Designing models for online use with Modelica and FMI

Pål Kittilsen[1,2]   Svein Olav Hauger[1]   Stein O. Wasbø[1]

[1]Cybernetica AS, 7038 Trondheim, Norway

[2]Statoil Research Centre, 7005 Trondheim, Norway

pkit@statoil.com

{svein.o.hauger, stein.wasbo}@cybernetica.no

Abstract

Model-based online applications such as soft-sensing, fault detection or model predictive control require representative models. Basing models on physics has the advantage of naturally describing nonlinear processes and potentially describing a wide range of operating conditions. Implementing adaptivity is essential for online use to avoid model performance degradation over time and to compensate for model imperfection. Requirements for identifiability and observability, numerical robustness and computational speed place an upper limit on model complexity. These considerations motivate that models for online use should be balanced-complexity, physically based with online adaption possible.

Despite potential benefits, the effort required to implement balanced-complexity models, particularly at large scales, may deter their use. This paper presents techniques used in the design of balanced-complexity models. A Modelica-based approach is chosen to reduce implementation effort by interfacing exported Modelica models with application code by means of the generic interface FMI. The suggested approach is demonstrated by parameter estimation for a process of offshore oil production: a subsea well-manifold-pipeline production system.

Keywords: modeling, process control, process models, process simulators, offshore oil and gas production, Modelica, subsea production, multiphase flow, balanced-complexity models, nonlinear model-predictive control, FMI

1 Introduction

In this paper the term *online model* refers to a model that tracks the state of a process over time and is implemented with adaptivity. *Adaptivity* in this paper can refer to either state estimation, parameter estimation, or both.

Applications that can benefit from online models include online simulators for "what-if" and look-ahead analysis, data reconciliation, soft-sensors, fault detection, advisory decision support systems, (nonlinear-) model predictive control (nMPC) and real-time optimization. Such applications have in common that real-time computations are performed on a model that hopefully represents the process with sufficient accuracy. Evaluating and comparing multiple simulation scenarios internally within real-time requirements place conditions on computational speed. Algorithms that evaluate models at different combinations of inputs, states and parameters place requirements on numerical robustness.

Unless the process is time-invariant and the fitted model matches the process perfectly, the model's ability to track process states will degrade over time. For industrial processes, both time-variation and model imperfections must be expected, which makes adaptivity a crucial factor in the maintenance of model-based online applications. Adaptivity can also be exploited to simplify aspects of modeling for online use, to be discussed.

Identifiability and observability considerations place limits on how many states and parameters that can be uniquely adapted to a given set of measurements of a process. As a consequence, adapting all the parameters and states that are uncertain or time-varying in complex models will often be an ill-posed problem for the available set of measurements. Some authors have suggested converting full-complexity engineering simulators into online models, see for instance [11], but few references are found in the literature of the use of such models for the online applications listed above.

Balanced-complexity models in this paper refer to models based mainly on physics which are specifically designed to adhere to requirements set by online use

for identifiability, observability, numerical robustness and computational speed. In control literature many references to purpose-built online models are found, some recent applications related to process control and oil and gas applications are; industrial batch process: [14], thin-rim oil reservoirs: [13] and [15], riser slugging in multiphase flows: [9] and drilling: [7].

Balanced-complexity models cited in the literature are usually quite small in scope, and for the applications listed above they typically describe a particular piece of equipment or a specific phenomenon of interest in a subsection of a larger plant. Often such models found in the literature are small-scale, on the order of 10 states or less and are feasible to hand-code. There may be synergies to monitoring and controlling large plants in a unified manner instead of as a series of smaller subsystems, a recent discussion of this idea applied to subsea fields is found in [1]. A balanced-complexity model of such larger systems can have hundreds of states, for instance when modeling an entire offshore processing plant, see [16]. At this scale, balanced-complexity models become challenging to code and maintain manually, and it can be challenging to re-use code and to collaborate on model design. *Large-scale* in this paper refers to balanced-complexity models which attempt to describe large systems, and where challenges related to the scale of the model can potentially deter their use.

