Feedback Loop Optimization for a Distillation System by applying C-Code Controllers with Dymola

Hansjörg Kapeller Dragan Simic
AIT Austrian Institute of Technology, Mobility Department, Electric Drive Technologies
Giefinggasse 2, 1210 Vienna, Austria
hansjoerg.kapeller@ait.ac.at dragan.simic@ait.ac.at

Abstract

The process engineering domain covers a large field of different disciplines such as thermodynamics, electrical engineering or chemistry. On one hand just the knowledge about these different disciplines to approach and develop standalone-solutions is a big challenge where on the other hand the configuration and the control of the system routines are crucial steps. Nowadays most applications in process engineering are automated and digital signal processors (DSP) are widely used to implement control modules for different automatic control systems in an efficient and flexible way. Nevertheless, without dynamically applicable and real-time capable models it seems nearly impossible to estimate real operating conditions (e.g. controller parameter settings) in process engineering according to real requirements.
This paper presents the simulation model of a thermal control circuit for determining the distillation properties of petrochemical end products modeled in *Modelica* and performed by using the simulation tool *Dymola*. This simulation environment improves the design of interdisciplinary models and allows the optimization of the entire feedback loop.

Keywords: process engineering, automatic control systems, simulation, interdisciplinary models, optimization

1 Introduction

The measurement device for determining the distillation properties of petrochemical end products is depicted schematically in Figure 1. It is a device in which different processes of thermodynamics, chemistry, mechanics, electrical measuring and control technology must be controlled. The mode of operation is based on the vaporization of the investigated medium, e.g. acetone, which condenses in a collecting vessel again. The heating energy is controlled by the heat controller and the condense level of the vessel is controlled by a stepper motor, respectively. In case of equilibrium and on condition that all controllers are working in a stable operating point the medium vaporizes and condensates in the vessel by keeping a constant level until all - in case of a pure substance - is exhausted or - in case of a mixture - the next component reaches the inherent evaporating temperature.

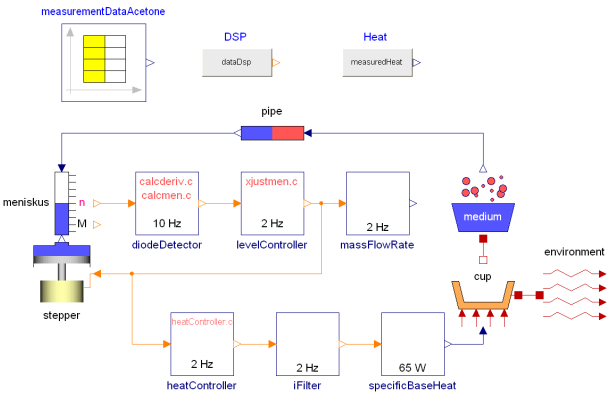

Figure 1: Scheme of the investigated measurement device for determining the distillation properties of petrochemical end products

The challenge in this process engineering application is to parametrize the level control and the heating control [1]. Both controllers are not independent: if the stepper motor controller does not work adequatly, e.g. the motor rotates too fast, the level in the vessel sinks too quickly and the operating point becomes unstable. The incapacity resp. inertia of the heat controller

leads to an insufficient vaporization of the medium and therefore to an insufficient condensation rate with the consequence, that the level decrease can not be compensated. A flexible, timely and systematic way is to model the entire measurement device using the object-oriented modeling language *Modelica* [2].

2 Modeling and Simulation in Dymola

For modeling and simulating integrated and complex systems the software platform *Dymola* is used. *Dymola* is an environment using the objects described in *Modelica* syntax, which allows the software designer to create models of any kind of objects that can be described by algebraic and ordinary differential equations. With *Dymola* simulations of the behavior and interaction betweens systems of different engineering fields, such as mechanical, electric, thermodynamic, hydraulic, pneumatic, thermal and control systems are possible. The modeling language itself is open which means that users are free to create their own model libraries or modify standard libraries to better match the users individual modeling and simulation needs.

