

ROBIN HENRY  
MATRICULATION NUMBER S1605269  
MENG PROJECT PHASE 2 REPORT  
DATA-DRIVEN ESTIMATION OF VOLTAGE  
SENSITIVITY COEFFICIENTS IN POWER  
DISTRIBUTION GRIDS  
12 JANUARY 2021

# Mission Statement

## Background

The increased displacement of conventional electricity generation in favor of renewable energy resources, such as photovoltaic, has increased the stochasticity in the generation mix, causing operational issues such as voltage regulation and congestion in lines and transformers. As a result, the management of electricity distribution networks today largely consists of designing decision-making strategies (control schemes) that adequately modulate the power injections from controllable resources connected to the grid such that the grid constraints are satisfied, while minimizing a combination of operating costs, risks, and energy losses.

Traditionally, distribution grid operators used droop-based control schemes that involved local sensing of the voltage-level and resulted in the actuation of the active and reactive power from the distributed energy resources. However, this scheme is sub-optimal and relies on local available data and grid parameters, as reported in [A]. Many works [A, B] proposed model-based grid-aware controls were the power setpoints are computed at discretised time steps in the order of sub-seconds to hours.

## Linearised OPF-based control

A common alternative approach is to solve a convex version of the original OPF by relaxing or linearising the non-convex grid constraints (e.g., nodal voltage and line current constraints), which can be efficiently solved by standard convex optimization softwares, thus allowing system operators to reach satisfying solutions in the amount of time available (e.g., 5 minutes) [B].

## Sensitivity coefficients

One proposed linearisation method is that of using so-called sensitivity coefficients [C] to model linear dependencies between controlled (e.g., node voltages, line currents) and control variables (e.g., active and reactive power injections). This way, system operators can compute the expected deviation of controlled variables (from their current values) as a function of control variable deviations, allowing them to modulate control variables in a way that ensures a secure operation of the grid (i.e., respecting network constraints such as voltage and current limits). Classical methods of dynamic sensitivity coefficient computations are based on load-flow computations, which require knowledge of the grid model (topology

and parameters) and of the network state [D]. Unfortunately, such knowledge is rarely available and up-to-date in the context of low-voltage grids, a result of outdated records and a lack of installed metering devices. Consequently, network operators must work with limited information.

As a result, research groups have started to look at alternative model-free/measurement-based methods of sensitivity coefficient computations. One notable work is that of [E], in which the authors developed a least squares (linear regression)-based approach.

## Project description

In this project, we consider the problem of managing a low-voltage electricity distribution network with limited knowledge about the topology and characteristics of the grid. In particular, we focus on developing a reliable model-free/measurement-based approach to estimate the sensitivity coefficients for (a) nodal voltages, (b) line currents, and (c) net power losses.

Broadly speaking, the project can be broken down into several stages:

1. Implementation of the least squares-based method for voltage sensitivities developed in [E] and its extension to line current sensitivities and net loss sensitivities, both for single-phase networks.
2. Investigation of other machine learning approaches to derive sensitivity coefficients (e.g., Support Vector Machines, neural networks).
3. Extend the different methods to 3-phase unbalanced networks.

Initially, each method will be tested and evaluated on a benchmark network, one for which we have the network model. A synthetic training dataset will be generated by solving load-flow computations (to obtain nodal voltage and line current values, based on nodal power injections). In order to generate a sufficiently large dataset and model real-world measurements, random noise will be added to the load-flow outcomes. The performance of each approach will be evaluated by comparing the sensitivity coefficients obtained through our model-free methods against the true ones obtained using the network model and state (as done in [B] and [D]).

Once satisfactory results are obtained on the benchmark network, the successful model-free methods will be integrated into the linearised OPF-based distributed real-time predictive control framework of [B], resulting in a novel model-free control scheme for low-voltage distribution networks.

## Later changes

The above Mission Statement is the original Mission Statement, as required by the provided MEng Project Phase 2 Guidelines. The actual work carried out and described in this report slightly differs, however, as summarised here:

1. Early on, we decided to focus exclusively on the estimation of voltage sensitivity coefficients. Reporting results for voltage, line current, and power loss sensitivity coefficients at the same time was

too time consuming and not worth it, since all estimation methods described in the sequel can be easily extended to compute line current and power loss sensitivities, too.

2. We did not extend our methods to 3-phase unbalanced networks. Instead, we decided to focus on applying the proposed estimation methods to achieve a model-less real-time voltage control of a distribution network.

## References

- [A] Christakou, K.: 'Real-Time Optimal Controls for Active Distribution Networks From Concepts to Applications', in: (2015), p.212, DOI: 10.5075/epfl-thesis-6795, URL: <http://infoscience.epfl.ch/record/214543>.
- [B] Guptal, R., Sossan, F. and Paolone, M.: 'Performance Assessment of Linearized OPF-based Distributed Real-time Predictive Control', in: *2019 IEEE Milan PowerTech*, IEEE, pp. 1-6.
- [C] Peschon, J., Piercy, D. S., Tinney, W. F. and Tveit, O. J.: 'Sensitivity in power systems', in: *IEEE Transactions on Power Apparatus and Systems* 8 (1968), pp. 1687-1696.
- [D] Christakou, K., LeBoudec, J.-Y., Paolone, M. and Tomozei, D.-C.: 'Efficient computation of sensitivity coefficients of node voltages and line currents in unbalanced radial electrical distribution networks', in: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 741-750.
- [E] Mugnier, C., Christakou, K., Jaton, J., De Vivo, M., Carpita, M. and Paolone, M.: 'Model-less/measurement-based computation of voltage sensitivities in unbalanced electrical distribution networks', in: *2016 Power Systems Computation Conference (PSCC)*, IEEE, pp. 1-7.

# Abstract

The management of low voltage distribution power grids with high penetration of intermittent renewable energy resources (e.g., photovoltaic panels) requires the design of efficient real-time control schemes accounting for the grid constraints. Since the modeling of these grid constraints is a non-linear and non-convex optimisation problem (widely known as optimal power flow), it is often tackled by linearising the non-convex grid constraints through the use of so-called sensitivity coefficients. These sensitivity coefficients can be efficiently computed by model-based methods. In some cases, however, the model of the network may be unknown or inaccurate, especially in the case of low voltage distribution networks. In this context, model-less/data-driven methods have been proposed in the literature as an alternative. In this work, the performance of (a) a least squares-based, (b) a feedforward neural network-based, and (c) a recurrent neural network-based model-less methods are investigated for the estimation of voltage sensitivity coefficients. These methods are empirically compared against one another and against an analytical model-based approach in terms of their estimation accuracy, computational requirements, and effectiveness when applied to a practical real-time voltage control problem.

# Declaration of Originality

I declare that this thesis is my  
original work except where stated.

Robin Henry   
.....

# Statement of Achievement

Throughout the project, I was provided with a number of 24-hour load flow simulations for different benchmark low voltage distribution networks (e.g., the CIGRE-13 grid referred to in this report) by my external supervisors from the Distributed Electrical Systems Laboratory group at École Polytechnique Fédérale de Lausanne, Switzerland. I therefore did not have to worry about generating realistic simulation data since it was already provided to me.

I was also provided with the true sensitivity coefficient values corresponding to those same 24-hour simulations. Although I describe the analytical method that was employed to compute them in Section 3.4.2, I did not implement it myself and used the provided values in my numerical experiments (e.g., to evaluate the accuracy of estimated sensitivity coefficients by other methods).

Finally, the basis of the implementation of the model predictive control framework considered in Chapter 6 was also provided to me as a collection of MATLAB files. I then had to adapt it to fit the particular voltage-control problem I was considering.

Apart from the aforementioned exceptions and unless otherwise stated, the remainder of this report describes my own achievements. These include:

1. A review of the existing literature on the subject of linearising the power flow equations using sensitivity coefficients in low voltage power distribution networks, with a particular focus on the voltage control problem in Chapters 2 and 3.
2. The design and/or implementation of the model-less/measurement-based sensitivity coefficient estimation methods investigated in Chapters 4 and 5.
3. The application of these estimation methods to a real-time voltage control problem in Chapter 6.

# Contents

<b>Mission Statement</b>	i
Background . . . . .	i
Project description . . . . .	ii
Later changes . . . . .	ii
<b>Abstract</b>	iv
<b>Declaration of Originality</b>	v
<b>Statement of Achievement</b>	vi
<b>List of Symbols</b>	x
<b>List of Abbreviations</b>	xiii
<b>1 Introduction</b>	1
1.1 Motivation . . . . .	1
1.2 Problem statement . . . . .	3
1.3 Research methods . . . . .	5
1.4 Contributions . . . . .	5
1.5 Structure of the thesis . . . . .	6
<b>2 Power Flow Model and Benchmark Power Distribution Network</b>	7
2.1 Introduction . . . . .	7
2.2 Power flow model . . . . .	7
2.2.1 Notations . . . . .	7
2.2.2 Bus types . . . . .	8
2.2.3 Branch model . . . . .	8
2.2.4 Power flow equations . . . . .	9
2.3 Simulation setup . . . . .	9
2.3.1 CIGRE-13 benchmark network . . . . .	9
2.3.2 Measurement noise model . . . . .	12

<b>2.4 Summary</b>	14
<b>3 Sensitivity Coefficient-based Real-Time Voltage Control in a Power Distribution Grid</b>	15
<b>3.1 Introduction</b>	15
<b>3.2 Linearisation of the power flow equations</b>	15
<b>3.3 Sensitivity coefficients</b>	16
<b>3.4 Model-based sensitivity computation</b>	17
<b>3.4.1 Classical computation of sensitivity coefficients</b>	17
<b>3.4.2 Analytical derivation of sensitivity coefficients</b>	19
<b>3.5 Real-time voltage control</b>	22
<b>3.5.1 Active vs. reactive power management</b>	22
<b>3.6 Voltage magnitude vs. real/imaginary sensitivity coefficients</b>	24
<b>3.7 Summary</b>	26
<b>4 Least Squares-based Sensitivity Coefficient Estimation</b>	27
<b>4.1 Introduction</b>	27
<b>4.2 Linear model 1</b>	28
<b>4.3 Linear model 2</b>	29
<b>4.4 Numerical stability issues</b>	30
<b>4.4.1 Measurement pre-filtering</b>	30
<b>4.4.2 Principal component regression (PCR)</b>	31
<b>4.5 Numerical results</b>	32
<b>4.6 Cramer-Rao lower bound (CRLB)</b>	36
<b>4.6.1 Lower bound derivation</b>	36
<b>4.6.2 Estimation variance: numerical results</b>	37
<b>4.7 Summary</b>	39
<b>5 Neural Network-based Sensitivity Coefficient Estimation</b>	40
<b>5.1 Introduction</b>	40
<b>5.2 Feedforward neural network (FNN) model</b>	40
<b>5.2.1 Feedforward neural networks</b>	40
<b>5.2.2 FNN-based sensitivity coefficient estimation</b>	42
<b>5.2.3 Numerical results</b>	44
<b>5.3 Long-Short Term Memory (LSTM) neural network model</b>	48
<b>5.3.1 LSTM neural networks</b>	48
<b>5.3.2 LSTM-based sensitivity coefficient estimation</b>	49
<b>5.3.3 Numerical results</b>	50
<b>5.4 Discussion</b>	53
<b>5.4.1 Computational considerations</b>	53
<b>5.4.2 Estimation variance</b>	57
<b>5.5 Summary</b>	57

<b>6 Application: Real-Time Voltage Control</b>	<b>59</b>
6.1 Introduction . . . . .	59
6.2 The voltage-control problem . . . . .	59
6.3 Simulation and results . . . . .	61
6.4 Summary . . . . .	64
<b>7 Impact and Exploitation</b>	<b>66</b>
7.1 Introduction . . . . .	66
7.2 The big picture . . . . .	66
7.3 Potential impact . . . . .	68
7.4 Summary . . . . .	68
<b>8 Conclusion</b>	<b>70</b>
8.1 Conclusions . . . . .	70
8.2 Future work . . . . .	71
<b>A Acknowledgements</b>	<b>72</b>
<b>References</b>	<b>78</b>
<b>A CIGRE-13 Network Parameters</b>	<b>79</b>

# List of Symbols

## Sets

- $\mathcal{B}$  Neural network training batch of size  $B$ .
- $\mathbb{C}$  Complex numbers.
- $\mathcal{D}$  Dataset of  $D$  neural network training input-output pairs.
- $\mathcal{E}$  Set of transmission lines.
- $\mathbb{N}$  Real numbers.
- $\mathcal{N}$  Set of  $N + 1$  buses  $\{0, 1, \dots, N\}$ .

## Operators

- $\mathbb{E}[X]$  Expected value of random variable  $X$ .
- $f(\cdot; \boldsymbol{\theta})$  Neural network function approximator with parameters  $\boldsymbol{\theta}$ .
- $f_l(\cdot)$  Nonlinear activation function of layer  $l$  of a neural network.
- $\mathcal{I}\{x\}$  Imaginary part of complex variable  $x \in \mathbb{C}$ .
- $L(\mathbf{y}, \hat{\mathbf{y}})$  Neural network training loss function.
- $\mathcal{R}\{x\}$  Real part of complex variable  $x \in \mathbb{C}$ .

## Variables

- $b_{ij}$  Total charging susceptance of branch  $(i, j)$ .
- $I_{ij}$  Complex phasor of directed branch current of branch  $(i, j)$ .
- $K_{P_{ij}}$  Voltage magnitude sensitivity coefficient at bus  $i$  w.r.t. active power injection at bus  $j$ .
- $K_{Q_{ij}}$  Voltage magnitude sensitivity coefficient at bus  $i$  w.r.t. reactive power injection at bus  $j$ .
- $N$  Number of non-slack buses in the power grid.

$\hat{p}_{g,t}$	Maximum power point (MPP) forecast of PV generation in voltage control.
$P_i$	Total active power injection at bus $i$ .
$P_{ij}$	Directed active power flow in branch $(i,j)$ .
$Q_i$	Total reactive power injection at bus $i$ .
$Q_{ij}$	Directed reactive power flow in branch $(i,j)$ .
$r_{ij}$	Total series resistance of branch $(i,j)$ .
$S_{ij}$	Sensitivity of dependent variable $x_i$ w.r.t. independent variable $u_j$ (i.e., $\partial x_i / \partial u_j$ ).
$\hat{S}_{ij}$	Estimation of true sensitivity coefficient $S_{ij}$ .
$t$	Operating timestep for which the MPC problem is formulated in voltage control.
$u_j$	The $j^{\text{th}}$ independent variable (e.g., power injection).
$V_i$	Complex voltage phasor at node $i$ .
$x_i$	The $i^{\text{th}}$ dependent variable (e.g., voltage magnitude).
$x_{g,t}$	Decision variable of curtailed active and reactive power injections from PV plant $g$ in voltage control.
$x_{ij}$	Total series reactance of branch $(i,j)$ .
$y_{ij}$	Total series admittance of branch $(i,j)$ .
$y_{ij}^{sh}$	Half of total shunt admittance of branch $(i,j)$ .

## Graphs

$G(\mathcal{N}, \mathcal{E})$  Directed graph representing the power grid.