Modelica has several advantages that can aid in the synthesis of large-scale balanced-complexity models for online use. First, Modelica is declarative and equation-based, meaning that models are expressed by writing differential and algebraic equations, and Modelica compilers interpret these equations into algorithmic code (usually to the C programming language). Second, Modelica is object-oriented and supports building larger models by connecting smaller sub-models. Third, Modelica supports collecting submodels into libraries that can be shared, re-used and combined as needed. Fourth, most Modelica environments support exporting models with functional-mockup interface(FMI), to be discussed in Section 3.

An earlier reference to work on interfacing translated Modelica code with online control applications is found in [8]. A reference to a similar vendor-specific approach is found in [6]. Several authors have considered interfacing translated Modelica code with optimization algorithms offline, see for instance [10] and [2] for trajectory planning in power plant control.

This paper is to a large extent motivated by development of nMPC for offshore oil and gas produc-

tion, however much of the discussion is independent of process and application. The excitation resulting from normal operation in offshore oil and gas fields can be very low as documented in [3], and this motivates the use of physical modeling and nNMPC, as this approach has reduced need for excited data, see [5]. Some recent applications of nMPC to smart wells are [12], who used a full reservoir simulator as a process model, and [15] who took a balanced-complexity modeling approach. Earlier references to work on large-scale balanced-complexity modeling for offshore oil and gas production are found in [8], which considered the topside processing system, and in [16] which considered a well-pipeline-riser-processing system.

Despite the widespread use of balanced-complexity models reported in control engineering literature, the idea that models for online use should be purpose-built is not widely accepted by industry practitioners with backgrounds in other engineering disciplines. Motivated by this observation, the first purpose of this paper is to present argumentation for the use of balanced-complexity models and then present techniques used in their design. Secondly, this paper discusses how Modelica can be used to simplify the process of synthesizing large-scale balanced-complexity models and to integrate them in online applications.

The paper is structured as follows: Section 2 outlines techniques for the design of balanced-complexity models. Next, Section 3 discusses techniques for interfacing models written in Modelica with control applications. Section 4 presents a case study of using Modelica to build a large-scale balanced-complexity model of an offshore processing plant for state estimation.

2 Synthesis of balanced-complexity models for online use

2.1 The purpose dictates the model

Modeling is to map a real world object into a simpler representation - in this context, into a set of equations. It is the modeler's choice which of the real object's properties and features the model should mimic. Emphasis on the purpose of the model leads naturally to a set of required model properties. Including details not contributing to fulfilling the model's purpose adds computational load, degrades identifiability and increases challenges of robustness.

Example 1. If the purpose of a model based tool is to control the pressure in a gas tank, it is sufficient to model the pressure with the ideal gas law (or poten-

tially modified with a compressibility factor), lumping all gas components into one pseudo-component. However, if the purpose is to control e.g. the CO_2 fraction, one needs to include a component balance and have at least two components: CO_2 and the 'remaining'-component.

2.2 Techniques for developing balanced-complexity models

This section introduces some techniques that can be useful for developing balanced-complexity models. The techniques are illustrated with examples from an in-house Modelica library developed for online use (see Section 3):

Adaptivity: Candidate adaptivity parameters have significant influence on the solution, yet are known to be difficult or complicated to model with accuracy and/or are slowly time-varying. Which parameters to adapt is determined by analysis of the equation set, literature and by comparison with real-world data. Adaptivity has the ability to reduce model complexity as it may reduce the need for complex empirical correlations in the equation set.

> **Example 2.** Modeling multiphase flow in pipelines is complex, as key parameters such as pressure drop coefficients and gas-liquid velocity distributions depend on many factors that may be difficult to describe accurately with experimental correlations, and as these parameters may also vary with time. The ratio of gas velocity to liquid velocity in multiphase flow can depend on many factors such as flow-regime, Reynolds-numbers, incline angles or others. By choosing the slip factor, the ratio of gas velocity to liquid velocity, as an adaption parameter the challenge of accurately modeling this ratio is mitigated. As modeling the gas-liquid velocity distribution can be complex and can add to model uncertainty, the resulting online model with adaption in slip ratio need not be less accurate than offline counterparts.

