Engineering, **E**conomic, and **E**nvironmental **E**lectricity **S**imulation **T**ool
E4ST 1.0b2
User's Manual

other contributors
Charles M. Marquet
Doug Mitarotonda
Yingying Qi
William D. Schulze
Richard E. Schuler
Di Shi
John Tabor
Daniel J. Tylavsky
Jubo Yan
Yujia Zhu

primary developers
Biao Mao
Carlos E. Murillo-Sánchez
Daniel L. Shawhan
Ray D. Zimmerman

May 11, 2016

Contents

List of Tables

1 Introduction

1.1 Background

The **E**ngineering, **E**conomic, and **E**nvironmental **E**lectricity **S**imulation **T**ool [1, 2], or E4ST (pronounced "east"), was developed by faculty and research staff at Cornell, Rensselaer Polytechnic Institute, Arizona State Universities and at Resources for the Future, with support from the U. S. Department of Energy's CERTS program as well as the Power Systems Engineering Research Center (PSERC). The code itself was developed primarily by Biao Mao, Carlos Murillo-Sánchez and Ray Zimmerman with contributions by others, but the E4ST project as a whole was a much larger enterprise with major contributions and efforts by a larger group.

E4ST is built on top of MATPOWER[1], a package of MATLAB® M-files for solving power flow and optimal power flow problems [3, 4] and is available openly, without charge from the E4ST home page:

`http://www.e4st.com/`

It consists of a set of software toolboxes that can be used to estimate present and future operating and investment states of an electric power system, including generator dispatches, generator entry and retirement, locational prices, fixed and fuel costs, air emissions, and environmental damages. The E4ST software toolboxes can be used with suitable data from any part of the world.

E4ST can be applied to detailed system models. Algorithms are included that simulate the economic operation of the power grid, in response to the model-user's projections of economic factors (e.g. fuel prices), government incentives or environmental regulations. Simultaneously, the algorithms project and implement the economical investment and retirement of generation over time, by location. The algorithms are designed to maintain the redundancy necessary for service reliability.

E4ST is useful for both energy and environmental-policy planning purposes. It accounts for short and long-term feedbacks between energy and environmental policies. It can be used to project the operation and evolution of the power system under any combination of prices, demand patterns, and policies specified by the user. It can calculate the net benefits of any policy simulated, and disaggregate them into the benefits or costs for customers, generation owners, the system operator, the government, public health, and the environment.

In addition, E4ST can be used as a transmission planning tool to explore the consequences of network changes. The existing electric transmission system is fixed

[1]See `http://www.pserc.cornell.edu/matpower/` for more information on MATPOWER.

throughout these simulations, and only the generator dispatches and customer loads respond endogenously, but the user can change the transmission network and re-run the simulation to calculate the effects of the change, potentially repeating this thousands of times to test many different transmission system investment scenarios.

1.2 License and Terms of Use

The code in E4ST is distributed under the 3-clause BSD license [5]. The full text of the license can be found in the `LICENSE` file at the top level of the distribution and reads as follows:

1.3 Citing E4ST

While not required by the terms of the license, we do request that publications derived from the use of E4ST explicitly acknowledge that fact by citing references [1, 2]:

> Daniel L. Shawhan, John T. Taber, Di Shi, Ray D. Zimmerman, Jubo Yan, Charles M. Marquet, Yingying Qi, Biao Mao, Richard E. Schuler, William D. Schulze, and Daniel J. Tylavsky, "Does a Detailed Model of the Electricity Grid Matter? Estimating the Impacts of the Regional Greenhouse Gas Initiative," *Resource and Energy Economics*, Volume 36, Issue 1, January 2014, pp. 191–207.
> `http://dx.doi.org/10.1016/j.reseneeco.2013.11.015`.
>
> Biao Mao, Daniel Shawhan, Ray Zimmerman, Jubo Yan, Yujia Zhu, William Schulze, Richard Schuler, Daniel Tylavsky. "The Engineering, Economic and Environmental Electricity Simulation Tool (E4ST): Description and an Illustration of its Capability and Use as a Planning/Policy Analysis Tool" *Proceedings of the 49th Annual Hawaii International Conference on System Sciences (HICSS)*, Koloa, HI, 2016, pp. 2317–2325.
> `http://dx.doi.org/10.1109/HICSS.2016.290`.