2.1 Simulation Models

To ensure, that the simulation reflects the measurement device as best as possible, all components - except the controllers - are modeled in *Modelica* code, whereas the C-code algorithms of the DSP controllers and all variable names have been retained unchanged and are coupled directly as external C-functions to the rest of the closed loop control system. Following this strategy, the model allows to tune the controllers in the same way as it occurs in the real measurement device by adapting the C-routines of the DSP controllers and the obtained simulation results based on the same C-routines.

To emulate the DSP properties in *Dymola*, the external C-routines (e.g. the heat controller) are invoked using a sample clock. The clock frequency is given by $T_{clock} = \frac{1}{f_{sample}}$ and corresponds to the processing time when the heat controller routine will be executed on the DSP. Thus every clock-event the *Dymola*-function `heatControl.mo` will be executed (cp. Figure 2).

In Figure 3 the source code of this *Dymola*-function in which the C-routine itself will be invoked and processed is presented (cp. the sequence beginning with `external "C"`).

```
model heatController

  import SI = Modelica.SIunits;
  parameter SI.Frequency fSample = 2 "controller frequency";

  Integer dsp(start=0);
  Integer dspDummy[3](start={0,0,0});
  Integer indexVar(start=0);
  Integer indexVarPre(start=0);

  Modelica.Blocks.Interfaces.IntegerInput dspIn
    inner Modelica_LinearSystems.Sampled.SampleClock
        sampleClock(sampleTime=1fSample, blockType=
        Modelica_LinearSystems.Sampled.Types.BlockType.Discrete,
        methodType=Modelica_LinearSystems.Types.Method.ExplicitEuler)
  Modelica.Blocks.Interfaces.IntegerOutput dspOut
    algorithm

  when sampleClock.sampleTrigger then
    dsp := dspIn;
    dspDummy := {dsp,pre(dsp),indexVar};
    (dspOut, indexVarPre) := dymolaFunction.heatControl(dspDummy);
    indexVar := indexVarPre;
  end when;

end heatController;
```

Figure 2: *Dymola* model of the heat controller. When the clock-event occurs, the *Dymola* function `heatControl.mo` will be executed

```
function heatControl

  input Integer dspDummy[3];
  output Integer dspOut;
  output Integer indexVarPre;

    external "C" heatControl(dspDummy,dspOut,indexVarPre);
       annotation (Include="#include <heatController.c>");

end heatControl;
```

Figure 3: *Modelica* code of the *Dymola* function `heatControl.mo`. Represents the interface to the external C-routine `heatController.c`

2.1.1 Level Meter – Diode Array

Figure 4 depicts the model of the vessel equipped with two input connectors. One connection provides the input for the vaporized distillate (cp. *Distillate_in*), the second one represents the control the input-variable for the level control (cp. *Stepper_connect*). In real life the level meter is realized by a diode array consisting of 16 diodes, where effectively 10 diodes are evaluated. The remaining 6 diodes are used, to delimit the upper and lower recoverable level positions. In the simulation environment a corresponding discrete signal is generated, which feeds the *diodeDetector*-model. The *diodeDetector*-model invokes the C-function routines `calcderiv.c` and `calcmen.c` and maps the discrete level signals into a proper range of values (nominal value amounts 5) which are available after the inherent DSP process time of $0.1s$ as output, again.

Figure 5 presents two generated outputs taken from a simulation run during evaluation procedures. The red curve shape (10 diodes evaluated) can be compared with a gray curve, which represents the obtained output of discrete position values, when all 16 diodes

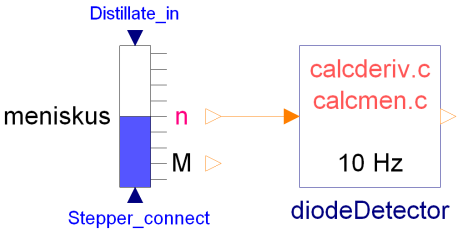

Figure 4: *Dymola* model of the vessel including the C-function routines for the level output generation