> **Example 3.** Centrifugal compressor models are static and based on compressor maps of polytropic head versus volumetric rate, parameterized in compressor speed. The compressor maps supplied by equipment vendors may be subject to inaccuracies and slow changes over time due to wear and tear. A single adaption parameter is introduced to linearly scale the compressor map.

Thereby inaccuracies and time-varying effects in the compressor can be adjusted for in online compressor models.

Explicit models: Deriving model equations from physics often results in models which are differential-algebraic equations sets (DAEs). Solving such equation sets can be both time consuming and subject to numerical stability issues. It is desirable to re-formulate such models as ordinary differential equation sets (ODEs) to improve numerical speed and stability. Especially implicit algebraic equations requiring dedicated solvers should be avoided. Simple algebraic relations can often be solved by rearranging equations. Artificial dynamic variables can be introduced in more challenging cases to break algebraic loops.

State selection: Another key to avoid implicit equations is to formulate the problem explicitly in terms of states. State variables should be selected so that other dependent properties can be calculated explicitly. This is a common challenge particularly when calculating thermodynamic properties. For instance, if thermodynamic relations are explicit in pressure and temperature, pressures and temperatures should be chosen as states. The Modelica language has support for setting preferred state variables while still formulating derivatives using other variables. A Modelica compiler will automatically differentiate the differential equations in order to change the state variables to the preferred set, see [4].

Smoothing: When models are used in conjunction with optimization algorithms it is important that they are continuous and differentiable. To ensure this property, all equations used must be analyzed with regard to smoothness before use, and where needed, artificial transition functions can be included to enforce smoothness.

Right level of detail: For efficient models, the level of detail for a specific phenomenon in the model should match the importance of that particular phenomenon. As discussed in Section 2.1, phenomena which do not contribute to fulfill the purpose of the model should be left out, illustrated by the example below:

> **Example 4.** A common approach in process simulators is to model hydrocarbon fluids with a multi-component mixture, often with 10 or more

components. The high number of components leads to a large number of thermodynamic state variables. For phase equilibrium calculations the common approach is to use iterative algorithms for solving the resulting equation set.

A multiphase medium in an in-house model library is implemented using a low number of components: The gas phase normally contains only one high and one low molecular weight component. This is sufficient to make any gas mixture with an average molecular weight between the two components. A similar approach is taken with the oil/condensate phases, optionally with a water component to be used if water content in oil/condensate is of interest. In addition to the low number of state variables resulting from this approach, an advantage is that a phase equilibrium in a two component mixture can always be calculated explicitly. This is considered as a sufficient level of detail for the purposes of pressure and level control.

Utilize operational conditions: Knowledge of the operational conditions for which the model is applied can simplify the model considerably. It is unnecessary to include descriptions of operational conditions which will never occur. For example, if it is known that the model will be used for a process with strict temperature control, it will be a good approximation to drop the energy balance and use constant temperature in the model.

Pre-computation of properties: A common model simplification technique is to tabulate complex relations, for instance thermodynamic properties. In this way, complex calculations can be pre-computed, and when used online models can access the ready solutions. If tables are used, attention should be paid to the selection of table interpolation algorithm as to avoid non-smooth derivatives of the interpolated functions. Since searching through large tables is time consuming, simple function approximations is a good alternative.

Data-driven modeling: Data from operation of a process can be used for selecting the right model. Process data with excitations can reveal hints of what model structures can emulate the process. One could either look for a physical phenomenon giving the same response as the data, or consider introducing a semi-empirical model component which replicates the observed response. For empirical equations, care should be taken when extrapolating.

