

Master controller for offshore wind power and hybrid grids

Rüdiger Franke¹ Marcin Bartosz² Rasmus Nyström³

ABB, ^{1,3}{ruediger.franke,rasmus.nystroem}@de.abb.com, ²marcin.bartosz@pl.abb.com

Abstract

Offshore power generation and transmission requires long subsea cables. When using AC, the inductance of the cables results in reactive power that contributes to the load of the cables and increases the voltage at bus-bars. At the same time, offshore power systems are becoming increasingly complex. They are evolving from dedicated collector grids and transmission cables for wind farms through shared use of transmission cables by multiple wind farms to hybrid offshore grids that connect wind farms to multiple countries.

This paper discusses operational challenges of offshore grids and how they are solved with model-based master control solving state estimation and optimal power flow problems in real-time.

The optimization model is built with Modelica using the PowerSystems library and FMI for its deployment in the control system. The optimization solver runs model evaluations and sensitivity analyses over different time intervals in parallel.

The Kriegers Flak Combined Grid Solution (KF CGS) serves as example. It is one of the world's first hybrid offshore grids. The methods have been implemented as Master Controller for Interconnected Operations (MIO). After extensive tests with a digital twin prototype, MIO has been deployed in a geo-redundant system with ABB Ability™ OPTIMAX® and System 800xA.

Keywords: *Offshore grids, Hybrid Interconnectors, Optimal Power Flow, Modelica, PowerSystems, state estimation, optimal control.*

1 Introduction

Offshore power and intercontinental power transmission are playing an important role for establishing climate neutral energy systems. The European Wind Power Action Plan sets ambitious goals for new installations. The North Sea Energy Consortium alone heads for at least 260 GW of offshore wind energy by 2050, with intermediate targets of at least 193 GW by 2040 and 76 GW by 2030, starting from 33 GW in 2023 (European Commission, 2023).

In December 2020, the Kriegers Flak Combined Grid Solution (KF CGS) was inaugurated by the transmission system operators 50Hertz, Energinet, the German Federal Minister of Economics and the Danish Minister for

Climate, Energy and Utilities. It transports wind power from the four offshore wind farms (Baltic 1 and 2, Kriegers Flak A and B) with a total capacity of 950 MW to shore. Additionally, it promotes energy trade between Germany and Denmark. At the heart of the connection of these two energy grids through the Baltic Sea there is a Master Controller for Interconnected Operations (MIO).

This paper discusses the methods and technologies employed in the first of its kind master controller for interconnected operation (Marten et al, 2018).

2 Kriegers Flak Combined Grid Solution

The KF CGS connects the transmission cables of the German wind farms Baltic 1 and Baltic 2 with the transmission cables of the Danish wind farms Kriegers Flak A and B. This creates a hybrid offshore grid for the collection of wind power and the exchange of power between the northern and the continental European transmission grids, see Figure 1.

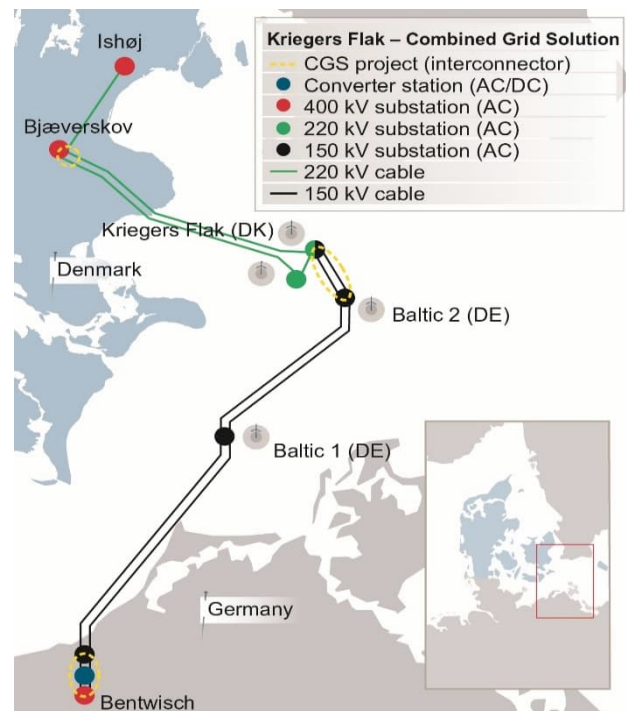


Figure 1: Overview of Kriegers Flak Combined Grid Solution (KF CGS) encircled in yellow. Source: 50Hertz

The KF CGS imposes several operational challenges. A Back-to-Back (BtB) HVDC converter station in Bentwisch enables the connection of the two unsynchronized AC transmission grids. It controls the power exchange at the interconnector in the Baltic Sea by manipulating power flow and voltage from the German coast. The meshed submarine grid interconnection is (n - 0) secure, meaning that a single failure can result in a system outage. Figure 2 shows a single line diagram.