## Vectors and Matrices

$\omega$	Vector of i.i.d. white Gaussian noise with standard deviation $\sigma$ .
$\Sigma$	Correlation matrix of the measurement errors.
$\mathbf{b}_l$	Bias vector of layer $l$ of a neural network.
$\mathbf{C}$	Covariance matrix of the measurement errors.
$\mathbf{I}$	Vector of nodal current injections into the network.
$\mathbf{I}_m$	An $m$ -by- $m$ identity matrix.

$\mathbf{S}_i$	Vector of voltage magnitude sensitivity coefficients w.r.t. active and reactive power injections (i.e., $\mathbf{S}_i = [\mathbf{K}_{\mathbf{P}}_i^T, \mathbf{K}_{\mathbf{Q}}_i^T]^T$ ).
$\hat{\mathbf{S}}_i$	Neural network prediction of $\mathbf{S}_i$ .
$\mathbf{S}$	Matrix of all voltage magnitude sensitivity coefficients w.r.t. nodal power injections, of size $(N, 2N)$ .
$\mathbf{V}$	Vector of nodal voltage phasors.
$\mathbf{W}_l$	Weight matrix of layer $l$ of a neural network.
$(\mathbf{x}_i, \mathbf{y}_i)$	The $i^{\text{th}}$ input-output pair used to train a neural network.
$\mathbf{x}^{(t)}$	Neural network input vector used to estimate sensitivity coefficients at time $t$ .
$\mathbf{Y}$	Admittance matrix of the network with elements $Y_{ij}$ .
$\hat{\mathbf{y}}_i$	Neural network output, given an input vector $\mathbf{x}_i$ .

### Parameters

$\alpha$	Maximum error introduced by the ITs on voltage and branch current magnitude measurements (in %).
$\beta$	Maximum error introduced by the ITs on voltage and branch current phasor angle measurements (in rad).
$\Delta t$	Time interval between subsequent control timesteps in voltage control.
$\gamma$	Threshold used during the measurement pre-filtering step.
$\tau$	Time window $[t_1, t_m]$ used in least squares.
$K$	Size of the time window of past measurements fed into a neural network.
$S_{max}^g$	Rated power of the PV controller in voltage control.
$v_{min}, v_{max}$	Voltage magnitude limits in voltage control.

### Other Symbols

$\Delta x(t)$	Deviation in variable $x$ between timesteps $t$ and $t + 1$ , i.e., $\Delta x(t) := x(t + 1) - x(t)$ .
$\tilde{x}$	Noisy measurement of variable $x$ .
$i \sim j$ or $(i, j)$	Transmission line linking buses $i$ and $j$ .

# List of Abbreviations

**AC** Alternative Current.

**CDF** Cumulative Distribution Function.

**CRLB** Cramer-Rao Lower Bound.

**DER** Distributed Energy Resource.

**DNO** Distribution Network Operator.

**FNN** Feedforward Neural Network.

**HV** High Voltage.

**LS** Least Squares.

**LSTM** Long-Short Term Memory.

**LV** Low Voltage.

**MPC** Model Predictive Control.

**MPP** Maximum Power Point.

**MSE** Mean Squared Error.

**MVU** Minimum Variance Unbiased.

**OPF** Optimal Power Flow.

**PCR** Principal Component Regression.

**PMU** Phasor Measurement Unit.

**PV** Photovoltaic Panel.

**RNN** Recurrent Neural Network.

**SVD** Singular Value Decomposition.

**SVM** Support Vector Machine.

# Chapter 1

## Introduction

### 1.1 Motivation

Over the last two decades, electricity networks, commonly referred to as *power grids* or *power systems*, have experienced significant structural and operational changes [1]. The liberalisation of electricity markets introduced a competitive aspect in their management, driving network improvements and cheaper energy generation [2]. The increasing contribution of distributed generators, such as wind turbines and photovoltaic panels, additionally perturbed the traditional model of decentralised generation, with the appearance of (virtual) microgrids creating local supply systems in which consumers are now also producers [3]. As a consequence, system operators are facing many new complex control problems (overvoltages, transmission line congestion, voltage coordination, investment issues, etc.). Developing efficient and reliable algorithms to manage power systems is thus particularly important to ensure a smooth transition to sustainable energy systems.

Almost all approaches currently used to tackle these decision-making problems rely on a mathematical model of the electricity flow in the network and on successfully resolving the numerous constraints imposed by the physical limitations of the grid components. In an alternating current (AC) power grid, the flow of electricity, which we refer to as the *power flow*, is governed by a set of nonlinear equations: the *power flow equations*. The physical limitations consisting of statutory limits on the voltage and limits on the lines and transformer capacity, commonly referred to as the *network constraints*, must be satisfied at all times for the whole range of operation of each generator or load connected in order to ensure a safe and reliable operation of the grid. Assuming that the physical model of the network is known (i.e., the admittance parameter and the topology), almost all decision-making problems in the management of power systems can be expressed as an instance of a class of optimization problems, known as *optimal*

*power flow* (OPF) problems, generally expressed as:

$$\underset{\mathbf{u}}{\text{minimize}} \quad f(\mathbf{u}, \mathbf{x}) \quad (1.1\text{a})$$

$$\text{subject to} \quad g(\mathbf{u}, \mathbf{x}) = \mathbf{0} \quad (1.1\text{b})$$

$$h(\mathbf{u}, \mathbf{x}) \leq \mathbf{0} \quad (1.1\text{c})$$

where  $\mathbf{u} \in \mathbb{R}^n$  and  $\mathbf{x} \in \mathbb{R}^m$  are the controllable and dependent system vector variables, respectively,  $f(\mathbf{u}, \mathbf{x}) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  is the objective function that network operators would like to minimize,  $g(\mathbf{u}, \mathbf{x}) = \mathbf{0}$  is the vector of equality constraints (e.g., energy balance), and  $h(\mathbf{u}, \mathbf{x}) \leq \mathbf{0}$  is the vector of inequality constraints (e.g., nodal voltage constraints, transmission line ratings). Since its introduction by Carpentier in 1962 [4], the OPF problem has become one of the most studied problem in the optimization literature and remains an active area of research [4–11], mainly due to the non-convexity of its original form. This non-convexity is a result of the fact that the power flow equations represent, in general, quadratic equality constraints (see Chapter 2). Examples of OPF problems include, but are not limited to, security-constrained unit commitment [12] and economic dispatch [13], optimal reactive power flow [5], and reactive power planning [14] problems.

Over the years, numerous methods have been proposed to convexify OPF problems in order to take advantage of the wide range of efficient convex programming solvers available today. A comprehensive survey of such approaches can be found in [8]. On the one hand, there exist convex relaxation approaches [9]. On the other hand, linearisation methods have been proposed. One particularly interesting method in the context of this work is that of using the first-order Taylor's approximation by so-called *sensitivity coefficients*, which Peschon et al. originally described as *the ratio  $\Delta x_i / \Delta u_j$  relating small changes  $\Delta x_i$  in some dependent variable  $x_i$  to small changes  $\Delta u_j$  in some controllable variable  $u_j$*  [15]. Under certain assumptions, these coefficients can be used in a localised linearisation of the power flow equations [16–33], which after relabeling sensitivity coefficient  $\Delta x_i / \Delta u_j$  as  $S_{ij}$  can generally be expressed as:

$$\Delta x_i \approx \sum_j S_{ij} \Delta u_j, \quad (1.2)$$

for some dependent variable  $x_i$  and all controllable variables  $u_j$ . Once sensitivity coefficients for all non-trivial  $(i, j)$  pairs are known, the dependent variables  $x_i$  in the OPF formulation (1.1) can be replaced by their linearised relationships to controllable variables  $u_j$ , using (1.2). This reformulation yields a linear constrained program, for which a global minimum can be efficiently found using known methods.

The main challenge of OPF linearisation through sensitivity coefficients remains in the computation of these coefficients. Often, the sensitivity coefficients are computed through the use of the power flow equations Jacobian matrix [15, 22, 24, 30], which explicitly provides sensitivity of power injections with respect to nodal voltage magnitudes and angles. Alternatively, sensitivity coefficients have also been computed using the adjoint matrix derived from circuit theory [20, 23, 26, 34]. As previously stated, these analytical methods all rely on the assumption that the network model, that is its topology and parameters, and state are available to network operators. In the sequel, we will refer to these approaches

as *model-based*. Unfortunately, model-based methods might be of little use in many low (or sometimes even medium) voltage networks, in which parts of the network model are often unknown due to, for example, erroneous or missing information about the status of breakers and line parameters [35, 36]. This lack of a complete and up-to-date model of the network makes model-based sensitivity coefficient computation methods inadequate for low voltage networks.

As an alternative, novel methods that do not rely on the full knowledge of the network model have been recently proposed. These new approaches have sometimes been called interchangeably *model-less*, *data-driven*, or *measurement-based* in the literature, referring to the fact that they derive sensitivity coefficients directly from a (often large) set of measurements. Until now, most data-driven methods derived the coefficients as a regression, or least squares, analytical solution to an overdetermined system of linear equations [31, 37, 38]. Although regression models are able to accurately reconstruct sensitivity coefficients in ideal conditions (i.e., in the absence of noise), the imperfection of measurement devices (e.g., smart meters, phasor measurement units (PMUs)) makes these approaches less suitable in practice where measurements are always noisy and outliers are often encountered [39]. An alternative approach, based on Support Vector Machines (SVM) [40], was also proposed in [39]. This SVM-based model was then used in a model predictive control for voltage and reactive power optimisation in [41]. Finally, although they do not explicitly compute sensitivity coefficients, the following references are also of close interest, as they build data-driven mappings between known and unknown variables in an attempt to linearise the power flow equations. In [42] and [43], such a mapping is built based on linear regression, and in [44] the authors learned a complete model of power flow using a deep convolutional neural network. Note that statistical processing of the measured data is also often required when using these data-driven methods, since changes in operating conditions may lead to different statistical distributions of available measurements.

Overall, few measurement-based methods of estimating sensitivity coefficients have been proposed so far. In addition, most proposed approaches derive them through linear regression methods, which are highly sensitive to measurement errors. It is therefore possible that more complex estimators may be able to extract more relevant information from the available measurements in order to model the nonlinear relationships between dependent and controllable variables more accurately. This is precisely what this project focuses on, as developed in later sections.

## 1.2 Problem statement

We consider a discrete-time setting in which we assume the power grid of interest is equipped with a measuring infrastructure that provides us with measurements at small regular intervals. We will assume that such measurements are provided by PMUs, which can already provide measurements at sub-second resolutions [45]. Other measurement devices could be used instead, as long as high-resolution (e.g., on the second-scale) measurements can be obtained so as to be able to track sensitivity coefficients in real time. Time steps are denoted by a discrete variable  $t = 0, 1, \dots$ .

Among the measurements available, we distinguish two categories of variables:

- *Independent variables*: the variables that are either controllable (e.g., generator output), or uncontrollable but fixed (e.g., load power injection).
- *Dependent variables*: the variables whose value is unknown but fully determined by the controllable variables and the power flow equations (e.g., voltage magnitude at a non-slack bus).

The general problem considered in this work is that of estimating sensitivity coefficients of dependent variables with respect to independent variables, in real-time and with as little knowledge of the power grid model as possible. This problem encompasses two main components:

- sensitivity coefficients vary over time and we would like to be able to track that variation,
- the methods that we develop should rely only on the availability of regular measurements from the measuring infrastructure and not on knowledge of network parameters (e.g., line parameters).

Note that the measurement infrastructure already assumes some knowledge of the network topology (e.g., the buses being monitored).

The aforementioned problem is rather general. In order to keep the scope of this project reasonable and concrete, we focus on a particular case:

- the independent variables are taken as the total net nodal power injections (both active and reactive) from all the non-slack buses and are all assumed voltage independent (e.g., constant power type loads), so that the true sensitivity coefficients can be derived using the analytical method of [30],
- the dependent variables, whose dependency with respect to the independent variables we would like to model, are the voltage magnitude at all non-slack buses.

In summary, the question tackled in this thesis can be formally expressed as:

*Let  $\mathbf{x}(t) = [x_1(t), \dots, x_N(t)] \in \mathbb{R}^N$  and  $\mathbf{u}(t) = [u_1(t), \dots, u_{2N}(t)] \in \mathbb{R}^{2N}$  denote the dependent (voltage magnitudes) and independent (power injections) variable vectors at time  $t$ , respectively, where  $N$  is the number of non-slack nodes in the network. Then let  $\tilde{\mathbf{x}}(t)$  and  $\tilde{\mathbf{u}}(t)$  denote the corresponding noisy measurements recorded by the PMUs. Given only the history of past measurements  $\mathcal{H}_t = \{(\tilde{\mathbf{x}}(s), \tilde{\mathbf{u}}(s)\}_{s=1}^t$ , how can we predict in real-time the sensitivity coefficients  $S_{ij}(t)$ , defined as:*

$$S_{ij}(t) := \frac{\partial x_i(t)}{\partial u_j(t)}, \quad (1.3)$$

*such that the next deviation in each dependent variable  $x_i$  can be predicted through a linear combination of the deviation in controllable variables:*

$$\Delta x_i(t) \approx \sum_{j=1}^{2N} S_{ij}(t) \Delta u_j(t), \quad (1.4)$$

*where  $\Delta x(t) := x(t+1) - x(t)$  and  $\Delta u(t) := u(t+1) - u(t)$ ?*

### 1.3 Research methods

In order to answer the question presented in the previous section, there are three broad families of model-less sensitivity coefficient estimation methods that are investigated in this work:

1. **Regression (least squares).** These methods are taken from the existing literature. In general, they estimate some vector of sensitivity coefficients by computing the least squares solution of an overdetermined problem in the presence of measurement noise (often assumed white Gaussian).
2. **Feedforward neural networks.** We developed a neural network-based estimation procedure in which a feedforward neural network is trained to predict vectors of sensitivity coefficients in real-time, given the recent history of measurements as input.
3. **Recurrent neural networks.** Motivated by the lack of a memory component in both the least squares and feedforward neural network methods, we investigated a method in which a Long-Short Term Memory (LSTM), a type of recurrent neural network (RNN), is used to estimate sensitivity coefficients.

In order to compare the performance of the different estimation methods, this thesis contains a number of numerical experiments. Each of these experiments is conducted on a realistic load flow simulation of a real low-voltage distribution network, as described in [Chapter 2](#). In [Chapter 3](#), the true sensitivity coefficients are computed using a model-based approach (i.e., based on the grid topology and parameters), which provides us with ground truth values against which to compare the coefficients predicted by the different models. Numerical experiments for the least squares methods are discussed in [Chapter 4](#) and for both neural networks in [Chapter 5](#).