2 Getting Started

The first step in beginning to use E4ST is to get familiar with MATPOWER, in particular with the details of using MATPOWER to run optimal power flows. This step is essential and this manual will assume familiarity with MATPOWER.

2.1 System Requirements

To use E4ST you will need:

- MATLAB® version R2013b or later[2]
- MATPOWER 5.1 or later[3]
- the E4ST code, distributed as a ZIP file such as `e4stNNN.zip`

It is also highly recommended that you install a high-performance LP/QP solver such as Gurobi, CPLEX, MOSEK, MATLAB's Optimization Toolbox or GLPK, described in the Appendix of the MATPOWER User's Manual.

2.2 Installation

Step 1: Install MATPOWER and any high performance solvers as described in the MATPOWER User's Manual.

Step 2: Unzip the file containing the E4ST distribution. It should be named `e4stNNN.zip`, where `NNN` depends on the version of E4ST.

Step 3: Move the resulting `e4stNNN` directory to the location of your choice. These files should not need to be modified, so it is recommended that they be kept separate from your own code. We will use `$E4ST` to denote the path to this directory.

Step 4: Add the following directories to your MATLAB path:

- `$E4ST` – core E4ST functions
- `$E4ST/t` – E4ST tests

[2]MATLAB is available from The MathWorks, Inc. (`http://www.mathworks.com/`). MATLAB is a registered trademark of The MathWorks, Inc.

[3]See the MATPOWER User's Manual for more information on the system requirements.

- (optional) `$E4ST/extras` – only for MATPOWER versions prior to v6

Step 5: At the MATLAB prompt, type `test_e4st` to run the test suite and verify that E4ST is properly installed and functioning. The result should resemble the following.

```
>> test_e4st
t_apply_changes....ok
t_e4st_solve.......ok
All tests successful (914 of 914)
Elapsed time 1.95 seconds.
```

2.3 Running a Simulation

Running a E4ST simulation involves (1) preparing the input data which defines all of the relevant power system parameters and policy data (2) invoking the function to run the simulation and (3) viewing and accessing the results saved in output data structures. Since E4ST is built upon MATPOWER, it is assumed that the user is already familiar with running OPF simulations in MATPOWER (see Section 2.3 in the MATPOWER User's Manual).

2.3.1 Preparing Input Data

The input data consists of (1) a standard MATPOWER case file, (2) a `CaseInfo` file, and (3) a set of E4ST options. Please refer to the MATPOWER User's Manual for more information about the MATPOWER case file data. The `CaseInfo` file defines additional data and parameters for the simulations that are not part of a standard MATPOWER case data file. Some of the contents of this file differ across networks. The detailed description of the `CaseInfo` file can be found in Appendix B.1.

2.3.2 Solving the Case

The unified command is `RunE4ST` and it can be used to make various simulations using the following format:

```
Results = RunE4ST(InterconnectionName, Option)
```

`InterconnectionName` - the name of the interconnection. There are four options now: `'ercot'`, `'wecc'`, `'ei'`, `'test3'`. Note that the interconnection data for ERCOT, WECC and EI are from various data sources, including Energy Visuals, and are not publicly available. The only available option with the E4ST distribution is `'test3'`, which represents a 3-bus example network.

`Option` - (optional) specify optional comma-separated pairs of *Name, Value* arguments. *Name* is the argument name and *Value* is the corresponding value. *Name* must appear inside single quotes (`' '`). You can specify several name and value pair arguments in any order as *Name1*, *Value1*, . . . , *NameN*, *ValueN*:

(1) `'policy'`: determine which polices are included in the simulation

1 *(default)* – base case
2 – environmental damages case

For `'test3'` case, only `'base case'` and `'environmental damages case'` are available. Users can customize policy parameters by modifying corresponding files.

