optimisable media functions like this before, suggesting it is a possible way to go.

Splines have the advantage that they allow for much more accurate evaluation. However the connection between different splines at the nodes may cause problems in simulation and optimisation. Furthermore depending on how the implementation is done the splines can allocate a lot of working memory. Function values and first and second derivatives in several directions have to be stored or calculated upon call for each grid-point which can add up to a significant amount of memory space, alternatively slowing down the media calls.

There has already been made some attempts with implementing splines, and there has been issues with the implementation creating oscillations. These oscillations gives rise to inaccuracies but perhaps more importantly they may cause simulations to run slow creating performance issues [5].

Based on previous experience from the different kinds of implementation, it is decided that polynomial approximation defined over regions will be the method investigated in this thesis.

POLYNOMIAL FITTING THROUGH LEAST SQUARES

The form of the polynomial fits that will be used in this thesis is as follows.

$$p(x_1, x_2) = \sum_{i=0}^k \sum_{j=0}^k b_{ij} x_1^i x_2^j \quad (4.1)$$

$$p(x) = \sum_{i=0}^k b_i x^i \quad (4.2)$$

Here k is the order of the polynomials.

The method used for approximating polynomials to the data sets is least-squares which fits a linear model to the specified data set. Linear in this case means that the fit is a linear combination of the monomials.

The objective of the least-square scheme is to minimise the square of the sum of the residuals S . The residual r_i of a data point is defined as

$$r_i = y_i - p(x_i, \mathbf{b}) \quad (4.3)$$

$$S = \sum_{i=1}^m r_i^2 \quad (4.4)$$

where y_i is the value of the function to be approximated corresponding to point x_i , \mathbf{b} the coefficients of the polynomial, p is the fitted polynomial and m is the number of data points. This can be written with matrix notation as

$$S = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \|\mathbf{y}\|^2 - 2\mathbf{y}^T \mathbf{X}\mathbf{b} + \mathbf{b}^T \mathbf{X}^T \mathbf{X}\mathbf{b} \quad (4.5)$$

where each row in the $m \times n$ matrix \mathbf{X} represents a data point and each column represents the term of the polynomial to be multiplied with the corresponding coefficient in the column vector \mathbf{b} . \mathbf{y} is the column vector corresponding to the values y_i . Since the Hessian of S , $\frac{\partial^2 S}{\partial \mathbf{b}^2} = \mathbf{X}^T \mathbf{X}$ is positive semidefinite the objective function is convex. Hence the minimum can be calculated through differentiating the objective and setting the partial derivatives to zero. In this way n equations are received, where n is the number of coefficients.

$$\frac{\partial S}{\partial \mathbf{b}} = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X}\mathbf{b} = \mathbf{0} \quad (4.6)$$

The resulting equation system, often referred to as the normal equations, can be written as follows.

$$\mathbf{X}^T \mathbf{y} = \mathbf{X}^T \mathbf{X}\mathbf{b} \quad (4.7)$$

Assuming $X^T X$ has full rank, the normal equations are solved for \mathbf{b} by

$$\mathbf{b} = (X^T X)^{-1} X^T \mathbf{y} \quad (4.8)$$

4.1 WEIGHTED LEAST SQUARES

If weights are used in the least-squares regression, certain data points can be given a greater importance in the fitting process. The expression for the residuals becomes

$$S = \sum_{i=1}^m w_i r_i^2 = \sum_{i=1}^m W_i r_i^2 \quad (4.9)$$

Where w_i is the weight associated with data point i and is the i th diagonal element of matrix \mathbf{W} . After this step the same procedure as above follows, the objective S is differentiated with respect to \mathbf{b} and minima is found where the gradient is zero. The least squares equation for the coefficient estimates then becomes

$$\mathbf{b} = (X^T \mathbf{W} X)^{-1} X^T \mathbf{W} \mathbf{y} \quad (4.10)$$

For more documentation and information about the curve fitting tool in MatLab, see documentation [6].

DYNAMIC OPTIMISATION PROBLEMS

The framework used for making the optimisations in this thesis is the open source Modelica platform JModelica.org. It uses Modelica-models to describe the dynamics of the systems and uses the Modelica extension Optimica [7] to transfer the problem into the optimisation formulation. From there the optimisation problem is given a symbolic representation with the aid of the open-source tool CasADi [8]. This representation can be transformed into a nonlinear program (NLP) by the use of different discretisation techniques. There are several ways of doing this but the method used and discussed in this thesis is going to be direct collocation. The NLP is later solved by IPOPT [9] using analytic derivatives from CasADi.

The type of dynamic optimisation problem (DOP) that will be considered in this thesis is finding optimal trajectories, using offline settings. These problems have three main features. The objective for the solver is to

- Find the optimal control variables that minimises the objective function of the problem over the prescribed time frame.
- Make sure all model equations are satisfied.
- Ensure that the problem constraints are fulfilled.

For a more thorough description of how dynamic optimisation works in JModelica.org than this thesis presents the reader is referred to [10].

5.1 PROBLEM FORMULATION

The systems considered in this thesis are described by differential-algebraic equation (DAE) systems. DAE systems are a class of differential equations that includes algebraic constraints, i.e non-differential equations. A DAE can be put generally on the form

$$F(\dot{x}(t), x(t), y(t), u(t), p) = 0, \quad \forall t \in [t_0, t_f] \quad (5.1)$$

where $x(t)$ are the differential variables ($\dot{x}(t)$ the corresponding time derivatives), $y(t)$ are the algebraic variables, $u(t)$ the control variables and p are free parameters of the system. $t \in [t_0, t_f]$ is the time variable defined over the considered time frame. To simplify the notation, the variable $z = (\dot{x}(t), x(t), y(t), u(t))$ is introduced such that the DAE system can be described by $F(z(t), p) = 0$. Initial conditions associated with the DAE system are denoted by $F_0(z(t_0), p) = 0$.

The DAE system describes the dynamics of the system, in practical use this is represented by the Modelica model associated with the DOP.

The DOP can then be stated as

$$\min_{x(t),y(t),u(t),t_0,t_f,p} \phi(t_0, t_f, z(t), z_T, p) \quad (5.2)$$

subject to

$$F(z(t), p) = 0 \quad F_0(z(t_0), p) = 0 \quad (5.3)$$

$$z_L \leq z(t) \leq z_U \quad p_L \leq p \leq p_U \quad (5.4)$$

$$g_e(t_0, t_f, t, z(t), z_T, p) = 0 \quad g_i(t_0, t_f, t, z(t), z_T, p) \leq 0 \quad (5.5)$$

$$G_e(t_0, t_f, t, z(t), z_T, p) = 0 \quad G_i(t_0, t_f, t, z(t), z_T, p) \leq 0 \quad (5.6)$$

ϕ is the objective function of the DOP, which is to be minimised. The main purpose of the objective is to penalise deviations from a desirable reference trajectory but can also be used to minimise the rate of change of the control input etc. z_L , z_U , p_L and p_U represents upper and lower bounds on the system variables. g_e and g_i are path constraints and G_e and G_i are point constraints only enforced at certain points in time. The new variable z_T corresponds to the value of $z(t)$ at certain time points, corresponding to the points where the point constraints are enforced.

In order to solve the DOP with IPOPT, there are continuity requirements on all of the functions involved. They all need to be twice continuously differentiable with respect to any of the DOP variables. This is of special importance to this thesis since the model used to describe the dynamics of the system is included in the function F describing the DAE. This implies that if a media model is used it also has to be twice continuously differentiable, which is the aim of this thesis to investigate.

5.2 COLLOCATION METHOD

As stated before, a discretisation of the infinite-dimensional DOP is needed in order to convert the problem into a finite-dimensional NLP. This is done by the use of local direct collocation. The main idea of local direct collocation is to divide the time horizon into n_e elements, and approximate the system variables with low order polynomials over each element.

In each element the system variable z is approximated by a polynomial, $z(t) = z_i(t)$, $\forall t \in [t_i, t_{i+1}]$. These polynomials are defined by n_c collocation points, which also determines the order of the polynomial. At these collocation points the approximated polynomials for the system variables are enforced to satisfy the DAE. This is done practically by the use of Lagrange basis polynomials l_k , which are constructed in a way that they equal one at the associated collocation point and equals zero at all the others. In this way the system variables in element i can be expressed as

$$z_i(t) = \sum_{k=1}^{n_c} z_{i,k} l_k(t), \quad \forall t \in [t_i, t_{i+1}] \quad (5.7)$$

Where $z_{i,k}$ is the value of the variable at collocation point k and l_k is the associated Lagrange basis polynomial. In this way $z_i(t_{i,k}) = z_{i,k}$ where $t_{i,k}$ is the time correspond-

ing to collocation point k in element i . In this way approximations for the x , y and u -variables can be obtained. The approximations for the time derivatives \dot{x} can be deduced by differentiating the expression for x .

The collocation points might be chosen in many ways, of which Gauss and Radau has support in JModelica.org. The Radau scheme chooses one point at the endpoint of each element and the rest in a way that numerical accuracy is maximised. To enforce continuity for the differential variables x , an extra collocation point is added for these variables at the start of each element. Since there is also a collocation point at the end of the previous element (because of the Radau scheme) this will ensure continuity.

With this discretisation of the problem, the problem can be transcribed into a NLP where all of the system variables $z_{i,k}$ is chosen as decision variables along with t_0 and t_f if they are free as well as the free parameters p . The form of the resulting NLP is

$$\min_x f(x) \tag{5.8}$$

subject to

$$x_L \leq x \leq x_U \tag{5.9}$$

$$g(x) = 0 \tag{5.10}$$

$$h(x) \leq 0 \tag{5.11}$$

Where equation 5.9 - 5.11 are bounds and constraints. The variable x in equation 5.8-5.11 is not to be confused with the differential variable of the initial DOP. This is merely a formulation of a general NLP, where x represents the decision variables of the program.

This thesis will not go further into the details on how to solve the resulting NLP. The interested reader is referred to [9].

