OATS: Optimisation and Analysis Toolbox for Power Systems

W. A. Bukhsh, *Member, IEEE*, C. Edmunds, *Student Member, IEEE*, and K. R. W. Bell, *Member, IEEE*

***Abstract*—Optimisation and Analysis Toolbox for power Systems analysis (OATS) is an open-source simulation tool for steady-state analyses of power systems problems distributed under the GNU General Public License (GPLv3). It contains implementations of classical steady-state problems, *e.g.* load flow, optimal power flow (OPF) and unit commitment, as well as enhancements to these classical models relative to the features available in widely used open-source tools. Enhancements implemented in the current release of OATS include: a model of voltage regulating on-load tap-changing transformers; load shedding in OPF; allowing a user to build a contingency list in the security constrained OPF analysis; implementation of a distributed slack bus; and the ability to model zonal transfer limits in unit commitment. The mathematical optimisation models are written in an open-source algebraic modelling language, which offers high-level symbolic syntax for describing optimisation problems. The flexibility offered by OATS makes it an ideal tool for teaching and academic research. This paper presents novel aspects of OATS and discusses, through demonstrative examples, how OATS can be extended to new problem classes in the area of steady-state power systems analysis.**

***Index Terms*—Power systems analysis; load flow analysis; optimal power flow; integer programming; unit commitment problem; linear programming; mixed integer programming.**

I. Introduction

THE structure and operation of power systems have changed significantly over the last two decades. Most developed countries have restructured their power sectors to separate the activities of generation, network management and retail. Moreover, the increased deployment of partly controllable weather-dependent renewable generation has brought about new challenges for the operation and management of power systems. Classical optimisation models, developed for the analysis and control of power systems with predominantly controllable thermal generation, need to adapt to account for the changes in the inputs brought about by intermittent renewable generation and liberalised electricity markets [1].

A number of classical modelling and optimisation formulations exist in power systems analysis literature: load flow analysis [2], optimal power flow [3] and unit commitment [4], to name a few. While the physics of power flow remains unchanged, the inputs have changed drastically over the years. For example, a significant percentage of thermal generation capacity in developed countries is being replaced by intermittent renewable generation, storage and demand response. On the network side, new technologies such as HVDC links are being introduced in place of traditional overhead transmission lines. These changes necessitate inputs from different disciplines, *e.g.* time-varying forecasts from statistics, nonlinear modelling and convex approximations from optimisation, analysis of electricity market principles from economics and techniques to solve large scale problems from high-performance computing.

In recent years, there has been significant academic interest in delivering open-source tools to facilitate research in power systems analysis [5]–[12]. These tools have made significant contributions in enabling researchers to propose new methods and demonstrate them in simulations. This paper is a continuation of such efforts to support and facilitate the next generation of models to tackle some of the issues that we face in representing real modern power systems. It presents an open-source toolbox, Optimisation and Analysis Toolbox for power Systems (OATS), that includes implementation of classical power systems analysis, as well as enhancements that make such analysis more realistic. The optimisation models are written in an open-source algebraic modelling language (AML) that is easy to comprehend and extend.

A. Motivation

The initial motivation for the development of OATS was to support students' projects at the department of Electronics and Electrical Engineering at the University of Strathclyde. The projects offered to the students often involve an extension of traditional steady-state analysis models. A need was felt for a power systems analysis tool with a quick learning curve for students, while remaining flexible, open-source and sophisticated enough to model real engineering challenges posed in the projects.

OATS has been successfully used on a range of power systems analysis studies [13]–[17] and research projects [18]–[20]. OATS is now ready to be shared with the wider power systems research community and has been made available through GitHub [21]. The version control system of GitHub, flexible nature of OATS and future collaboration with the wider research community will ensure that future enhancements to the toolbox are made in a modular way.

B. Existing open-source tools

Some of the well known power systems analysis software in common use in universities include MATPOWER [5], PSAT [6], PST [7], PYPOWER [9], minpower [8] and pandapower [10]. MATPOWER, PSAT and PST are MATLAB based open-source tools. MATLAB is a commercial software, however, an open source tool GNU Octave [22], can be used for MATPOWER, PSAT and PST. PYPOWER [9] is a port of MATPOWER to Python but is no longer maintained.

The authors are with the Department of Electronic and Electrical Engineering, University of Strathclyde, 99 George Street, Glasgow, G1 1RD (e-mail: waqquas.bukhsh@strath.ac.uk)

`pandapower` [10] is a Python based tool which relies on PYPOWER for solving the load flow and optimal power flow problems. Python for Power System Analysis (PYPSA) is also a Python based tool that can solve DC and AC load flow problems as well as DC security constraint optimal power flow [12]. EGRET is another Python based tool that uses the Pyomo modelling language to specify mathematical models and is capable of solving unit commitment, DC optimal power flow and AC optimal power flow problems [23]. PowerModels.jl is a Julia programming language based tool that can solve a range of steady-state power network optimisation problems [11]. A comparison of the modelling capabilities of OATS and selected open-source power systems analysis tools is presented in Appendix A.

The existing tools lack some key features that are used by power systems utilities to carry out steady state analysis, *e.g.* distributed slack bus and voltage regulation via on-load tap-changing transformers. OATS aims to fill this gap and provide a platform for extending the classical power systems optimisation models.

Furthermore, a key limitation of most of the tools mentioned above is that changes in constraints, variables and the objective function cannot be made in a straightforward manner. PowerModels.jl and pandapower have addressed this issue to a certain extent by allowing users to modify and extend implemented models in a finite number of ways. Through the interactive tutorials in the online help [24] and extended models provided in the GitHub repository [21], users of OATS can learn how to interact directly with the model files using the PYOMO modelling language.

C. Contribution of OATS

In order to support current research interest in the planning and operation of power systems, it is important to design a simulation framework that can adapt to various needs. As argued earlier, most of the available open-source software for power systems analysis lacks the flexibility to enhance the embedded models.

OATS is specifically designed to be a portable and fully accessible simulation toolbox for application to real-world steady state power systems problems. All the ingredients of OATS and its dependencies are open-source and are compatible with Python 2 and Python 3. This flexibility means that OATS is a useful tool for the power systems community in academia and industry for analysing, extending and simulating important research questions. The following are the key contributions of OATS:

- a framework with the implementation of classical models that is easy to comprehend and extend;
- the ability to easily switch solvers to suit the demands of a particular problem;
- a tool designed to have a fast learning curve for student projects and enough capability for significant research in the area of power systems analysis;
- implementation of a distributed slack bus and on-load tap changing transformers in load flow analysis;
- security constrained optimal power flow with the ability to perform AC pre-fault and DC post-fault analysis;
- the ability to model voltage dependent loads in optimal power flow and zonal transfer constraints in a unit commitment model;
- the inputs and outputs of OATS are managed using spreadsheets, which is a convenient way of handling data.