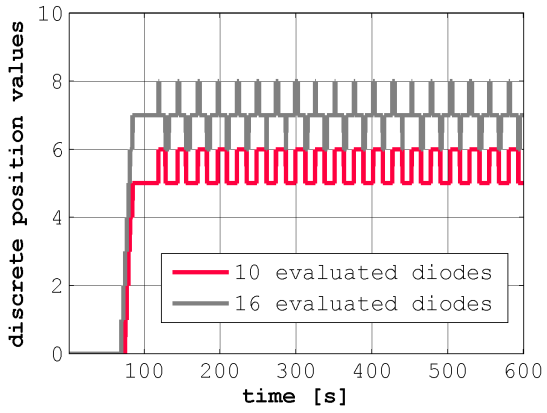

Figure 5: Obtained output of the level meter in *Dymola*. Comparison of two curve shapes: red curve 10 evaluated diodes, gray curve 16 evaluated diodes

were evaluated. A diode array with higher resolution gives rise to a higher frequent, albeit smaller ripple, which leads to positive effects on one hand for the stepper motor - less influence due to the mass inertia - on the other hand for the heat controller, too.

2.1.2 Level Control

Figure 6 depicts the previous model extended with a PI-controller which provides the control variable for the level control thus the control loop for adjusting the frequency of stepper motor can be closed. The stepper motor controller is implemented as external C-routine and corresponds identically to the implemented PI controller on the digital signal processor. The DSP process time for this C-code amounts $0.5\,s$, the process time for *diodeDetector*-model amounts $0.1\,s$ and is implemented corresponding to the real measurement device. With other words: the provided input sample rate for the level controller is higher $(10\,Hz)$ than the generated output of the level controller $(2\,Hz)$ which acts as input for the stepper motor, again.

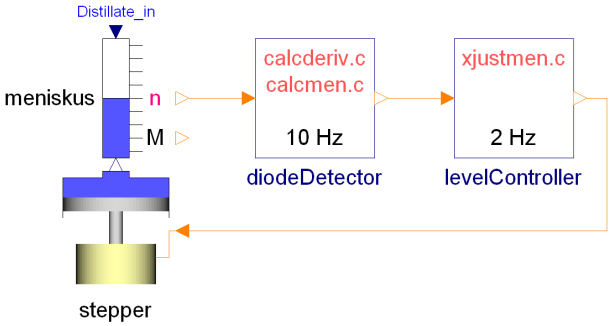

Figure 6: Extended vessel model with additional PI-controller which provides the control variable for the level control

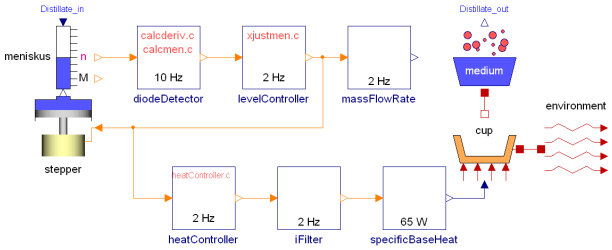

Figure 7: Implementation of the heat controller closed loop in *Dymola* which corresponds to the real measurement device

2.1.3 Heat Controller

The control signal for the stepper motor is also used as input for the heat controller (`heatController.c`) which is implemented as PID-controller. The output of the heat controller is filtered (*iFilter*) in order to smooth the control variable if unexpected discontinuities occurs. This signal will be charged with an acetone-specific base heat and is connected to the cup-heater. Thus the controlled heating energy causes the evaporation of the medium (acetone) and a desired flow rate will be achieved (*Distillate_out*). All components regarding the heat controller are depicted schematically in Figure 7.

2.2 Simulation of the Entire Measurement Device Model

If the two interfaces *Distillate_out* and *Distillate_in* are connected using a pipe model (*pipe*), then the feedback loop for the entire model can be closed (anticipated in Figure 1 already) and simulated in order to ensure the validity comparing measurement data with simulation data. Only after a successful validation, modifications in parameterization or other appropriate measures can be taken into account. The measurement data are imported in *Dymola* using the block