 IMPLEMENTING THE MEDIA MODEL

The approach chosen was to approximate the thermodynamical functions with polynomials over different operating regions in the p - h , p - s , and d - T plane. These approximations are then connected via smooth step functions (see section 9.2) from one region to another. In that way the functions are twice continuously differentiable over the whole working regime.

The functions that were implemented in the model can be divided into 1D-functions and 2D-functions. 1D-functions are typically functions that describes the saturated behaviour of the media in the two-phase region. Temperature, bubble enthalpy and dew enthalpy are quantities that can be calculated directly from the pressure at saturation. 2D-functions on the other hand are functions, as the name suggests, that needs two inputs. In the following table 2 all of the implemented functions with descriptions are defined.

Function name	Input	Output
temperature_ph	p, h	T
density_ph	p, h	ρ
x_ph	p, h	x
pressure_dT	ρ, T	p
specificEnthalpy_dT	ρ, T	h
specificEnthalpy_pT	p, T	h
specificEnthalpy_ps	p, s	h
specificEntropy_ph	p, h	s
velocityOfSound_ph	p, h	h
specificHeatCapacityCp_ph	p, h	C_p
specificHeatCapacityCv_ph	p, h	C_v
Tsat_p	p, h	T_{sat}
psat_T	T	p_{sat}
hsub_p	p	h_{sub}
hdew_p	p	h_{dew}
ddph_ph	p, h	$\frac{\partial \rho}{\partial p} _h$
ddhp_ph	p, h	$\frac{\partial \rho}{\partial h} _p$
ddhp_ph	p, h	$\frac{\partial \rho}{\partial h} _p$

Table 2: All implemented thermodynamic property functions

6.1 REGIONS

The regions are referred to as liquid, vapour and two-phase region. The liquid and regions are divided into sub- and super-critical areas. The region of a certain point is decided by its p and h values. The regions are defined according to table 3.

$h < h_{bub}$	liquid
$h > h_{dew}$	vapour
$h_{bub} < h < h_{dew}$	two-phase
$p < p_{crit}$	subcritical
$p > p_{crit}$	supercritical

Table 3: Region division

Figure 1 shows the phase diagram in the $p-h$ plane with all of the regions marked out.

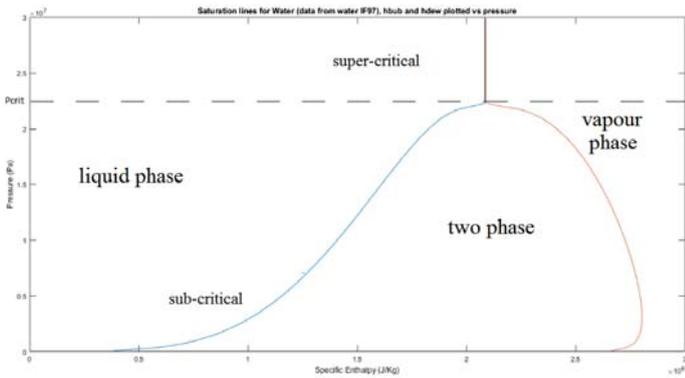


Figure 1: Phase diagram of water, saturation lines is drawn with approximated functions. Regions are divided into super- and sub-critical for both liquid and vapour phase.

There are some exceptions from this division, for example the specific entropy function, which could be approximated using one single region. The reason for this is that some properties does not change so "dramatically" when crossing the phase borders. Furthermore it was noticed later in the process that accurate media calls were needed for very low pressures, why a super low pressure region were added to the functions `density_ph`, `hdew_p`, `hbub_p`, `Tsat_p` and `ddew_p`.

6.2 THE SMOOTH STEP FUNCTION

The method used for making a smooth transition between regions in the media implementation in this thesis is via smooth step functions. The idea is to multiply the

polynomial defining the function over a certain region with a function so that the polynomial assumes its own value within the region and goes to zero outside this specific region. For the implementation considered this function has one input variable and will be referred to as $S(x), x \in \mathbb{R}$. The desired properties of S are

$$S(x) = \begin{cases} 0, & x \leq 0 \\ p(x), & 0 < x < 1 \\ 1, & x \geq 1 \end{cases} \quad (6.1)$$

The right- and left borders of the step has been chosen to 1 and 0 for easy implementation and the scaling can be done when calling the function by scaling the parameter x .

Since the overall goal with this implementation is to make the media implementation twice continuously differentiable the step function S must also be C^2 . To make a smooth connection between the left- and right borders of equation 6.1, the generic 5th order polynomial, $p(x)$ can be used.

$$p(x) = \sum_{i=0}^5 a_i x^i \quad (6.2)$$

The task is now to chose the coefficients a_i properly. From 6.1 there are the following constraints on $p(x)$.

$$p(0) = 0, \quad p(1) = 1 \quad (6.3)$$

$$p'(0) = 0, \quad p'(1) = 0 \quad (6.4)$$

$$p''(0) = 0, \quad p''(1) = 0 \quad (6.5)$$

$$(6.6)$$

These constraints results in the equation system

$$a_5 + a_4 + a_3 + a_2 + a_1 + a_0 = 1 \quad (6.7)$$

$$5a_5 + 4a_4 + 3a_3 + 2a_2 + a_1 = 0 \quad (6.8)$$

$$20a_5 + 12a_4 + 6a_3 + 2a_2 = 0 \quad (6.9)$$

$$a_2 = a_1 = a_0 = 0 \quad (6.10)$$

$$(6.11)$$

The solution of this equation system is

$$a_5 = 6, a_4 = -15, a_3 = 10, a_2 = a_1 = a_0 = 0 \quad (6.12)$$

And $S(x)$ is hence defined as

$$S(x) = \begin{cases} 0, & x \leq 0 \\ 6x^5 - 15x^4 + 10x^3, & 0 < x < 1 \\ 1, & x \geq 1 \end{cases} \quad (6.13)$$

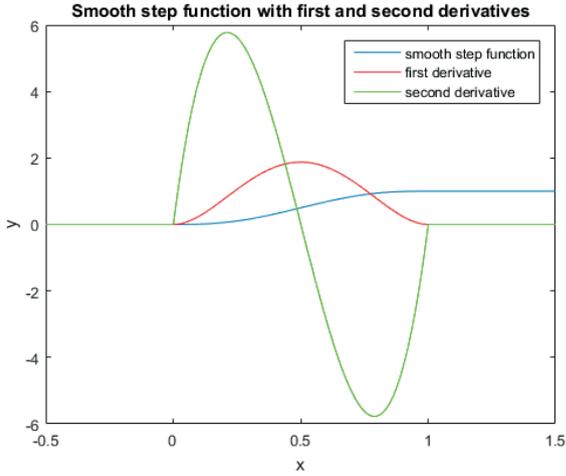


Figure 2: Smooth step function with its first and second derivative.

The plot of $S(x)$ and its first and second derivative can be viewed in figure 2.

When defining which region the step function is active two parameters are needed, the activation point x_{edge} and the active region Δ_x . These parameters can then be adjusted to the needs of the two functions that are to be connected by calling the smooth step function by $S(\frac{x-x_{edge}}{\Delta_x})$. See figure 3.

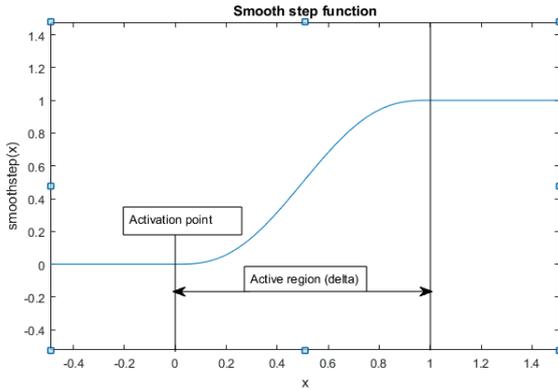


Figure 3: Smooth step function activation point and active region.

6.3 POLYNOMIAL APPROXIMATIONS

The method used for doing the polynomial approximations was polynomial regression as described in chapter 4. For this purpose MatLab's own built-in curve fitting

tool was used to construct the actual fits. The data needed for making the polynomial fits was extracted from the Water IF97 standard already implemented in Dymola. This is of course a simplification as the IF97 standard also is an approximation. But since the model has been proven accurate [3] and the aim of this project is to analyse whether it is possible to make new implementations optimisable, it is a reasonable simplification to make.

The grid in the p - h plane that was used for data extraction was 100×100 points and linear along the h -axis with range $[1.0e5, 4.0e6]$ (J/kg). A logarithmic scale was used for the p -axis according to following formula: $\pi = 1.06215^{i-95} p_{crit}$ making the pressure range $[7.6245e4, 2.9827e7]$ (Pa). Here p_{crit} being the critical pressure of water and i ranges from 1–100. The logarithmic scale was used because of the nature of the media at small pressure levels. Properties such as temperature and density change more rapidly at small p -values. The table data for the 2D-functions was extracted at the grid points shown in figure 4. As mentioned earlier in section 6.1 there also added a low pressure region for certain media functions, this grid added data points in the range $[1e3, 1e5]$ to extend the valid area for function calls.

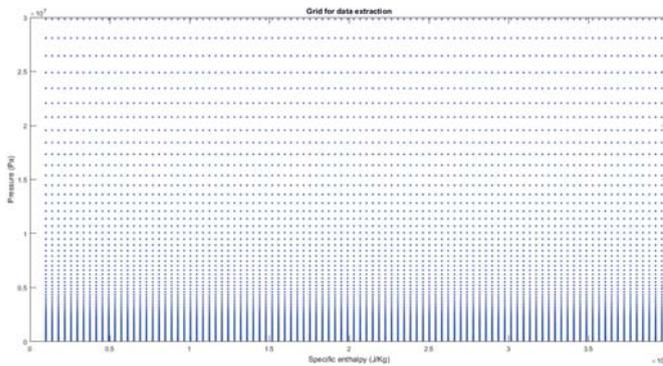


Figure 4: Grid used for extracting data to surface fits.