OATS is distributed under GNU General Public License 3.0 license, which is a strong copyleft license and requires extensions to be made under the same license conditions.

The rest of the paper is organized as follows: Section II presents an overview of OATS including it's architecture, data format and solvers. The modelling capabilities of OATS are presented in Section III. A selection of existing models are demonstrated using a 24-bus IEEE reliability test network in Section IV. Conclusions are presented in Section V.

II. OATS

OATS is a collection of optimisation models and Python scripts for analysing and solving a range of power systems problems. OATS makes use of an open-source, Python based, algebraic modelling language, PYOMO, for describing mathematical models [25]. A key advantage of using PYOMO is the similarity of it's syntax to the mathematical notation of implemented optimisation problems.

OATS can be installed as a python package (pip installable), or can be installed directly from the source (GitHub repository [21]) depending on the desired level of customisation. The documentation of OATS[24], contains detailed explanations of models, syntax and example code. Users can report issues with the code using the GitHub issue reporting system to improve the quality and usability of the code.

A. Architecture

Figure 1 presents the architecture of OATS. The mathematical models and data for test networks are defined in the model library and test case library, respectively. The model files are written in the high-level syntax of PYOMO. A model file of PYOMO does not solve the models directly; instead, it calls an appropriate external solver to obtain a solution.

A number of Python scripts have been written that handle the flow of information between OATS, PYOMO and a solver. A user specifies an OATS model file along with a test case and set of options. Further Python scripts convert the user-defined dataset into a PYOMO readable data file, which, along with the model file and user-specified options, is passed to a solver. After a problem is solved, a final set of Python scripts process the output and write the results to a spreadsheet.

Users can either specify in-build OATS models (*e.g.* OPF, UC) or their own extensions of OATS models for solving bespoke problems. A number of extensions are made available in the GitHub repository [21]. OATS allows users to directly interact with the model files and extend the models in a way that the problem at hand demands. A down-side to the need for interaction with the model file is that the user needs to become familiar with the PYOMO syntax of defining variables, parameters and sets. However, PYOMO being an algebraic modelling language, makes it an intuitive language to learn. In our experience, the students that have some

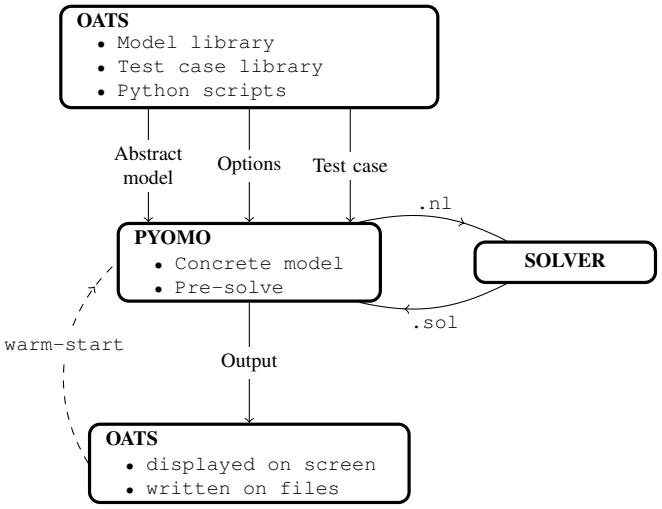


Fig. 1. Architecture and flow of information in OATS.

knowledge of Python programming language did not have any issues working with PYOMO. Moreover, in the online documentation of OATS [24], a number of examples are provided for extending the classical models built in to OATS.

OATS allows a user to warm-start problems, which means that a user-defined initial-guess or a previous solution from a comparable system condition can be passed as an input. This feature is particularly useful for networks where a gradient based solver may struggle to find a feasible solution and expert knowledge may be used to provide guidance to the solver. This feature of warm-starting can also reduce run times of batch simulations, where the next iteration could be warm-started with the solution of a previous iteration.

B. Data structure

The system data and problem solution are stored in a spreadsheet format, which is a convenient and easy way to interact with data. A variety of open-source and commercial tools are available for reading data in the spreadsheet format.

The input data takes into account all the necessary information required for the steady-state analysis of a power systems. The data format is divided into 8 different categories. A brief description of these sheets is presented in Appendix B and more details can be found in the documentation of OATS [24].

In the OATS data format, each electrical component (*e.g.* transformer, generator, bus) is assigned a unique string name (often MATLAB based tools only allow integer identifiers), making it easier to work with the inputs and outputs. Status flags (online or offline) and contingency flags (included or not included in the contingency analysis) are defined for selected electrical components. This allows a user to easily modify the network state using a spreadsheet, without needing to dig deep into the Python code to manipulate constraints. OATS allows users to import test case data from a MATPOWER [26] test library and a script is provided in [21] that automatically translates MATPOWER input files (`.m`) into OATS input files (`.xlsx`).

The model solution output is written on a spreadsheet, which has the same structure as the input file. A summary of the results is displayed on the screen.

TABLE I
A LIST OF OPTIMISATION SOLVERS COMPATIBLE WITH OATS.

Solver name	Capability	License
`glpk`	LP, MILP	Open source [28]
`cbc`	LP, MILP	Open source [29]
`lp_solve`	LP, MILP	Open source [30]
`ipopt`	LP, NLP	Open source [31]
`bonmin`	LP, NLP, MILP, MINLP	Open source [32]
`couenne`	LP, NLP, MILP	Open source [33]
`gurobi`	LP, QP, MILP	Free academic license [34]
`CPLEX`	LP, QP, MILP	Free academic license [35]

C. Solvers

A standard optimisation problem can be written as follows:

$$\min_x f(x) \tag{1a}$$

subject to

$$g(x) \leq 0 \tag{1b}$$

$$h(x) = 0 \tag{1c}$$

$$X^- \leq x \leq X^+ \tag{1d}$$

where (1a) is an objective function, (1b) and (1c) are the equality and inequality constraints respectively, and (1d) defines the bounds on the decision variables x. Depending on the objective function and constraints, an optimisation problem could be characterized as a linear programming problem (LP), nonlinear programming problem (NLP) or mixed integer linear programming problem (MILP).

A solver is required to solve the mathematical model described using PYOMO in OATS. OATS provides users with two options for selecting a suitable solver. The first option is the installation of a local solver. A number of open-source solvers (or free for academic use) can be used with OATS depending on the type of the problem. A list of compatible solvers is provided in Table I. The second option is to use NEOS [27]. NEOS is a free web-based server and allows an OATS user to access approximately 60 state-of-the-art solvers.