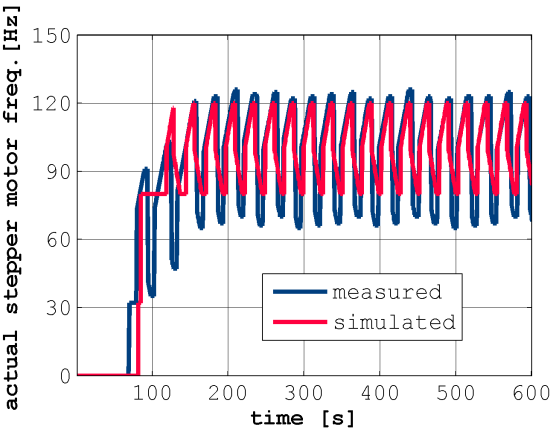

Figure 8: Comparison of measured and simulated actual stepper motor frequency

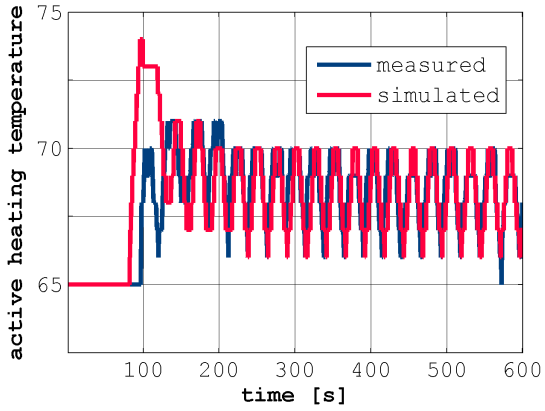

Figure 9: Comparison of measured and simulated active heat temperature

measurementDataAcetone, the recorded DSP output is also available. The comparison between measurement and simulation is the subject matter of chapter 3.

2.2.1 Validation of the Stepper Motor Control

In Figure 8 the actual stepper motor frequency (blue curve) - plotted from the real measurement device when determining the distillation properties of acetone - and the corresponding simulation curve shape (red curve) are presented. The comparison of both, the measured curve shape and the simulated curve shape shows a good approximation of the developed measurement device-model in the simulation environment *Dymola*.

2.2.2 Validation of the Heat Controller

In Figure 9 the active heating temperature regulated by the heat controller (blue curve from measurement data) - when determining the distillation properties of acetone - and the corresponding simulation curve shape (red curve) are presented. The comparison of both, the measured curve shape and the simulated curve shape shows a good approximation of the developed measurement device-model in the simulation environment, again.

3 Discussion of the Simulation Results

As in Figure 8 and Figure 9 already illustrated, the validation of the simulated measurement device is ensured adequately. The further investigation leads to a first conclusion, that the significant oscillating heating tempertuare destabilizes the entire measurement device. A suitable way to smooth this signal is not to smooth the heat controller output as anticipated in chapter 2.1.3 already and realized here, but to smooth the heat controller input, corresponding to the common approach recommended by several control engineering theories [3]. As mentioned above, the heat controller input signal (implemented as PID-controller) is derived from the control signal for the stepper motor and is a strong oscillating signal (cp. Figure 8, nominal value: $120 Hz$) and from point of reasonableness it makes sense to smooth this signal.
A different approach, but with similar effect, can be achieved, when a diode array with higher resolution is used (e.g. 16 evaluated diodes), which leads to less ripple in the provided heat controller input signal. These modification and further effects of controller and filter parameter changes will be investigated in the next chapter.

3.1 Parameter Tuning - Improvements

The measurement device for determining the distillation properties of petrochemical end products is depicted in Figure 10 again, though with the already mentioned modification to smooth the *heatController* input. To enhance the filtering effect, the number of averaged signals in the filter block is increased from 32 values up to 64. Thus the smoothed filter output is generated by averaging the last 64 values and implies an additional improvement for the entire system performance. To reduce the signal-ripple distinctly (allows more effective signal-smoothing), a diode array with higher resolution (e.g. 16 evaluated diodes) could