(2) `'year'`: determine the periods of simulation

1 *(default)*

For example, to run a simple Year 0 case with default options on the 3-bus system defined in `test3.m`, at the MATLAB prompt, type:

```
et = RunE4ST('test3')
```

2.3.3 Accessing the Results

The solution is stored in a results object array available as a member of the return value from the simulation functions. Each row of the results array represents the result of one year while each column represents the result from one policy. You can access the results by following commands. The detailed members of the output data can be found in Appendix B.3.

```
result = et.resultArr
```

For example, quantities like the averaged price by load in the whole network, the total CO_2 emissions, the total generation can be extracted as follows.

```
averaged_price_byload = result(1, 1).averageLoadPrice;
total_CO2 = result(1, 1).totalCO2;
total_load = result(1, 1).totalLoad;
```

2.3.4 Setting Options

The standard MATPOWER options struct is used to set options such as the amount of progress output to display and algorithm to use to solve the problem. For example, you can set the options struct using the `mpoption` function.

```
mpopt = mpoption('verbose', 2, 'opf.dc.solver', 'GUROBI');
et = RunE4ST('test3', 'option', mpopt);
```

2.4 Documentation

The primary sources of documentation for E4ST is this manual, which gives an overview of the capabilities and structure and describes the problem formulation. It can be found in your E4ST distribution at `$E4ST/docs/E4ST-manual.pdf`.

3 Problem Formulation

3.1 Nomenclature

Sets

L	Set of all lines
N	Set of all nodes
G	Set of all generators
D	Set of all demands
H	Set of all representative hours
T	Set of all generator types

Indices

$i \in G$	Generator index
$j \in N$	Node index
$d \in D$	Demand index
$k \in H$	Representative hour index
$t \in T$	Generator type index

Variables

p_{ik}	Power output from generator i during representative hour k
p_{dk}	Price-responsive demand d during representative hour k
r_i	Used or invested capacity of generator i

Parameters

p_i^0	Capacity of generator i already existing at the beginning of the year
R_i	Capacity of generator i retired during the year
I_i	Capacity of generator i newly built during the year
c_i^F	Variable cost per megawatt-hour of generator i, including cost of fuel and variable operations and maintenance costs
c_i^T	Annual fixed costs per megawatt, including taxes, insurance, and fixed operating and maintenance costs
c_i^I	Levelized per-year cost per megawatt of new investment in generator i if its total investment cost is amortized over ten years
H_k	Hours per year represented by representative hour k
e_i	Carbon dioxide emission rate for generator i, tons/megawatt-hour
a_{ik}	Carbon dioxide emission price for generator i in hour k, $/ton
α_i^{min}	Minimum generation fraction of capacity for generator i
γ_{ik}	Availability factor (maximum capacity factor) for generator i in hour k
K_i	Maximum megawatts of generation capacity that can be built or used for generator i
B_{dk}	Piecewise linear consumer surplus function associated with demand d in hour k.
$F_{jj'}$	Flow limit on line from node j to node j'
$S_{jj'}$	A constant parameter known as the susceptance of the line from node j to node j'
Θ_{jk}	Phase angle of node j in hour k
GrossCS_k	Gross consumer surplus in hour k

FuelVOMCost_k	Fuel cost plus variable operation and maintenance cost ($) in hour k
InvFOMCost	Annualized investment cost (for new generators only) plus fixed operation and maintenance cost, taxes and insurance payments ($)
EnvEmissCost_k	Environmental emission cost ($) in hour k imposed by a government
ET	Emission limits; for CO_2 the unit is tons; for NO_x or SO_2 the unit is lbs
E_{ik}	Indicator to show whether generator i at hour k is included in the emission constraint. If it equals 1, generator i is included in the emission constraint at hour k. Otherwise, it equals zero

3.2 Objective function

The model maximizes annual gross consumer benefits minus the sum of variable operating cost, environmental emission cost, fixed operating cost, and annualized investment cost, as shown in the following forms:

$$\begin{aligned}\max_{p_{ik},p_{dk},r_i} \sum_k \text{GrossCS}_k - \sum_k \text{FuelVOMCost}_k \\ - \sum_k \text{EnvEmissCost}_k - \text{InvFOMCost}\end{aligned} \tag{3.1}$$

We use a piecewise linear function to represent the consumer surplus. It is the integral of the demand function mentioned above. Then the gross consumer surplus could be calculated as follows. B_{dk} is the consumer surplus function of price-responsive demand p_{dk}:

$$\text{GrossCS}_k = \sum_d H_k B_{dk}(p_{dk}) \tag{3.2}$$

We combine the fuel cost and variable operation and maintenance cost per megawatt-hour (MWh) together into the parameter c_i^F, then use it to calculate FuelVOMCost:

$$\text{FuelVOMCost}_k = \sum H_k c_i^F p_{ik} \tag{3.3}$$

Based on the heat rates and fuel types of the generators, we calculate the CO_2 emission rates for each generator. We have the emission rates of nitrogen oxides and sulfur dioxide largely from data reported to the Environmental Protection Agency. In the case of certain environmental policies, a tax will be imposed on emissions, and the associated tax payments of generators will be deducted from the objective function:

$$\text{EnvEmissCost}_k = \sum H_k a_{ik} e_i p_{ik} \tag{3.4}$$

For existing generators, the fixed cost contains fixed operation and maintenance cost and taxes and insurance payments, which together we denote c_i^I. For newly built generators, we need to add the construction cost c_i^T, which can be converted from the overnight capital costs reported by EIA, to the total fixed cost. The investment I and retirement R for each generator can be calculated from used capacity r_i and initial capacity p_i^0. Then the total fixed costs can be calculated as follows:

$$I_i = \begin{cases} 0, & \text{if generator } i \text{ is existing generator} \\ r_i, & \text{if generator } i \text{ is newly built generator} \end{cases} \tag{3.5}$$

$$R_i = \begin{cases} p_i^0 - r_i, & \text{if generator } i \text{ is existing generator} \\ 0, & \text{if generator } i \text{ is newly built generator} \end{cases} \tag{3.6}$$

$$\text{InvFOMCost} = \sum c_i^T (p_i^0 + I_i - R_i) + c_i^I I_i \tag{3.7}$$

3.3 Constraints

Constraints of the optimization problem are listed as follows:

$$p_{ik} \geq \alpha_i^{min} (p_i^0 + I_i - R_i) \tag{3.8}$$

$$p_{ik} \leq \gamma_{ik} (p_i^0 + I_i - R_i) \tag{3.9}$$

$$I_i < K_i \tag{3.10}$$

$$\sum_i p_{ik} - \sum_d p_{dk} - \sum_{j'} S_{jj'} (\Theta_{jk} - \Theta_{j'k}) = 0 \tag{3.11}$$

where i is the index of generators at node j, d is the index of demands at node j

$$|S_{jj'} (\Theta_{jk} - \Theta_{j'k})| \leq F_{jj'} \tag{3.12}$$

$$\sum_k \sum_i H_k E_{ik} e_i p_{ik} \leq \text{ET} \tag{3.13}$$

Constraints (3.8), (3.9) and (3.10) are capacity limits. Constraints (3.9) are upper bounds of energy outputs of generator i in hour k, which equal the total installed capacity multiplied by the availability factor of the generator i in hour k. For some types of generators such as coal and hydro, there are lower bounds (3.8) to ensure that minimum energy outputs of the generator i during hour k are more than a fraction of the installed capacity. For newly built generators, the invested capacities cannot exceed the maximum buildable capacities, as shown in constraint (3.10). Each line of these constraints represents a set of capacity limits which are applied to each generator and each hour. Constraints (3.11) and (3.12) are energy security and transmission line constraints. At any node of the electricity system, the generation minus the consumption should equal the power flowing out of the node, which is shown in the equality constraints (3.11). Moreover, the electricity flow on each line cannot exceed the flow limits of the line, as shown in constraints (3.12). Constraints (3.13) are emission constraints, which are set by the environmental policy. E_{ik} is an indicator vector to indicate of the generators are included in the emission constraints during representative hour k. Therefore, these constraints can be regional or seasonal emission constraints. We can include multiple emission constraints for different regions or emission types into the model.

4 Acknowledgments

This work was supported in part by the Consortium for Electric Reliability Technology Solutions (CERTS) and the Office of Electricity Delivery and Energy Reliability, Transmission Reliability Program of the U.S. Department of Energy under the National Energy Technology Laboratory Cooperative Agreement No. DE-FC26-09NT43321.