III. MODELLING CAPABILITIES OF OATS

Steady-state analysis optimisation models in power systems have some similarity when it comes to defining the set of constraints. For example, all models that account for power flows include a set of nonlinear power flow equations (or a linear approximation of those) in the constraint set. Upper and lower bounds are required for all the decision and state variables. The objective function may vary depending on the problem under consideration, but most often the objective function is to minimize the cost of meeting demand.

Table II presents selected model types implemented in the current open-source release of OATS along with characterisation of each model. Depending on the type of the optimisation problem, a suitable solver can be selected. The

TABLE II
STEADY STATE ANALYSIS OPTIMISATION MODELS AVAILABLE IN OATS. ID REFERS TO UNIQUE IDENTIFIERS USED IN OATS TO CALL THE MODELS.

ID	Model	Classification	Reference
DCLF	DC load flow	LP	[36], [37]
DCOPF	DC optimal power flow	LP	[38]
SCOPF	Security constrained OPF	LP	[39]
ACLF	AC load flow	NLP	[36], [37]
ACOPF	AC optimal power flow	NLP	[40], [41]
UC	Unit commitment problem	MILP	[42]

models implemented in OATS can be broadly classified into three power systems problems: load flow analysis, the optimal power flow problem and unit commitment. In the following, a brief description of each category is provided.

A. Load flow analysis

The load flow problem is solved by fixing all the real and reactive power demands, the voltages at all voltage controlled buses, the voltage magnitude and phase angle at the reference bus, and the real power generator outputs at all generator buses (generator buses can be chosen to be voltage or reactive power regulating). If a distributed slack bus (see below) is not used, the reference bus can be chosen to become the slack bus and real power generation or demand at the bus is also undefined. We then get a system of $2n^{\mathrm{B}}$ nonlinear equations with $2n^{\mathrm{B}}$ variables, where n^{B} is the number of buses in a network. This system of equations is solved for the remaining variables by numerical iterative methods [43]. It is well known that the load flow problem can have 0, 1 or multiple solutions [44].

The load flow problem is implemented as a constrained optimisation problem in OATS, which allows a robust way of modelling distributed slack buses and voltage regulating tap-changing transformers, as briefly described in the next sections. The exact details of the implementation of the load flow as an optimisation problem are provided in [36] and the reader is referred to [40] for details regarding the relationship between load flow and OPF.

1) Distributed slack: A slack bus in load flow analysis is a bus at which power mismatches are corrected. In view of the likelihood of there being a positive injection of power at this bus, it is often chosen to be one at which generation is located. When the modeller chooses to use just one slack bus, it is typically chosen to be the same bus as the reference bus. If the outputs at all generators add up to the total demand in a system then the slack bus output is the contribution of overall line losses in the network. The idea behind a *distributed slack bus* in a load flow problem is that a set of selected generators contribute to the system losses instead of a single slack bus to balancing out the total mismatch [45].

It is important to make a clear distinction between a slack-bus and a reference-bus. A reference bus is a slack bus where the voltage angle is *fixed*, and it is unique within an area; a distributed slack bus is not unique and an area may have 0, 1 or more buses making up the distributed slack.

One real-world analogy for the one or more buses that compensate for mismatches, *i.e.,* a single or distributed slack bus, is the set of generators that regulate system frequency. When there is a system imbalance, system frequency will rise or fall. Frequency responsive generators that are in a governor controlled mode or are part of an Automatic Generator Control (AGC) scheme will, based on frequency deviations and governor droop or AGC settings, respond to mismatches and correct them. This behaviour can be mimicked via a distributed slack with rules defined by the modeller to determine how much of the mismatch is picked up at each bus that is part of the distributed slack. For example, in [46], each bus that is part of the distributed slack is a generator that provides a frequency regulation or frequency containment service and its output is adjusted in proportion to the generator's rating such that the total correction matches the total system mismatch. Because the network losses will have changed as a consequence, an iterative approach is used until the total slack bus deviation is under a given threshold. Particularly in the case of a contingency where a generator or load is tripped, use of a single slack bus can give very different results from when a distributed slack is used. Depending on the frequency or inter-area transfer management policy used in the real system being modelled, a suitably defined distributed slack will give much more realistic results than a single slack.

OATS allows the user to specify any generator as a distributed slack bus in the input spreadsheet (by setting generator `type` to 2). A policy (objective function) must be set defining how much, and in which order, the distributed slack buses will contribute to balancing out the system. Some possible policies for this are overall minimum transmission line loss or minimum deviation of slack generators from a given set-point. The default in OATS is to minimise the overall deviation of a distributed slack bus from a given set-point.

2) On-load tap-changers: Transformers are commonly equipped with on-load tap changers (OLTCs) to regulate secondary voltage at grid supply points by changing the tap ratios at the high-voltage (HV) side of a transformer. The tap ratio of a transformer τ is a discrete variable that typically can take values from a given discrete set of pre-defined turn ratios $\{\tau_0, \tau_1, \cdots, \tau_n\}$. In order to avoid computational complexity associated with integer programming problems, the variable tap ratios in OATS are modelled as continuous variables taking values from the continuous interval $[\tau_0, \tau_n]$. After a solution is reached, the optimal tap ratio found can be rounded to the nearest discrete set-point and the problem can be re-run with fixed tap-ratios to obtain a new solution.

An alternative to the above approach is to modify the load flow model in OATS to model tap-changers with discrete set-points. This involves defining the following binary variables for each on-load tap-changer transformer in a network $\{\gamma_0, \gamma_1, \cdots, \gamma_n\}$ that models the status of each tap-ratio The following constraint ensures that a single tap-ratio is selected from the set: $\gamma_0 + \gamma_1 + \cdots + \gamma_n = 1$. This problem formulation results in a mixed integer nonlinear programming problem (MINLP) and will require a solver that can solve MINLP problems, *e.g.* BONMIN. Due to the complexities involved in solving the MINLP problems, the current implementation of on-load tap-changers in OATS uses continuous variables to model tap-ratios.

OATS has two models of a transformer: a 2-winding transformer with fixed tap-ratios ($\texttt{type} = 1$), and a tap-changing transformer ($\texttt{type} = 2$). OATS allows a user to fix the voltage at the secondary end of the transformer and specify bounds for off-nominal tap settings. The load-flow analysis determines the setting of the tap-ratios within the specified bounds that achieve the target voltage at the secondary bus of the tap-changing transformer.

B. The optimal power flow problem

The optimal power flow problem is a well-known power systems optimisation problem that seeks to find a steady state operating point of a system which minimizes the cost of electricity generation while satisfying operating constraints and meeting demand. The nonlinear ACOPF problem in polar coordinates is implemented in OATS. A linear approximation of the ACOPF problem that is commonly known as DCOPF is also available in OATS.