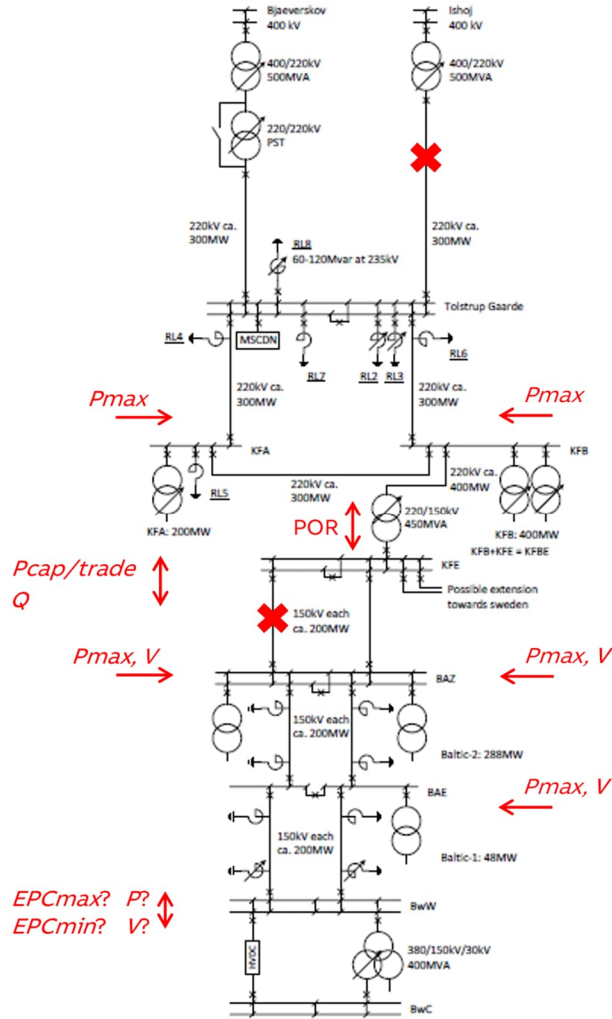


Figure 2: Single line diagram of KF CGS outlining operational challenges. Source: 50Hertz

This results in several tasks for the master controller. First, it provides forecasts for available transfer capacity P_{cap} at the point of reference (POR) using wind forecasts as input. Second, it controls the traded power P_{trade} and reactive power Q in real time, thereby accounting for cable load limits, forecast errors, contingencies and other disturbances in the system. Primarily, the master controller manipulates active power and voltage at the BtB converter. Additionally, it may limit the maximum power P_{max} of wind farms. Finally, it provides limits for emergency power control (EPC) in real time.

3 Model-based control with Modelica

Modelica has been widely used for model-based control for many years. Applications range from the treatment of optimal control problems (Franke, 2002) to the creation of inverse models and the generation of embedded controller code (Otter et al, 2012).

Modelica is also used for the modeling and simulation of power systems ranging from microgrids (Fachini et al 2023) to transmission systems (Feghali et al, 2023).

This section outlines the technologies used to implement the master controller for KF CGS.

3.1 PowerSystems library

The Modelica PowerSystems library (R. Franke, H. Wiesmann, 2014) provides a flexible way to model electric power systems at different levels of detail, up to advanced modeling of electromagnetic transients (B. Bachmann, H. Wiesmann, 2000).

This is achieved by different phase systems as shown in the library structure in Figure 3.

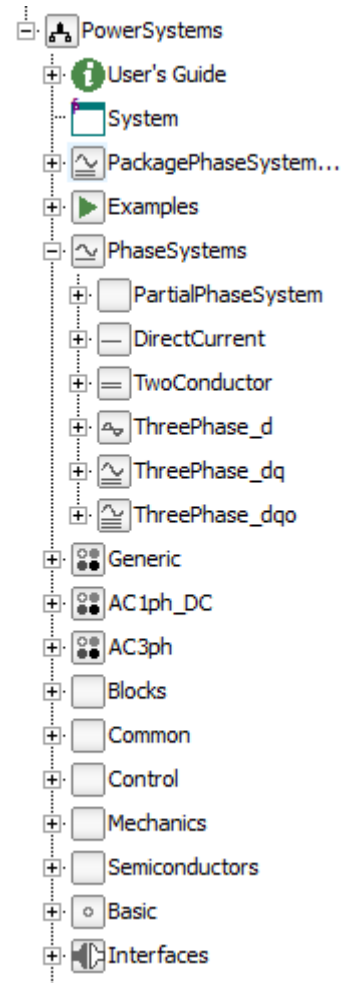


Figure 3: PowerSystems library structure.

Each phase system defines the number n of independent current and voltage components, the number m of reference angles and appropriate supporting functions like the j “operator”.

The interfaces define a general power terminal.

```
connector Terminal
  replaceable package PhaseSystem;
  PhaseSystem.Voltage
    v[PhaseSystem.n];
  flow PhaseSystem.Current
    i[PhaseSystem.n];
  PhaseSystem.ReferenceAngle
    theta[PhaseSystem.m]
    if PhaseSystem.m > 0;
end Terminal;
```

Table 1 summarizes the PhaseSystems that are predefined in the PowerSystems library.

Table 1: Phase systems by PowerSystems library.