### 1.4 Contributions

Throughout this work, the goal has been to provide a rigorous comparison between the least squares estimation methods presented in the literature and two novel neural network-based methods that we propose. Concretely, we believe this project contributes to research in the field of power systems in the following manner:

1. We propose two novel model-less/data-driven methods of voltage sensitivity coefficient estimation, one based on feedforward neural networks and one on recurrent neural networks. These methods do not rely on knowledge of the power grid model (except knowledge of the location of measuring devices) and can be used to estimate sensitivity coefficients in real-time.
2. We extensively compare our methods against the existing model-less least squares methods presented in the literature, in terms of (a) accuracy of sensitivity coefficient estimations, (b) accuracy of dependent variable estimations, (c) estimation variance, and (d) required computational time.
3. These three data-driven sensitivity coefficient estimation methods are applied to a model-less real-time voltage control problem, in which a PV power plant must be curtailed so as to avoid overvoltage

at the same bus. We compare the performance of each of these control schemes against the case where the network model is known and sensitivity coefficients are computed in a model-based manner.

4. Our results suggest that (a) all three model-less estimation methods are suitable to replace the model-based approach and (b) the feedforward neural network model outperforms the other two.

## 1.5 Structure of the thesis

The remainder of this thesis is organised as follows. In **Chapter 2**, the power grid model assumed throughout this work is reviewed, along with the 13-bus low voltage benchmark that we use in our numerical experiments. The simulation procedure to generate synthetic noisy PMU measurements is also explained.

**Chapter 3** then starts by introducing the concept of sensitivity in power systems, with a particular focus on voltage sensitivities. We then review the most widely used model-based methods for computing sensitivity coefficients. The chapter ends with a discussion about a hypothetical real-time voltage control problem to illustrate the usefulness of sensitivity coefficients in practice.

In **Chapter 4**, we explore the first type of measurement-based sensitivity coefficient estimation methods: least squares methods. Two variants are considered, each computing sensitivity coefficients as the least squares solution of an overdetermined system of linear equations.

Both neural network-based estimation methods are then introduced and investigated in **Chapter 5**. For each method, we provide a brief overview of the type of neural networks used, before describing the approach taken in this work. The chapter ends with a comparison of the performance of these novel methods against the least squares approaches from the previous chapter.

Next, **Chapter 6** discusses the application of model-less sensitivity coefficient estimation methods to a practical real-time voltage control problem. The performance of the least squares and neural network methods presented in this work are then compared.

In order to put the findings of this project into the larger picture of power systems management, **Chapter 7** discusses the importance of this research, as well as its potential societal impact.

Finally, **Chapter 8** concludes this work and presents a few potential future research directions.

## Chapter 2

# Power Flow Model and Benchmark Power Distribution Network

### 2.1 Introduction

In this chapter, we introduce the power flow model assumed throughout the remainder of this work, as well as the benchmark network used for our numerical experiments. Section 2.2, mostly provided for the reader who may be unfamiliar with power system analysis, describes the power grid model and the resulting power flow equations. Our notation, although chosen to be as close to the conventional notation used in most of the power system literature as possible, is also formally introduced in Section 2.2. Section 2.3 then describes the 13-bus CIGRE low voltage network that we use as a test-case benchmark. Finally, Section 2.3 describes the procedure followed to simulate real-world PMU measurements.

### 2.2 Power flow model

#### 2.2.1 Notations

An electricity network, or power system, can be represented as a directed graph  $G(\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N} = \{0, 1, \dots, N\}$  is a set of positive integers representing the  $N + 1$  buses (or nodes) in the network, and  $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$  is the set of directed edges linking buses together. We use the notation  $(i, j) \in \mathcal{E}$ , or simply  $i \sim j$ , to denote the branch linking a sending bus  $i \in \mathcal{N}$  and a receiving bus  $j \in \mathcal{N}$ . With each bus  $i \in \mathcal{N}$ , we associate several variables: a voltage level  $V_i$  (complex phasor), a current injection  $I_i$  (complex phasor), an active power injection  $P_i$ , and a reactive power injection  $Q_i$ . Similarly, we associate a set of variables with each branch  $(i, j) \in \mathcal{E}$ : a current flow  $I_{ij}$  (complex phasor), an active power flow  $P_{ij}$ , and a reactive flow  $Q_{ij}$ . Note that  $I_{ij}$ ,  $P_{ij}$ , and  $Q_{ij}$  are directed flows going from bus  $i$  to bus  $j$  as measured at bus  $i$ . We also write  $I_{ji}$ ,  $P_{ji}$ , and  $Q_{ji}$  to denote the directed flows from branch  $j$  to branch  $i$  as seen at bus  $j$ .

We are interested in the change of electrical quantities over time, which we denote by a discrete time index variable  $t = 0, 1, \dots$ . When the time index is not required, we omit to include it in the derivations for clarity of notation. Otherwise, we denote the quantity of any variable  $x$  at time  $t$  as  $x(t)$ . We also use  $\Delta x(t) := x(t+1) - x(t)$  to refer to the deviation in variable  $x$  between timestep  $t$  and  $t+1$ .

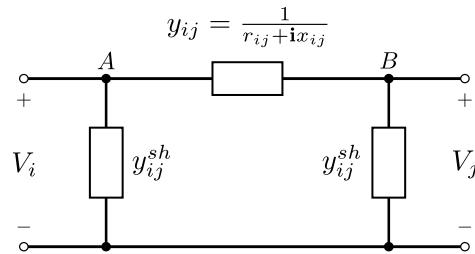
We denote the imaginary number by  $\mathbf{i}$ , with  $\mathbf{i}^2 = -1$ , and the complex conjugate of  $x \in \mathbb{C}$  as  $x^*$ .

### 2.2.2 Bus types

The state of each bus  $i$  can be fully expressed in terms of the four real-valued variables  $\{|V_i|, \angle V_i, P_i, Q_i\}$ . In power systems, we often differentiate between three types of buses: (a) slack buses, (b) PQ (or load) buses, and (c) PV (or generator) buses. Slack buses refer to buses where the variables  $\{|V_i|, \angle V_i\}$  are fixed, PQ buses where the variables  $\{P_i, Q_i\}$  are fixed, and PV buses where the variables  $\{|V_i|, P_i\}$  are fixed. The non-fixed, or unknown, variables refer to the quantities that must be determined when solving the power flow equations. As is commonly done in power system analysis, we assume that there is a single slack bus, that it is the bus with index  $i = 0$ , and that its voltage phasor is fixed to  $V = 1\angle 0^\circ$ . We also assume that all other buses are PQ buses, i.e., that their power injections are fixed and known during the power flow computation (see Section 2.2.4).

### 2.2.3 Branch model

We model each branch  $i \sim j$  by its nominal  $\pi$ -model, as shown in Figure 2.1. The branch is characterised by a series resistance  $r_{ij}$ , a series reactance  $x_{ij}$ , and a total charging susceptance  $b_{ij}$ . The branch series admittance is then given by  $y_{ij} = (r_{ij} + \mathbf{i}x_{ij})^{-1}$  and each shunt admittance by  $y_{ij}^{sh} = \mathbf{i}\frac{b_{ij}}{2}$ . We assume that the network considered does not contain any off-nominal transformers.



**Figure 2.1:** Branch nominal  $\pi$ -model.

### 2.2.4 Power flow equations

The flow of electricity in the network is dictated by the power flow equations, which we now derive. Applying Kirchhoff's current law at nodes A and B of Figure 2.1 yields:

$$\begin{cases} I_{ij} = V_i y_{ij}^{sh} + (V_i - V_j) y_{ij} \\ I_{ji} = V_j y_{ij}^{sh} + (V_j - V_i) y_{ij} \end{cases} \iff \begin{bmatrix} I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} y_{ij} + y_{ij}^{sh} & -y_{ij} \\ -y_{ij} & y_{ij} + y_{ij}^{sh} \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix}, \quad \forall (i, j) \in \mathcal{E}, \quad (2.1)$$

which form the branch current flow equations. Applying Kirchhoff's current law at each bus  $i \in \mathcal{N}$ , we obtain the nodal current injection equations:

$$I_i = \sum_{j:i \sim j} I_{ij} - \sum_{j:j \sim i} I_{ij}, \quad \forall i \in \mathcal{N}. \quad (2.2)$$

We can now formulate the power flow equations in matrix form:

$$\mathbf{I} = \mathbf{Y}\mathbf{V}, \quad (2.3)$$

where  $\mathbf{I} = [I_0, \dots, I_N]^T$  is the vector of nodal current injections,  $\mathbf{V} = [V_0, \dots, V_N]^T$  is the vector of corresponding bus voltages, and  $\mathbf{Y} \in \mathbb{C}^{(N+1) \times (N+1)}$  is the nodal admittance matrix with elements:

$$Y_{ij} = \begin{cases} -y_{ij}, & \text{if } i \neq j \text{ and } i \sim j \text{ or } j \sim i, \\ \sum_{k:i \sim k} (y_{ik} + y_{ij}^{sh}) + \sum_{k:k \sim i} (y_{ki} + y_{ki}^{sh}), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad (2.4)$$

Another alternative formulation often found in the literature is expressed in terms of nodal power injections and voltage levels, removing the need to compute current injections:

$$P_i + \mathbf{i}Q_i = V_i I_i^* = V_i \begin{bmatrix} Y_{i0}^* & Y_{i1}^* & \dots & Y_{iN}^* \end{bmatrix} \begin{bmatrix} V_0^* \\ V_1^* \\ \vdots \\ V_N^* \end{bmatrix}, \quad \forall i \in \mathcal{N}. \quad (2.5)$$

Formulation (2.5) is often referred to as the *bus injection model* [46].

## 2.3 Simulation setup

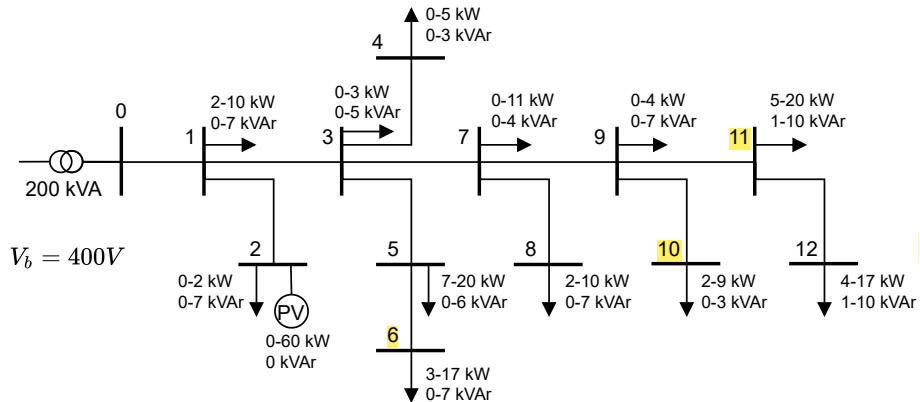
### 2.3.1 CIGRE-13 benchmark network

All the experiments presented in this work are conducted on the benchmark low voltage 200kV/400V three-phase power distribution network [47] shown in Figure 2.2 and referred to as CIGRE-13 throughout this work. It is connected to the main grid through a 200kVA transformer at the slack bus ( $i = 0$ ). The

power flow considered in our experiments never exceeds 200kVA at Bus 0, which we can therefore assume to be an ideal slack bus (i.e., with no upper bound on power flow). All other buses have a load component and Bus 2 also has a 60kW three-phase photovoltaic panels (PV) installation injecting active power into the grid. It is assumed that a PMU is installed at each of the 13 buses.

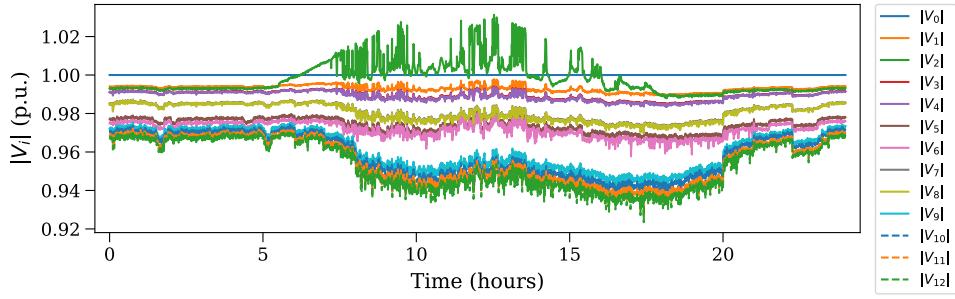
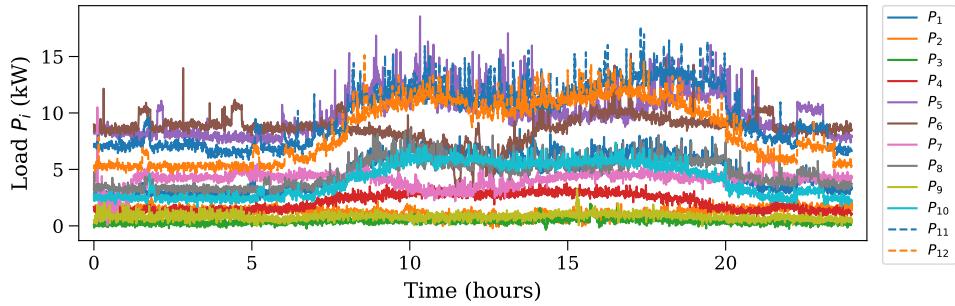
Although the network is radial, the methods investigated in this work can be applied to meshed networks. In particular, the model-based analytical method for estimating sensitivity coefficients (from [30]) discussed in Section 3.4.2 works with meshed networks and none of the three data-driven estimation methods investigated in Chapters 4 and 5 make the radial network assumption. In addition, for simplicity, the network is taken to be a single-phase network or, equivalently, a balanced three-phase one on which all derivations and experiments are conducted on one of its phases. In practice, low voltage grids are often unbalanced, but all methods investigated here can be applied to such cases without any changes. In case of unbalanced systems, sensitivity coefficients can be estimated for each phase separately (see, e.g., [31]).

The network transmission line parameters are summarised in Appendix A.

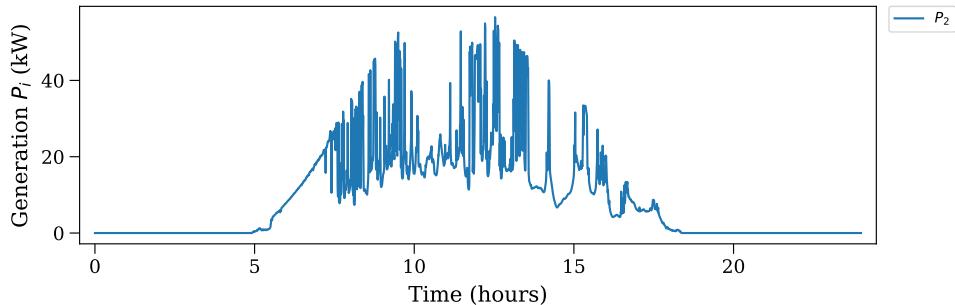


**Figure 2.2:** Network topology of the CIGRE-13 three-phase low voltage power distribution network [47], in which PMU devices are present at each bus. The low voltage network is connected to the main power grid through a 200kV/400V transformer, a 60kW three-phase PV is installed at Bus 2, and the range (minimum and peak) demands are reported at each bus.