Appendix A E4ST Files and Functions

Table A-1: E4ST Files, Classes and Functions

name	description
`RunE4ST`	interface to run the simulations, which takes users' inputs and call low-level functions
`InputFiles`	interface to define the format of input data
`LtInputFiles`	implements `InputFiles` interface and defines input data for long term cases
`SigInputFile`	inherits from `LtInputFiles` class but only for one policy
`E4STResults`	processes the results and converts E4ST data to output format
`E4STOutput`	output and save results data to files
`CaseInfo`	defines simulation parameters and system-specified input data
`PolicyInfo`	defines policy-specified input data
`E4STOption`	defines E4ST option for the solver such as solver type, enable/disable crossover, etc
`E4STFactory`	interface for one complete set of runs with different polices
`LtE4STFactory`	implements `E4STFactory` interface for long term runs
`E4STDirector`	interface for one simulation with several decades for one policy
`LtE4STDirector`	implements `E4STDirector` for long terms runs
`E4STBuilder`	interface for one E4ST run with one decade for one policy
`InitE4STBuilder`	implements `E4STBuilder` interface for default year
`FirstE4STBuilder`	implements `E4STBuilder` interface for year 0
`NextE4STBuilder`	implements `E4STBuilder` interface for decade 1, decade 2, etc
`E4STPlan`	interface to define a basic E4ST object which includes mpc, offer, contab and option
`DRBuilder`	processes demand response
`GenOperation`	processes generator retirement and investment
`HelpWrap`	extra functions to finish some operations such as modifying gencost for different fuel prices
`LoadScaler`	scale loads for different purposes: total scaler, load growth, load availability
`MPCBuilder`	create mpc data from input data
`OfferBuilder`	create offer data from input data
`PolicyBuilder`	apply polices to the E4ST object
`docs/`	
`CHANGES`	E4ST change history
`E4ST-manual.pdf`	E4ST 1.0b2 Users' Manual

Appendix B Data Structures and Definitions

B.1 Data Structure of CaseInfo

CaseInfo data defines additional data and parameters for the simulations, which are not part of a standard MATPOWER case data file. The detailed definitions:

- caseInfo.fuelname – the fuel names corresponding to different types of generators.

- caseInfo.gen_info – the columns are: cost to keep($/MW/h), cost to build($/MW/h), can be built flag, gencost,tax($/MW/h), insurance($/MW/h), buildable capacity.

- caseInfo.gen_aux – auxilliary generator data for new generators. They have the same data structure as mpc.gen_aux_data. Type help idx_gen_aux for more information.

- caseInfo.gen_damages – damages of NO_x and SO_2 from each bus. The columns are: bus Number, NO_x (damage/lb), SO_2 (damage/lb).

- caseInfo.new_wind_cap – new wind buildable capacities and the coulumns are: bus number, new wind capacity (MW).

- caseInfo.af_gen – availability factors for different types of generators. A generator can have a different availability factor in each representative hour. Its availability factor equals the its maximum capability in a particular hour divided by its maximum rated capacity. Wind and solar generators often have availability factors different than 1 because there is not enough wind or sun for them to generate at their full rated capacity. Thermal generators have availability factors less than one because we derate each generator's capacity in each hour by an estimate of its probability of outage in that hour. Each row contains an availability factor for one specific hour. And the columns are:(1)coal (2)oil (3)ng(including ngct and ngcc) (4)hydro (5)nuclear (6)wind (7)refuse (8)na (9)Storage (10)Biomass (11)Unknown (12)Solar.

- caseInfo.af_probability – proportion of each year represented by each representative hour. This is a vector with number of elements equal to the number of representative hours.

- `caseInfo.af_ex_wind` – availability factors for existing windfarms. The code is written to accommodate a maximum of one wind farm at each bus, and we have aggregated wind farms in the input data accordingly. Each row is corresponding to one wind farm and each column is corresponding to one representative hour. There are similar definitions for AF of new wind, existing solar and new solar.
- `caseInfo.af_new_wind` – availability factors for newly built windfarms.
- `caseInfo.af_ex_solar` – availability factors for existing solar.
- `caseInfo.af_new_solar` – availability factors for newly built solar.
- `caseInfo.af_scalemode` – "scale mode" means whether availability factors vary by multi-bus area (0) or by individual bus (1) for exWind, exSolar, newWind, newSolar. It's a 4-element vector.
- `caseInfo.af_hours` – actual representative hours for each contingency
- `caseInfo.totalscale` – total scale factor for whole network. For example, if the loads we have are from 2013 but we are simulating 2015 and there is expected to be a 2% load increase from 2013 to 2015 in the region covered by the network in question, then this scale factor is 1.02.
- `caseInfo.loadscale` – each row is the load scale factor for each representative hour, since different representative hours have different loads. If there are multi areas each column is the corresponding scale for each area.
- `caseInfo.loadgrowth` – assumed load growth rate per year. Differs from network to network or from area to area.
- `caseInfo.dr_elasticity` – elasticity value.
- `caseInfo.dr_distcost` – assumed average retail cost per MWh on top of the wholesale generation price, in dollars per MWh.
- `caseInfo.dr_pricevector` – vector of multiples of the base-case price at which load changes.