PhaseSystem	n	m	Description
DirectCurrent	1	0	One voltage and one current component in natural coordinates
TwoConductor	2	0	Two voltage and two current components for Spot AC1ph_DC components
ThreePhase_d	1	0	One modal component for active power — like DirectCurrent, but converting voltage values to three phase
ThreePhase_dq	2	1	Two modal components for active and reactive power; one reference angle for frequency — cf. complex phasors with variable frequency
ThreePhase_dq0	3	2	Three modal components for active, reactive and dc power; two reference angles for transient dq0 components

Note the use of the function j that generalizes complex calculations known from quasi-static AC models to arbitrary phase systems. ThreePhase_dq, with two model components for active and reactive power defines a multiplication with the complex j :

```
function j
  input Real x[n];
  output Real y[n];
algorithm
  y := {-x[2], x[1]};
end j;
```

The simpler ThreePhase_d neglecting reactive power defines:

```
function j
  input Real x[n];
  output Real y[n];
algorithm
  y := zeros(n);
end j;
```

The more detailed PhaseSystem_dq0 (direct-quadrature-zero) also considers a component for dc power in asymmetric systems, besides active and reactive power. It defines:

```
function j
  input Real x[n];
  output Real y[n];
algorithm
  y := cat(1, {-x[2], x[1]},
           zeros(size(x,1)-2));
end j;
```

A general base model with two terminals reads

```
partial model PartialTwoTerminal
  replaceable package PhaseSystem =
    PackagePhaseSystem;
  package PS = PhaseSystem;
  function j = PhaseSystem.j;
  Terminal term_p(
    redeclare package PhaseSystem = PS);
  Terminal term_n(
    redeclare package PhaseSystem = PS);
  SI.AngularFrequency omegaRef;
equation
  v = term_p.v - term_n.v;
  i = term_p.i;
  if PS.m > 0 then
    omegaRef =
      der(PS.thetaRef(term_p.theta));
  else
    omegaRef = 0;
  end if;
  Connections.branch(term_p.theta,
    term_n.theta);
end PartialTwoTerminal;
```

A generic steady-state impedance can now be formulated as

```
model Impedance
  extends PartialTwoTerminal;
  parameter SI.Resistance R;
  parameter SI.Inductance L;
equation
  v = R*i + omegaRef*L*j(i);
  zeros(PS.n) = term_p.i + term_n.i;
```

```
term_p.theta = term_n.theta;
end Impedance;
```

Accordingly, a generic steady-state admittance reads

```
model Admittance
  extends PartialTwoTerminal;
  parameter SI.Conductance G;
  parameter SI.Capacitance C;
equation
  i = G*v + omegaRef*C*j(v);
  zeros(PS.n) = term_p.i + term_n.i;
  term_p.theta = term_n.theta;
end Admittance;
```

A forced switch can be modeled with curve parameter s

```
model Switch
  extends PartialTwoTerminal;
  Modelica.Blocks.Interfaces.RealInput
    closed(min=0, max=1);
protected
  Real[PS.n] s;
equation
  {v, i} = {(1-closed)*PS.v_nominal*s,
    closed*PS.i_nominal*s};
end Switch;
```

The package AC3ph provides many relevant component models like Lines, Nodes, Breakers, Shunts and Transformers for the application at hand, see Figure 4.

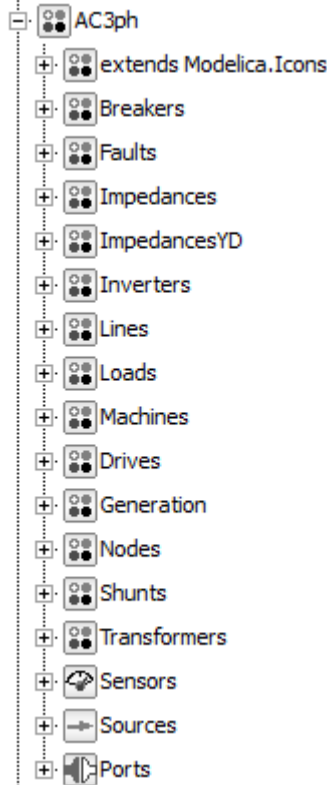


Figure 4: PowerSystems AC3ph component models.

3.2 Dynamic Optimization

Modelica distinguishes between equation-based models and executable simulation code. A model translator generates executable code. Beyond simulation runs in the Modelica tool at hand, the executable code can also be exported to other tools. It has been used for dynamic optimization in several industrial applications, e.g. (Franke, 2002). The introduction of the Functional Model Interface (FMI) and synchronous language features standardized the interface of executable discrete-time models (Franke et al, 2017).

Alternatively, (Åkesson et al, 2010) proposed a syntactical extension of the equation-based Modelica language to Optimica for optimization. Recent work couples Modelica with Matlab to implement model predictive control (Alizadeh et al, 2023).

Numerical solution as large-scale nonlinear program

The nonlinear solver HQP used by ABB OPTIMAX[®] treats discrete-time optimal control problems as large-scale mathematical programs. Discrete-time states and controls are collected in the state vector x and the control vector u , respectively. This gives the discrete-time optimal control problem:

$$J = f_0(x^K) + \sum_{k=0}^{K-1} f_0(x^k, u^k) \rightarrow \min_{x^0, u^k}$$

with the discrete-time state equations

$$\begin{aligned} x^{k+1} &= f^k(x^k, u^k), \\ y^k &= g^k(x^k, u^k), \quad k = 0, \dots, K-1 \end{aligned}$$

and the constraints

$$\begin{aligned} c^k(x^k, u^k) &\geq 0, \quad k = 0, \dots, K-1 \\ c^K(x^K) &\geq 0 \end{aligned} \tag{1}$$

The executable model imported as Functional Model Unit (FMU) for model exchange implements the state equations f^k . It is important to note that the explicit relationship between f^k and x^{k+1} results in linear optimization constraints that can be exploited for efficient parallel optimization. The model can be evaluated in parallel in each time step. Additionally, the FMU provides optimization objective functions f_0 and the constraints c^k as outputs.