A solution of an ACOPF problem is not necessarily secure against network outages. Security constrained optimal power flow (SCOPF) is an extension of a normal OPF, which yields a solution that guarantees that the network operates within prescribed limits, not only intact, but also after an occurrence of a credible contingency. OATS includes implementation of two versions of SCOPF: (i) a linear DC-SCOPF, and (ii) a non-linear SCOPF that models the pre-fault operation of a system using nonlinear AC power flow equations, and models the post-fault operation using DC-power flow equations. The linear DC-SCOPF model is built-in to OATS and the non-linear SCOPF is available as an extension on the Github repository [21]. The purpose of the non-linear SCOPF is to ensure that the pre-fault state of a system is feasible from a voltage and reactive power point of view while still respecting a large number of thermal security constraints.

The set of contingencies includes single outages of transmission lines, transformers and generating units. A user can customise a contingency set by choosing which single contingencies to include in the analysis.

The optimal power flow problem can be extended to model optimal balancing actions of a system operator: minimisation of the cost of readjusting a solution provided by the electricity market so that it respects network and security constraints. The OPF functionality of OATS has been used to model such a problem in a part of the Belgian transmission network [15].

C. Unit commitment

Unit commitment is an optimisation problem with the objective of determining the operating schedule of generating units at every time-step of a given time horizon, with varying demand for electricity and under temporal and network constraints. Traditional implementations of the unit commitment problem do not include constraints on zonal transfer limits. OATS allows a user to define zones and impose separate *to* and *from* net transfer capacity (NTC) limits. The generation parameters related to the unit commitment problem (*e.g.* ramp rates and minimum up and down times) are defined in hourly units. However, OATS allows users to specify the temporal resolution of a problem and a script within OATS scales the input parameters accordingly.

The current implementation of OATS uses three binary variables to represent on/off commitment, start-up and shut-down status of the conventional generators. The implementation is taken from [42]. Other formulations of the UC problem exist in the literature that have been demonstrated to perform computationally better than the traditional UC formulation, *e.g.* in [47]. These enhanced formulations of the UC problem include an extra set of inequalities that provide a tighter description of the feasible region of a UC problem. Such constraints can be included in the existing UC implementation of OATS by modifying the constraints within the model file using the method described in the online documentation [24].

D. Other extensions

OATS can be extended in various other ways for different power systems problems and applications. The following are some research directions in which OATS has been extended.

- Risk-based SCOPF: incorporating a risk-index in the standard formulation of the SCOPF.
- Three stage SCOPF: SCOPF model that balances the cost of pre-fault preventive actions and post-fault corrective actions, as described in [15].
- Storage model: Modelling of a generic storage model within the multiperiod optimal power flow model. This model is being developed for the RESTLESS project [19].
- HVDC transmission model: ability to model interconnections between two different synchronous areas and 'embedded HVDC' connecting two different locations within a single synchronous area. A demonstration of this model is provided in [16].
- Demand response: Modelling of the ability of demand to flex in response to changes in price, as demonstrated in [17].

IV. DEMONSTRATION

Selected features of OATS are demonstrated on a 24-bus IEEE reliability test system [48]. The 24-bus network is shown in Figure 2. Three case studies are presented: the first case study demonstrates extension of OPF problem by modelling voltage dependent loads; the second case study demonstrates the use of OPF to optimise reactive power margins and use of voltage regulating on-load tap changing transformers (OLTCs); and the third case study demonstrates an application of the SCOPF model for checking maintainability of transmission lines in the 24-bus network. The extension models for the three case studies can be found within the OATS GitHub repository [21].

A. Voltage dependent loads in OPF

The steady-state analysis models implemented in OATS have a constant power model of electricity load. The ease of extension in OATS is demonstrated by modelling of a ZIP load in the ACOPF model. A ZIP model of electricity load consists of constant impedance (Z), constant current (I) and constant

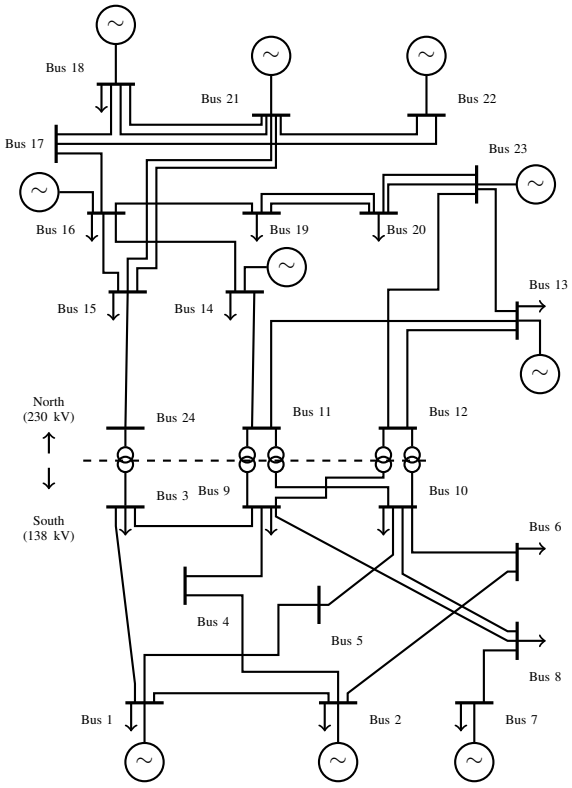


Fig. 2. Single line diagram of the IEEE 24-bus reliability test system [48].

power (P) load components and are represented by a second-order polynomial in bus voltage magnitude v_b as follows [43]:

$$p_d^{\mathrm{D}}(v_d) = \mathrm{P}_d^{\mathrm{D}}\left(a_d^{\mathrm{P}} v_d^2 + b_d^{\mathrm{P}} v_d + c_d^{\mathrm{P}}\right) \tag{2a}$$

$$q_d^{\mathrm{D}}(v_d) = \mathrm{Q}_d^{\mathrm{D}}\left(a_d^{\mathrm{Q}} v_d^2 + b_d^{\mathrm{Q}} v_d + c_d^{\mathrm{Q}}\right) \tag{2b}$$

where $a^{\mathrm{P,Q}}, b^{\mathrm{P,Q}}, c^{\mathrm{P,Q}}$ are the active and reactive power coefficients of the quadratic polynomial, respectively. Parameters $a^{\mathrm{P,Q}}$ represent the relative participation of constant impedance load, $b^{\mathrm{P,Q}}$ the relative participation of constant current load, and $c^{\mathrm{P,Q}}$ the relative participation of constant power load.