3 Efficient large-scale modeling by the use of Modelica

The approach to efficient large-scale modeling considered in this paper is outlined in Figure 1. The approach is based on implementing the *Functional-mock up interface (FMI)*[1] in software used in online control applications. An FMI-standard model component is shared as a *functional mock-up unit (FMU)*.
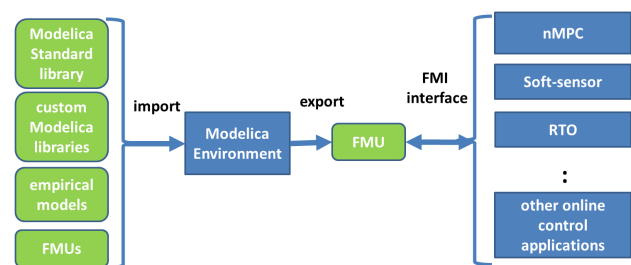

Figure 1: Flow of information between models (rounded edges) and applications (straight edges).

Since the translation from Modelica to FMI is done by a compiler, and as all low-level code to interface model and online application is model-independent and re-usable, the transition from Modelica to online applications can be made in a matter of minutes. This framework supports an iterative modeling work flow, as repeating the conversion from model to application multiple times is not workload-intensive.

Aside from the advantages of Modelica listed in Section 1, a benefit of designing models in a Modelica environment is that sub-modules can be imported from multiple external sources. The ability to import modules as FMUs means that the process owner, equipment suppliers or others can supply proprietary models as pre-compiled FMUs. This also opens an avenue for suppliers of process simulators to export their models seamlessly into control applications, provided they implement support for export of models as FMUs. For the reasons mentioned in Section 1, it will still be advantageous for such models to be designed with the techniques discussed in Section 2.

When designing large-scale models, it is often desirable to model selected subsystems or components
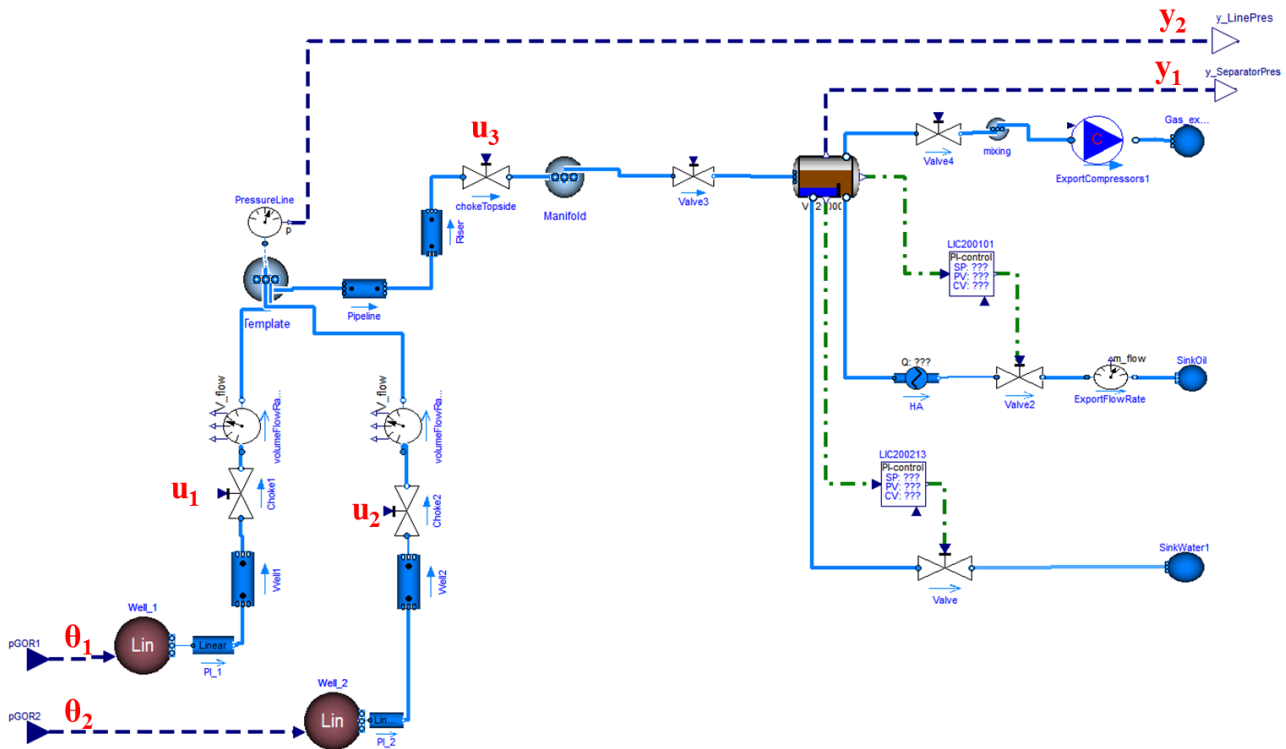
---