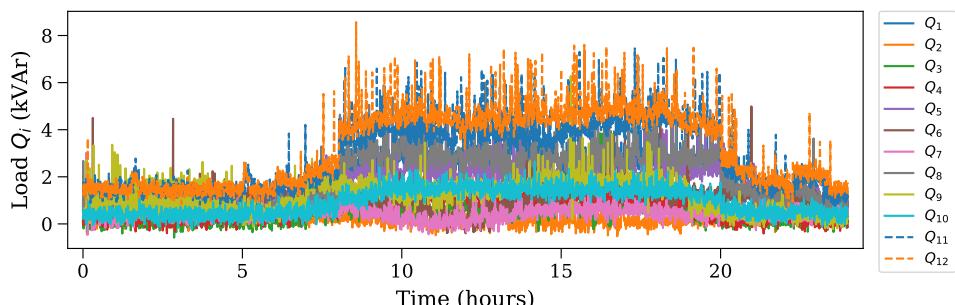
The demand and PV generation data are collected from experimental measurements at Distributed Electrical Systems Laboratory (DESL) at École Polytechnique Fédérale de Lausanne, Switzerland. Because there were many zero-injection nodes in the original CIGRE-13 system, loads were artificially placed at those buses so that sensitivity coefficients could be computed. In addition, all loads used are voltage independent (i.e., constant power type) loads. The measurements were recorded at a time-resolution of 1 measurement per second and the profiles for a single day (consisting of  $24 \times 3600 = 86400$  discrete timesteps) are shown in Figure 2.3.

(a) Nodal voltage magnitudes  $|V_i|$ .

(b) Nodal active power demand.



(c) PV generation power injection.



(d) Nodal reactive power demand.

**Figure 2.3:** Original (non-noisy) 24-hour load flow simulation on CIGRE-13.

### 2.3.2 Measurement noise model

The original CIGRE-13 simulation measurements are noiseless (i.e., they satisfy the power flow equations exactly). In order to simulate realistic scenarios, we start by adding noise in polar coordinates to the provided load flow solutions, so as to simulate noisy PMU measurements. In order to reflect real-world scenarios, in which PMUs provide measurements in polar coordinates, the simulation measurement noise is added in polar coordinates, i.e., to both the phase and angle of phasors. We assume the availability of nodal voltage phasor and branch current phasor measurements only [48]. As described in [49], the noise in the measurements comes from two sources: (a) at the instrument transformers (ITs) and (b) at the PMUs. The total noise, which is often simply referred to as the *PMU measurement error*, is thus actually the sum of the noises at the ITs and at the PMUs. In practice, IT errors are often significantly larger than PMU errors. For that reason, we only consider measurement errors introduced by ITs in the sequel (although we refer to them as the PMU measurement error).

For each benchmark network, we construct time series of noisy measurements following the procedure presented by **Algorithm 2.1** for all timesteps. We first generate random i.i.d. white Gaussian noise and add it to both the magnitude and phase of each nodal voltage and branch current phasor. Then, nodal power injection measurements are computed using the noisy voltage and current phasors. In the algorithm,  $\alpha$  and  $\beta$  refer to the maximum errors introduced by the ITs on the magnitude (in %) of the measured phasors and on the phase of those phasors (in rad), respectively [50, 51]. Values of  $\alpha$  and  $\beta$  are summarised in Table 2.1 for both IT sensor classes that we consider.

---

**Algorithm 2.1:** Noisy measurement generation

---

```

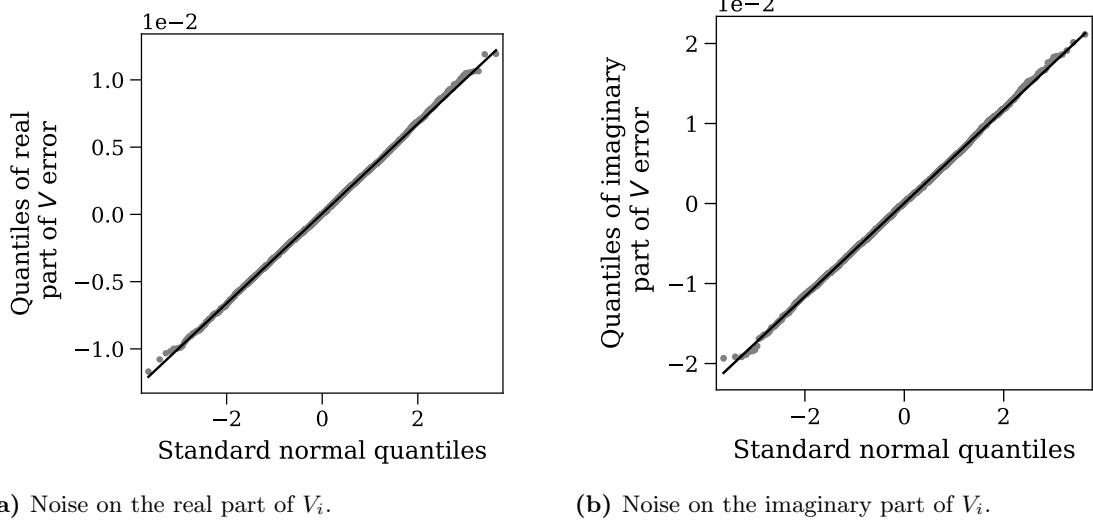
Input:  $\alpha, \beta, V_i = |V_i|e^{i\theta_{V_i}}$  ( $i = 0, \dots, N$ ),  $I_{ij} = |I_{ij}|e^{i\theta_{I_{ij}}}$  ( $(i, j) \in \mathcal{E}$ )
Output:  $\tilde{V}_i, \tilde{P}_i, \tilde{Q}_i$  ( $i = 0, \dots, N$ ),  $\tilde{I}_{ij}$  ( $(i, j) \in \mathcal{E}$ )
for  $i \leftarrow 0$  to  $N$  do
     $\epsilon_\alpha \leftarrow \mathcal{N}(0, \alpha/3)$ 
     $\epsilon_\beta \leftarrow \mathcal{N}(0, \beta/3)$ 
     $\tilde{V}_i \leftarrow (1 + \epsilon_\alpha)|V_i|e^{i(\theta_{V_i} + \epsilon_\beta)}$ 
end
for  $(i, j) \in \mathcal{E}$  do
     $\epsilon_\alpha \leftarrow \mathcal{N}(0, \alpha/3)$ 
     $\epsilon_\beta \leftarrow \mathcal{N}(0, \beta/3)$ 
     $\tilde{I}_{ij} \leftarrow (1 + \epsilon_\alpha)|I_{ij}|e^{i(\theta_{I_{ij}} + \epsilon_\beta)}$ 
end
for  $i \leftarrow 0$  to  $N$  do
     $\tilde{I}_i \leftarrow \sum_{j: i \sim j} I_{ij} - \sum_{j: j \sim i} I_{ij}$ 
     $\tilde{S}_i \leftarrow \tilde{V}_i \tilde{I}_i^*$ 
     $\tilde{P}_i, \tilde{Q}_i \leftarrow \mathcal{R}(\tilde{S}_i), \mathcal{I}(\tilde{S}_i)$ 
end

```

---

Although the noise added to the voltage and current phasors is Gaussian in polar coordinates and its transformation from polar to rectangular coordinates is nonlinear (i.e., Gaussianity is not preserved) [49], the quantile-quantile (QQ) plots of Figure 2.4 show that its quantiles (scaled by the standard deviation)

Sensor class	Max. magnitude error ( $\alpha$ ) [%]	Max. phase error ( $\beta$ ) [rad]
0.5	0.5	$9 \times 10^{-3}$
1	1	$18 \times 10^{-3}$

**Table 2.1:** Maximum errors for different instrument transformer sensor classes.**Figure 2.4:** QQ plots of the measurement noise in rectangular coordinates of the voltage phasors.

in rectangular coordinates are very close to those of a Gaussian distribution. These QQ plots are for the highest measurement noise sensor class (sensor class 1.0) and were numerically obtained on the CIGRE-13 network in steady-state. The observation that the noise in rectangular coordinates is close-to being Gaussian will be important when considering the usage of magnitude versus real/imaginary sensitivity coefficients in Section 3.6.

It must be noted that this measurement generation procedure assumes that errors occur as i.i.d. Gaussian noise on both the phase and magnitude of phasors at each timestep, and that each error is below its specified maximum value ( $\alpha$  or  $\beta$ ) with probability 0.9973, i.e., from a Gaussian distribution with a standard deviation of  $\frac{\alpha}{3}$  or  $\frac{\beta}{3}$ . Moreover, because nodal power injections are nonlinear in terms of voltages and branch currents (the only phasors we assume access to), as given by:

$$P_i + iQ_i = V_i \left( \sum_{j:i \sim j} I_{ij} - \sum_{j:j \sim i} I_{ij} \right)^*, \quad (2.6)$$

the error term on  $P_i$ , i.e.,  $\tilde{P}_i - P_i$ , and on  $Q_i$ , i.e.,  $\tilde{Q}_i - Q_i$ , are in general not normally distributed.

## 2.4 Summary

In this chapter, we laid down the foundations of the experimental framework used throughout this work to model real-world distribution power grids as accurately as possible. We started by introducing the power flow model assumed in Section 2.2, which led to the derivation of the well-known power flow equations. Section 2.3 then outlined the simulation setup that was used for all numerical experiments and which included (a) a description of the CIGRE-13 distribution network considered and (b) the procedure followed to simulate real-world noisy PMU measurements.

In the next chapter, we will look into methods that compute sensitivity coefficients by relying on the knowledge of this model of the distribution grid.

## Chapter 3

# Sensitivity Coefficient-based Real-Time Voltage Control in a Power Distribution Grid

### 3.1 Introduction

In this chapter, we introduce the notion of sensitivity in power systems and formulate a real-time voltage control problem through the use of voltage sensitivity coefficients. We consider this problem to be one of the main potential usages of the sensitivity coefficient estimation methods investigated in later chapters, which is why we believe it is important to introduce it at this stage.

### 3.2 Linearisation of the power flow equations

As briefly mentioned in the Introduction chapter, it is often advantageous to approximate the power flow equations in order to solve the OPF problem using efficient computation methods. One such approximation often found in the literature is the Taylor-series expansion of voltage magnitude in terms of nodal power injections:

$$\Delta|V_i| = \sum_{j=1}^N \left( \frac{\partial|V_i|}{\partial P_j} \Delta P_j + \frac{\partial^2|V_i|}{\partial(P_j)^2} (\Delta P_j)^2 + \dots \right) + \sum_{j=1}^N \left( \frac{\partial|V_i|}{\partial Q_j} \Delta Q_j + \frac{\partial^2|V_i|}{\partial(Q_j)^2} (\Delta Q_j)^2 + \dots \right), \quad (3.1)$$

In some cases, second-order expansions have been investigated [15, 30, 32]. Most derivations and methods derived in this work consider the first-order Taylor-series expansion only, however, since second-order coefficients are very small in practice, do not contribute significantly to voltage deviations, and ignoring them greatly simplifies the derivations. In addition, the resulting linear approximation, expressed

as:

$$\Delta|V_i| = \sum_{j=1}^N \frac{\partial|V_i|}{\partial P_j} \Delta P_j + \sum_{j=1}^N \frac{\partial|V_i|}{\partial Q_j} \Delta Q_j, \quad (3.2)$$

leads to a linear constraint in the OPF problem.

### 3.3 Sensitivity coefficients

Because we do not have access to the exact partial derivatives in (3.2), we can only approximate them. In general, we denote by  $S_{ij}$  the sensitivity coefficient that represents the deviation in some dependent variable  $x_i$  with respect to a small deviation in some independent variable  $u_j$ , i.e.:

$$S_{ij} := \frac{\partial x_i}{\partial u_j} \approx \frac{\Delta x_i}{\Delta u_j}, \quad (3.3)$$

for some small changes in dependent and independent variables  $\Delta x_i$  and  $\Delta u_j$ , respectively.

For simplicity, we focus on one type of sensitivity coefficients in particular, voltage magnitude sensitivities with respect to active and reactive nodal power injections:

$$K_{P_{ij}} := \frac{\partial|V_i|}{\partial P_j} \approx \frac{\Delta|V_i|}{\Delta P_j} \quad \text{and} \quad K_{Q_{ij}} := \frac{\partial|V_i|}{\partial Q_j} \approx \frac{\Delta|V_i|}{\Delta Q_j}, \quad (3.4)$$

as used in expression (3.2). We also briefly consider sensitivity coefficients of the real and imaginary parts of the voltage phasors (see Section 3.6):

$$K_{P_{ij}}^{re} := \frac{\partial \mathcal{R}\{V_i\}}{\partial P_j}, \quad K_{Q_{ij}}^{re} := \frac{\partial \mathcal{R}\{V_i\}}{\partial Q_j}, \quad K_{P_{ij}}^{im} := \frac{\partial \mathcal{I}\{V_i\}}{\partial P_j}, \quad K_{Q_{ij}}^{im} := \frac{\partial \mathcal{I}\{V_i\}}{\partial Q_j}. \quad (3.5)$$

We write the vectors of magnitude sensitivity coefficients of a node  $i$  with respect to active and reactive power injections at all nodes as:

$$\mathbf{K}_{\mathbf{P}_i} := \begin{bmatrix} K_{P_{i1}} & \dots & K_{P_{iN}} \end{bmatrix}^T \in \mathbb{R}^N \quad \text{and} \quad \mathbf{K}_{\mathbf{Q}_i} := \begin{bmatrix} K_{Q_{i1}} & \dots & K_{Q_{iN}} \end{bmatrix}^T \in \mathbb{R}^N, \quad (3.6)$$

which we sometimes aggregate into a vector  $\mathbf{S}_i \in \mathbb{R}^{2N}$ :

$$\mathbf{S}_i := \begin{bmatrix} \mathbf{K}_{\mathbf{P}_i} \\ \mathbf{K}_{\mathbf{Q}_i} \end{bmatrix} = \begin{bmatrix} K_{P_{i1}} & \dots & K_{P_{iN}} & K_{Q_{i1}} & \dots & K_{Q_{iN}} \end{bmatrix}^T. \quad (3.7)$$

Finally, we denote the full matrix of voltage sensitivity coefficients as  $\mathbf{S} \in \mathbb{R}^{N \times 2N}$ :

$$\mathbf{S} := \begin{bmatrix} \mathbf{S}_1^T \\ \vdots \\ \mathbf{S}_N^T \end{bmatrix}. \quad (3.8)$$

Note that all sensitivity coefficients are time-varying; the above formulations omitted time indices for clarity. It is also worth noting that all methods investigated in this work can be easily adapted to other sensitivity coefficients, such as branch current sensitivity coefficients. Whereas voltage sensitivities are useful in, for instance, voltage control problems, current sensitivity coefficients need to be considered for line congestion problems.

## 3.4 Model-based sensitivity computation

The notion of sensitivity in power systems has been used for a long time. Since their first introduction by Peschon et al. in [15] in 1968, sensitivity coefficients have been applied to a wide range of applications, as seen in the Motivation section of Chapter 1. Until recently, the computation of these coefficients relied on knowledge of the power grid parameters. Numerous such model-based methods of sensitivity estimation have been proposed over the years. The goal of this chapter is to review the most widely used of those methods. Similarly to the rest of this report, we focus on voltage sensitivities in single-phase networks, although all the methods described in the chapter can be extended to multiphase networks and other dependent variables.

