OPTIMIZATION TOOL CHAIN IN JMODELICA.ORG

Toivo Henningsson, Modelon

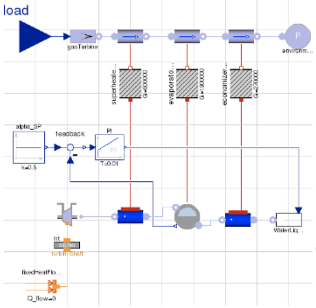
OUTLINE

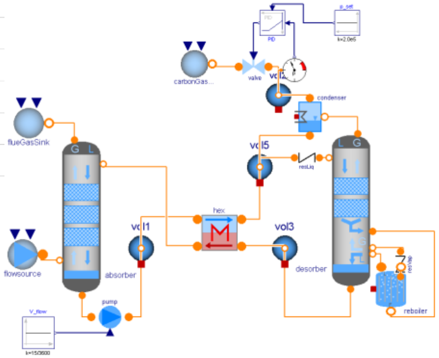
- JModelica.org
- Using dynamic optimization
  - Problem formulation
  - Optimica
  - Python tool box
  - Example
- Solution algorithm – collocation
- Application examples

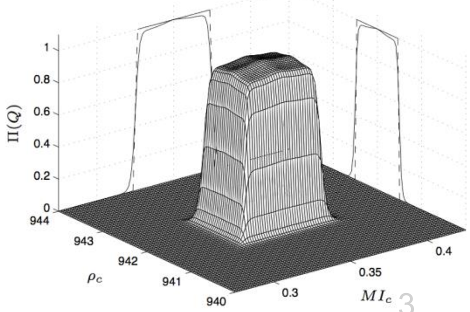
THE JMODELICA.ORG OPEN SOURCE PROJECT

JModelica.org is an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems.

- Modeling with Modelica
- Simulation with FMI
- Optimization
- Analysis
- Visualization
- Industrial applications







DYNAMIC OPTIMIZATION

- **Simulation** means finding solution to

$$F(\dot{x}, x, y) = 0, x(0) = x_0$$

  - Unique solution given initial conditions
  - Constraint $F(\dot{x}, x, y) = 0$ is the model
- **Dynamic optimization** adds inputs u and/or parameters p:

$$\min \int_{t_0}^{t_f} L(\dot{x}, x, y, u, p)dt$$

$$\text{s.t.}\, F(\dot{x}, x, y, u, p) = 0, x(0) = x_0$$

  - Find u, p that minimize cost
  - Possible to add extra constraints such as $x_L \leq x \leq x_u$

DYNAMIC OPTIMIZATION WITH JMODELICA.ORG

- Write models in Modelica
- Add optimization information in *Optimica*
    - A small extension of Modelica for optimization
    - Allows to add optimization information:
        - Cost function
        - Additional constraints
        - Parameters to optimize – including time horizon length
        - Initial guesses
        - …
- Work with optimization problems from Python

EXAMPLE

```
model Integrator
    Real x(start = 2, fixed = true);
    input Real u;
equation
    der(x) = -u;
end Integrator;
```

Let’s add optimization information!

THE OPTIMICA EXTENSION

```
optimization Integrator(
    objective = finalTime,
    objectiveIntegrand = x^2 + u^2,
    startTime = 0,
    finalTime(free=true,min=0.5,max=2,initialGuess=1))
    Real x(start = 2, fixed = true);
    input Real u;
equation
    der(x) = -u;
constraint
    u <= 2;
    x(finalTime) = 0;
end Integrator;
```