The real and reactive power demand are modelled as parameters in the ACOPF model of OATS (written in PYOMO syntax) on the set of demands D, as follows:

```
model.PD = Param(model.D, within=Reals)
model.QD = Param(model.D, within=Reals)
```

In order to model the dependence of electricity demand on voltages, the real and reactive power parameters are modelled as variables `pD` and `qD`, as follows.

```
model.pD  = Var(model.D, within=Reals)
model.qD  = Var(model.D, within=Reals)
model.aPD = Param(model.D, within=NonNegativeReals)
model.bPD = Param(model.D, within=NonNegativeReals)
model.cPD = Param(model.D, within=NonNegativeReals)
model.aQD = Param(model.D, within=NonNegativeReals)
model.bQD = Param(model.D, within=NonNegativeReals)
model.cQD = Param(model.D, within=NonNegativeReals)
```

The coefficients of the quadratic function are defined as parameters given by the user. Equations (2) are implemented in the ACOPF model of OATS to model the ZIP load as follows:

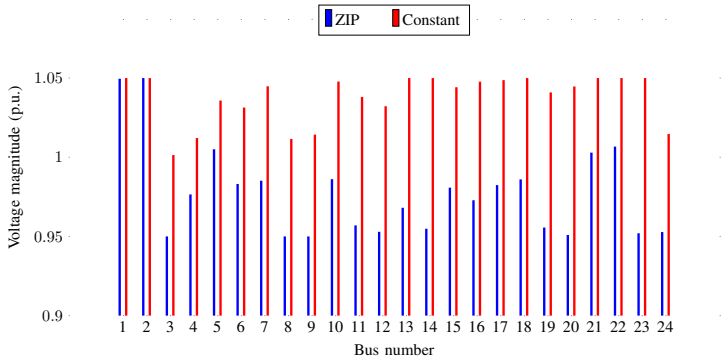


Fig. 3. Voltage profile of 24-bus IEEE reliability test network with constant load and ZIP load.

```
def real_power_demand(model,d):
    return model.pD[d] == model.PD(
           model.aPD[d]*model.v[b]**2+
           model.bPD[d]*model.v[b]+model.cPD[d])
def reactive_power_demand(model,d):
    return model.qD[d] == model.QD(
           model.aQD[d]*model.v[b]**2+
           model.bQD[d]*model.v[b]+model.cQD[d])
```

The ZIP model of the load is implemented in OATS and is solved on the 24-bus case. The following parameters for modelling residential load were used for the ZIP model [49]: $a_d^{\mathrm{P}} = 0.29, b_d^{\mathrm{P}} = 0.10, c_d^{\mathrm{P}} = 0.61$ and $a_d^{\mathrm{Q}} = 3.22, b_d^{\mathrm{Q}} = -4.53, c_d^{\mathrm{Q}} = 2.31$.

Figure 3 presents the voltage profile with constant and ZIP load models. The voltage profile with the ZIP load model is comparatively lower than with a constant load. This is because the load is dependent on voltage and lowering the voltage results in lowering the overall demand and hence the total cost of meeting that demand. Due to the decrease in bus voltages, the overall system demand has decreased by 46 MW and the line losses increased by 9.8 MW. The objective value of the ZIP load model OPF decreased by 2.9% as compared to the constant load OPF.

B. Using OPF to optimise reactive power margin

The AC-OPF is extended to model the balancing actions of a system operator. In this extension, the cost of adjusting a solution from a given target (e.g. an electricity market solution) is minimised so that the dispatch respects the network constraints. Four different cases, as described in Table III, are considered for a choice of objective function and modelling of the OLTC transformers. The first two cases consider fixed tap transformers and minimise the cost of balancing real power generation and the cost of balancing real and reactive power generation, respectively. The third and fourth case are the first two cases but with all the five transformers modelled as OLTCs.

The real power initial conditions are obtained by solving a UC model with half hourly periods for 24-hours. The demand curve is taken from [50] and the UC solution for the peak demand hour is selected to be used for the balancing model. Representative bid and offer prices are used from Great British (GB) balancing mechanism data for the generator

TABLE III
FOUR CASES CONSIDERED TO QUANTIFY THE REACTIVE POWER MARGIN IN THE IEEE 24-BUS NETWORK

	Objective Function	On-load tap changers
case1	Only real power cost	No
case2	Real and reactive power cost	No
case3	Only real power cost	Yes
case4	Real and reactive power cost	Yes

TABLE IV
REACTIVE POWER MARGIN AND CHANGE IN OBJECTIVE FUNCTION VALUE FOR THE FOUR CASES. THE CHANGE IN OBJECTIVE FUNCTION VALUE IS RELATIVE TO CASE1.

	Reactive power (Q) margin (Mvar)			Change in Objec func (%)	
	Q-leading	Q-lagging	Q-RMSE	Overall	Real Power
case1	1067.7	1243.3	101.9	-	-
case2	1081.0	1230.0	29.9	9.2	2.4
case3	1065.0	1246.0	102.7	-0.5	-0.5
case4	1084.6	1226.4	28.3	8.8	2.2

TABLE V
CHECKING MAINTAINABILITY OF TRANSMISSION LINES OF THE 24-BUS SYSTEM. A TRANSMISSION LINE IS MAINTAINABLE IF NO LOAD SHEDDING OCCURS IN PRE-FAULT STATE WHEN IT IS TAKEN OUT OF THE SYSTEM.

From	To	Maintainable	Load shedding(%)	Critical Buses
1	3	No	0.2	{3,24}
1	5	No	2.5	{5}
2	4	No	2.6	{4}
2	6	No	4.8	{6}
3	9	No	0.2	{7,24}
4	9	No	2.6	{4}
5	10	No	2.5	{5}
6	10	No	4.8	{6}
8	9	No	6.0	{8}
8	10	No	6.0	{8}
11	13	No	-	-
11	14	No	6.8	{14}
14	16	No	6.8	{14}
15	24	No	0.2	{3,24}
17	22		——— Infeasible ———	
22	17		——— Infeasible ———	

types (nuclear, coal, etc.) in the 24-bus system [51]. The reactive power cost of 3.4 £/Mvarh (2018 average reactive power service price in GB [52]) is used for the reactive power balancing from the mid-point of the generators.

Table IV presents the total leading reactive power margin (margin to Qmin) and the total lagging reactive power margin (margin to Qmax) for the 4 cases. The total leading and lagging reactive power margin does not change significantly, however, the root mean square error (RMSE) of reactive power from the mid-point of generators reduces by approximately 70% in case2 and case4 as compared to case1 and case3, respectively. This demonstrates that a system can be dispatched in such a way that provides a solution with reactive power margins spread at a number of locations. Such a solution is valuable for maintaining system voltages after a fault has occurred. The last column in Table IV shows that such a solution will cost approximately 2% more because of higher real power transmission system losses.