In Section 3.4.1, we review the three main classical methods for estimating sensitivity coefficients in power systems. Because these methods were not used during the project, we keep their description short, with the goal of providing the reader with an overview of the history of sensitivity coefficient estimation methods. Section 3.4.2 then introduces a more recent, analytical method for estimating voltage sensitivity coefficients. We describe this method in more detail, since it is the one that was used to compute the ground-truth values of sensitivity coefficients in our numerical experiments.

### 3.4.1 Classical computation of sensitivity coefficients

#### Method 1: load flow simulations

The first method to consider consists in solving a series of load flow problems [52]. Starting from a power grid in steady-state, all voltage sensitivity coefficients can be estimated as follows:

1. Apply a small deviation of either the active or reactive power injection of a single bus in the network,
2. Solve the load flow, keeping all other power injections constant,
3. Estimate the voltage sensitivity coefficients as the observed voltage deviations at each bus,
4. Repeat for the active and reactive power injection of all buses.

The above method can be formalised as:

$$\frac{\partial|V_i|}{\partial P_j} \approx \left. \frac{\Delta|V_i|}{\Delta P_j} \right|_{\substack{\Delta P_{l,l \neq j}=0 \\ \Delta Q_l=0}} \quad \text{and} \quad \frac{\partial|V_i|}{\partial Q_j} \approx \left. \frac{\Delta|V_i|}{\Delta Q_j} \right|_{\substack{\Delta P_l=0 \\ \Delta Q_{l,l \neq j}=0}}. \quad (3.9)$$

### Method 2: inversion of the Jacobian matrix

The second method often used relies on the computation of the inverse of the Jacobian matrix associated with the load flow problem [15, 22, 24]. This approach can be summarised as follows:

1. Solve a load flow problem using the Newton-Raphson method, which makes use of the Jacobian matrix associated with the load flow, defined as:

$$J = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial |\mathbf{V}|} & \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mathbf{Q}}{\partial |\mathbf{V}|} & \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}} \end{bmatrix}, \quad (3.10)$$

where  $\mathbf{V}$ ,  $\boldsymbol{\theta}$ ,  $\mathbf{P}$ ,  $\mathbf{Q}$  are the vectors of nodal voltage magnitudes, voltage angles, active power injections, and reactive power injections, respectively.

2. Inverse the Jacobian matrix and extract the submatrices of voltage magnitude sensitivity coefficients  $\frac{\partial|\mathbf{V}|}{\partial \mathbf{P}}$  and  $\frac{\partial|\mathbf{V}|}{\partial \mathbf{Q}}$ .

Comparatively to the first method, this approach has the advantage of requiring only a single load flow problem to be solved. However, it also has the disadvantage of having to inverse the Jacobian matrix, which may potentially be large for large networks.

Traditionally, when the ratio of series line resistance over reactance (i.e.,  $R/X$ ) was negligible, the voltage magnitudes were largely dependent on reactive power flows. For that reason, it is common in the power system literature to find works that solve voltage control problems using the Jacobian submatrix  $\frac{\partial \mathbf{Q}}{\partial |\mathbf{V}|}$  only, ignoring the dependency between active power and voltage magnitude given by  $\frac{\partial \mathbf{P}}{\partial |\mathbf{V}|}$ . This is further discussed in Section 3.5.1.

### Method 3: adjoint networks

The last classical method that we consider combines the Tellegen theorem [53] with the concept of adjoint networks in power systems [26, 34]. An overview of the method is provided here:

1. From a modified version of the Tellegen theorem, derive the adjoint network associated with the original power grid,
2. Compute a base case load flow solution for the original network,
3. Express the admittance matrix of the original network  $\mathbf{Y}$  and use it to derive the admittance matrix of the adjoint network  $\mathbf{Y}_{ad}$ ,
4. Using  $\mathbf{Y}_{ad}$ , compute the bus voltage magnitudes of the adjoint network,

5. Derive the voltage magnitude sensitivity coefficients as given by an analytical formula (equation (8) of [26]).

Similarly to the Jacobian matrix-based method, this approach relies on the computation of a base case load flow solution. It must be noted, however, that it does not include any matrix-inversion step, which may be a computational advantage for some large networks. In general, however, Method 2 is often preferred due to its simpler implementation. The Jacobian matrix is usually already available as part of the Newton-Raphson load flow computation, and there is no need to form the adjoint network associated with the power grid.

### 3.4.2 Analytical derivation of sensitivity coefficients

In this section, we describe the analytical derivation of voltage sensitivity coefficients originally introduced by Christakou et al. in [30]. This method has been used in numerous later works, due to the advantages it has over the classical methods of sensitivity computation described in Section 3.4.1. Namely, (a) a single load flow solution must be computed to compute nodal voltage magnitudes and angles, (b) the method can be applied to multiphase unbalanced networks, and (c) it is more computationally efficient. In addition, the authors proved that the resulting sensitivity coefficients are unique for radial networks. Because the CIGRE-13 benchmark network that we consider is radial, it makes the work of [30] the ideal method for computing the sensitivity coefficients that we would like to later estimate using measurement-based approaches.

We now derive the voltage sensitivity coefficients using this analytical method. The original derivation was presented for three-phase networks and multiple slack buses. For simplicity, the derivation presented below was simplified to account for a single slack bus (bus  $i = 0$ ) in a single-phase network. The original full method can be found in [30].

The method starts by assuming that nodal power injections are constant and voltage independent. That is, we assume that any deviation in generation or load from one or more bus(es) does not impact the power injected at other buses. This assumption is key to most model-based methods for computing sensitivity coefficients, as it leads to the observation that any changes in voltage magnitude can be explained by sensitivity coefficients. If that were not the case, higher-order dynamics would become non-negligible and the first-order Taylor-series expansion assumed in (3.2) would no longer be accurate.

The first step in the derivation of the voltage sensitivity coefficients is to observe that the coefficients at the slack bus are null, i.e.:

$$\frac{\partial|V_0|}{P_i} = \frac{\partial|V_0|}{Q_i} = 0, \quad \forall i \in \mathcal{N}. \quad (3.11)$$

This observation results directly from the assumption that the slack bus voltage magnitude is kept constant and equal to the network rated value, with a null reference angle, i.e.,  $V_0 = 1\angle 0$  (see Section 2.2.2).

The next step is to derive the sensitivity coefficients analytically, starting from the power flow equa-

tions in the bus injection model given by (2.5):

$$S_i = V_i \sum_{j \in \mathcal{N}} (Y_{ij} V_j)^* \quad (3.12)$$

$$\iff S_i^* = V_i^* \sum_{j \in \mathcal{N}} Y_{ij} V_j. \quad (3.13)$$

Considering sensitivity coefficients with respect to active power injections first, we first note that:

$$\frac{\partial(S_i^*)}{\partial P_j} = \frac{\partial\{P_i - \mathbf{i}Q_i\}}{\partial P_j} = \mathbb{1}_{(i=j)} = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j, \end{cases} \quad (3.14)$$

since we previously assumed that power injections were constant and independent of voltage. Taking the derivative of both sides of (3.13) w.r.t.  $P_j$  gives:

$$\frac{\partial(S_i^*)}{\partial P_j} = \frac{\partial}{\partial P_j} \left( V_i^* \sum_{j \in \mathcal{N}} Y_{ij} V_j \right) \quad (3.15)$$

$$= \frac{\partial V_i^*}{\partial P_j} \sum_{j \in \mathcal{N}} Y_{ij} V_j + V_i^* \sum_{j \in \mathcal{N}} Y_{ij} \frac{\partial V_i}{\partial P_j} \quad (3.16)$$

$$= \frac{\partial V_i^*}{\partial P_j} \sum_{j \in \mathcal{N}} Y_{ij} V_j + V_i^* \sum_{j \in \mathcal{N} - \{0\}} Y_{ij} \frac{\partial V_i}{\partial P_j}, \quad (3.17)$$

where the last equality holds because of (3.11). Substituting (3.14) into (3.17) gives:

$$\mathbb{1}_{(i=j)} = \frac{\partial V_i^*}{\partial P_j} \sum_{j \in \mathcal{N}} Y_{ij} V_j + V_i^* \sum_{j \in \mathcal{N} - \{0\}} Y_{ij} \frac{\partial V_i}{\partial P_j}, \quad \forall i, j \in \mathcal{N} - \{0\}. \quad (3.18)$$

A similar result can be obtained for sensitivity coefficients w.r.t. reactive power injection, by replacing  $P_j$  with  $Q_j$ . The only difference comes from (3.14), which becomes  $\frac{\partial(S_i^*)}{\partial Q_j} = -\mathbf{i}\mathbb{1}_{(i=j)}$ . The equation for reactive power injections then becomes:

$$-\mathbf{i}\mathbb{1}_{(i=j)} = \frac{\partial V_i^*}{\partial Q_j} \sum_{j \in \mathcal{N}} Y_{ij} V_j + V_i^* \sum_{j \in \mathcal{N} - \{0\}} Y_{ij} \frac{\partial V_i}{\partial Q_j}, \quad \forall i, j \in \mathcal{N} - \{0\}. \quad (3.19)$$

Note that equality (3.18) (ref. (3.19)) is linear with respect to  $\partial V_i / \partial P_j$  and  $\partial V_i^* / \partial P_j$  (ref.  $\partial V_i / \partial Q_j$  and  $\partial V_i^* / \partial Q_j$ ). As a results, it is also linear in rectangular coordinates. That is, after expanding the partial derivatives into rectangular coordinates, the equality would be linear with respect to  $\partial \mathcal{R}\{V_i\} / \partial P_j$ ,  $\partial \mathcal{I}\{V_i\} / \partial P_j$  (ref.  $\partial \mathcal{R}\{V_i\} / \partial Q_j$ ,  $\partial \mathcal{I}\{V_i\} / \partial Q_j$ ). For a fixed bus  $i \in \mathcal{N} - \{0\}$ , expression (3.18) (ref. (3.19)) defines a linear system of  $N$  equations. This linear system, as proved in [30], has a unique solution for radial networks.

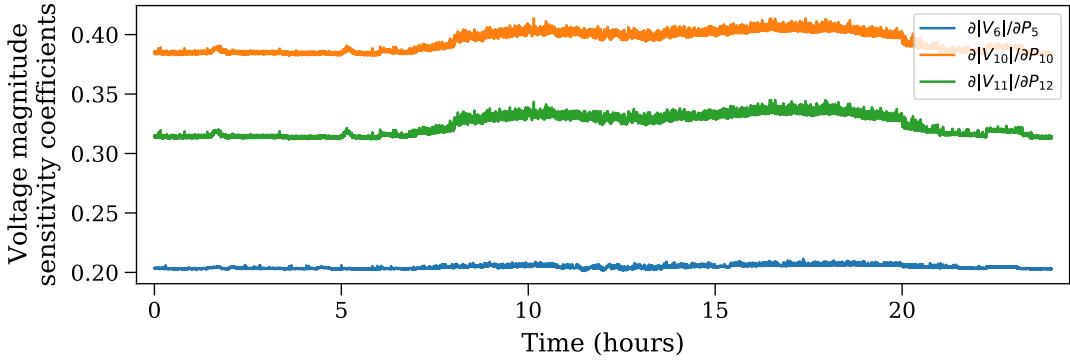
Once both linear systems are solved, the voltage magnitude sensitivity coefficients that we are inter-

ested in can be obtained as:

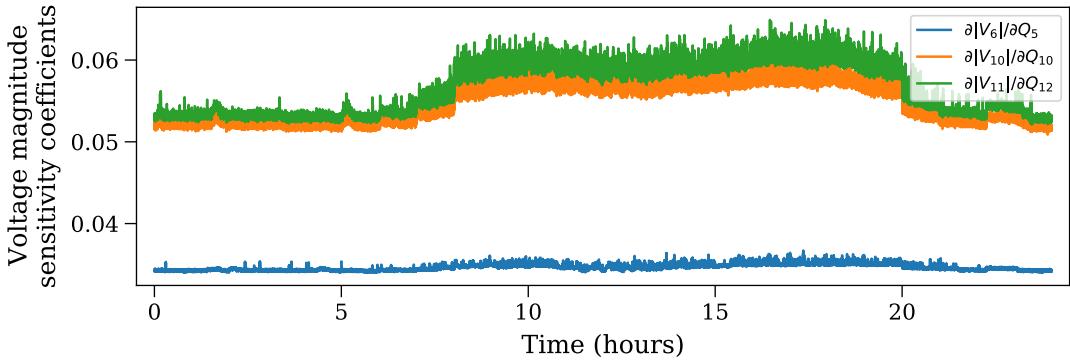
$$\frac{\partial|V_i|}{\partial P_j} = \frac{1}{|V_i|} \mathcal{R}\{V_i^* \frac{\partial V_i}{\partial P_j}\} \quad \text{and} \quad \frac{\partial|V_i|}{\partial Q_j} = \frac{1}{|V_i|} \mathcal{R}\{V_i^* \frac{\partial V_i}{\partial Q_j}\}. \quad (3.20)$$

In terms of computational requirements, the method outlined in this section outperforms the classical approaches described in Section 3.4.1, since it needs only a single load flow computation (to get the voltage phasors) and the admittance matrix of the network.

Figure 3.1 shows three sensitivity coefficients computed using the analytical method described in this section for the CIGRE-13 24-hour load flow.



(a) Sensitivity coefficients with respect to active power injections:  $\partial|V_6|/\partial P_5$ ,  $\partial|V_{10}|/\partial P_{10}$ , and  $\partial|V_{11}|/\partial P_{12}$ .



(b) Sensitivity coefficients with respect to reactive power injections:  $\partial|V_6|/\partial Q_5$ ,  $\partial|V_{10}|/\partial Q_{10}$ , and  $\partial|V_{11}|/\partial Q_{12}$ .

**Figure 3.1:** Some voltage magnitude sensitivity coefficients with respect to nodal power injections, computed using the model-based analytical method of [30] for the CIGRE-13 24-hour simulation.

### 3.5 Real-time voltage control

In order to illustrate the usefulness of computing sensitivity coefficients in distribution networks, we now discuss a practical real-time voltage control problem, inspired from [31].