The states and the control inputs of all time intervals are collected into one large vector of optimization variables

$$v = (x^0, u^0, x^1, u^1, \dots, x^{K-1}, u^{K-1}, x^K). \tag{2}$$

This results in the mathematical program

$$\begin{aligned} J(v) &\rightarrow \min_v & J: \mathbb{R}^{n_v} &\rightarrow \mathbb{R}^1 \\ h(v) &= 0 & h: \mathbb{R}^{n_v} &\rightarrow \mathbb{R}^{m_e} \\ g(v) &\geq 0 & g: \mathbb{R}^{n_v} &\rightarrow \mathbb{R}^{m_i} \end{aligned} \quad (3)$$

HQP treats large-scale nonlinear optimization with Sequential Quadratic Programming (SQP).

Basing on the Lagrangian

$$\begin{aligned} L(v, \lambda, \mu) &= J(v) - \lambda^T h(v) - \mu^T g(v) \\ L: \mathbb{R}^{n_v} \times \mathbb{R}^{m_e} \times \mathbb{R}^{m_i} &\rightarrow \mathbb{R}^1 \end{aligned} \quad (4)$$

the solution must fulfill the Karush Kuhn Tucker (KKT) conditions

$$\begin{aligned} \nabla_v L(v, \lambda, \mu) &= \nabla J(v) - \nabla h(v)^T \lambda - \nabla g(v)^T \mu = 0 \\ \nabla_\lambda L(v, \lambda, \mu) &= -h(v) = 0 \\ g(v) &\geq 0 \\ \mu &\geq 0 \\ g(v)^T \mu &= 0 \end{aligned} \quad (5)$$

HQP applies Lagrange Newton iterations

$$\begin{aligned} \nabla^2 L(v, \lambda) \begin{pmatrix} \Delta v \\ \Delta \lambda \end{pmatrix} &= -\nabla L(v, \lambda) \\ \begin{pmatrix} v^+ \\ \lambda^+ \end{pmatrix} &:= \begin{pmatrix} v + \Delta v \\ \lambda + \Delta \lambda \end{pmatrix} \end{aligned} \quad (6)$$

to find the solution. The Lagrange Newton iteration is given here for the case $m_i=0$. HQP augments the Lagrangian to treat inequality constraints with an Interior Point method. The Jacobian $\nabla J(v)$ and the Jacobian matrices $\nabla h(v)$ and $\nabla g(v)$ are formed analytically exploiting directional derivatives provided by the FMU per time interval. The Jacobians of each time step are collected into large sparse matrices for the overall dynamic optimization problem giving

$$\nabla h(v) = \begin{pmatrix} \frac{\partial f^0}{\partial x^0} \frac{\partial f^0}{\partial u^0} & -I & \\ & \ddots & \\ & & \frac{\partial f^{K-1}}{\partial x^{K-1}} \frac{\partial f^{K-1}}{\partial u^{K-1}} & -I \end{pmatrix} \quad (7)$$

and

$$\nabla g(v) = \begin{pmatrix} \frac{\partial \bar{c}^0}{\partial x^0} \frac{\partial \bar{c}^0}{\partial u^0} & & \\ & \ddots & \\ & & \frac{\partial \bar{c}^{K-1}}{\partial x^{K-1}} \frac{\partial \bar{c}^{K-1}}{\partial u^{K-1}} & \\ & & & \frac{\partial \bar{c}^K}{\partial x^K} \end{pmatrix} \quad (8)$$

The Hessian of the Lagrangian $\nabla^2 L(v, \lambda, \mu)$ is formed numerically applying a rank 2 update in each time interval based on the progress over subsequent iterations. As the linear coupling between subsequent time intervals plays no role for second order derivatives in the Hessian, the large-scale nonlinear program is partial separable. This allows an efficient multi-rank update of the large Hessian of the Lagrangian. This is also why no analytical second order derivatives are required.

4 Application to KF CGS

A model of the Kriegers Flak Combined Grid Solution was implemented in OpenModelica using the PowerSystems library. For the operational optimization considered here, we assume symmetric load of the three phases and quasi-stationary behavior. This is achieved with the phase system ThreePhase_dq. Once selected globally, the connectors and most components adjust to it automatically, for instance cables, shunts and breakers. Some components, such as inverters, have multiple implementations in the PowerSystems library. The simplified InverterAverage uses time-averaged variables, whereas the rigorous Inverter model implements fast switching in the time domain. The phase system ThreePhase_dq0 would cover transient scenarios and unsymmetric loads at the cost of larger models involving fast dynamics.

The model is then exported as FMU for model exchange with the C++ runtime. Several activities are configured for state estimation, predictive planning optimization and real-time optimization.

Figure 5 shows a screenshot of the model. To keep the size of the model diagram better manageable, some details were encapsulated into sub models. Particularly a generic AC transmission line was formed from a π -line combined with an optional shunt and an optional switch at each end, see Figure 6.

The KF CGS model was tested in OpenModelica. It consists of 4722 variables and equations. Many of them are trivial equalities going back to connections between component models.

The translator identifies 788 non-trivial equations, 144 model inputs and 335 model outputs. For dynamic optimization, these numbers multiply with the number of time steps K , such as $K=288$ for three days with a step size of 15 minutes.