For 2D-functions that use d , T and s as inputs the response data from IF97 for constructing the functions that calculates these properties from p and h was used instead. This was done since it is hard to construct a grid that does not contain points outside of the domain of definition for these properties.

It is of extra importance that the polynomial fits have high accuracy close to the borders to other regions, since they are to be connected to another polynomial function there. Big differences in the values of the different surfaces close to the border will lead to a "leap" in the function value at the border. Even though the step function smooths this leap out and make sure the function is continuous it is of extra importance that this difference is made as small as possible since the model is to be used in optimisation algorithms which can get stuck at inconsistencies like this. The approach used for handling this problem is to give data points at the borders between

different regions a higher weight to make the linear regression generate polynomials which are accurate at the border. There is however a setback with this approach, if the data points are weighted to hard then polynomial "overshoots" might occur in the rest of the region. This makes it hard to use this approach for certain data sets as the accuracy inside the region becomes unacceptable when trying to force the polynomial to coincide with the data points on the border of the region.

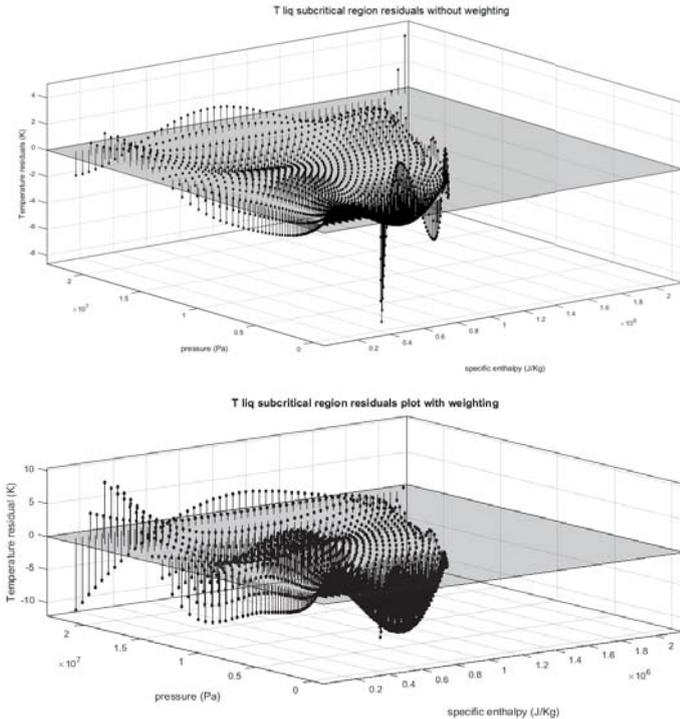


Figure 5: Residual plot of weighted fit (below) and residual plot of unweighted fit (above)

As can be seen in the unweighted plot in figure 5, the error is larger on the border than inside the region, but still much smaller than the error close to the border for the weighted fit. This is due to the weighting of this fit heavily emphasises the residual at the region border, making errors in the rest of the region less significant. This method was used only when this phenomenon did not cause large residuals inside the considered region.

Another use of the weighting is to be able to make the least-square algorithm minimise the relative errors instead of absolute. This is of use since some of the approximated functions ranges largely in value. Data points which has small response reference values (close to zero) will get very large relative errors if weighting is not performed. But by adding weights proportional to the inverse of the data value at ev-

ery point, points with low values will be given larger weights and hence this problem can be dealt with. With notation from chapter 4, the weight for data point y_i from 4.9 are defined as

$$w_i = 1/y_i^2 \quad (6.14)$$

Hence, the objective function S can be written as

$$S = \sum_{i=1}^m w_i r_i^2 = \sum_{i=1}^m \left(\frac{y_i - p(x_i, \mathbf{b})}{y_i} \right)^2 \quad (6.15)$$

The impact of this in the relative error can be viewed in figure 6. This has a big impact in the region for the density function as the density values in this region ranges from values very close to zero up to a couple of hundred (Kg/m^3).

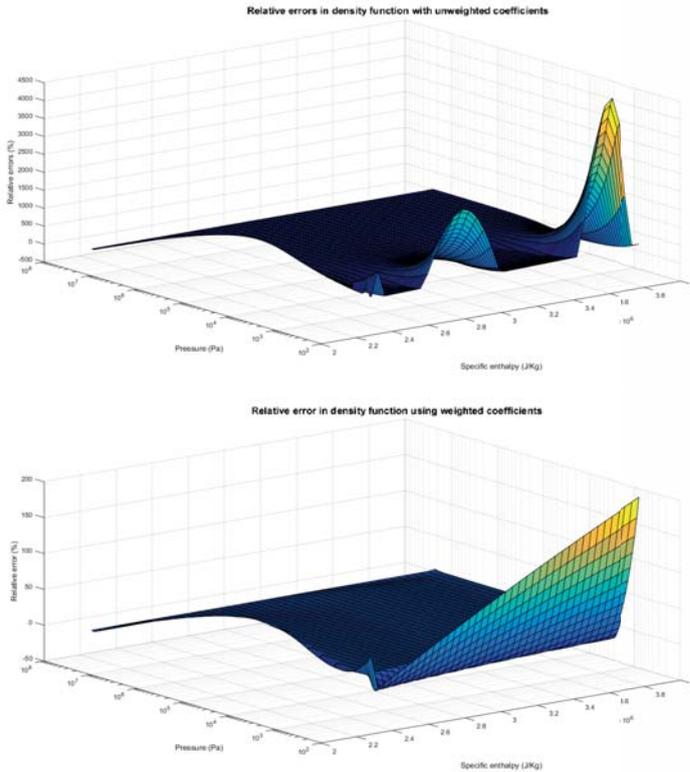


Figure 6: Weighted (below) versus unweighted relative error plot for the density function in the region.

There is still a rapid increase in relative error for the weighted coefficients at very low pressures, but it appears at much lower pressures and is much smaller than for the weighted coefficients.

6.4 ACCURACY OF IMPLEMENTED MEDIA FUNCTIONS

The resulting approximations of the thermodynamical-property functions will be presented in this section. Since there are so many functions that have been implemented, a few important examples will be accounted for in this section.

6.4.1 Temperature

The temperature function has a maximum relative error of around 0.8% according to figure 7. The red line in the figure represents the phase border.

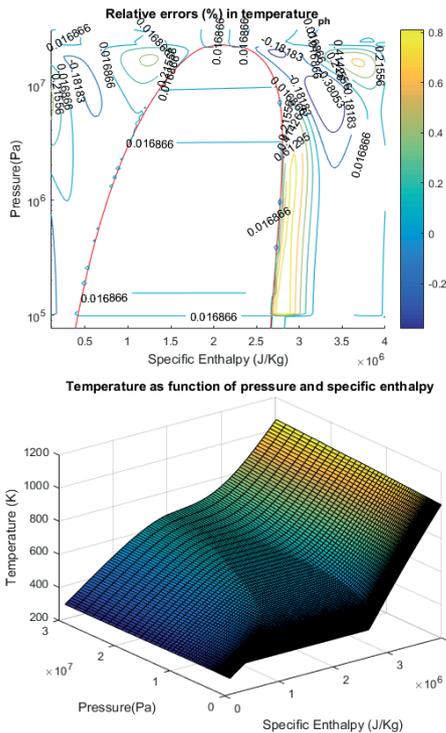


Figure 7: Contour plot of relative errors (above) and the approximated temperature function (below).

There are a couple of interesting things to note from the relative error plot. At low pressures and high specific entropies there is a distinct drop in the relative error. This

is due to the fact that a new region was added for sub 1 bar pressures in the vapour region. This was due to problems when simulating models including components such as condensers which operates at very low pressures. Another thing to note is the that the highest relative error is located after the phase border at the vapour side. The coefficients here have been weighted in a way to be consistent with the saturated properties at the phase border. This weighting might cause this bulge as the least squares-algorithm prioritises minimising the error at the border instead of inside the region but with the benefit of better consistence at the phase border.

6.4.2 Density

The density function can be viewed in figure 8. As can be seen, there is a "spike" in the relative error around the critical point, this is due to that 3 regions are connected there, and the value there is an interpolation between the function approximation in all three regions.

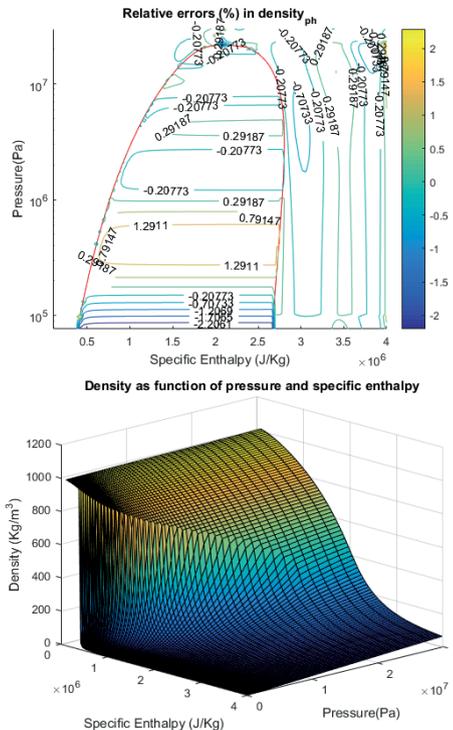


Figure 8: Contour plot of relative errors (above) and the approximated density function (below).

The same behaviour at low pressures in the vapour region can be seen in the relative error plot as in the temperature function due to the same reasons, an added region at low pressures. The spike in relative errors is due to the fact that 3 regions meet at the critical point. If the density function is compared with the temperature function, which has a similar point, it can be seen that the temperature function is rather flat at the critical point where the density is rather steep. This might lead to the inconsistencies between the different regions that can be observed in figure 8

6.4.3 Specific entropy

The function approximation and the relative errors of the specific entropy function can be seen in figure 9. There is no regions since the specific entropy of the media does not change dramatically with phase changes. This makes it possible to use one single region for evaluation of specific entropy for all pressures and specific enthalpies.