C. Checking maintainability of assets using security constrained optimal power flow

Periodic inspection of assets and their maintenance are preventive actions that a system operator takes to decrease the frequency of occurrence of failures. In order to carry out maintenance activity, the asset which is being maintained needs to be taken out of service as a planned outage. In general, such outages reduce the transmission capacity of a network and increase the risk of system problems. For this reason, the European transmission system operators perform security analysis to plan outages [53].

In this demonstrative case study, the linear DC security constrained OPF model, provided in OATS, is used to check the *maintainability* of the 33 transmission lines of the IEEE 24-bus network for the given demand and generation. As noted earlier, a contingency column is provided in the input sheet that allows a user to include a contingency in the list of contingencies for analysis. Single contingencies can be applied on transmission lines, transformers and/or generating units.

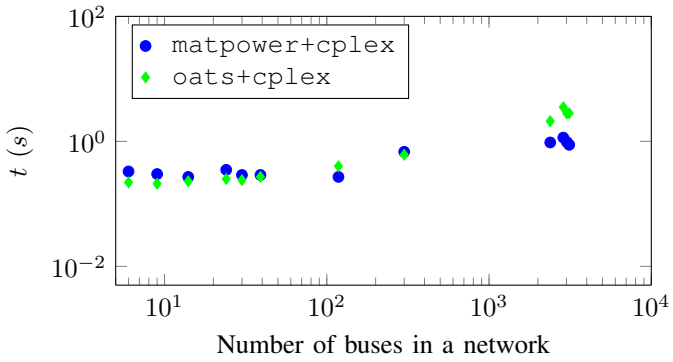
OATS allows users to switch off electrical components by changing their status to 'off-line' in the input data file. In this demonstration, maintainability of a transmission line is checked by switching it out of the system and then solving the DC SCOPF model. A transmission line is labelled maintainable if the system can be operated securely without any need of load shedding. Table V presents the results of the SCOPF model by taking each line out of service for maintenance. Critical buses are those at which the marginal price equals the value of lost load indicating that load must be shed in order that an (N-1) secure condition can be achieved.

Eighteen out of thirty three transmission lines are maintainable (54.4%). Most of these are in the lower voltage (138 kV) area. Two lines connecting bus 22 are not maintainable. This is due to non-zero minimum stable operating points of six generating units located at bus 22. As may be expected from inspection of the network diagram, the problem is infeasible because it is not (N-1) secure when one of the two transmission lines are on the planned outage.

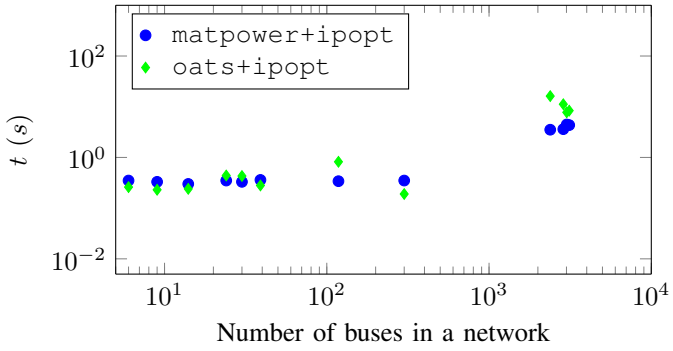
Bus 7 is connected to the rest of the system by a single transmission line. If islanded operation is permitted, the generator at bus 7 could satisfy the load there when the single line from bus 7 is out of service. Moreover, the solution of a SCOPF on the intact network yields a marginal price of 48.0 $/MWh at bus 7 and 49.9 $/MWh at other buses in the system, which implies that bus 7 is constrained in the solution because there is no adjustment allowed from pre-contingency to post-contingency operation, therefore the output of generator at bus 7 is reduced to match the demand at bus 7 in this case.

D. Solution times

In this section, the computational performance of OATS and MATPOWER are compared for DCOPF and ACOPF models. Test cases ranging from 6 to 3120 buses are taken from the MATPOWER test case library. The problems were solved on



(a) Solution times for DCOPF problems using OATS and MATPOWER with the solver `cplex`.



(b) Solution times for ACOPF problems using OATS and MATPOWER with the solver `ipopt`.

Fig. 4. Comparison of solution times for DCOPF and ACOPF problems using OATS and MATPOWER

a Intel Core i5-4590 Linux machine at 3.30 GHz using a flat initial point i.e. the point at which all bus voltage angles 0, all bus voltage magnitudes 1 and all generator injections at the midpoint of the bounds. The reported times are averaged over 20-runs for each considered test network and they include solver time plus the problem setup time.

Figure 4(a) presents solution times for MATPOWER and OATS for solving the DCOPF model with the increasing number of buses using `cplex` as a solver. We observe that for small to medium size networks (i.e. networks with ≤ 300 buses), OATS performs marginally better than MATPOWER, however, for larger networks MATPOWER performs better than OATS. For the large networks, OATS solver times are still faster than MATPOWER but the problem setup time is greater than MATPOWER. This is because OATS reads the problem data from a spreadsheet and then writes a datafile to be passed on to a solver, which takes considerable time, especially for large networks.

Figure 4(b) presents solution times for the ACOPF problem using `ipopt` as a solver for MATPOWER and OATS. Again the solution times of OATS and MATPOWER are similar for small to medium size networks and MATPOWER performing better on large networks. This is again because of the OATS overhead time spent in preparing the data file to be passed to the solver. We are currently researching ways to reduce the problem setup time in OATS for large networks.

V. Conclusions

Although a number of open-source power system tools exist, there is a need both for enhanced features allowing more accurate representation of real power systems operation and the ability to easily make modifications to the modelling framework. A new flexible, open-source platform for the steady-state analysis of power systems is presented in this paper. OATS includes implementation of standard steady-state power systems optimisation models: AC load flow, OPF and unit commitment. The AC load flow and OPF have been enhanced to include representation of a distributed slack bus and voltage regulating on-load tap-changing transformers. The OPF has the ability to solve non-linear equations in a pre-contingency state and to represent post-contingency constraints in a security-constrained OPF via linear approximations. The most important features of OATS are its fast learning curve, ease of extending existing models and no third-party dependence on commercial products.

The input and output data of OATS is managed using spreadsheets, which provides a convenient way of data handling. Switches are provided in the input data file for selected electrical components and for contingencies, which allows a user to investigate selected outages and impact of certain contingencies on a solution. The models are written in a high-level algebraic modelling language, which is easy to comprehend and extend.