[1]see http://www.modelisar.com/

Figure 2: Overview subsea-pipeline-riser-separator system as implemented in DYMOLA, with piping (solid), handles to the estimator via FMI (dashed), and PI-control (dashdot).

of the larger system using empirical models, for instance fitted curves or state-space models inferred from data. An efficient manner of incorporating such sub-models in a larger Modelica-based framework is to express empirical models in the Modelica language. Exporting empirical models in Modelica-form is a task that can be automated by software for system identification. The modular buildup of Modelica allows such exported models to be seamlessly integrated with physics-based Modelica models.

4 Case-study: Estimation of gas-oil ratio in offshore oil and gas production

The aim of this case study is to illustrate that a large-scale balanced-complexity model which has been designed along the principles outlined in Section 2 can be implemented efficiently by the methods outlined in Section 3. The case considered is *stylized* in that for demonstration purposes, the estimator used has a relatively low number of fitted parameters and measurements.

The system considered is shown in Figure 2, and consists of the joint production of oil, gas and water from two different wells. The fluids from the wells are mixed in a subsea template before traveling along a horizontal pipeline, through a vertical riser, into a topside manifold before reaching the topside processing plant. The production rates from the two wells are not measured directly, yet these flow rates are of great interest as they determine production revenues and the feed rates to which the process plant must adapt.

There is a significant static pressure drop from the reservoir to the sea bead (elevation often being of order thousands of meters) and from the sea bed to the floating production unit (elevation often of order hundreds of meters). The static pressure dropdepends on the ratio of gas-to-liquid, and as the proportion of total production that is water is often fairly constant, it should be possible to infer about the gas-oil ratio by modeling its relationship to pressure in well and pipeline. Since pressure in the pipeline depends on the settings of chokes on each well and upstream of the separator, these chokes must also be modeled.

A typical full-complexity multiphase pipe flow simulator could for the well-pipeline-separator system considered have hundreds or thousands of control volumes, and a full-complexity thermodynamic model could have on the order of 20 states for each control volume. Thermodynamic relations would in a full-

complexity model depend on implicit relations, and a large number of different empirical closure relations for different conditions would be used in multiphase flow models.

From our perspective such a full-complexity model would be unsuitable for the purposes of estimating gas-oil ratio online, due to the issues mentioned in Section 1.

4.1 Modeling

Modules from an in-house Modelica library were used and put together with the aim of finding the right level of detail to achieve the desired goal of estimating gas-oil ratio. It was elected to model flow as a two-phase flow, lumping oil and water flows into a single liquid flow. Modules describing wells, horizontal and vertical pipelines and chokes were combined to create the large-scale model. Each of these modules were originally designed by combining first-principles with empirical closure relations from the literature that were revised for simplicity, to obtain smoothness and to avoid implicit relations. The number of different closure relations was kept as low as possible, and the resulting models were validated module-for-module against real-world data. The modules include handles for introducing adaptivity as needed through adjustable parameters such as gas-liquid velocity ratios, valve coefficients and friction factors. Adapting the mentioned parameters was omitted here for simplicity.