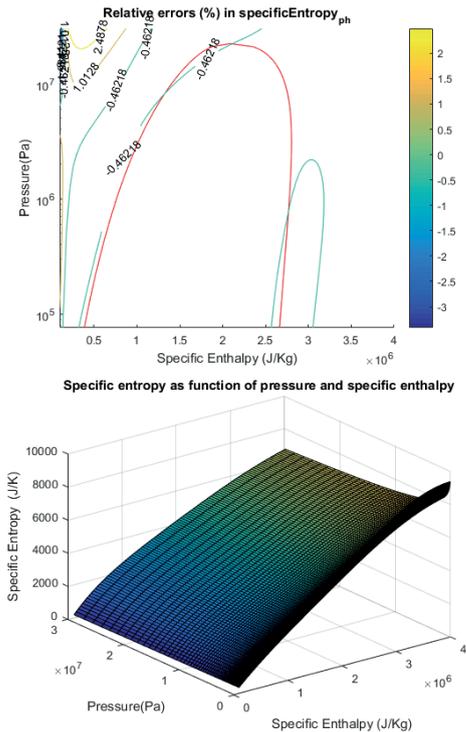


Figure 9: Contour plot of relative errors (above) and the approximated specific entropy function (below).

This approximation was made with the logarithm of the pressure values. The approximated function is a polynomial of the logarithm of the pressure and the specific

entropy. The figure showing the function shape shows clearly that there is a very rapid (exponential) increase in specific entropy at low pressures. Logarithmic pressure values smooths this rapid slope. This made it easier to get a good fit, but at the expense that the logarithm of the pressure must be calculated every time this function is called which adds to the computational time when the media model is used in simulations.

6.5 IMPLEMENTING THE MODELON.MEDIA INTERFACE

The function approximations was combined into a Modelica package that implements the Modelon.Media interface. The most significant feature of this implementation is the state record formulation as described in section 2.5. However it proved to be computationally disadvantageous to to "set a state" (calculate most thermodynamic properties at once) every time a media function is called. The reason for this is that the different media functions are not sharing the same computations in the same way they do in the IF97 formulation (where all properties are derived from a fundamental function). Hence it was decided to alter the implementation in a way that this state formulation was removed and the thermodynamic properties are calculated directly upon call of the related media function.

 OPTIMISATION BENCHMARKING CASE: START-UP OF A HEAT RECOVERY STEAM GENERATOR (HRSG)

For testing the implementation in optimisation applications, a model describing a start-up phase of a heat recovery steam generator (HRSG) has been chosen.

7.1 DESCRIPTION OF THE HRSG MODEL

The model used in this thesis has been built upon a model developed for a tutorial, for further information and material from this tutorial please see [11]. A similar model and optimisation problem is investigated thoroughly in a master thesis previously written in cooperation with Modelon [12].

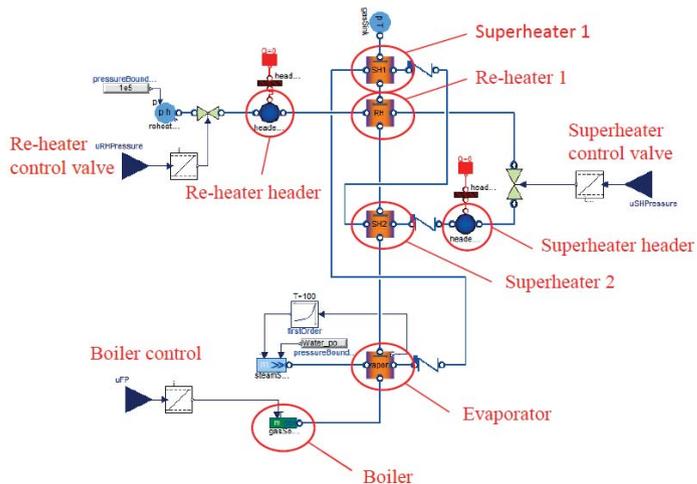


Figure 10: Original HRSG model.

The main working components of the model is a series of heat exchangers (two superheaters and one reheater), transferring the heat from a flue gas source to the water medium until the steam reaches a desired working state. The flue gas is lead into an evaporator where the water is evaporated into steam. A feedback loop connected to the evaporator keeps the water level in the evaporator on a constant level. From here,

the steam goes through two superheaters where the steam pressure and temperature increases in order to reach the desired working levels. After superheater 2 the steam is collected in a superheater header, where in reality the pipes are collected and the steam can be redirected into a turbine step. There is also a wall model connected to the header. One of the main issues with the start-up phase of the power plant is the exposure of thermal stresses in the components, and the wall models allows for the modelling of these stresses. There is a valve located after the header which can be used to control the temperature and pressure inside the header. When the steam has reached high enough temperature and pressure it can instead of going through the control valve, be redirected into a turbine step. In the turbine step pressure and temperature drops. To maximise energy output of the plant, the steam is therefore lead into a reheater step where the steam is reheated once more. After going through the reheater header it could once again be lead through another turbine step. Again a control valve is added to be able to control the pressure and the temperature inside the header.

The controllable inputs of the model is the valve position of the two control valves located after superheater 2- and the reheater header. These two can be used to control the pressure and temperature inside the headers and consequently can be used to limit the thermal stresses inside the header walls. As can be seen from figure 10 integrator steps are added to the control inputs. This was done in order to be able to put constraints on the rate of change of the optimising input signals.

This model was altered in order to be able to test more of the implemented media functions. The components in the original model only made use out of the temperature functions, so density dependency was added to the valve models and a turbine step was added which make use of the entropy functions. In the end, functions that are used in the model can be seen in table 4.

Function:	In component:
<code>temperature_ph</code>	Heat exchangers, evaporator, headers
<code>hdew_p</code>	Evaporator
<code>density_ph</code>	Turbine, valves
<code>specificEnthalpy_ps</code>	Turbine
<code>specificEntropy_ph</code>	Turbine

Table 4: Used media property functions in the HRSG optimisation.

The turbine step was added after the superheater control valve. This is not how it would be done in an actual plant where steam is not going through the turbine step without having the right quality first, making the actual simulation results from this model have less practical meaning. However, it can be used to test the robustness and properties of the optimisability of the model including the implemented media functions. A diagram of this altered model is shown in figure 11.

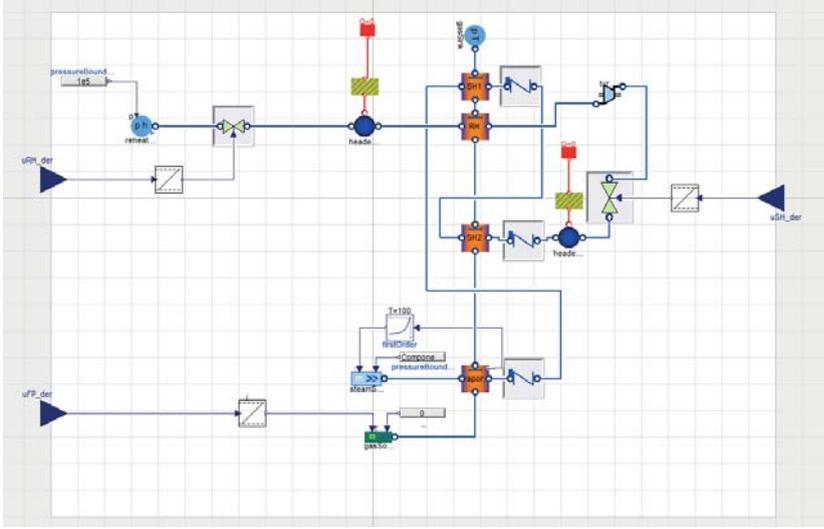


Figure 11: Altered HRSG Modelica model.

7.1.1 Optimisation problem formulation

The aim during the start up is to take the plant from the initial operating point to another operating point. The preferred properties of this process to reach this point are:

- Control the system from the initial operating point to a point where the steam in the plant has high enough quality to be redirected to turbines.
- The thermal stresses inside the header walls should be limited to extend the lifespan of the components.
- The controllable inputs have rate of change constraints which must be obeyed.

The optimal control problem defined over the time interval $[0, t_f]$ is therefore stated as

$$\begin{aligned} \min \int_0^{t_f} & w_{T_{SH2}} (T_{SH2} - T_{SH2ref})^2 + w_{p_{SH2}} (p_{SH2} - p_{SH2ref})^2 + \\ & w_{p_{RH}} (p_{RH} - p_{RHref})^2 + w_{SH2v} \dot{u}_{SH2v}^2 + w_{RHv} \dot{u}_{RHv}^2 + w_b \dot{u}_b^2 dt \end{aligned} \quad (7.1)$$

subject to

$$F(x, \dot{x}, u, w) = 0 \quad (7.2)$$

$$dT_{SH2} < dT_{SH2}^{max} \quad (7.3)$$

$$dT_{RH} < dT_{RH}^{max} \quad (7.4)$$

$$|\dot{u}_{SH2v}| < \dot{u}_{SH2v}^{max} \quad (7.5)$$

$$|\dot{u}_{RHv}| < \dot{u}_{RHv}^{max} \quad (7.6)$$

$$|\dot{u}_b| < \dot{u}_b^{max} \quad (7.7)$$

$$(7.8)$$

The first three terms of the objective function corresponds to the penalties on temperature and pressure deviations from the desired values inside the heat exchangers (same as in the headers). T_{SH2} and p_{SH2} is the temperature and pressure inside superheater 2 (T_{SH2ref} and p_{SH2ref} the desired values), p_{RH} the pressure inside the reheater. w are the corresponding weights. The last three terms represents the derivative of the control inputs, \dot{u}_{RHv} is the reheater valve signal, \dot{u}_{SH2v} the superheater 2 signal and \dot{u}_b the boiler control signal.