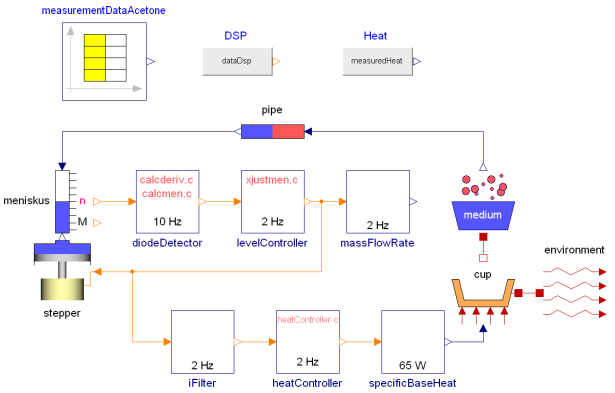

Figure 10: Scheme of the supposed measurement device for determining the distillation properties of petrochemical end products

be used. Without improvements, the achieved active heating temperature from the heat controller indicates curve shapes, as shown in Figure 9 already. With the modifications and when the heat controller parameters are tuned in a proper way, too, the controlled heating energy leads to an optimized evaporation of the medium (acetone) and therefore a more constant flow rate can be achieved, however.

3.1.1 Consequences

Figure 11 depicts the achieved improvement when the *heatController* input is filtered by using an enhanced filter, where the number of averaged signals in the filter block is increased from 32 values up to 64 and when the heat controller parameters are tuned in experimental way: the derivative control component is deactivated and the proportional gain is increased double the amount. When applying these modifications (note that the heat controller is a PI-controller and not a PID-controller anymore) the obtained active heating temperature curve shape shows a significant less oscillating curve shape.

Finally the investigation of the resulting flow rate in the system is of interest and the comparison of the suggested improvements are presented in Figure 12. The red curve shape shows the flow rate of acetone (desired flow rate $10 \mu l/s$) simulating the entire model without improvements and where all parameters remain unchanged.

The green curve represents the flow rate of acetone when the *heatController* input is filtered by using the enhanced filter and when the heat controller parameters are tuned. This graph shows a signficant improvement in the flow rate regarding minor oscillating and closer approximating to the desired flow rate value.

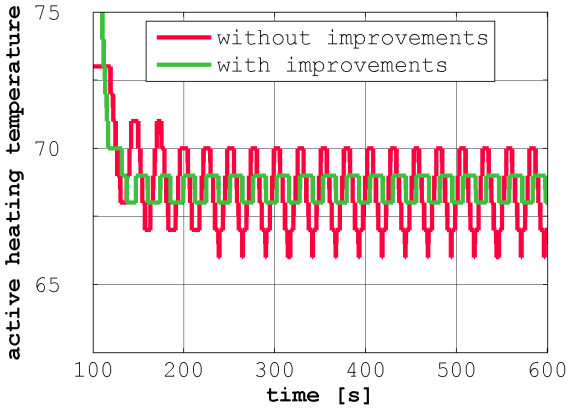

Figure 11: Comparison of the simulated active heating temperature curves - without improvements and with improvements

Figure 12: Comparison of the of the simulated curves for the acetone flow rate - without improvements and with improvements

The gray curve shape shows a further enhanced time behavior and results if additionally to the proposed improvements a diode array with higher resolution (e.g. 16 diodes) is used.

4 Conclusions

The purpose of this contribution was the implementation of the entire distillation model in such way, that the simulation reflects the measurement device as best as possible. Following this strategy, the model allows to improve the measurement device and to tune the controllers in the same way as it occurs in the real measurement device by adapting the C-routines of the DSP controllers, whereas time can be saved and a benefit in flexibility can be obtained. The entire *Modelica* model

of the measurement device for determining the distillation properties of petrochemical end products was presented and through the use of the simulation environment *Dymola* the design of interdisciplinary models and the design of the entire feedback loop can be revised. The recommended parameter modification especially for the heating controller and the changes regarding the control structure (i.e. the right selection which signals should be filtered) show a significant improvement in the total system behavior.

References

[1] O. Föllinger, *Regelungstechnik*, Hüthig Verlag, Heidelberg, 8 edition, 1994.

[2] Peter Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, Piscataway, NJ, 2004.

[3] O. Föllinger, *Optimierung dynamischer Systeme*, R. Oldenbourg Verlag, München, 1985.