OATS is built using an algebraic modelling language (AML) PYOMO, which provide users of the tool to easily switch solvers and extend the implemented models. OATS share these features with other open-source power system analysis tools that make use of an AML e.g. PyPSA, PowerModels.jl and EGRET. By modifying constraints, the objective function and variables with relative ease in OATS, as demonstrated in this paper and online [24], a traditional model can be completely transformed to model a new problem. OATS is under continuous development and more models, case studies and real-world test case data is planned to be made available through the OATS GitHub repository [21].

Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme project Generally Accepted Reliability Principle with Uncertainty modelling and through probabilistic Risk assessment (GARPUR) and Grant Agreement No 608540, and by the UK Engineering and Physical Research Council (EPSRC) through the Realising Energy Storage Technologies in Low-carbon Energy Systems (RESTLESS) project under Grant agreement number EP/N001893/1.

Appendix A
Comparison of Open-Source Power Systems Analysis Software

A number of open-source steady-state power systems analysis toolbox exist in literature. Table VI presents a comparison of modelling capabilities of OATS with a selection of commonly used open-source tools.

TABLE VI
COMPARISON BETWEEN OPEN-SOURCE POWER SYSTEMS ANALYSIS SOFTWARE

		OATS 1.0.1	*PyPSA 0.16.0*	*minpower 4.4.0*	*PSAT 2.1.10*	*MATPOWER 6.0*	*PYPOWER 5.1.2*	*PowerModels.jl 0.13.1*	*pandapower 2.1.0*	*MOST 1.0.2*	*EGRET 0.1*
	Citation	[21]	[12]	[8]	[6]	[5]	[9]	[54]	[10]	[55]	[23]
	DC Load Flow	✓	✓			✓	✓	✓	✓	✓	
	AC Load Flow	✓	✓		✓	✓	✓	✓	✓		
	DC Optimal Power Flow	✓	✓	✓		✓	✓	✓	✓	✓	✓
	AC Optimal Power Flow	✓			✓	✓	✓	✓	✓		✓
Methods	Multiperiod DC OPF	✓	✓					✓		✓	
	Continuation power flow				✓	✓					
	Unit commitment	✓	✓	✓						✓	✓
	DC SCOPF	✓	✓							✓	
	AC Pre-fault, DC Post-Fault SCOPF	✓									
	Distributed slack	✓	✓								
	On-load tap changing transformer	✓			✓				✓		
Models	Storage	✓	✓					✓		✓	
	Three winding transformers								✓		
	DC line model		✓		✓	✓			✓		

APPENDIX B
TEST CASE FORMAT

The test case data of OATS is assembled in a form of a spreadsheet (`.xlsx`). The spreadsheet consists of 8 sheets. A brief description of the sheets is provided in Table VII. It is important to note that certain sheets are only used for specific models, and are redundant for other models. For example, `timeseries`, `zone`, `zonalNTC` sheets are only used for the unit commitment problem.

TABLE VII
DETAILS OF SHEETS IN A OATS TEST CASE.

Sheet name	Description
`bus`	Bus (node) data
`demand`	Demand data
`branch`	Transmission line data
`transformer`	Transformer data
`baseMVA`	baseMVA for conversion into p.u. system
`wind`	Wind generators
`generator`	Generation data
`shunt`	Shunt data
`zone`	Zonal data
`zonalNTC`	Transmission constraints between zones
`timeseries`	Time-series data

REFERENCES

[1] K. W. Hedman, A. S. Korad *et al.*, "The application of robust optimization in power systems," Power Systems Engineering Research Center (PSERC), Tech. Rep., 2014. [Online]. Available: https://pserc.wisc.edu/documents/publications/reports/2014_reports/S-51_Final-Report_Sept-2014.pdf

[2] B. Stott, "Review of load-flow calculation methods," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 916–929, July 1974.

[3] J. Carpentier, "Contribution a l'etude du dispacting economique," *Bull. Soc. Francaise des Electriciens*, vol. 3, pp. 431–447, 1962.

[4] G. S. Lauer, N. R. Sandell *et al.*, "Solution of large-scale optimal unit commitment problems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 1, pp. 79–86, Jan 1982.

[5] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[6] F. Milano, "An Open Source Power System Analysis Toolbox," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1199–1206, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1490569

[7] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Transactions on Power Systems*, vol. 7, no. 4, pp. 1559–1564, Nov 1992.

[8] A. Greenhall, R. Christie, and J.-P. Watson, "Minpower: A power systems optimization toolkit," in *Power and Energy Society General Meeting, 2012 IEEE*, July 2012, pp. 1–6.

[9] R. Lincoln. Pypower. Accessed: 2015-10-30. [Online]. Available: https://pypi.python.org/pypi/PYPOWER

[10] L. Thurner, A. Scheidler *et al.*, "pandapower - an open source python tool for convenient modeling, analysis and optimization of electric power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2018.

[11] C. Coffrin, R. Bent *et al.*, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference (PSCC)*, June 2018, pp. 1–8.

[12] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for Power System Analysis," *Journal of Open Research Software*, vol. 6, no. 4, 2018. [Online]. Available: https://doi.org/10.5334/jors.188

[13] W. A. Bukhsh, G. S. Hawker *et al.*, "Adequacy assessment of future electricity networks," in *2016 Power Systems Computation Conference (PSCC)*, June 2016, pp. 1–8.

[14] W. A. Bukhsh, A. Papakonstantinou, and P. Pinson, "A robust optimisation approach using cvar for unit commitment in a market with probabilistic offers," in *2016 IEEE International Energy Conference (ENERGYCON)*, April 2016, pp. 1–6.

[15] W. A. Bukhsh, K. R. W. Bell *et al.*, "Enhanced, risk-based system development process: a case study from the Belgian transmission network," in *2018 Power Systems Computation Conference (PSCC)*, June 2018.