$F(x, \dot{x}, u, w)$ are the model equations describing the system. dT_{SH2} is the thermal gradient in the superheater 2 header wall and dT_{RH} is its counterpart in the reheater header wall. The last three constraints put upper and lower limits on the derivatives of the three control signals.

As mentioned earlier, integrator steps has been added to the control inputs of the model, which makes the control input to the system the time derivative of the control signal. This forces the control signal to be continuous as well as it makes it easy the user to put constraints directly on to the time derivative of the control signals by just constraining the input signal.

7.2 INITIALISATION

The phase of the start-up considered starts after the plant has reached a certain working level, where the system is already warm. This is achieved by simulating the plant for 5000 s with constant control inputs. The state of the plant model after this initialisation simulation will be used as initial state for the optimisation. The considered time frame for the start-up will be 2000 s. After this initial heat up phase of the optimisation run, a good initial guess trajectory of the problem is needed for robust convergence of the algorithm. This is due to the fact that the problem is not convex and hence the optimisation algorithm used in JModelica.org only finds local optimums (see chapter 5). The initial guess trajectory is achieved by simulating the plant over the time horizon of the optimisation, controlling the reheater valve (see figure 10 and 11) so that the pressure in the reheater p_{RH} assumes the reference value p_{RHref} via the control variable \dot{u}_{RHv} . The other control variable derivatives, \dot{u}_{SH2v} and \dot{u}_b are kept constant. The tolerance used for the initialisation simulation was 10^{-6} using the CVode solver.

As can be seen from the initial guess trajectories from the model in figure 12, the pressure in the reheater assumes the reference value after a period of time. This mean that it might be used as a good initial guess for the actual optimisation. Note the other two control variables, "SH valve position" (corresponding to u_{SH2v}) and "T gas" (u_b), which has been given constant time derivatives and does not change with the state of the system.

Since the initialisation phase of the optimisation only run the simulation over a period of time with predefined control inputs and does not involve any optimisation algorithms, a comparison with the reference media, Water IF97 was made. This corresponds to the dotted line in figure 12. The initial guess trajectories from the Water-IF97 simulation seems to match trajectories from the newly implemented medium well.

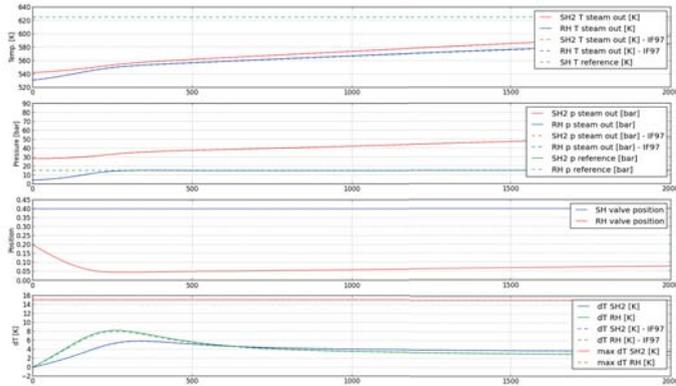


Figure 12: Initial guess trajectories from the optimisation of the HRSG model from figure 11, compared with initial guess trajectories using media functions from Water IF97.

7.3 OPTIMISATION

Once the initial operating point and initial guess trajectory have been calculated, the actual optimisation can be performed. The time horizon is divided into 12 elements, using 4 collocation points in each element. The placement of the element grid points are located in such a way that they are closer together in the first part of the time horizon, in order to better capture the transient behaviour at the beginning of the start-up. 3/4 of the elements are located in the first 3/8 of the time horizon and 1/4 in the last 5/8.

There were compatibility issues when using the Modelon.Media interface in the Modelica extension for optimisation OCT, why the media functions used in the model were called upon as separate functions. This has the disadvantage that it is harder to interchange different media models in the model.

The transferred optimisation problem had the following characteristics (output from IPOPT).

```

Number of nonzeros in equality constraint Jacobian...: 13087
Number of nonzeros in inequality constraint Jacobian.: 0
Number of nonzeros in Lagrangian Hessian.....: 2205

Total number of variables.....: 4204
    variables with only lower bounds: 512
    variables with lower and upper bounds: 2494
    variables with only upper bounds: 98
Total number of equality constraints.....: 4060
Total number of inequality constraints.....: 0
    inequality constraints with only lower bounds: 0
    inequality constraints with lower and upper bounds: 0
    inequality constraints with only upper bounds: 0

```

The optimisation statistics from the solving of the problem.

```

Number of Iterations....: 29

                                (scaled)                (unscaled)
Objective.....: 6.7577879472577536e+001  6.7577879472577536e+001
Dual infeasibility.....: 1.3691127508359579e-011  1.3691127508359579e-011
Constraint violation....: 2.4876372456485769e-009  6.9729052484035492e-005
Complementarity.....: 9.1006727498621969e-010  9.1006727498621969e-010
Overall NLP error.....: 2.4876372456485769e-009  6.9729052484035492e-005
    
```

```

Number of objective function evaluations      = 30
Number of objective gradient evaluations     = 30
Number of equality constraint evaluations     = 30
Number of inequality constraint evaluations   = 0
Number of equality constraint Jacobian evaluations = 30
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations    = 29
Total CPU secs in IPOPT (w/o function evaluations) = 2.320
Total CPU secs in NLP function evaluations   = 11.976
    
```

```

EXIT: Optimal Solution Found.
time spent in eval_f: 1.002 s. (30 calls, 33.4 ms. average)
time spent in eval_grad_f: 1.094 s. (31 calls, 35.2903 ms. average)
time spent in eval_g: 1.067 s. (30 calls, 35.5667 ms. average)
time spent in eval_jac_g: 2.586 s. (32 calls, 80.8125 ms. average)
time spent in eval_h: 6.337 s. (30 calls, 211.233 ms. average)
time spent in main loop: 14.42 s.
time spent in callback function: 0 s.
time spent in callback preparation: 0.015 s.
    
```

Initialization time: 63.90 seconds

```

Total time: 78.89 seconds
Pre-processing time: 0.00 seconds
Solution time: 14.49 seconds
Post-processing time: 0.50 seconds
    
```

The obtained trajectories from the optimisation can be seen in figure 13.

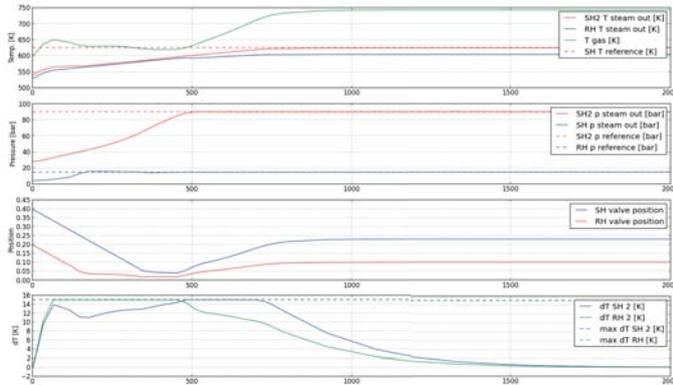


Figure 13: Optimal trajectories obtained from the optimisation problem

The optimal input signals calculated in optimisations can be exported and used in simulations. This is one way to examine if the element and collocation point division is done in an acceptable way or not, as the optimisation algorithm does not capture behaviour between the collocation points which a simulation on the other hand can. The optimal input signals were exported and used in both the model used in the optimisation algorithm and also with the medium model interchanged against Water-IF97 media functions. The trajectories for these simulations are displayed in figure 14 alongside the trajectory from the optimisation in figure 13.

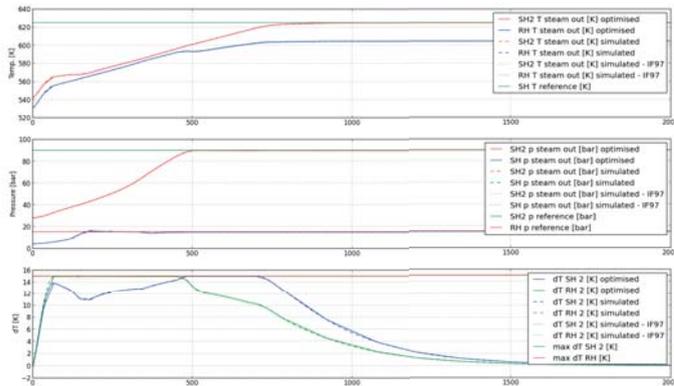


Figure 14: Simulations using the optimal input signals as input, using both new implementation and Water IF97 thermodynamic property functions.

The simulation results seem to match the optimised trajectory well, indicating that the elements have been placed in an efficient way. The simulation with the IF97 yields similar results, indicating that the media functions implemented has high enough accuracy to capture the dynamics of the model.

As mentioned in chapter 3 there is a previous implementation of a thermodynamic property model that is currently used for optimisation purposes at Modelon. These media functions were used in the same optimisation case and yielded the following statistics.