Let us assume that the Distribution Network Operator (DNO) has access to a number  $N_{DER}$  of controllable distributed energy resources (DER) and that the DNO is interested in controlling those generators with the goal of minimising the voltage deviations from the network rated value  $V_r$ , i.e., to do real-time voltage control. If we denote by  $\mathcal{H}_j$  the set of feasible  $(P_j, Q_j)$  power injections from the  $j^{\text{th}}$  generator and by  $\Delta\mathbf{u} = [\Delta\mathbf{P}_{DER}, \Delta\mathbf{Q}_{DER}]$  the vector of both active and reactive power injection controllable by the DNO, the control problem can be expressed as:

$$\underset{\Delta\mathbf{u}}{\text{minimize}} \quad \sum_{i=1}^N (|V_i| + \Delta|V_i| - V_r)^2 \quad (3.21a)$$

$$\text{subject to} \quad (P_j, Q_j) \in \mathcal{H}_j, \quad j = 1, \dots, N_{DER}. \quad (3.21b)$$

In the implementation of a control scheme tailored to the above problem, the challenge arises from the modelling of the voltage magnitude deviations  $\Delta|V_i|$ , which is governed by the nonlinear power flow equations derived in the previous section. This is where sensitivity coefficients become handy, as they allow us to linearise the dependencies between controlled (i.e.,  $\Delta|V_i|$ ) and control (i.e.,  $\Delta\mathbf{u}$ ) variables:

$$\Delta|V_i| \approx \mathbf{K}_{\mathbf{P}_i} \Delta\mathbf{P} + \mathbf{K}_{\mathbf{Q}_i} \Delta\mathbf{Q}, \quad (3.22)$$

which can be substituted into (3.21a) and solved using well-established linear programming methods. Note that  $\Delta\mathbf{P}$  and  $\Delta\mathbf{Q}$  include injections from both the controllable DERs and from other uncontrollable generators and loads. In practice, the performance of the control scheme provided by (6.1) and (3.22) thus also relies on the ability of the DNO to predict future uncontrollable power injections.

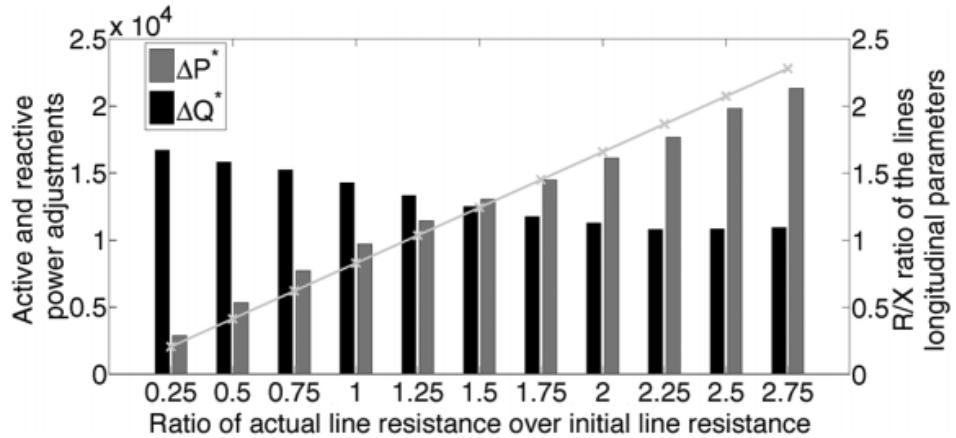
It is important to point out than any sensitivity coefficient-based voltage control scheme will only be as good as the linearised dependencies given by (3.22) are accurate. There may be certain distribution networks for which voltage deviations cannot be accurately estimated using 1st-order linear approximations only. This can be the case if, for example, nodal power injections express non-negligible voltage dependencies. If power injections are voltage dependent, then estimations of 2nd-order sensitivity coefficients would be required to adequately model such dependencies, and thus provide good approximations of  $\Delta|V_i|$ .

#### 3.5.1 Active vs. reactive power management

Traditionally, voltage control has been mostly done through reactive power management (e.g., static var compensators) [54]. This strategy works effectively in power networks that have small series  $R/X$  transmission line ratios, such as high voltage (HV) transmission networks that have negligible resistance  $R$  compared to their reactance  $X$ . Medium and low voltage distribution networks, on the other hand, are

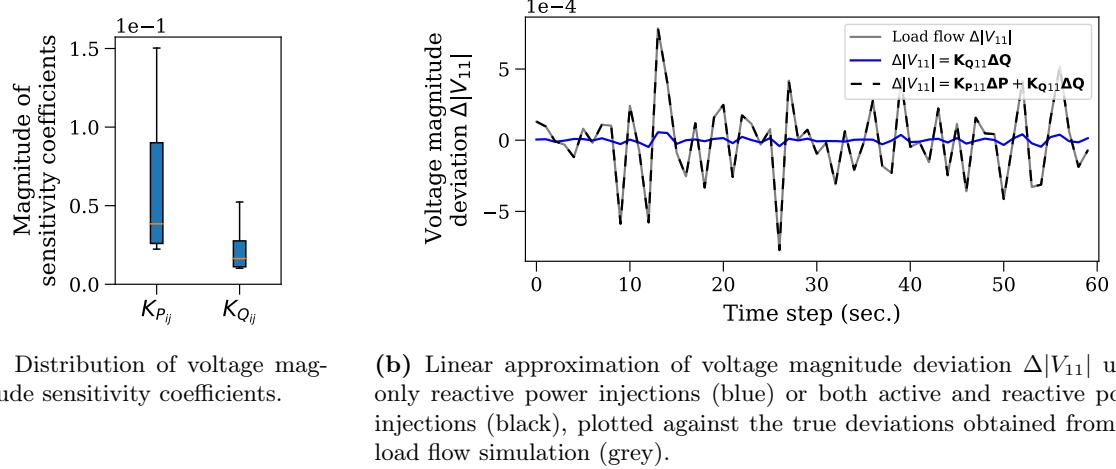
often characterised by short transmission lines with higher resistance, leading to larger  $R/X$  ratios. In such power grids, it is no longer possible to consider active and reactive power injections to be decoupled with respect to voltages. Voltage control schemes must thus also include an active power management component [55].

Whereas further simplifying (3.22) as  $\Delta|V_i| \approx \mathbf{K}_{\mathbf{Q}_i} \Delta \mathbf{Q}$  is often encountered in traditional voltage control with reactive power (volt-var control), doing so would lead to inaccurate voltage deviation estimations. This claim is illustrated by Figure 3.2 (taken from [55]), which illustrates the changes in active and reactive power needed to improve the voltage magnitude of a network bus by 2% as a function of the  $R/X$  ratio of the network lines. This figure was created by varying the line resistances from 0.25 to 2.75 times their initial values while keeping the line reactances constant. From left to right, as the  $R/X$  line ratio increases (light grey line), the active power dependency of the voltage magnitude also increases, leading to situations in which active power management becomes more important than reactive power management for voltage control.



**Figure 3.2:** Optimal active and reactive power adjustments necessary to improve the voltage by 2% as a function of the line parameters (taken from [55]).

We are interested in the computation of voltage sensitivity coefficients in networks for which we do not have adequate knowledge of the power grid parameters, such as low voltage distribution networks. Consequently, it will often also be the case that those networks fall into the category of grids that have an  $R/X$  transmission line ratio large enough that active power management cannot be ignored (i.e., right side of Figure 3.2). This is further illustrated by Figure 3.3, which was made using our CIGRE-13 simulation data and for which the average  $R/X$  line ratio is 8.7. Figure 3.3a compares the distribution of the magnitude of voltage sensitivity coefficients with respect to active power injections,  $K_{P_{ij}}$ , and with respect to reactive power injections,  $K_{Q_{ij}}$ . The fact that the  $K_{P_{ij}}$  coefficients are often larger in magnitudes than the  $K_{Q_{ij}}$  coefficients confirms our earlier assumption that active power injections cannot be ignored. Figure 3.3b complements this observation with a plot of the reconstructed  $\Delta|V_i|$  at node  $i = 11$ . The blue curve represents the estimated voltage if only reactive power injections are



(a) Distribution of voltage magnitude sensitivity coefficients.

(b) Linear approximation of voltage magnitude deviation  $\Delta|V_{11}|$  using only reactive power injections (blue) or both active and reactive power injections (black), plotted against the true deviations obtained from the load flow simulation (grey).

**Figure 3.3:** A comparison between voltage magnitude sensitivity with respect to active and reactive power injections in terms of (a) the magnitude of the coefficients and (b) the quality of the resulting linear approximation of  $\Delta|V_{11}|$ .

considered (i.e.,  $\Delta|V_{11}| = \mathbf{K}_{Q_{11}}\Delta\mathbf{Q}$ ), whereas the black one takes into account both active and reactive power (i.e.,  $\Delta|V_{11}| = \mathbf{K}_{P_{11}}\Delta\mathbf{P} + \mathbf{K}_{Q_{11}}\Delta\mathbf{Q}$ ). Clearly, the true voltage magnitude deviations cannot be expressed independently of active power injections. On the contrary, if both active and reactive power injections are considered, the local linear approximation is almost indistinguishable from the true voltage deviations computed from the load flow, shown in grey.

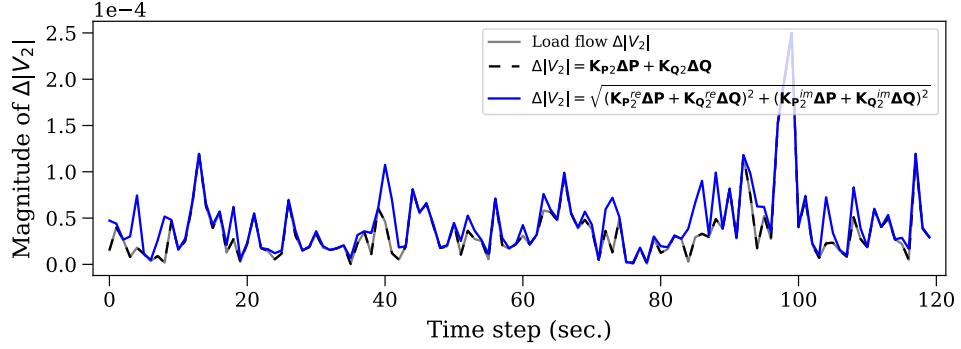
For those reasons, all estimation methods and results presented in the remainder of this work will assume the usage of both types of sensitivity coefficients in (3.22).

### 3.6 Voltage magnitude vs. real/imaginary sensitivity coefficients

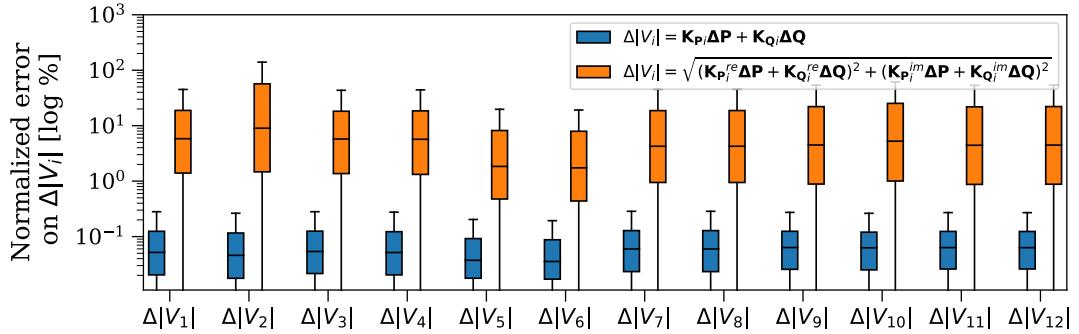
As previously stated, our end goal is to linearise the dependencies between voltage magnitude and nodal power injections in order to, for instance, formulate a voltage control problem as a linear program. An alternative to directly computing the voltage magnitude sensitivity coefficients  $K_{P_{ij}}$  and  $K_{Q_{ij}}$  is to compute the sensitivity coefficients of the real and imaginary parts of the voltage phasor with respect to nodal power injections, as given by (3.5) and from which voltage magnitude deviations can be estimated as:

$$\Delta|V_i| \approx \sqrt{(\mathbf{K}_{P_i}^{re}\Delta\mathbf{P} + \mathbf{K}_{Q_i}^{re}\Delta\mathbf{Q})^2 + (\mathbf{K}_{P_i}^{im}\Delta\mathbf{P} + \mathbf{K}_{Q_i}^{im}\Delta\mathbf{Q})^2}. \quad (3.23)$$

Although (3.23) is no longer a linear combination of nodal power injections, it remains a convex expression and can thus still be substituted into (6.1) to yield a convex optimization problem that can be efficiently solved. Similarly to voltage magnitude sensitivity coefficients, the coefficients in rectangular coordinates defined by (3.5) can also be computed using both the model-based and model-free estimation methods



(a) Magnitude of the linear approximation of  $\Delta|V_{11}|$  using voltage magnitude sensitivity coefficients (black) or real and imaginary sensitivity coefficients of the voltage phasor (blue), plotted against the true deviations obtained from the load flow simulation (grey).



(b) Distribution of the normalised error on the estimation of  $\Delta|V_i|$  using voltage magnitude sensitivity coefficients (blue) or real and imaginary sensitivity coefficients of the voltage phasor (orange).

**Figure 3.4:** Comparison between approximations of the voltage magnitude deviation  $\Delta|V_i|$  using voltage magnitude sensitivity coefficients or real and imaginary sensitivity coefficients of the voltage phasor.

presented in this work.

For the CIGRE-13 benchmark network that we use, however, it was observed that voltage magnitude coefficients led to a better linearisation of voltage in terms of power injections. This is illustrated in Figure 3.4a, in which the magnitude of the voltage deviation  $\Delta|V_2|$  is estimated using the voltage magnitude sensitivity coefficients (black) or the coefficients in rectangular coordinates (blue) over the first 2 minutes of the day. Figure 3.4b then shows the distribution over the 24 hours of the normalised errors on the estimation of  $\Delta|V_i|$ , computed as:

$$\left| \frac{\Delta|V_i^{(true)}| - \Delta|V_i|}{\Delta|V_i^{(true)}|} \right|, \quad (3.24)$$

for all non-slack buses. Both figures were constructed using the noiseless power injection measurements.

Clearly, the voltage magnitude coefficients lead to better approximations. It must be noted, however,

that this may not always be the case. Various changes in network topology and parameters may affect the quality of the estimations using voltage magnitude sensitivity coefficients, and vice-versa. This is however not something that was investigated in this work. Instead, we decided to use sensitivity coefficients in polar coordinates for most of our analyses.

### 3.7 Summary

In this chapter, we discussed the usage of voltage sensitivity coefficients as a means to linearise the power flow equations in distribution power grids, with a particular focus on the problem of real-time voltage control. We also illustrated the fact that in low voltage networks with high  $R/X$  series transmission line ratio, voltage control through reactive power only is not sufficient and active power must also be considered. Finally, we compared the adequacy of using voltage magnitude versus real/imaginary sensitivity coefficients and concluded that the former yielded better linear approximations, at least for the CIGRE-13 simulation we consider.

In the next chapter, we will look at a first measurement-based approach that estimates sensitivity coefficients by computing the least squares solution of an overdetermined system of linear equations.

## Chapter 4

# Least Squares-based Sensitivity Coefficient Estimation

### 4.1 Introduction

The goal of this project was to investigate data-driven sensitivity coefficient estimation methods and in particular whether neural networks could be used for that task. With this objective in mind, an important portion of the project (around a third) was first spent working on the implementation and validation of a model-less method already present in the literature and which could be used to provide a benchmark performance. We refer to this method, which is explored in detail in this chapter, as the *least squares* approach. It involves computing the least squares solution of an overdetermined system of linear equations. Because the least squares approach has already been used in several works in the literature (see [31, 37, 38]), it makes it an ideal benchmark against which to compare the performance of neural network-based methods introduced in later chapters.