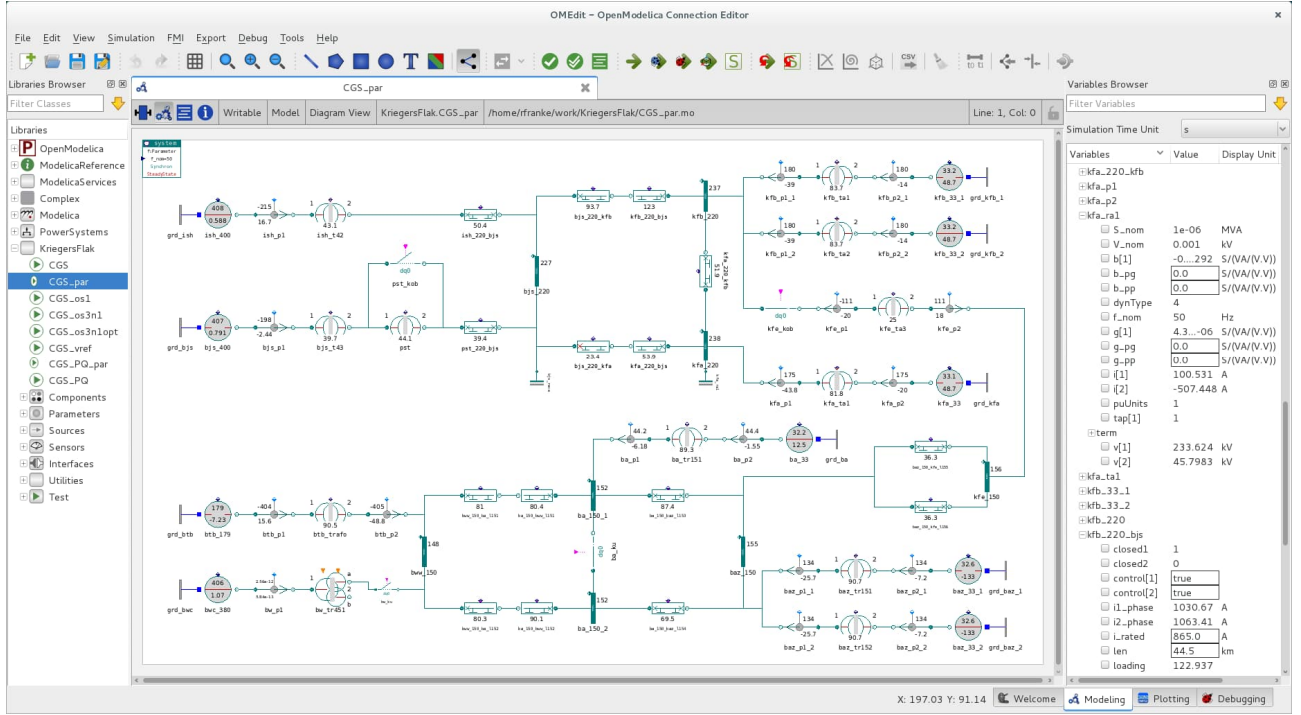


Figure 5: Screenshot of OpenModelica with KF CGS model.

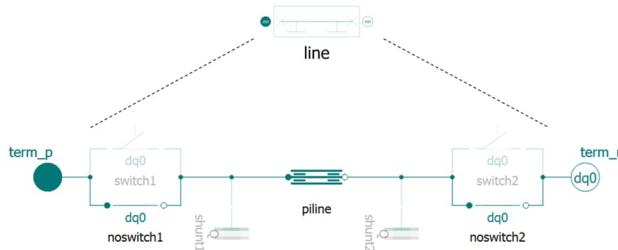


Figure 6: Generic sub model of an AC line with optional shunts and switches. Model icon at the top, model diagram at the bottom.

The variables relevant for optimization objective and constraints are quadratic functions of the internal voltage and current phasors. This gives active power and reactive power

$$\begin{aligned} P &= v \cdot i \\ Q &= -j(v) \cdot i \end{aligned} \quad (9)$$

along with absolute voltages, e.g. measured at busbars, and absolute currents, e.g. determining cable loads

$$\begin{aligned} V &= \sqrt{v \cdot v} \\ I &= \sqrt{i \cdot i} \end{aligned} \quad (10)$$

The steady-state system model depicted in Figure 5 leads to algebraic loops. Tearing reduces mutual dependencies to one linear equation system of dimension 81.

The variables of this equation system are curve parameters s of switches at the lines, besides currents i and voltages v of transformers, see Section 3.1. The system is linear thanks to the use of rectangular dq coordinates. Overall, this gives a quadratically constrained quadratic optimization problem that allows a fast and robust numerical solution.

Figure 7 shows the principle of the closed loop control. Measurements are absolute voltages, currents and power flows. Tuned model inputs u are voltage phasors in dq coordinates. Optimized variables y are set points for the BtB converter and wind farms.

The translated model is deployed with ABB OPTI-MAX[®] in a geo-redundant setup. Several optimization activities are running using multiple instances of the same model.

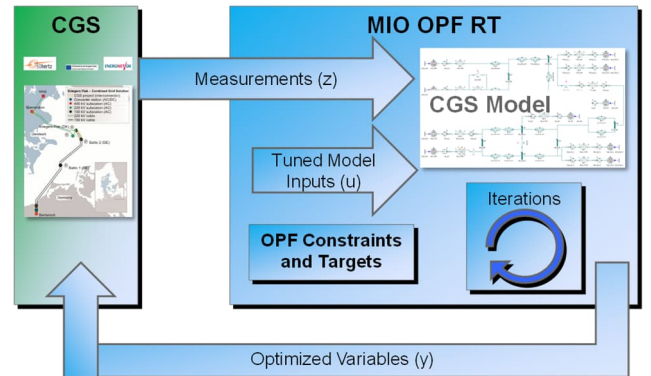


Figure 7: Principle of closed loop control with optimal power flow (OPF).