$$\min t_f + \int_0^{t_f} (x(t)^2 + u(t)^2)dt$$

$$\text{s.t. } \dot{x} = -u, \qquad u \leq 2,$$

$$x(0) = 2, \quad x(t_f) = 0,$$

$$½ \leq tf \leq 2$$

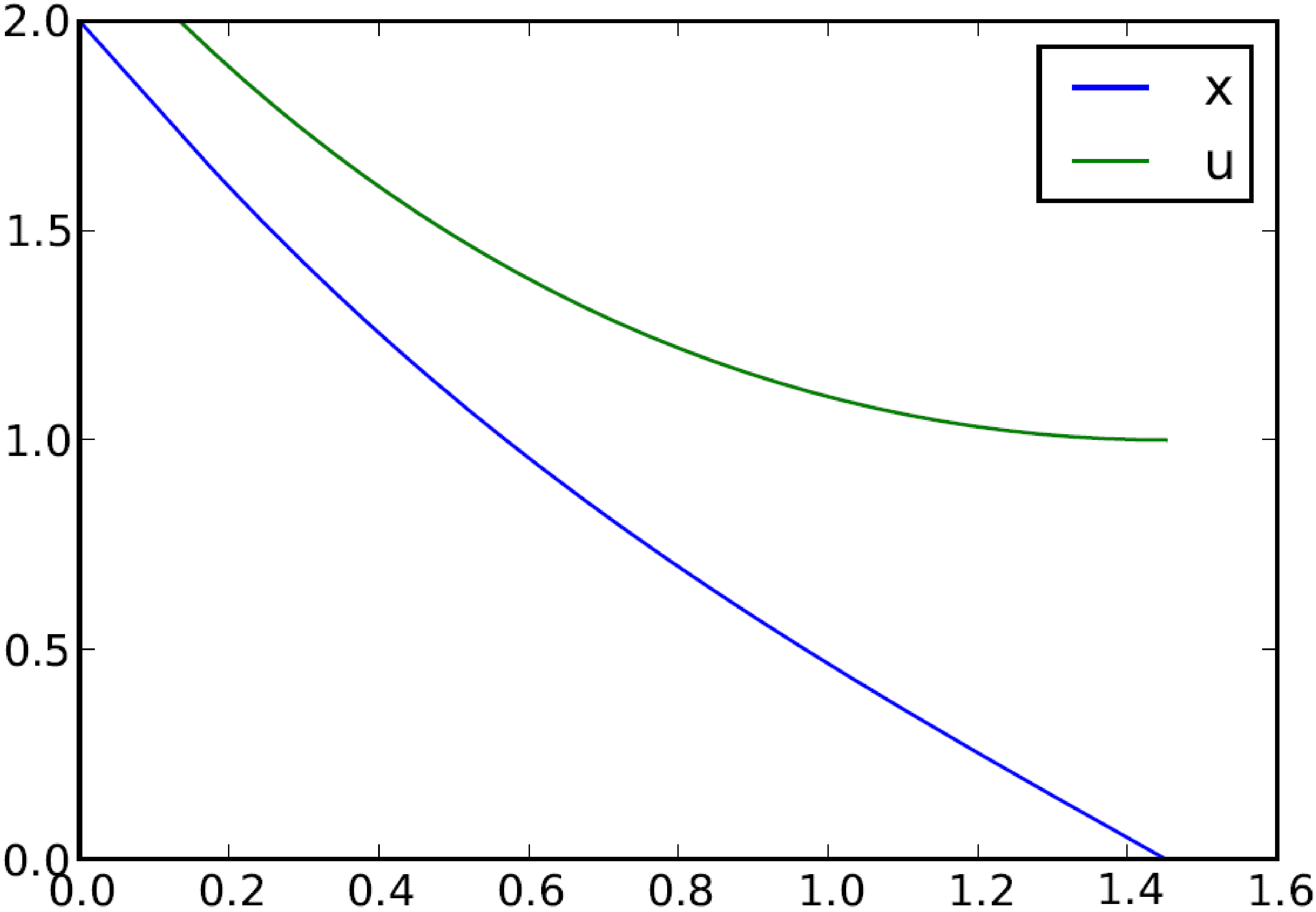
OPTIMIZING IN PYTHON

```
from pyjmi import transfer_optimization_problem
import matplotlib.pyplot as plt

op = transfer_optimization_problem('Integrator',
         'Example.mop')
res = op.optimize()

t = res['time']
x = res['x']
u = res['u']
plt.plot(t,x, t,u)
```

OPTIMIZATION RESULT



PYTHON API FOR OPTIMIZATION

- Import
- Inspect
- Manipulate – add/change cost, constraints…
- Solve
- Or extract equations and use in custom solver
  - Represented using CasADi

Additional options

- Initial and nominal trajectories, e.g. from simulation
- External data
- Warm starting – reuse discretization, change parameters/data
- Blocking factors
- Delay constraints

SOLUTION ALGORITHM

Collocation

- Divide time horizon into elements
- Approximate system variables by low-order polynomials in each element
  $\Longrightarrow$ Solution given by a finite number of variables
- Use entire trajectory as unknown variable

Solution

- Collocation result: A very big nonlinear program (NLP)
- Solve, e.g. using Ipopt
  $\Longrightarrow$ need C^2 continuous problem formulation

THERMODYNAMIC APPLICATIONS

Optimization projects with JModelica.org

- Model predictive control of CO_2 post-combustion capture plants
- Grade change of polyethylene production
- Production planning for district heating plants
- Optimal start-up of
  - Combined gas cycles
  - Coal fired plants

OPTIMIZATION CAPABLE MODELS

Library of optimization friendly thermodfluid components

- Accumulator
- Turbine
- Valves
- Heat exchangers
- Condensers
- Static and dynamics pipes
- Pipes with delay
- Media models