Some examples of the balanced-complexity principles in the current case-study follows:

- **Exclude flashing.** From experience and analysis of real-world data similar to this case, the flashing (evaporation of dissolved gas in the oil) as the pressure drops in the pipeline is not expected to be significant relative to amount of free gas. Excluding flashing from the model was therefore judged to be the right level of detail.

- **Few discrete mixing volumes.** Riser and pipeline models are finite-volume spatial discretizations of the underlying partial-differential flow equations, and the number of discrete volumes for each of these modules are design parameters that the user should select at design while evaluating resulting model accuracy. It is our experience that no fine discretization is required for estimators such as considered here to work. Lumping pipeline submodels into two or even just one volume is often found to be the right level of detail. For each volume in each sub-model, a mass-balance equation is formulated and a simplified thermodynamic relation with a low number of components that is smooth and explicit, as described in Example 4, was used.

- **Limiting the scope of the model.** The three-phase separator model uses a thermodynamic equilibrium equation for flashing/vaporization, in combination with a mass balance that takes in account separator geometry. Since the estimator considers the portion of the offshore oil and gas system spanning from wells to the separator, it was not considered necessary to model further downstream process equipment for the desired estimator, motivated by the concept of the purpose dictating the model.

All the models were expressed in equation-form in the Modelica language, and the translation capabilities of DYMOLA were used to convert this equation-based model into an imperative, C-language code that is suitable for online use. The model shown as drawn by DYMOLA is shown in Figure 2. That the imperative code of the model is generated rather than hand-coded directly is useful for iteratively deciding the right level of detail in the model. The degree of model detail is easily adjustable in the high-level, modular, equation based language Modelica, from which multiple estimators based on different low-level implementations of the model in C can be compared.

4.2 Estimation

Simulations were done for a model with only a single node for pipeline and riser. The resulting model has 48 states, DYMOLA choosing five states (pressure + 4 component mass fractions) for each of the nodes: well 1, well 2, subsea manifold, pipeline, riser, topside manifold and inlet separator.

Pressures at the topside separator (y_1) and subsea manifold (y_2) were chosen as outputs. Parameters were chosen as gas-oil ratios of well 1 (θ_1) and well 2 (θ_1). Choke openings of valves on well 1 (u_1), well 2 (u_2) and the topside valve (u_3) are varied during the simulation. The estimator used is a recursive Extended Kalman Filter (EKF). The model was implemented in Modelica, compiled as an FMU using DYMOLA, and interfaced with a generic and re-usable recursive Extended Kalman Filter (EKF).