Table 2: Speedup obtained with parallel optimization and different solver configurations.

Speedup for different solver configurations	Finite Differences	AD with refactoring	AD with factor reuse
Sequential shooting, Dense model blocks, no AD, 1 CPU	1.0		
Sequential shooting, Sparse model blocks, 1 CPU	1.9	1.3	3.9
Parallel multiple shooting, Sparse model blocks, 2 CPUs	2.8	2.2	4.6
Parallel multiple shooting, Sparse model blocks, 5 CPUs	4.6	3.7	6.3
Parallel multiple shooting, Sparse model blocks, 20 CPUs	7.0	6.5	7.6

4.1 Planning Optimization

Based on given wind power forecasts and maintenance schedules, the maximum remaining power transfer capacities from Denmark to Germany and from Germany to Denmark are obtained, maximizing the power flow through the system subject to utilization of wind, constraints on cable loadings and limits on busbar voltages.

The planning optimization turns out to be the most time critical task as the coverage of a time horizon of up to three days with a resolution of 15 minutes results in a large-scale dynamic optimization problem. The formulation as discrete-time optimal control problem (1) with the large vector of variables (2) allows the separation of the time horizon and the evaluation of the model in parallel for each time step. The method is known as Multiple Parallel Shooting.

Table 2 summarizes results obtained on a 20-core high performance server. A speedup of 1.9 is achieved by not only exploiting sparsity from the time staggered structure of the discrete-time optimal control problem, but also sparsity of Jacobian matrices at each time step as given in the modelDescription.xml file of the FMU. Coloring in particular leads to a reduction of the number of directional derivatives required to obtain the whole Jacobian matrix. Interestingly, Automatic Differentiation (AD) turned out slower than Finite Differences initially. A speedup to 3.9 is achieved when re-using factors of the 82-dimensional linear equation system for subsequent determinations of directional derivatives.

The solution time can be further reduced and a speedup of up to 7.6 is achieved by increasing the number of CPU cores to 20. The speedup of 7.0 for Finite Differences is almost as high with 20 CPU cores. This is achieved by higher utilization of parallel cores for repeated factorizations of the linear equation system.

Automatic Differentiation with re-use of factors achieves a high speedup of 6.3 already with 5 CPU cores. This leaves the remaining cores for other tasks, like the solution of two capacity planning problems towards Denmark and towards Germany at the same time.

4.2 Real-time Optimal Control

Uncertainties of wind forecasts, the intermittent nature of wind as well as possible disturbances are addressed by re-optimizing set points in closed loop control during the actual operation. Additionally, optimization is employed to determine limits for the underlying control during a possible emergency. Limits for power transfer change dynamically with the infeed of wind as well as with the switching structure of cables, shunts and transformers.

State estimation: existing measurements of absolute voltages and currents are used to estimate system wide voltages and currents in dq coordinates. This is achieved by minimizing the residuals between measurements of absolute quantities and respective model outputs. From a control point of view, the electrical system is in steady-state. Dynamic states arise from rate-of-change bounds of controlled electrical assets.

Real-time optimization: starting from the estimated state, optimal power and voltage set points for the BtB converter and possibly required curtailments of wind farms are determined every 5 seconds. The underlying optimal power flow problem maintains the given target value P_{trade} at the point of reference while minimizing losses in the system subject to constraints on cable loadings and busbar voltages.

Emergency Power Control (EPC): starting from the estimated state, the maximum power transfer capacities towards Denmark and towards Germany are determined subject to constraints on cable loadings and busbar voltages in real-time and communicated as EPC limit to the BtB converter.

4.3 Digital Twin Prototype

A digital twin was employed during the engineering of MIO. It comprised the FMI model, the OPTIMAX[®] runtime and the communication protocol along with existing grid simulation software of 50Hertz and Energinet. This made it possible to test a vast number of operational scenarios with simulation prior to the commissioning of the actual system. It was key to shortening the commissioning time and for ensuring the reliable operation of this complex offshore grid.

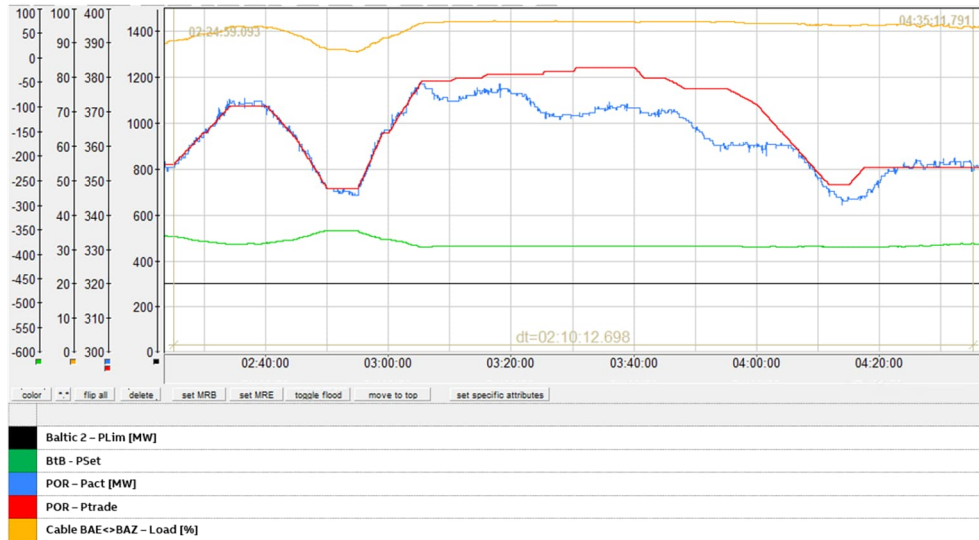


Figure 8: Real-time optimization: Overseeing planned energy transfer, respectful of network constraints.