OPTIMIZATION CAPABLE MODELS

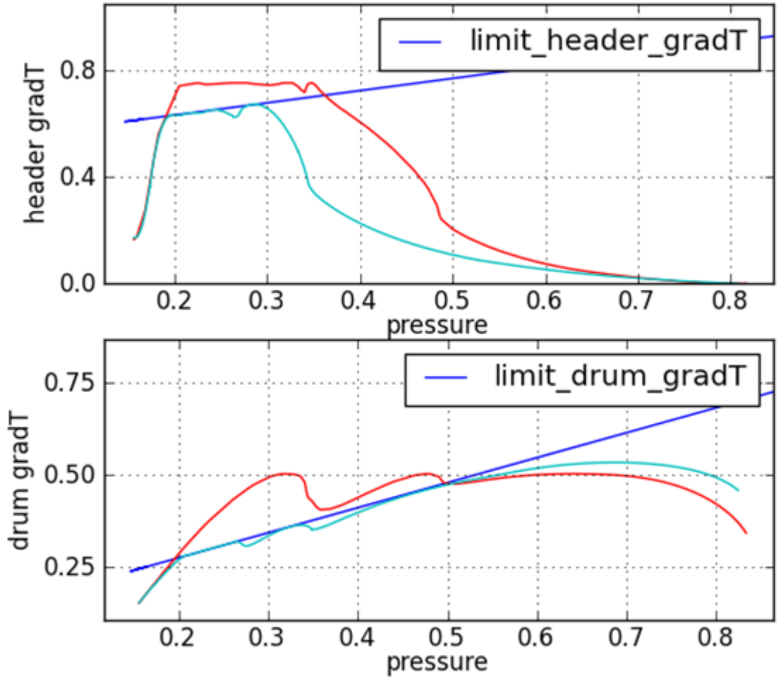
- High fidelity water & steam properties
  - Polynomials in pressure and enthalpy
  - Range: 0.1-184 bar
  - Valid in liquid, gas and 2 phase regions
- Simple water model
  - Constant properties
  - District heating water

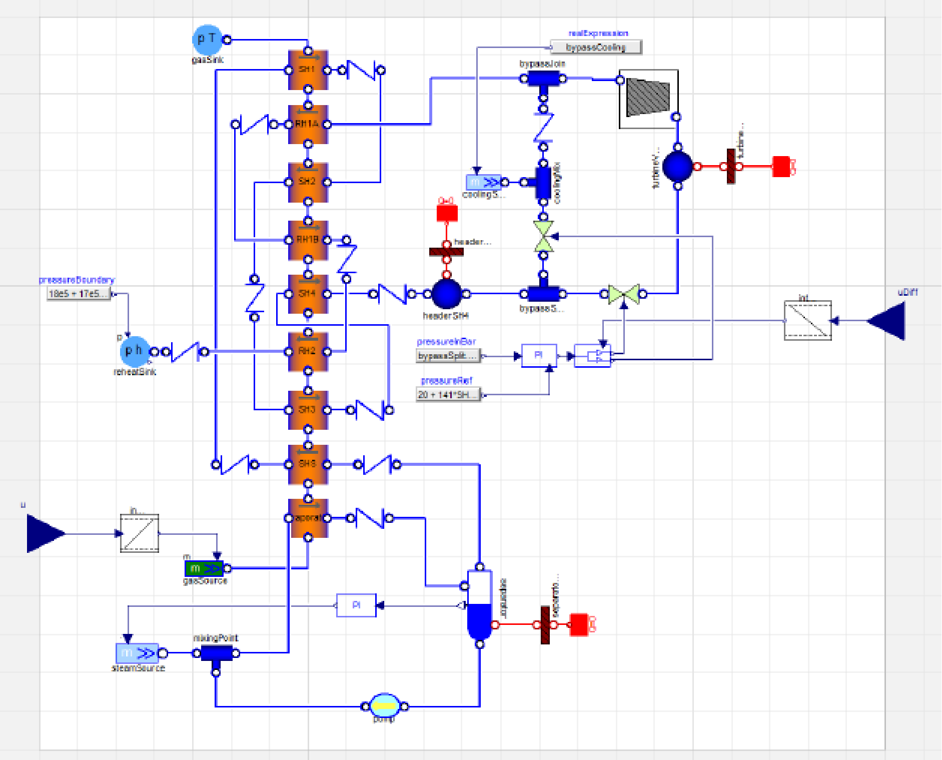
OPTIMAL START-UP OF POWER PLANTS

- Several EU research projects (Siemens, Vattenfall)
- Minimize the start-up time and limit stress in Heat Recovery Steam Generator
- Method:
  - Development of dynamic models in **Dymola**
    - Detailed steam system (evaporator, re-heater, superheater, turbine)
  - Dynamic optimization of the load
    - Drive boiler or gas turbine to full load in minimum time
    - Stress monitored in header and evaporator drum