Characteristics of the optimisation problem:

Number of nonzeros in equality constraint Jacobian...	34255
Number of nonzeros in inequality constraint Jacobian...	0
Number of nonzeros in Lagrangian Hessian.....	20874
Reallocating memory for MA57: lfact (448843)	
Reallocating memory for MA57: lfact (478309)	
Total number of variables.....	10966
variables with only lower bounds:	512
variables with lower and upper bounds:	2494
variables with only upper bounds:	98
Total number of equality constraints.....	10822

```
Total number of inequality constraints.....: 0
      inequality constraints with only lower bounds: 0
      inequality constraints with lower and upper bounds: 0
      inequality constraints with only upper bounds: 0
```

Optimisation statistics:

```
Number of Iterations.....: 29

                                (scaled)                (unscaled)
Objective.....: 6.7468782487163736e+001  6.7468782487163736e+001
Dual infeasibility.....: 4.5232689244787391e-010  4.5232689244787391e-010
Constraint violation.....: 2.9809505974753847e-009  5.8559235185384757e-005
Complementarity.....: 9.7106991538623417e-010  9.7106991538623417e-010
Overall NLP error.....: 2.9809505974753847e-009  5.8559235185384757e-005

Number of objective function evaluations      = 30
Number of objective gradient evaluations     = 30
Number of equality constraint evaluations     = 30
Number of inequality constraint evaluations   = 0
Number of equality constraint Jacobian evaluations = 30
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations    = 29
Total CPU secs in IPOPT (w/o function evaluations) = 3.794
Total CPU secs in NLP function evaluations   = 5.260
```

```
EXIT: Optimal Solution Found.
time spent in eval_f: 0.51 s. (30 calls, 17 ms. average)
time spent in eval_grad_f: 0.665 s. (31 calls, 21.4516 ms. average)
time spent in eval_g: 0.497 s. (30 calls, 16.5667 ms. average)
time spent in eval_jac_g: 1.077 s. (32 calls, 33.6563 ms. average)
time spent in eval_h: 2.574 s. (30 calls, 85.8 ms. average)
time spent in main loop: 9.102 s.
time spent in callback function: 0 s.
time spent in callback preparation: 0.033 s.
```

```
Initialization time: 43.27 seconds
```

In figure 15 the trajectories from the optimisation are displayed.

From the statistics there are a couple of interesting details that can be highlighted. The total time in NLP function evaluations is roughly halved if compared to the optimisation using the model implemented in this thesis (5.260 vs. 11.976). One possible explanation for this might be that the old media implementation includes less regions than the new. In the new implementation there are polynomials for evaluating thermodynamic properties for super-critical pressures for example. Another notable thing from the translation log is the very high amount of variables that are created when using the old model. The amount of variables are more than doubled compared to the new implementation (10966 vs. 4204). This could also be a reason to the difference in computation time. What the reason is for this big difference would be interesting to investigate further, but since this was discovered late in the process it was not allowed by the remaining time-frame.

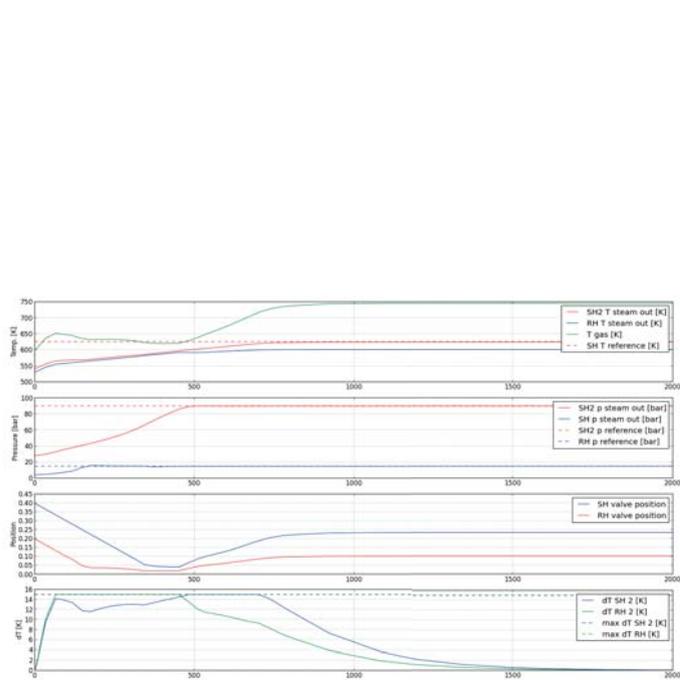


Figure 15: Optimisation using previously implemented media functions.

8.1.1 *Description of model*

The first case that was used, is an example plant describing a heat recovery steam generator (HRSG), a diagram of the model layout is demonstrated in figure 16. The HRSG is essentially a series of heat exchangers that are designed to maximise the heat recovery and supply steam at desired temperature and pressure at the steam outlet. The model also includes a control system designed to deliver steam t the outlet with desired quality and a control to keep the water level in the boiler constant.

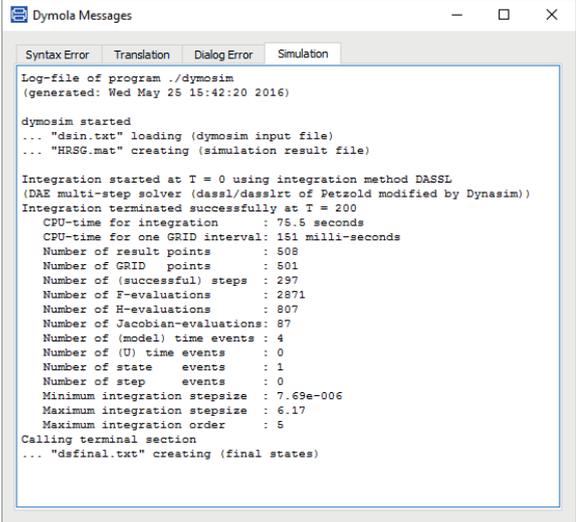
In the model, water is supplied by a water-source with prescribed temperature and pressure. It goes consequently through the economiser, drum (heated by the boiler) and superheater, heating and raising the quality of the water at each step. In the economiser the water is preheated before going into the drum by the exhaust gas. In the drum the water goes through the boiler and changes phase and goes back into the drum. The steam is let out at the outlet and lead into a superheater step further raising the temperature and quality of the steam to desired level.

In comparison with the model of an HRSG in the optimisation test case, this model includes much more complex dynamics and describes the considered system more thoroughly as it uses components designed for simulation and does not have to fulfil requirements for optimisation purposes. In comparison this model has 39 continuous time states compared to the 7 of the optimisation HRSG. For example the drum, boiler and economiser in the simulation model which models the transition of the working fluid from liquid form into vapour form is replaced by the much simpler evaporator model in the optimisation case.

8.1.2 *Result from simulation*

The simulation was done with an increasing water flow rate with a ramp from 5 to 80 seconds, where the set point for the steam output temperature increases from 400 to 450 deg C. The control system attached to the model, maintains the pressure and water levels in the drum so that steady operation is ensured.

The simulation was made both with the new media function and the standard Water-IF97 implementation, in order to investigate eventual differences and flaws in the implementation of the new media. The solver used was DASSL and the tolerance was set to 10^{-5} . The simulation statistics from the simulations can be seen in figure 17.



```

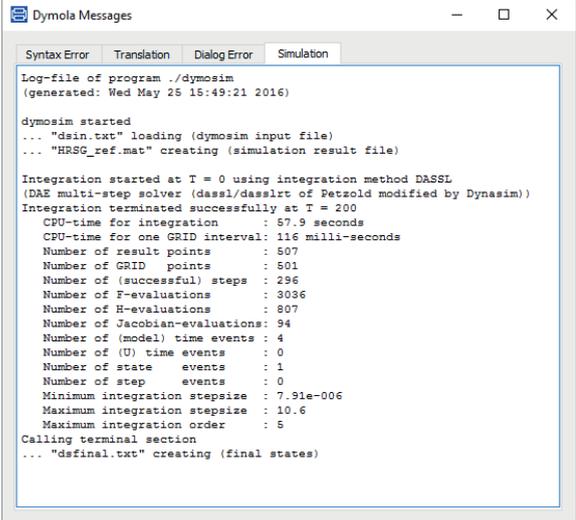
Dymola Messages
Syntax Error  Translation  Dialog Error  Simulation
Log-file of program ./dymosim
(generated: Wed May 25 15:42:20 2016)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "HRSRG.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 200
CPU-time for integration      : 75.5 seconds
CPU-time for one GRID interval: 151 milli-seconds
Number of result points      : 508
Number of GRID points       : 501
Number of (successful) steps : 297
Number of F-evaluations      : 2871
Number of H-evaluations      : 807
Number of Jacobian-evaluations: 87
Number of (model) time events : 4
Number of (U) time events    : 0
Number of state events      : 1
Number of step events       : 0
Minimum integration stepsize : 7.69e-006
Maximum integration stepsize  : 6.17
Maximum integration order     : 5
Calling terminal section
... "dsfinal.txt" creating (final states)

```

(a) New media implementation



```

Dymola Messages
Syntax Error  Translation  Dialog Error  Simulation
Log-file of program ./dymosim
(generated: Wed May 25 15:49:21 2016)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "HRSRG_ref.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 200
CPU-time for integration      : 57.9 seconds
CPU-time for one GRID interval: 116 milli-seconds
Number of result points      : 507
Number of GRID points       : 501
Number of (successful) steps : 296
Number of F-evaluations      : 3096
Number of H-evaluations      : 807
Number of Jacobian-evaluations: 94
Number of (model) time events : 4
Number of (U) time events    : 0
Number of state events      : 1
Number of step events       : 0
Minimum integration stepsize : 7.91e-006
Maximum integration stepsize  : 10.6
Maximum integration order     : 5
Calling terminal section
... "dsfinal.txt" creating (final states)

```

(b) Water IF97 media implementation

Figure 17: Simulation statistics from simulation of HRSG model with Water IF97 media implementation

The curves showing the total power transferred from the exhaust gas to the steam through all three heat exchanger stages can be displayed in figure 18

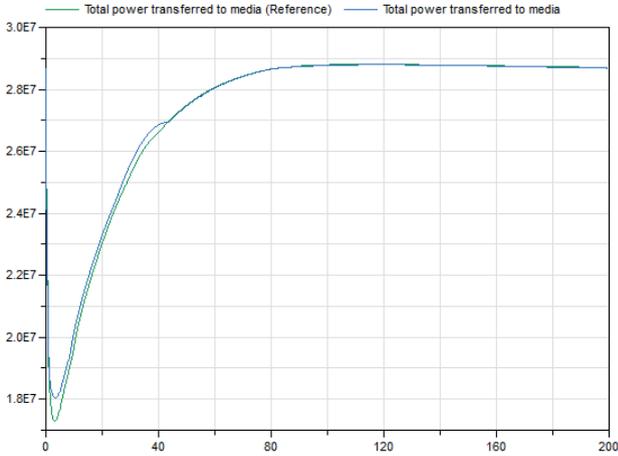


Figure 18: Total power transfer from exhaust gas to steam

The pressure level in the drum for the two different media models are displayed in figure 19