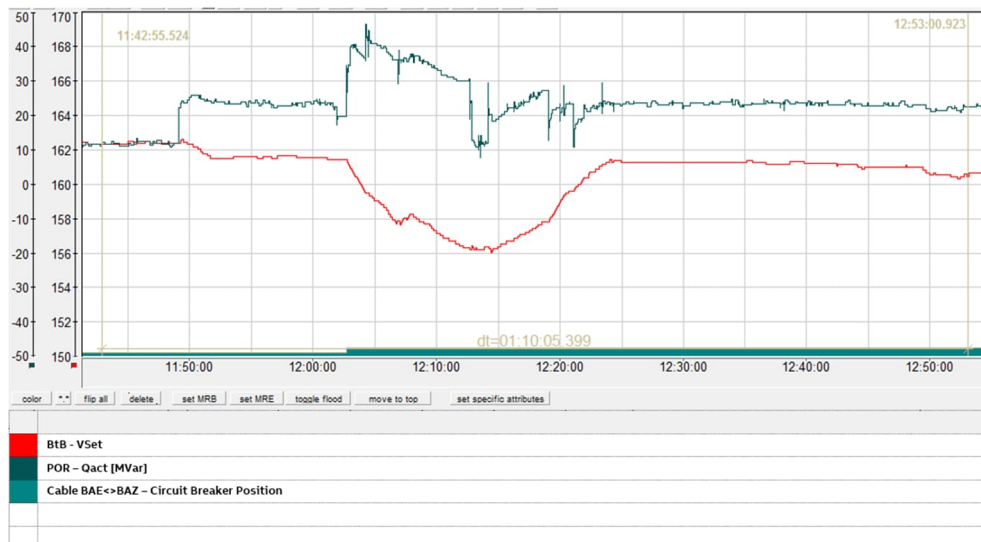


Figure 9: Real-time optimization: BtB voltage control and POR reactive power constraints.

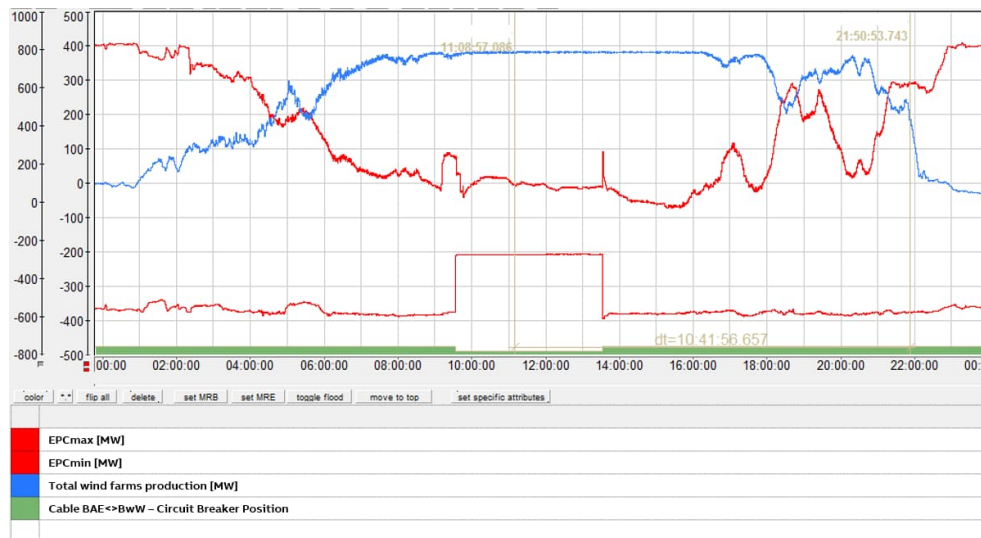


Figure 10: Emergency Power Control: Adjustment of the safe operating limit considering operational conditions.

5 Operational Examples

This section discusses several examples of the actual operation of the Kriegers Flak hybrid offshore grid with real-time optimal power flow, maintaining planned power transfers while prioritizing wind power and considering limits on cable loadings. See also the single line diagram in Figure 2 for relating acronyms used in this section to the overall system.

The plots show active power in MW, reactive power in Mvar, voltage in kV, line loading in % and switch position as 0 (off) or 1 (on).

Real-time optimization: Figure 8 shows an example, where power transfer is optimized. MIO manipulates the BtB such that the actual power transfer via POR follows the planned transfer until 03:05. Then the load on cable BAE—BAZ reaches its limit as the actual wind power is higher than expected during the planning. The wind farm Baltic 2 is not curtailed. Instead, MIO manipulates the BtB to maintain the cable load limit, resulting in lower than planned power transfer until 4:20.

Figure 9 illustrates BtB voltage control during the connection of the BAE—BAZ cable at 12:03. MIO manipulates the BtB to limit the reactive power flow through POR by reducing the voltage. Simultaneously, the underlying control system adjusts transformer tap positions and reactive power compensators. This is why the BtB can return to a higher voltage until 12:24.