OPTIMAL START-UP OF COMBINED CYCLES

- Faster start-up and limited stress in critical components

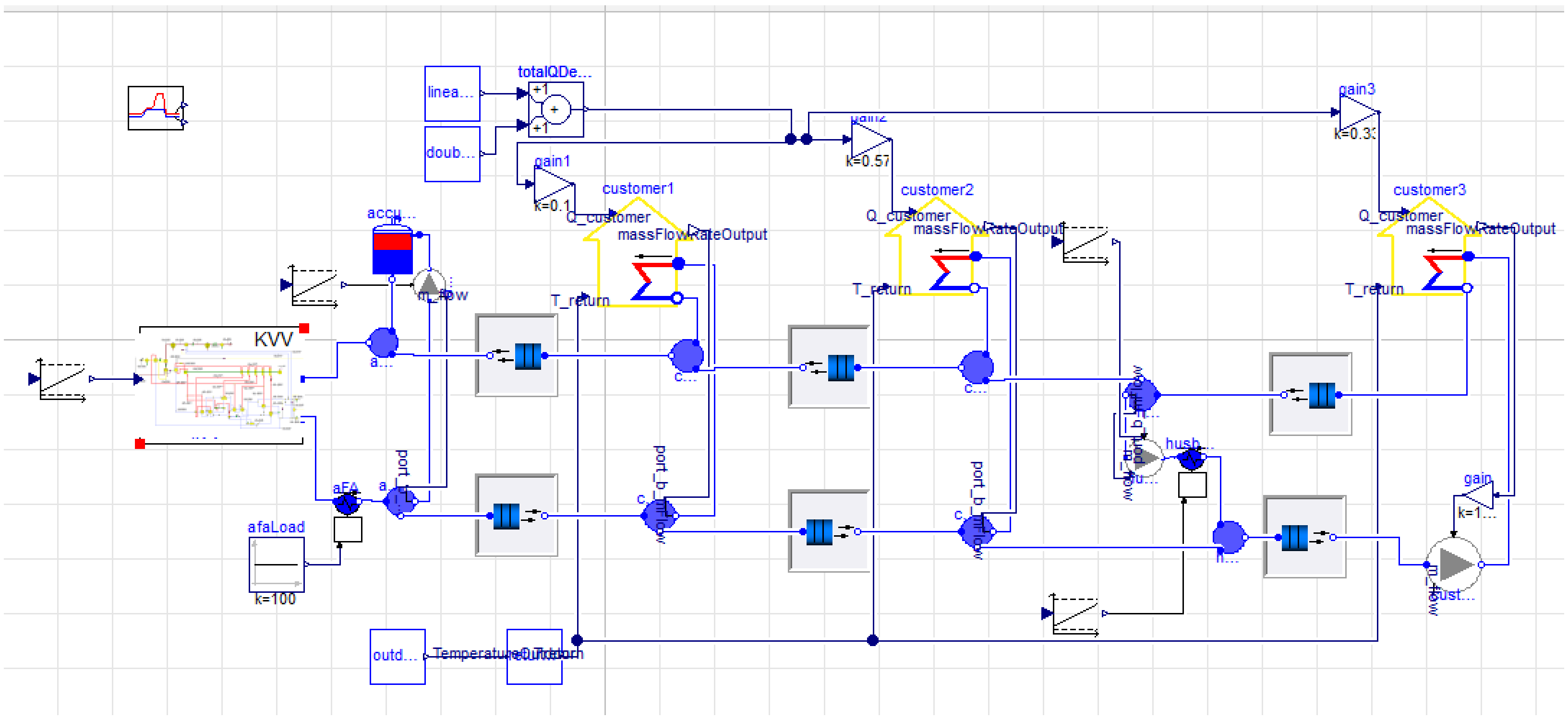




PRODUCTION PLANNING

- Research project together with Vattenfall
- Motivation
  - Standard approach: simple linear plant models
  - Degrees of freedom: power flows
  - No representation of temperature, flowrate, pressure
  - But impact on electrical production, energy in network and accumulators
- Objective: Short-term production planning using nonlinear programming
- Method
  - Physical modeling in Modelica
  - Simulation in Dymola
  - Optimization wrt. plant economics in JModelica.org

DISTRICT HEATING NETWORK



PRODUCTION PLANNING: RESULTS

- Lower supply temperature
- Higher supply flows
- Lower power peaks at production sites
- Higher electricity production
  - Use of heat storage
  - Lower return temperature
  - No turbine by-pass
- Constraints on
  - Customer temperature
  - Pump capacity
  - Network capacity
  - Condenser pressure

toivo.henningsson@modelon.com

Want to know more?

jmodelica.org

> Users > User's Documentation > Ch. 6. Optimization