The dataset considered is synthetic, generated by simulating a copy of the model where the gas-oil ratios of both wells were set equal to 811. Noise of 2% of average amplitude was added to both pressures.
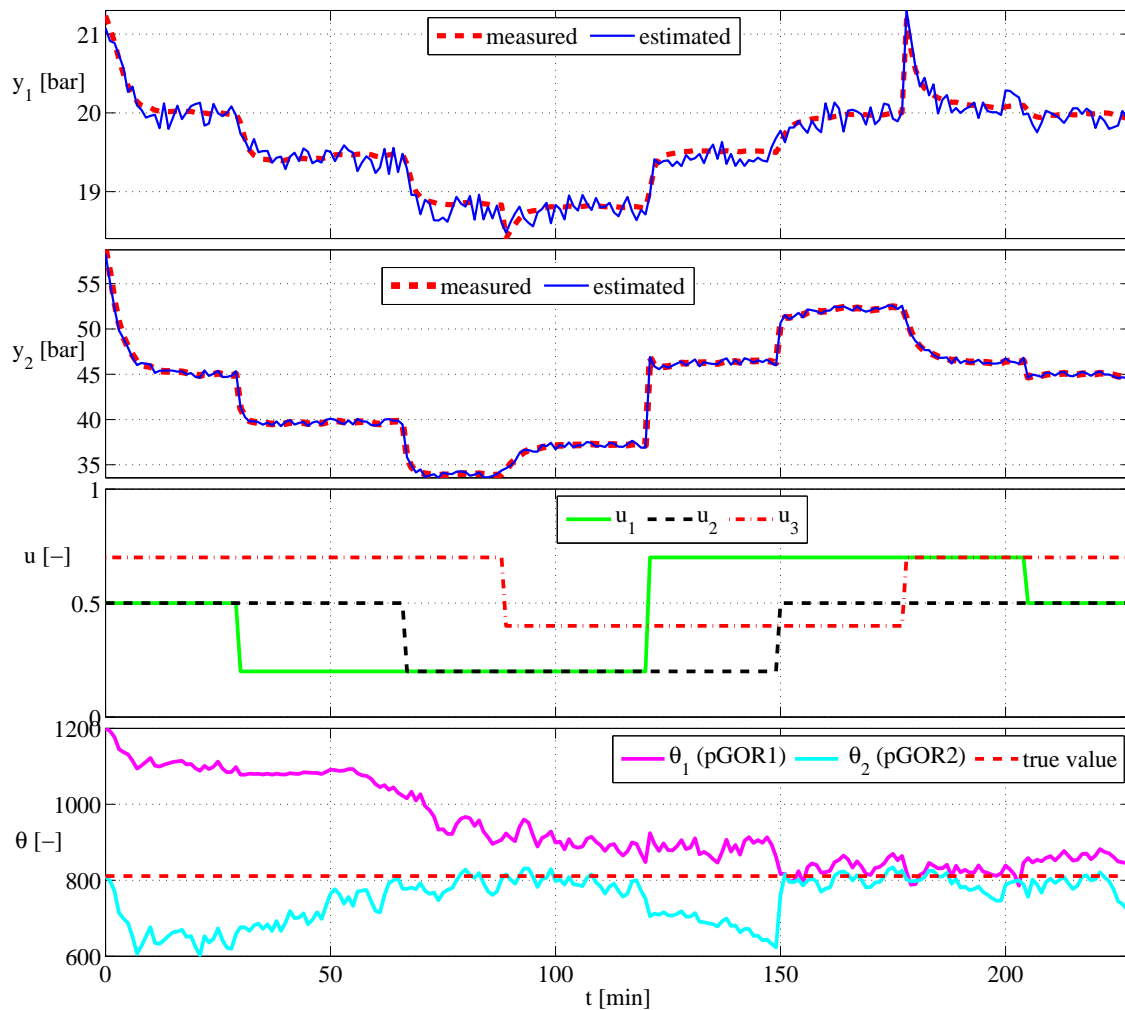
Figure 3: Simulation results. Top subplot shows measured and estimated separator pressures. Subplot 2 shows measured and estimated line pressures. Subplot 3 shows relative choke opening of wells 1 and 2 (u_1 and u_2) and of the topside valve (u_3). Subplot 4 shows recursive estimates of gas-oil ratios of wells 1 and 2 compared with the true value.

4.3 Simulations

Estimated and measured pressures and estimated gas-oil ratios for wells 1 and 2 are shown in Figure 3. The initial estimate for the gas-oil ratio of well 1 was set to 1200, while the actual gas-oil ratio for both wells is 811. The inaccurate initial estimate of gas-oil ratio resulted in an offset between measured and modeled pressures, which the estimator attempts to correct during simulation. The excitation shown in Figure 3 made it possible to uniquely determine gas-oil ratios for both wells from the data set, and as the simulation progresses, the estimated gas-oil ratios move toward the real value of 811.

4.4 Discussion

The main contribution of this case study is the technology and workflow used to implement an online model including Kalman Filter estimators. The solution was implemented in a low-level language suitable for online use, yet no line of low-level code was manually written. The model used has 48 states, and manually implementing low-level model code would be a challenging task already at this scale if you consider that modeling requires several design iterations, collaboration among multiple designers, code-reuse and code validation. Our experience indicates that the approach could accommodate working efficiently on

much larger models as well.