Emergency Power Control: Figure 10 illustrates the adaptive EPC limits responding to dynamic variations in wind power and contingencies. Starting from 01:00, MIO reduces EPC_{max} gradually to accommodate for increasing wind power. Additionally, unforeseen contingencies are directly impacting power transmission capabilities. This is seen as example by the loss of the BAE—BwB cable at 09:35. Consequently, MIO reduces EPC_{min} until 13:30

6 Conclusions

The increasing installation of offshore wind power results in more demanding power collection and transmission tasks. Multiple wind farms get connected to offshore grids. Hybrid offshore grids connect the wind farms to multiple countries and thus serve for power transmission between those countries as well. This increases the operational complexity.

This paper shows how hybrid offshore grids can be operated with model-based nonlinear control, solving optimal power flow problems in real-time. The methods were implemented in the Master Controller for Interconnector Operation (MIO) for the Kriegers Flak Combined Grid Solution (KF CGS). KF CGS connects multiple wind farms in the Baltic Sea to the northern and conti-

nental European transmission grids. MIO was tested extensively with simulations using a digital twin prototype and has been in successful operation in the real system since December 2020.

As a result, the transmission of power through offshore grids is maximized, complex flexibilities arising from connections to multiple wind farms and multiple countries are exploited and the reliability is improved.

The formulation of the electrical models in dq coordinates results in linear equation systems for voltage and current phasors with quadratic output relationships for absolute voltages, e.g. as measured at busbars, and currents, e.g. determining cable loads. The absolute quantities are constrained in the optimal power flow problem. This results in a quadratically constrained quadratic optimization problem, allowing a reliable and fast numerical solution for the application in closed loop real-time control.

Significant speedups are achieved with parallel multiple shooting in dynamic optimization problems for the planning of the operation. Parallel automatic differentiation and optimization had been addressed in the PARADOM project.

Current research investigates the extension of future offshore installations with the production of green hydrogen and the use of gas pipelines complementing electric cables. The OpenSCALING project addresses the treatment of large-scale green hydrogen production with Modelica.

Acknowledgements

This work was supported in parts by the Federal Ministry of Education and Research (BMBF) within the projects PARADOM and OpenSCALING – BMBF funding codes: 01IH15002E and 01IS23062B, respectively.

References

- European Commission: European Wind Power Action Plan, Brussels, October 2023. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52023DC0669&qid=1702455143415>
- A. Marten, V. Akmatov, T. Bentzon Sorensen, R. Stornowski, D. Westermann, C. Brosinsky: Kriegers flak combined grid solution: coordinated cross-border control of a meshed HVAC/HVDC offshore wind power grid, IET Renewable Power Generation, 2018, Vol. 12 Iss. 13, pp. 1493-1499. <https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/iet-rpg.2017.0792>
- R. Franke: Formulation of dynamic optimization problems using Modelica and their efficient solution, 2nd International Modelica Conference, 2002. https://modelica.org/events/Conference2002/papers/p39_Franke.pdf
- M. Otter, B. Thiele, H. Elmqvist: A Library for Synchronous Control Systems in Modelica, 9th International Modelica Conference, Munich, 2012. <https://www.ep.liu.se/ecp/076/002/ecp12076002.pdf>

- F. Fachini, S. Bhattacharjee, M. Aguilera, L. Vanfretti, G. Laera, T. Bogodorova, A. Moftakhar, M. Huylo, A. Novoselac: Exploiting Modelica and the OpenIPSL for University Campus Microgrid Model Development, 15th International Modelica Conference, Aachen, October 2023. <https://ecp.ep.liu.se/index.php/modelica/article/view/936/844>
- J.E. Feghali, Q. Cossart, G. Bureau, B. Letellier, I. Menezes, F. Rosiere, M. Chiaramello: An Open-Source Benchmark of IEEE Test Cases for Easily Testing a New Approach for Steady State Calculations in Power Systems, 15th International Modelica Conference, Aachen, October 2023. <https://ecp.ep.liu.se/index.php/modelica/article/view/974/882>
- B. Bachmann, H. Wiesmann: Advanced Modeling of Electromagnetic Transients in Power Systems, *Modelica Workshop 2000*, Lund, September 2000. <https://modelica.org/events/workshop2000/proceedings/Bachmann.pdf>
- R. Franke, H. Wiesmann: Flexible modeling of electrical power systems – the Modelica PowerSystems library, 10th International Modelica Conference, Lund, 2014. <https://www.ep.liu.se/ecp/096/054/ecp14096054.pdf>
- R. Franke, S.E. Mattsson, M. Otter, K. Wernersson, H. Olsson, L. Ochel, T. Blochwitz: Discrete-time models for control applications with FMI, 12th International Modelica Conference, Prague, 2017. <https://www.ep.liu.se/ecp/132/057/ecp17132507.pdf>
- J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, H. Tummescheit. Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problems. *Computers and Chemical Engineering*, 34(11):1737–1749, November 2010.
- M. H. Alizadeh, A. M. Sahlodin, A. Palanisamy, F. Casella, P. Fritzson: Application of the OpenModelica-Matlab Interface to Integrated Simulation and Successive Linearization Based Model Predictive Control, 15th International Modelica Conference, Aachen, October 2023. <https://ecp.ep.liu.se/index.php/modelica/article/view/937/845>