We consider two distinct formulations of the least squares approach, which we refer to as *Model 1* and *Model 2*, respectively, and which differ in terms of the formulation of the linear model used to represent the dependency between voltage magnitude and power injections. Model 1 is described in Section 4.2 and Model 2 in Section 4.3. Both methods rely on the assumption that sensitivity coefficients vary little over time, such that we can estimate their value at time  $t$  using measurements collected during the previous  $m$  time steps  $\{t - m, \dots, t\}$ , for some large  $m$ . All our experiments on CIGRE-13 suggest that this assumption is reasonable (see, e.g., Figure 3.1 in which coefficients slowly change over time). In addition, as we discovered through our experiments, both linear models suffer from non-negligible numerical stability issues, which leads to generated sensitivity coefficients with large estimation variances. We investigate ways to deal with those problems in Section 4.4.

The performance of the least squares approach on the CIGRE-13 benchmark network is illustrated in Section 4.5. Finally, Section 4.6 compares Model 1 and Model 2 in terms of the lowest estimation variance theoretically achievable by these unbiased estimators, also known as their Cramer-Rao lower

bounds.

## 4.2 Linear model 1

The first model that we consider is inspired from the work of Mugnier et al. in [31].

Let  $\Delta\tilde{\mathbf{x}}_{i,\tau} \in \mathbb{R}^m$  be the vector of dependent variable  $x_i$  measurement deviations and let  $\Delta\tilde{\mathbf{u}}_\tau \in \mathbb{R}^{m \times 2N}$  be the matrix of measurement deviations of independent variables over a time window  $\tau = [t_1, t_m]$ , defined as:

$$\Delta\tilde{\mathbf{x}}_{i,\tau} := \begin{bmatrix} \Delta\tilde{x}_i(t_1) \\ \vdots \\ \Delta\tilde{x}_i(t_m) \end{bmatrix}, \quad \Delta\tilde{\mathbf{u}}_\tau := \begin{bmatrix} \Delta\tilde{u}(t_1)^T \\ \vdots \\ \Delta\tilde{u}(t_m)^T \end{bmatrix} = \begin{bmatrix} \Delta\tilde{u}_1(t_1) & \dots & \Delta\tilde{u}_{2N}(t_1) \\ \vdots & \ddots & \vdots \\ \Delta\tilde{u}_1(t_m) & \dots & \Delta\tilde{u}_{2N}(t_m) \end{bmatrix}. \quad (4.1)$$

In the case of voltage sensitivity coefficients with respect to nodal power injections, (4.1) refers to matrices:

$$\Delta\tilde{\mathbf{x}}_{i,\tau} := \begin{bmatrix} \Delta|\tilde{V}_i(t_1)| \\ \vdots \\ \Delta|\tilde{V}_i(t_m)| \end{bmatrix}, \quad \Delta\tilde{\mathbf{u}}_\tau := \begin{bmatrix} \Delta\tilde{P}_1(t_1) & \dots & \Delta\tilde{P}_N(t_1) & \Delta\tilde{Q}_1(t_1) & \dots & \Delta\tilde{Q}_N(t_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Delta\tilde{P}_1(t_m) & \dots & \Delta\tilde{P}_N(t_m) & \Delta\tilde{Q}_1(t_m) & \dots & \Delta\tilde{Q}_N(t_m) \end{bmatrix}. \quad (4.2)$$

Provided that the number of measurements  $m$  in the time window  $\tau = [t_1, t_m]$  satisfies  $m > 2N$ , where  $2N$  is the number of independent variables, we can formulate an over-determined system of linear equations to represent the linear dependencies between the measured deviations in dependent and independent variables, in terms of the sensitivity coefficient vector  $\mathbf{S}_i$ :

$$\Delta\tilde{\mathbf{x}}_{i,\tau} = \Delta\tilde{\mathbf{u}}_\tau \mathbf{S}_i + \boldsymbol{\omega}, \quad (4.3)$$

where  $\boldsymbol{\omega}$  is the vector of measurement errors.

In general,  $\boldsymbol{\omega}$  accounts for the measurement noise present on the measurements of both dependent ( $\Delta\tilde{\mathbf{x}}_{i,\tau}$ ) and independent ( $\Delta\tilde{\mathbf{u}}_\tau$ ) variables. For the least squares approach presented here to be valid, we must assume that the effect of the errors on  $\Delta\tilde{\mathbf{u}}_\tau$  are negligible compared to their effect on  $\Delta\tilde{\mathbf{x}}_{i,\tau}$  [31]. As a result,  $\boldsymbol{\omega}$  is often assumed to correspond to the noise added to the voltage magnitudes only. In addition,  $\boldsymbol{\omega}$  is generally assumed to be white Gaussian (zero-mean) with a variance that is specified by the PMU manufacturer in terms of % of the true quantity, similarly to the simulation procedure previously introduced in Section 2.3.2. This assumption is, however, not always realistic and is one of the reasons why alternatives to least squares-based methods may lead to better results.

Given these assumptions, the vector of sensitivity coefficients of interest  $\mathbf{S}_i$  can be estimated as the least squares analytical solution of (4.3):

$$\mathbf{S}_i = (\Delta\tilde{\mathbf{u}}_\tau^T \Sigma^{-1} \Delta\tilde{\mathbf{u}}_\tau)^{-1} \Delta\tilde{\mathbf{u}}_\tau^T \Delta\tilde{\mathbf{x}}_{i,\tau}, \quad (4.4)$$

for all  $i = 1, \dots, N$  and where  $\Sigma \in \mathbb{R}^{m \times m}$  is the correlation matrix of the errors  $\boldsymbol{\omega}$ . The correlation

matrix  $\Sigma$  thus expresses the correlation between measurement errors at different time steps within the time window  $\tau$ . It takes the form:

$$\Sigma = \begin{bmatrix} 1 & -0.5 & & \\ -0.5 & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots \\ 0 & \ddots & \ddots & -0.5 \\ & & -0.5 & 1 \end{bmatrix} \quad (4.5)$$

and its full derivation can be found in [31].

### 4.3 Linear model 2

The second model that we consider is derived from the work of Valverde et al. in [56], in which they formulate the over-determined system of linear equations not in terms of measurement deviations between subsequent timesteps, but instead express the voltage at bus  $i$  as a linear combination of the power injections at other buses.

Let  $\tilde{\mathbf{x}}_{i,\tau} \in \mathbb{R}^m$  be the vector of dependent variable  $x_i$  measurements and let  $\tilde{\mathbf{u}}_\tau \in \mathbb{R}^{m \times 2N}$  be the matrix of independent variables measurements over the time window  $\tau = [t_1, t_m]$ , defined as:

$$\tilde{\mathbf{x}}_{i,\tau} := \begin{bmatrix} \tilde{x}_i(t_1) \\ \vdots \\ \tilde{x}_i(t_m) \end{bmatrix}, \quad \tilde{\mathbf{u}}_\tau := \begin{bmatrix} \tilde{u}(t_1)^T \\ \vdots \\ \tilde{u}(t_m)^T \end{bmatrix} = \begin{bmatrix} \tilde{u}_1(t_1) & \dots & \tilde{u}_{2N}(t_1) \\ \vdots & \ddots & \vdots \\ \tilde{u}_1(t_m) & \dots & \tilde{u}_{2N}(t_m) \end{bmatrix}. \quad (4.6)$$

In the case of voltage sensitivity coefficients with respect to nodal power injections, (4.6) refers to matrices:

$$\tilde{\mathbf{x}}_{i,\tau} := \begin{bmatrix} |\tilde{V}_i(t_1)| \\ \vdots \\ |\tilde{V}_i(t_m)| \end{bmatrix}, \quad \tilde{\mathbf{u}}_\tau := \begin{bmatrix} \tilde{P}_1(t_1) & \dots & \tilde{P}_N(t_1) & \tilde{Q}_1(t_1) & \dots & \tilde{Q}_N(t_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \tilde{P}_1(t_m) & \dots & \tilde{P}_N(t_m) & \tilde{Q}_1(t_m) & \dots & \tilde{Q}_N(t_m) \end{bmatrix}. \quad (4.7)$$

Similarly to Model 1, provided that the size of the window  $m$  satisfies  $m > 2N$ , we can formulate a new over-determined system of linear equations:

$$\tilde{\mathbf{x}}_{i,\tau} = \tilde{x}_{i,\tau}^{(0)} + \tilde{\mathbf{u}}_\tau \mathbf{S}_i + \boldsymbol{\omega}, \quad (4.8)$$

where  $\tilde{x}_{i,\tau}^{(0)} \in \mathbb{R}^m$  denotes the base value for variable  $x_i$ , also known as the *bias* or *intercept* term in linear regression, and  $\boldsymbol{\omega}$  contains the measurement errors, again assumed white Gaussian with a known variance.

In the case of voltage sensitivity coefficients, a good approximation for the bias terms would be the transformer voltage magnitude, as suggested in [56]. Because we may not have access to measurements

of  $x^{(0)}$ , however, Valverde et al. proposed to replace it by the mean value  $\bar{x}_{i,\tau}$  of the  $\tilde{x}_i(t)$  measurements over the interval  $\tau$ . Although this approximation is often close to the actual value, a better approach would be to include the intercept term as part of the parameter vector to be estimated, as is regularly done in linear regression. This leads to a new overdetermined system of equations, with an additional unknown quantity to be estimated compared to (4.8):

$$\tilde{\mathbf{x}}_{i,\tau} = \tilde{\mathbf{v}}_\tau \begin{bmatrix} \mathbf{S}_i \\ x_{i,\tau}^{(0)} \end{bmatrix} + \boldsymbol{\omega}, \quad \text{with } \tilde{\mathbf{v}}_\tau = \begin{bmatrix} \tilde{\mathbf{u}}_\tau & \mathbf{1}_m \end{bmatrix}, \quad (4.9)$$

where  $\mathbf{1}_m \in \mathbb{R}^m$  is a column vector of 1's.

A least squares solution to (4.9) can be found similarly to the solution presented for Model 1 (given by (4.4)). The only difference in the formulation of the solution is that the correlation matrix  $\Sigma$  can be replaced by the identity matrix, since all measurement errors between different timesteps in  $\boldsymbol{\omega}$  are assumed independent. This yields the simpler solution:

$$\begin{bmatrix} \mathbf{S}_i \\ x_{i,\tau}^{(0)} \end{bmatrix} = (\tilde{\mathbf{v}}_\tau^T \tilde{\mathbf{v}}_\tau)^{-1} \tilde{\mathbf{v}}_\tau^T \tilde{\mathbf{x}}_{i,\tau}. \quad (4.10)$$

## 4.4 Numerical stability issues

Computing the vector  $\mathbf{S}_i$  of sensitivity coefficients through the analytical formulae (4.4) and (4.10) for each time window is straightforward and easy to implement in practice. As suggested by [56] and confirmed by some of our experiments on simulated data, however, both methods suffer from numerical stability issues. This happens when the matrix to be inverted ( $\Delta \tilde{\mathbf{u}}_\tau^T \Sigma^{-1} \Delta \tilde{\mathbf{u}}_\tau$  or  $\tilde{\mathbf{v}}_\tau^T \tilde{\mathbf{v}}_\tau$ ) is ill-conditioned (almost singular), which may lead to large numerical errors in the computation of the analytical solution.

### 4.4.1 Measurement pre-filtering

In both models, this can happen when there is little variation in measurements between subsequent timesteps. This leads to deviation matrices  $\Delta \tilde{\mathbf{u}}_\tau$  and  $\Delta \tilde{\mathbf{x}}_{i,\tau}$  essentially composed of noise in the case of Model 1 and matrices  $\tilde{\mathbf{v}}_\tau$  and  $\tilde{\mathbf{x}}_{i,\tau}$  with highly correlated rows in the case of Model 2.

In an attempt to deal with this issue and therefore to increase the numerical stability of their method, Mugnier et al. proposed to include a pre-filtering step to discard timesteps during which measurement deviations were small enough to be likely consisting of noise only [31]. We replicate their idea in our experiments, as summarised by Algorithm 4.1. In essence, our pre-filtering step decreases the measurement resolution in the case of power grids in which little variation is observed.

The algorithm works as follows. We start by saving the measurements from the initial time step  $t = 1$ . We then iterate through all timesteps and discard measurements for which the power injections do not vary significantly compared to those recorded at the last measurements saved. The comparison is achieved by computing the average deviation in power injections and comparing it to a pre-specified threshold value  $\gamma \geq 0$ . In the case of Model 1, we also later update the correlation matrix  $\Sigma$  accordingly

by setting the entries corresponding to non-subsequent timesteps to 0.

---

**Algorithm 4.1:** Measurement pre-filtering step.

---

**Input:**  $\gamma$ ,  $\tilde{x}_i(t)$  ( $i = 1, \dots, N$ ),  $\tilde{u}_j(t)$  ( $j = 1, \dots, 2N$ ) for  $t = t_1, \dots, t_m$   
**Output:**  $\tilde{x}_i(t)$  ( $i = 1, \dots, N$ ),  $\tilde{u}_j(t)$  ( $j = 1, \dots, 2N$ ) for pre-filtered timesteps in  $\tau$   
 $\mathbf{PQ} \leftarrow [\tilde{u}_1(1), \dots, \tilde{u}_{2N}(1)]^T$   
**for**  $t_k \leftarrow 2$  **to**  $T$  **do**  
   $\Delta \leftarrow \frac{1}{2N} \sum_{l=1}^{2N} |\mathbf{PQ}_l - u_l(t_k)|$   
  **if**  $\Delta < \gamma$  **then**  
    Discard  $\tilde{x}_i(t_k)$  for all  $i = 1, \dots, N$   
    Discard  $\tilde{u}_j(t_k)$  for all  $j = 1, \dots, 2N$   
  **else**  
     $\mathbf{PQ} \leftarrow [\tilde{u}_1(t_k), \dots, \tilde{u}_{2N}(t_k)]^T$   
  **end**  
**end**

---

#### 4.4.2 Principal component regression (PCR)

For both Model 1 and Model 2, the matrix to be inverted may also become ill-conditioned due to a multicollinearity problem between nodal power injections, in which case different columns of  $\Delta\tilde{\mathbf{u}}_\tau$  or  $\tilde{\mathbf{v}}_\tau$  are highly correlated. This may happen when a bus retains a near-constant power factor over the time interval  $\tau$  for example, or when the power injections of several nodes follow a very similar pattern, such as nearby photovoltaic panel installations.

Let us consider the general least-squares solution:

$$\boldsymbol{\beta}^{LS} = (\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \quad (4.11)$$

which may refer to either (4.4), or to (4.10) with  $\boldsymbol{\Sigma} = \mathbf{I}$ , and where  $\mathbf{X}$  is  $m$ -by- $2N$ . A well-established numerical method to deal with multicollinearity in (4.11) is *Principal Component Regression* (PCR) [56, 57]. In PCR, the original matrix  $\mathbf{X}$  is transformed into a matrix  $\mathbf{X}_k$  of its  $k$  non-correlated principal components. The least-squares problem is solved in the principal component domain using  $\mathbf{X}_k$  and the solution is then transformed back to the original domain, avoiding to explicitly compute the inverse of  $\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X}$ .