[16] A. S. C. Leavy, W. A. Bukhsh, and K. R. W. Bell, "Optimal operation of

the western link embedded HVDC connection," in *2018 Power Systems Computation Conference (PSCC)*, June 2018.
[17] C. Edmunds, W. A. Bukhsh, and S. Galloway, "The impact of distribution locational marginal prices on distributed energy resources: An aggregated approach," in *2018 IEEE 15th Internal Conference on European Energy Markets*, June 2018.
[18] GARPUR Consortium, "Current practices, drivers and barriers for new reliability standards," 7th framework programme, EU Commission grant agreement 608570, Tech. Rep., May 2014. [Online]. Available: http://www.garpur-project.eu/deliverables
[19] RESTLESS Consortium, "Realising Energy Storage Technologies in Low-carbon Energy Systems (RESTLESS)," Engineering and Physical Sciences Research Council (EPSRC), Grant agreement EP/N001893/1, Tech. Rep., May 2015. [Online]. Available: http://www.restless.org.uk
[20] ICE Consortium, "The value of Interconnection in a Changing EU Electricity system (ICE)," Engineering and Physical Sciences Research Council (EPSRC), Grant agreement EP/R021333/1, Tech. Rep., April 2018. [Online]. Available: https://gow.epsrc.ukri.org/NGBOViewGrant.aspx?GrantRef=EP/R021333/1
[21] W. Bukhsh, "Oats release 0.1," Sep. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3387594
[22] S. H. John W. Eaton, David Bateman and R. Wehbring, *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2014, ISBN 1441413006. [Online]. Available: http://www.gnu.org/software/octave/doc/interpreter
[23] (2019) EGRET - Tools for building power systems optimization problems. [Online]. Available: https://github.com/grid-parity-exchange/Egret
[24] W. Bukhsh and C. Edmunds, "Online documentation of OATS," Sep. 2019. [Online]. Available: https://oats.readthedocs.io
[25] W. E. Hart, C. D. Laird *et al.*, *Pyomo–optimization modeling in python*, 2nd ed. Springer Science & Business Media, 2017, vol. 67.
[26] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower's extensible optimal power flow architecture," in *Power Energy Society General Meeting, 2009. PES '09. IEEE*, July 2009, pp. 1–7.
[27] J. Czyzyk, M. P. Mesnier, and J. J. Moré, "The neos server," *IEEE Journal on Computational Science and Engineering*, vol. 5, no. 3, pp. 68–75, 1998.
[28] "GLPK (GNU linear programming kit)," 2006. [Online]. Available: http://www.gnu.org/software/glpk
[29] J. Forrest and R. Lougee-Heimer, *CBC User Guide*, ch. Chapter 10, pp. 257–277. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/educ.1053.0020
[30] "lp_solve: Documentation 5.52.5," 2016. [Online]. Available: http://web.mit.edu/lpsolve/doc/
[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.
[32] "bonmin (basic open-source nonlinear mixed integer programming)," 2005. [Online]. Available: https://www.coin-or.org/Bonmin/
[33] P. Belotti, J. Lee *et al.*, "Branching and bounds tightening techniques for non-convex MINLP," *Optimization Methods and Software*, vol. 24, no. 4-5, pp. 597–634, 2009. [Online]. Available: https://projects.coin-or.org/Couenne
[34] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com
[35] "IBM ILOG CPLEX Optimizer," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, Last 2010.
[36] W. Bukhsh, *On Solving the Load Flow Problem as an Optimization Problem*. Tech. Report, University of Strathclyde, May 2018. [Online]. Available: https://strathprints.strath.ac.uk/64156/
[37] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016. [Online]. Available: https://doi.org/10.1080/0740817X.2016.1189626
[38] A. J. Wood, *Power generation, operation, and control [internet resource]*, third edition ed., 2014.
[39] D. Phan and J. Kalagnanam, "Some efficient optimization methods for solving the security-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 863–872, March 2014.
[40] W. Bukhsh, A. Grothey *et al.*, "Local solutions of the optimal power flow problem," *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4780–4788, Nov 2013.
[41] M. B. Cain, R. P. O'Neil, and A. Castillo, "History of optimal power flow and formulations, optimal power flow paper 1," 2012.
[42] G. Morales-Espana, J. Latorre, and A. Ramos, "Tight and compact milp formulation for the thermal unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4897–4908, Nov 2013.
[43] P. Kundar, N. Balu, and M. Lauby, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill Professional, 1994.
[44] Y. Tamura, H. Mori, and S. Iwamoto, "Relationship between voltage instability and multiple load flow solutions in electric power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 5, pp. 1115–1125, May 1983.
[45] W. R. Barcelo and W. W. Lemmon, "Standardized sensitivity coefficients for power system networks," *IEEE Transactions on Power Systems*, vol. 3, no. 4, pp. 1591–1599, Nov 1988.
[46] D. S. Kirschen, K. R. W. Bell *et al.*, "Computing the value of security," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 150, no. 6, pp. 673–678, Nov 2003.
[47] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39–46, Feb 2012.
[48] C. Grigg, P. Wong *et al.*, "The ieee reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1010–1020, Aug 1999.
[49] Report to CIGRE WG C4.605, "Modelling and aggregation of loads in flexible power networks," Tech. Rep., 2014.
[50] W. A. Bukhsh, C. Zhang, and P. Pinson, "An integrated multiperiod opf model with demand response and renewable generation uncertainty," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1495–1503, May 2016.
[51] Delivering the balancing and settlement code (bsc). [Online]. Available: https://www.elexon.co.uk/
[52] National Grid. obligatory reactive power service (orps). [Online]. Available: https://www.nationalgrideso.com/document/143116/download
[53] Deliverable 5.2, Generally Accepted Reliability Principle with Uncertainty modelling and through probabilistic Risk assessment (GARPUR), "Pathways for mid-term and long-term asset management," Tech. Rep. [Online]. Available: https://www.sintef.no/globalassets/project/garpur/deliverables/garpur-d5.2-pathways-for-mid-term-and-long-term-asset-management.pdf
[54] C. Coffrin, R. Bent *et al.*, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference (PSCC)*, June 2018, pp. 1–8.
[55] C. E. Murillo-Snchez, R. D. Zimmerman *et al.*, "Secure planning and operations of systems with stochastic sources, energy storage, and active demand," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2220–2229, Dec 2013.

Waqquas Bukhsh is a research associate at the Department of Electronics and Electrical Engineering, University of Strathclyde. He has a BSc degree in Mathematics from COMSATS University, and a Ph.D. degree in Operational Research from the University of Edinburgh. His research interests are in data analytics, optimisation problems and their applications to energy systems.

Calum Edmunds is a Ph.D. researcher at the University of Strathclyde wind and marine centre for doctoral training, specialising in markets to maximise the integration of renewables. He graduated with the M.Eng. in Chemical Engineering with Energy Engineering from Heriot-Watt University in 2010. He has worked as a Process Engineer in fine chemicals and the energy industry. His research interests include optimal power flow, electricity markets and system balancing.

Keith Bell holds the ScottishPower Chair in Smart Grids at the University of Strathclyde and is a co-Director of the UK Energy Research Centre (UKERC) and a Chartered Engineer. Before joining Strathclyde, he worked as a system development planner in the electricity supply industry in England and as a researcher in Bath, Naples and Manchester. He has provided advice on power systems issues to parties including the Scottish Government, the Office of Gas and Electricity Markets in Britain, the UK Government and the Government of Ireland and is active in CIGRE.