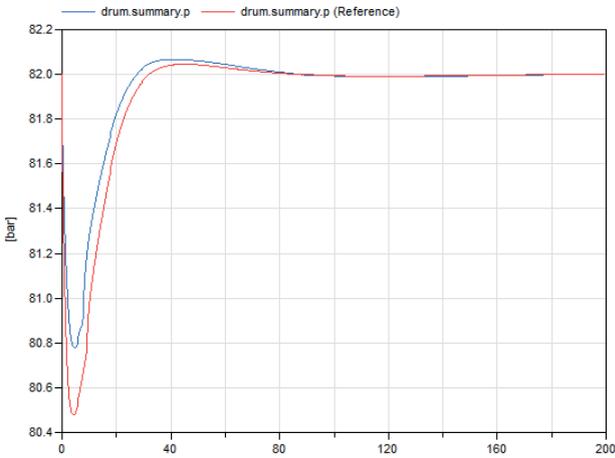


Figure 19: Pressure level in drum

From the simulation statistics it can be seen that the new media implementation is considerably slower than water IF97. There can be several reasons for this. One of the main advantages of the IF97 formulation is that most thermodynamic properties are calculated at once through the state formulation described in section 2.5. When larger systems like this are simulated there is much to gain through such an implementation compared to the one that has been used in the new implementation where the differ-

ent properties are calculated separately. Another advantage that the IF97 formulation has is that it only evaluates the fundamental function in one region compared to the new implementation that evaluates the polynomials associated to all regions in order to keep the functions C^2 .

If the trajectories from the simulation are compared, the results seem to match well. The dynamics is similar and the steady state solutions are the same. The largest deviations are at the transients. This is somewhat expected as there are controlled parameters in the model.

8.2 CASE 2: RANKINE CYCLE

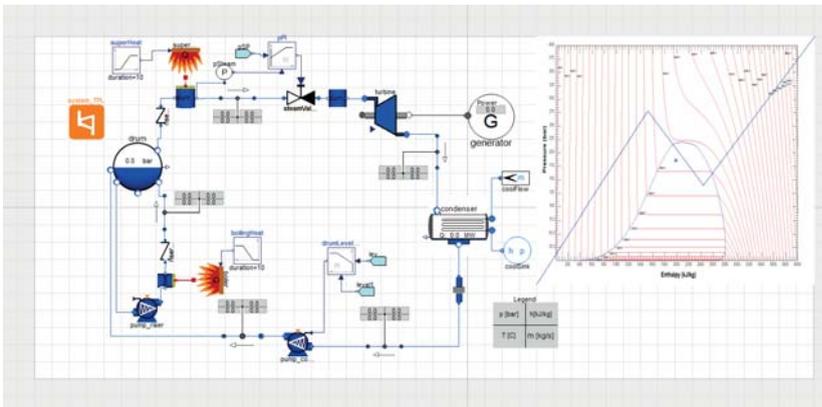


Figure 20: Diagram of the closed steam cycle used for benchmarking

8.2.1 Description of model

The second model used is a model of a closed steam cycle. The model includes a drum, condenser, turbine step with supporting control systems aiming at controlling the pressure and water level in the drum. This is done via the feed water pump to the drum and the high pressure steam valve located next to the turbine step.

At the start of the simulation there will be some transients due to that the system is not started in a steady state. At $T = 500$ the heat load increases from the boiler. This will initially increase the pressure in the drum before the control system controls the pressure back to the set-point. The increased heat load will make it possible to open the steam valve further and result in a greater power output from the turbine.

8.2.2 Result from simulation

The simulation was run with both the reference medium Water IF97 and the newly implemented media functions. The model translation yielded following translation message, which describes the complexity and size of the equation system. Since changing the media models is not changing any of the system characteristics such as number of equations or variables, both simulation will receive the same translation log. The solver used was DASSL and the tolerance was set to 10^{-6} .

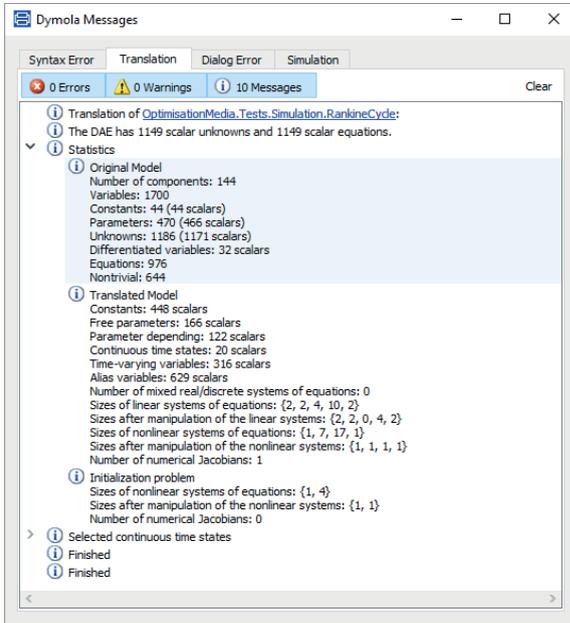
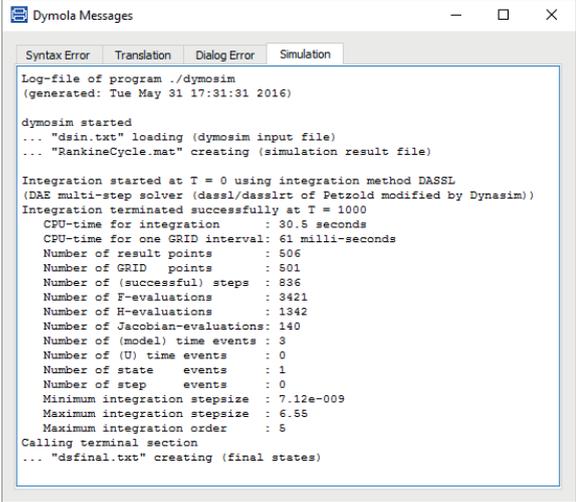


Figure 21: Translation log for the closed steam cycle model.

The translated models can then be simulated. The simulation statistics for both simulations can be seen in figure 22.

As can be seen from the simulation statistics, the new media implementation seems to be considerably slower than Water IF97, a little bit more than twice the simulation time. The main feature of the medium is however not to be as fast as possible but to be compatible with optimisation algorithms. However, if the medium is too slow it might become impractical to use. The reasons might be, except for the reasons mentioned in the previous simulation case, that this model includes components (condenser) which operate at very low pressure levels. The accuracy of some of the media functions here are worse. As mentioned in 6.1 there was an extra very low pressure region added to some of the functions for this reason. This extra region might very well add to the stiffness of the system, which decreases the step size used in the solver. From figure 22 it can be seen that the new implementation has a lower maximum step size and



```

Dymola Messages
Syntax Error  Translation  Dialog Error  Simulation

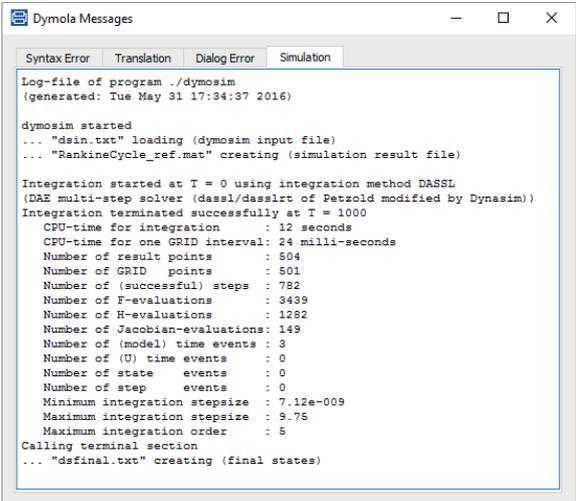
Log-file of program ./dymosim
(generated: Tue May 31 17:31:31 2016)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "RankineCycle.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 1000
CPU-time for integration      : 30.5 seconds
CPU-time for one GRID interval: 61 milli-seconds
Number of result points      : 506
Number of GRID points       : 501
Number of (successful) steps : 836
Number of F-evaluations      : 3421
Number of H-evaluations      : 1342
Number of Jacobian-evaluations: 140
Number of (model) time events : 3
Number of (U) time events    : 0
Number of state events       : 1
Number of step events        : 0
Minimum integration stepsize : 7.12e-009
Maximum integration stepsize  : 6.55
Maximum integration order    : 5
Calling terminal section
... "dsfinal.txt" creating (final states)

```

(a) New media implementation



```

Dymola Messages
Syntax Error  Translation  Dialog Error  Simulation

Log-file of program ./dymosim
(generated: Tue May 31 17:34:37 2016)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "RankineCycle_ref.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 1000
CPU-time for integration      : 12 seconds
CPU-time for one GRID interval: 24 milli-seconds
Number of result points      : 504
Number of GRID points       : 501
Number of (successful) steps : 782
Number of F-evaluations      : 3439
Number of H-evaluations      : 1282
Number of Jacobian-evaluations: 149
Number of (model) time events : 3
Number of (U) time events    : 0
Number of state events       : 0
Number of step events        : 0
Minimum integration stepsize : 7.12e-009
Maximum integration stepsize  : 9.75
Maximum integration order    : 5
Calling terminal section
... "dsfinal.txt" creating (final states)

```

(b) Water IF97 media implementation

Figure 22: Simulation statistics from simulation of the closed steam cycle model with Water IF97 media implementation

uses more steps to simulate the system over the time frame.

To compare the result from the different media implementation a few important variables of the model were plotted in figure 23 and 24.