5 Conclusion

Balanced-complexity modeling is an approach to bring physics-based models online while adhering to requirements for online use. Modelica and FMI have advantages that aid the development of such systems: efficient model development; reuse of models; and efficient integration with other software. By calling attention to this topic it is hoped for an increasd recognition for online applications with purpose-built models developed with Modelica and FMI.

References

[1] B. Bringedal, E. Storkaas, M. Dalsmo, M. Aarset, and H. M. With. Recent developments in control and monitoring of remote subsea fields. In *SPE Intelligent Energy Conference and Exhibition*, Utrecht, The Netherlands, 2010.

[2] F. Casella, F. Donida, and J. Akesson. Object-oriented modeling and optimal control: A case study in powerplant start-up. In *Proc. 18th IFAC World Congress, Milano, Italy*, volume 18, 2011.

[3] S. Elgsæter, O. Slupphaug, and T. A. Johansen. Challenges in parameter estimation of models for offshore oil and gas production optimization. In *International Petroleum Technology Conference*, Dubai, 2007.

[4] H. Elmqvist, H. Tummescheit, and M. Otter. Object-oriented modeling of thermo-fluid systems. *Proceedings of 3rd Int. Modelica Conference*, pages 269–286, 2003.

[5] B. A. Foss and T. S. Schei. Putting nonlinear model predictive control into use. In *Assessment and Future Directions Nonlinear Model Predictive Control*, LNCIS 358, pages 407–417. Springer Verlag, 2007.

[6] R. Franke, B. S. Babji, M. Antoine, and A. Isaksson. Model-based online applications in the abb dynamic optimization framework. In *Modelica'2008*, Bielefeld, Germany, 2008.

[7] J. M. Godhavn, A. Pavlov, G. O. Kaasa, and N. L. Rolland. Drilling seeking automatic control solutions. In *Proc. 18th IFAC World Congress, Milano, Italy*, volume 18, Milano,Italy, 2011.

[8] L. Imsland, P. Kittilsen, and T. S. Schei. Model-based optimizing control and estimation using modelica models. In *Modelica'2008*, Bielefeld, Germany, 2008.

[9] E. Jahanashahi and S. Skogestad. Simplified dynamical models for control of severe slugging in multiphase risers. In *Proc. 18th IFAC World Congress, Milano, Italy*, volume 18, Milano,Italy, 2011.

[10] K. Krueger, M. Rode, R. Franke, and B. A. Foss. Optimization of boiler start-up using a nonlinear boiler model and hard constraints. *Energy*, 29(12-15):2239–2251, 2004.

[11] S. McArdle, D. Cameron, and K. Meyer. The life cycle simulator: From concept to commissioning... and beyond. In *SPE Intelligent Energy Conference and Exhibition*, pages 246–267, Utrecht, The Netherlands, 2010.

[12] P. Meum, P. Tøndel, J. M. Godhavn, and O. M. Aaamo. Optimization of smart well production through nonlinear model predictive control. In *Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands, 2008.

[13] A. Mjaavatten, Robert Aasheim, Steinar Saelid, and Oddvar Gronning. Model for gas coning and rate-dependent gas/oil ratio in an oil-rim reservoir. *SPE Reservoir Evaluation & Engineering*, 11(5), 2008.

[14] Z. K. Nagy, B. Mahn, R. Franke, and F. Allgöwer. Nonlinear model predicitive control of batch processes: an industrial case study. In *Proc. 16th IFAC World Congress, Prague, Czech Republic*, volume 16, 2005.

[15] R. van der Linden and A. Leemhuis. The use of model predictive control for asset production optimization: Application to a thin-rim oil field case. In *SPE Annual Technical Conference and Exhibition*, Florence, Italy, 2010.

[16] A. Willersrud, L. Imsland, S. O. Hauger, and P. Kittlsen. Short-term production optimization of offshore oil and gas production using nonlinear model predictive control. In *Proc. 18th IFAC World Congress, Milano, Italy*, volume 18, Milano,Italy, 2011.