B.2 Data Structure of PolicyInfo

Policy data defines different policies' input data as follows:

- `policy.damage` – damage values for CO_2 ($/ton), NO_x (scaling) and SO_2(scaling) emissions.
- `policy.damage_new` – damage values for emissions from newly built generators.
- `policy.emission_price` – include or exclude emission prices in the objective function.
- `policy.adj_ex_genfuel, policy.adj_ex` – set the fuel price adjustments (per million Btu) relative to the fuel prices reflected in the basecost column of the MATPOWER case data file.
- `policy.adj_new_genfuel, policy.adj_new` – specify the fuel prices of newly built generators.
- `policy.wind_sub` – wind subsidy($).
- `policy.elasticity` – price elasticity of demand.
- `policy.is_nuclear` – enable or disable the indian point nuclear power plant.
- `policy.is_newline` – enable or disable the new line between New York and Quebec.
- `policy.newline_data` – parameters of the new line.
- `policy.year_delta` – number of years between each stage of simulations.
- `policy.retirement` – establish whether capacity that was not used in the prior time period will be removed from the generator dataset prior to the beginning of this time period.
- `policy.pv_costtb` – set the hourly cost-to-build per MW of solar.

B.3 Data Structure of E4STResult

Table B-1: Data Structure of E4STResult

name	description
`busID`	bus number in `mpc.bus`
`genBus`	bus number in `mpc.gen`
`fuelTypes`	unique fuel type in `mpc.genfuel`
`genSum`	generation summed by fuel type
`genFuels`	fuel types for each generator
`genTable`	installed capacity, used capacity and generation for each generator
`buildableCapSum`	buildable capacity summed by fuel type
`usedCapTabSum`	used capacity summed by fuel type
`InvestCapSum`	invested capacity summed by fuel type
`RetireCapSum`	retired capacity summed by fuel type
`actGenTabSum`	generation summed by fuel type
`loadTable`	loads at each bus
`lmpTable`	LMP at each bus
`averageLoadPrice`	system average price weighted by load
`averageGenPrice`	system average price weighted by generation
`totalEmissions`	total emissions for CO_2, NO_x and SO_2 respectively
`totalCO2`	total CO_2 emission
`totalNOX`	total NO_x emission
`totalSO2`	total SO_2 emission
`damageCO2`	total CO_2 emission damages
`damageSONO`	total NO_x and SO_2 emission damages
`totalTax`	total tax, which is part of cost to keep
`totalInsurance`	total insurance, which is part of cost to keep
`totalFixedcost`	total fixed cost
`totalVariablecost`	total variable cost
`totalLoad`	total loads
`objValue`	objective function values
`genRawTable`	complete raw data for each generator

References

[1] Daniel L. Shawhan, John T. Taber, Di Shi, Ray D. Zimmerman, Jubo Yan, Charles M. Marquet, Yingying Qi, Biao Mao, Richard E. Schuler, William D. Schulze, and Daniel J. Tylavsky, "Does a Detailed Model of the Electricity Grid Matter? Estimating the Impacts of the Regional Greenhouse Gas Initiative," *Resource and Energy Economics*, Volume 36, Issue 1, January 2014, pp. 191–207.
http://dx.doi.org/10.1016/j.reseneeco.2013.11.015. 1.1, 1.3

[2] Biao Mao, Daniel Shawhan, Ray Zimmerman, Jubo Yan, Yujia Zhu, William Schulze, Richard Schuler, Daniel Tylavsky. "The Engineering, Economic and Environmental Electricity Simulation Tool (E4ST): Description and an Illustration of its Capability and Use as a Planning/Policy Analysis Tool" *Proceedings of the 49th Annual Hawaii International Conference on System Sciences, Computer Society Press, forthcoming January 2016*. DOI: *(forthcoming)* 1.1, 1.3

[3] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. 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. 1.1

[4] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower's Extensible Optimal Power Flow Architecture," *Power and Energy Society General Meeting, 2009 IEEE*, pp. 1–7, July 26–30 2009. 1.1

[5] The BSD 3-Clause License. [Online]. Available: http://opensource.org/licenses/BSD-3-Clause. 1.2