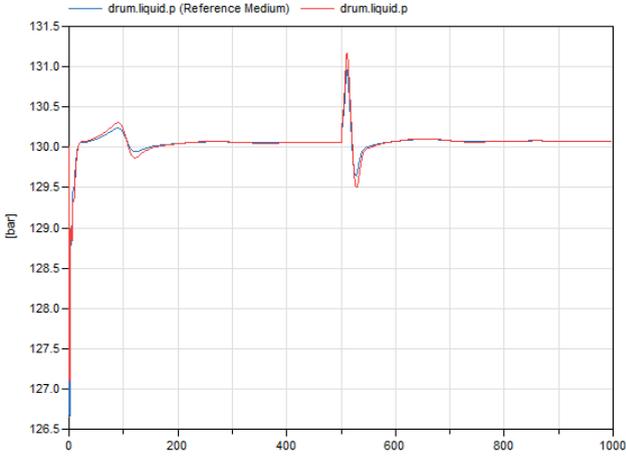


Figure 23: Pressure level in the drum for the two different media model

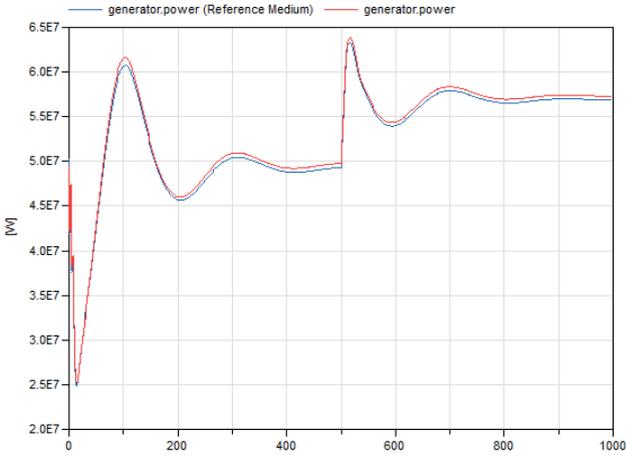


Figure 24: The power generated in the generator in the closed steam cycle model for the two different media implementations.

The results from the two media implementations seems to match fairly well. The same dynamics can be observed. However there seem to be a small difference in the steady state solution for non-controlled properties, that indicates small differences in the media functions.

CONCLUSION

9.1 OUTCOME

The aim of this thesis was to investigate and implement a new model of the thermodynamic properties of pure water, compatible with optimisation algorithms. This was successfully done. Furthermore there were plans on making this new model implements the Modelon.Media Interface, in order for easy use in existing simulation models. This was also done but it proved difficult to make this implementation compatible with the optimisation platform OCT, why a standalone function package was used in the optimisation benchmarking case. However the media interface implementation was used successfully when used in simulation. There were also some hopes on making the new media implementation faster than the old IF97 formulation. This proved hard in the simulation cases as the old implementation has an advantage that it may calculate many media properties at once as they all are related through the fundamental functions (Gibbs free energy function, etc.). Another contributing factor that adds up to the computational time of the new media implementation is the fact that it evaluates all the approximated polynomials for the different regions for every media call in order to make the implementation C^2 , while the IF97 formulation only calculates for one region at a time. These were contributing factors that made the new media implementation significantly slower than the IF97 formulation in simulation applications.

In the optimisation benchmarking case (chapter 8), it was shown that the implemented media functions could be used to provide results that coincides well with IF97 simulation results using the resulting optimal control inputs. This shows that such implementations as suggested in this thesis can yield reliable results. However the benchmarking case only tests some of the implemented media functions (see table 4) and only in a certain working region.

The simulation benchmarking test cases in chapter 7 aimed at comparing the accuracy and performance of the new implementation with the existing IF97 media implementation. The results from these simulations shows that there are some slight deviations in the results between the implementations. However the dynamics of the system is captured accurately and the relative errors are small. The largest deviations are observed at rapid transients. That there are deviations is of course expected as the new implementation is an approximation of the old one. The question is whether these differences are small enough to yield acceptable results and in the tested simulation models this seems to be the case. If the simulation statistics are compared, the new implementation seems to be significantly slower this old one. Reasons for this probably includes the earlier mentioned issue with using the state formulation in

the new media implementation as well as the fact that it calculates the polynomials linked to all the regions each time a media function is called. Another contributing factor might be the use of polynomials based on logarithmised inputs.

9.2 FUTURE WORK AND POSSIBLE IMPROVEMENTS

There are two features of the media implementation that would be interesting to investigate the impact of further when used in optimisation applications.

As mentioned earlier in the thesis it is desirable that the media is accurate at the phase borders. When connecting two regions, smooth step functions are used. To define the active region of the smooth step function two parameters are needed, one of these is the active region of the step function earlier referred to as Δ_x in section . This parameter impacts the smoothness of the functions in the implementation. A smaller delta makes the transition between the polynomials go faster and hence making the function "less smooth" even though the implementation in theoretical sense still is C^2 . However making this parameter to big will instead decrease the accuracy in a larger region around the region borders.

When modelling thermodynamic properties there are many natural laws to take into account, which might not totally be satisfied by the approximations made as there is no check on whether such relations are fulfilled. An example of this is that by nature the density must increase with increased pressure if the temperature is kept constant. Iterative solvers that uses gradients based on the function approximations might be affected if there are inconsistencies in such relations. This also is something that might be worth investigating further.

Furthermore the choice of functional form in the least squares approximations might be investigated. In this thesis only the use polynomials of the form described in equation 4.1 have been used. There might be better forms of functionals to represent the functions. In [13] the use of Chebyshev Rational polynomials [14] are proposed for fast evaluation of thermodynamic properties. The use of different functional forms might be a way of making the implementation faster and more accurate.

9.3 EXPERIENCES FROM THE WORK

In order for easy implementation of similar models describing the thermodynamic properties of other media than water, it is desirable to standardise the implementation. Ideally the whole work-flow would be automated so that the only thing that would have to be provided in order to create a new media model is the tables containing the thermodynamic property data. This has however been hard to achieve, as the many different functions that have been approximated have different shapes and appearances making it hard to construct an automated form for all these functions. It has been necessary to make specialised forms and adaptations for many of the functions in order to achieve good accuracy. This is also something that is hard to achieve before having worked with the models for a while realising what may be a good design of the function approximations. This along with designing and adapting a least-squares

algorithm to fit the purpose of fitting thermodynamic property functions would be improvements that simplify the work flow when constructing optimisation-friendly media models.

BIBLIOGRAPHY

- [1] Michael J. Moran and Howard N. Shapiro. *Fundamentals of Engineering Thermodynamics*. Wiley, 2006.
- [2] Yunus A. Çengel and Michael A. Boles. *Thermodynamics : an engineering approach*. New York, NY : McGraw-Hill Education, 2015.
- [3] Wolfgang Wagner and Hans-Joachim Kretzschmar. *International Steam Tables. [electronic resource] : Properties of Water and Steam Based on the Industrial Formulation IAPWS-IF97*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2008.
- [4] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization. [Elektronisk resurs]*. Springer Series in Operations Research and Financial Engineering. New York, NY : Springer New York, 2006.
- [5] Joakim Larsson. *Implementation of smooth interpolation for optimization*. Lund University, Lund, 2015. Master's thesis.
- [6] The MathWorks Inc. *Curve Fitting Toolbox™ User's Guide*. The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, r2015b edition, 2015.
- [7] Johan Åkesson. Optimica—an extension of modelica supporting dynamic optimization. In *Proceedings of the 6th International Modelica Conference*, pages 57–67, Bielefeld, Germany, July 2008.
- [8] Björn Lennernäs. *A CasADi based toolchain for JModelica.org*. Department of Automatic Control, Lund University, Lund, 2013. Master's Thesis.
- [9] Andreas Wächter and T. Lorenz Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.
- [10] Fredrik Magnusson and Johan Åkesson. Dynamic optimization in JModelica.org. *Processes*, 3(2):471, 2015.
- [11] Per-Ola Larsson. Report from the modelon tutorial at the 2015 modelica conference. <http://www.modelon.com/blog/articles/report-from-the-modelon-tutorial-at-the-2015-modelica-conference/>, October 2015.
- [12] Håkan Runvik. *Modelling and start-up optimization of a coal-fired power plant*. Lund University, Lund, 2014. Master's thesis.
- [13] Vikrant Aute and Reinhard Radermacher. Standardized polynomials for fast evaluation of refrigerant thermophysical properties. In *International Refrigeration and Air Conditioning Conference at Purdue*, number 1499, Purdue, Indiana, July 2014.
- [14] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials. [Elektronisk resurs]*. Boca Raton, Fla. : Chapman Hall/CRC, cop. 2003, 2003.

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> July 2016	
		<i>Document Number</i> ISRN LUTFD2/TFRT--6011--SE	
<i>Author(s)</i> Marcus Åberg		<i>Supervisor</i> Johan Windahl, Modelon AB Fredrik Magnusson, Dept. of Automatic Control, Lund University, Sweden Anders Rantzer, Dept. of Automatic Control, Lund University, Sweden (examiner)	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Optimisation-friendly modelling of thermodynamic properties of media			
<i>Abstract</i> <p>The use of dynamic modelling and simulation of thermal power systems has grown recent years and as the performance of computers have increased optimisation of such systems has become possible. When performing such optimisations, there are high numerical requirements on the model used in the system. Due to the nonlinear nature of thermodynamic properties of working fluids, extra consideration has to be made when modelling these. In this thesis such a media model has been implemented through polynomial approximation. It has been shown that such an implementation can be used in optimisation algorithms yielding reliable results. A benchmarking optimisation test case has been constructed in order to test the implementation's accuracy and performance. The physical models were implemented in Modelica used for dynamic optimisation using the JModelica.org platform. Furthermore it has also been benchmarked against existing media models designed for simulation in complex thermal power systems in order to verify the accuracy of the new model of thermodynamic properties of media.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-57	<i>Recipient's notes</i>	
<i>Security classification</i>			