The PCR method works as follows. For the  $m$ -by- $2N$  matrix  $\mathbf{X}$ , we start by normalising each column by its mean value to form  $\mathbf{X}'$ , so that all the columns of  $\mathbf{X}'$  are zero-centered. The singular value decomposition (SVD) of  $\mathbf{X}'$  is then computed:

$$\mathbf{X}' = \mathbf{U}_X \boldsymbol{\Sigma}_X \mathbf{V}_X^T, \quad (4.12)$$

where  $\mathbf{U}_X$  and  $\mathbf{V}_X$  are  $m$ -by- $m$  and  $2N$ -by- $2N$  orthogonal matrices that contain the left and right-singular vectors of  $\mathbf{X}'$ , respectively, and  $\boldsymbol{\Sigma}_X$  is a diagonal matrix with the singular values of  $\mathbf{X}'$  in descending order on its diagonal. The second step of PCR is to compute the matrix  $\hat{\mathbf{X}}^{(k)} = \mathbf{X}' \mathbf{V}_X^{(k)}$ ,

where  $\mathbf{V}_X^{(k)}$  is the  $2N$ -by- $k$  matrix consisting of the first  $k$  columns of  $\mathbf{V}_X$ , which correspond to the largest singular values. Decreasing the value of  $k$  from  $2N$  will ignore the smallest singular values, which can be thought of as ignoring the columns of  $\mathbf{X}$  that are highly correlated, leading to a less ill-conditioned matrix to be inverted and a more numerical stable solution.

The least-squares solution in the principal component domain  $\hat{\boldsymbol{\beta}}^{LS}$  can then be obtained by solving (4.11) after replacing  $\mathbf{X}$  by  $\hat{\mathbf{X}}^{(k)}$ . Finally, the final solution can be obtained by transforming  $\hat{\boldsymbol{\beta}}^{LS}$  back to the original domain, using:

$$\boldsymbol{\beta}^{LS} = \mathbf{V}_X^{(k)} \hat{\boldsymbol{\beta}}^{LS}. \quad (4.13)$$

Because the CIGRE-13 simulation introduced in Section 2.3.2 does not present significant multicollinearity between power injections, we did not observe any improvements on the estimation of the sensitivity coefficients using PCR.

## 4.5 Numerical results

In this section, we present numerical results on the estimation of voltage sensitivity coefficients using the least squares methods. Because it is not practically feasible to show estimations of all sensitivity coefficients, we selected three coefficients that will be looked at in this section and later ones, too. These coefficients were selected for their large magnitude compared to most other coefficients. Their model-based computation were previously presented in Figure 3.1. As previously mentioned, it is assumed that PMUs are installed at all buses of the network.

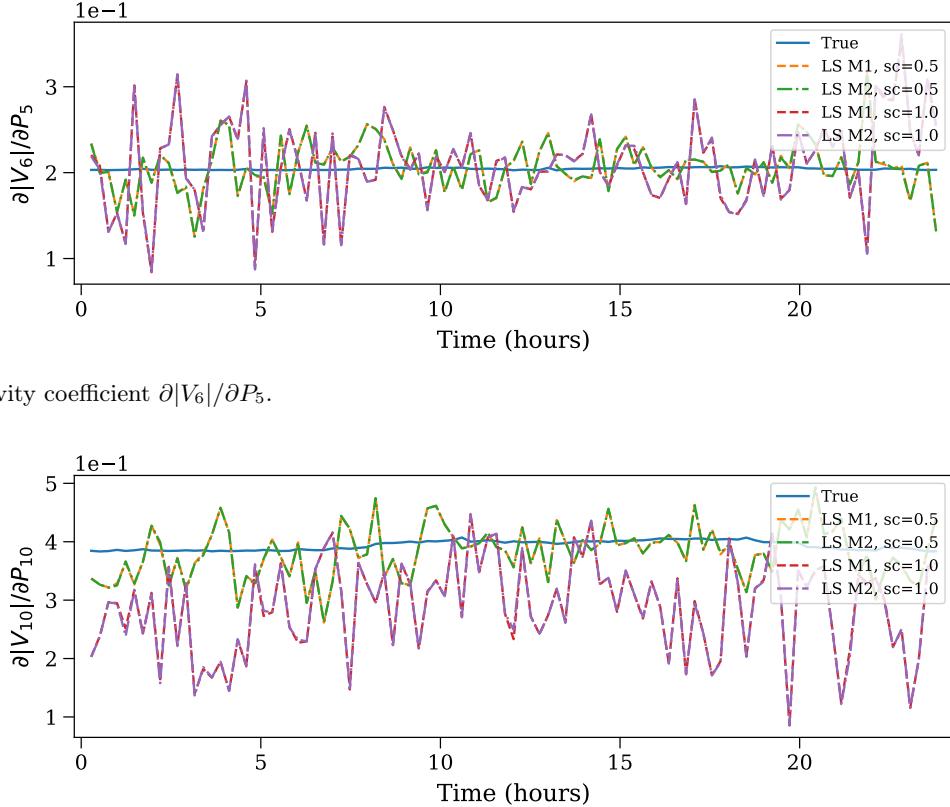
Figure 4.1 shows the estimated sensitivity coefficients using least squares (LS) Model 1 and Model 2, in the case where the PMUs are of the measurement error sensor class (sc) 0.5 or 1.0 (see Table 2.1). The coefficients were estimated using time windows of size  $|\tau| = 1000$  and after first filtering the measurements using the technique outlined in Section 4.4.1. Different time window sizes  $|\tau|$  were experimented with and 1000 was kept as the best compromise between accuracy (time windows  $|\tau| < 300$  led to poor estimation performance and  $|\tau| > 1500$  to negligible improvements) and computational requirements (larger values of  $|\tau|$  led to exponentially-growing matrix-inversion times).

Figure 4.2 then presents the distribution of the normalised estimation errors (in %) resulting from the estimation of those same sensitivity coefficients. The normalised errors are computed as:

$$100 \times \left| \frac{S_{ij} - \hat{S}_{ij}}{S_{ij}} \right|, \quad (4.14)$$

where  $S_{ij}$  and  $\hat{S}_{ij}$  refer to the true (computed using the model-based approach of Section 3.4.2) and estimated coefficients, respectively.

From Figures 4.1 and 4.2, we can observe that Model 1 and Model 2 lead to almost identical sensitivity coefficient estimations. Indeed, estimation curves in Figure 4.1 are overlapping and the error distributions of Figure 4.2 are highly similar. Although this is true for measurements that contain a significant noise

(a) Sensitivity coefficient  $\partial|V_6|/\partial P_5$ .(b) Sensitivity coefficient  $\partial|V_{10}|/\partial P_{10}$ .

**Figure 4.1:** Least squares estimation of voltage sensitivity coefficients using Model 1 (M1) and Model 2 (M2) for measurement error sensor classes (sc) 0.5 and 1.0, plotted against the true value computed using the model-based method of Section 3.4.2.

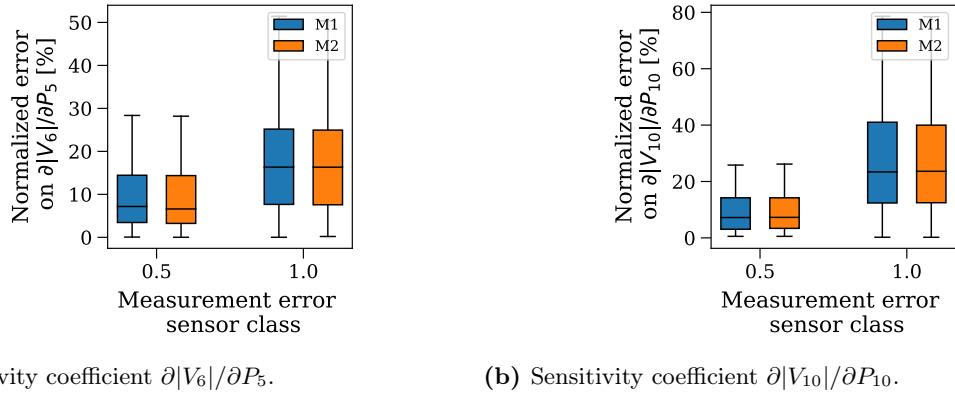
component (e.g., classes 0.5 and 1.0), we have observed better performance of Model 1 when measurement noise is close to zero. This is due to the fact that Model 2 is characterised by a larger estimation variance, as later discussed in Section 4.6.

Another conclusion that can be drawn from Figures 4.1 and 4.2 is that, as expected, both least squares estimation methods lead to worse estimations when the amount of measurement error is increased. This is particularly evident in the case of coefficient  $\partial|V_{10}|/\partial P_{10}$ , where the median normalised error jumps from 7% for sensor class 0.5 to 24% for sensor class 1.0 (see Figure 4.2).

As previously discussed, the end goal is to approximate voltage magnitude deviations  $\Delta|V_i|$  through sensitivity coefficients as:

$$\Delta|V_i| \approx \mathbf{K}_{\mathbf{P}_i} \Delta \tilde{\mathbf{P}} + \mathbf{K}_{\mathbf{Q}_i} \Delta \tilde{\mathbf{Q}}. \quad (4.15)$$

Figure 4.3 compares the true deviation  $\Delta|V_8|$  from the original noiseless load flow simulation against the



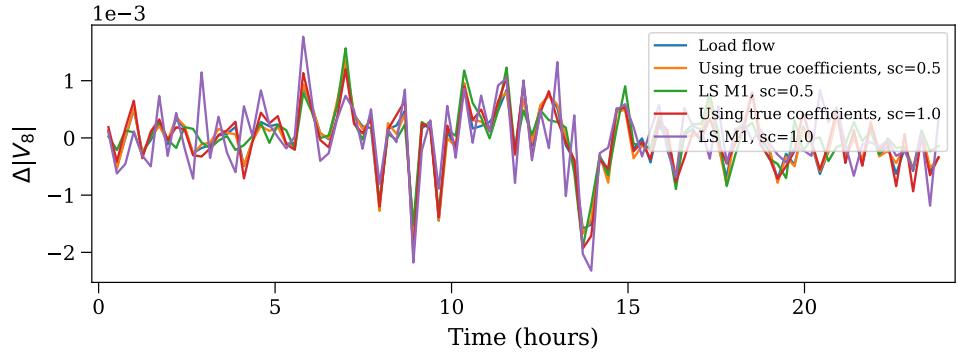
**Figure 4.2:** Normalised error (given by (4.14) in %) of the estimation of voltage magnitude sensitivity coefficients using Model 1 (M1) and Model 2 (M2) for measurement error sensor classes 0.5 and 1.0.

linear approximations (4.15) with sensitivity coefficients obtained using either the model-based approach of Section 3.4.2 (true coefficients) or the least squares Model 1 method. Results are presented for both measurement error sensor classes 0.5 and 1.0, and results obtained through Model 2 are not shown due to their high similarity with those of Model 1.

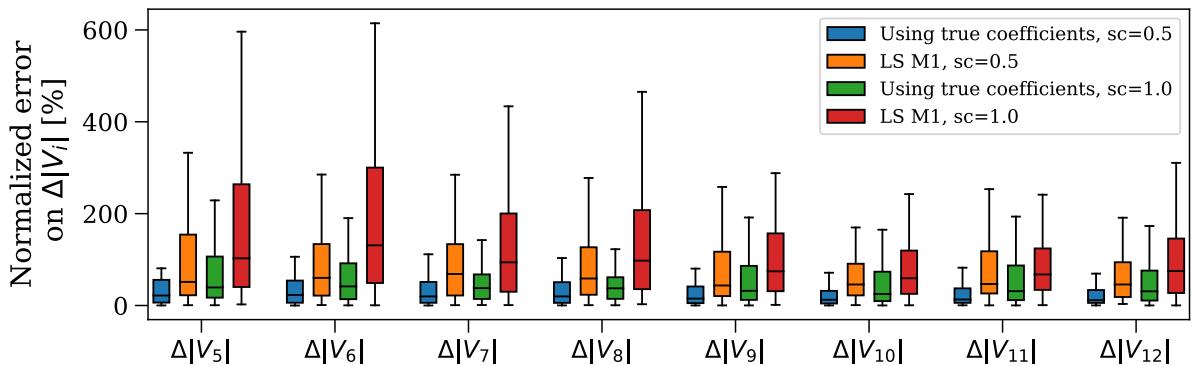
Apart from the visually apparent errors in the approximated  $\Delta|V_8|$  when Model 1 is used with sensor class 1.0, it is hard to visually assess the quality of the estimations from the other sensitivity coefficients. Figure 4.4 is provided to that end. The distribution of the normalised errors (in %, using expression (4.14) with  $\Delta|V_i|$  instead of  $S_{ij}$ ) is presented for  $\Delta|V_i|$  at eight different buses. As expected, the approximations that use sensitivity coefficients estimated through the LS method are less accurate than those that use the true sensitivity coefficients (blue vs. orange and green vs. red).

In addition, as previously hypothesised from Figure 4.3, the  $\Delta|V_i|$  estimation through least squares sensitivity coefficients is significantly less accurate in the case of sensor class 1.0 (red) compared to the other three cases, for nodes  $\{5, 6, 7, 8\}$  in particular. Although this was expected, it must be noted that the gap between the errors when using the true versus the LS sensitivity coefficients is larger in the case of sensor class 1.0 (green vs. red) than in the case of sensor class 0.5 (blue vs. orange). This observation highlights the fact that the LS estimation methods presented in this chapter are highly sensitive to measurement noise.

Finally, the last observation to be made about Figure 4.4 is that the accuracy of the linear approximations of  $\Delta|V_i|$  using (4.15) remain largely dependent on the noise present in the power injection measurements  $\Delta\tilde{\mathbf{P}}$  and  $\Delta\tilde{\mathbf{Q}}$  available from the PMUs. Even if we have access to the sensitivity coefficients computed using model-based methods, there remains a non-negligible error in the estimated  $\Delta|V_i|$  (blue and green). For instance, the median of the errors on  $\Delta|V_6|$  is as high as 68% and 117% for measurement sensor classes 0.5 and 1.0, respectively.



**Figure 4.3:** Linear approximation of  $\Delta|V_8|$  using the true (model-based) or least squares (Model 1) sensitivity coefficients, for measurement error sensor classes 0.5 and 1.0.



**Figure 4.4:** Distribution of the normalised error  $100 \times \left| \frac{\Delta|V_i| - \Delta|\hat{V}_i|}{\Delta|V_i|} \right|$  (in %) between the true deviation  $\Delta|V_i|$  obtained from the noiseless load flow and linear estimations  $\Delta|\hat{V}_i|$  obtained using sensitivity coefficients. For both measurement error sensor classes 0.5 and 1.0, the errors are presented for the estimations that use the true sensitivity coefficients (model-based computation) and those that use the sensitivity coefficients estimated using the least squares Model 1 method.

## 4.6 Cramer-Rao lower bound (CRLB)

From an estimation theory standpoint, the goal of this project can be summarised as finding the best estimator  $\hat{\mathbf{S}}_i$  of the true sensitivity coefficient vector  $\mathbf{S}_i$ . In estimation theory, the best estimator is usually considered to be the *minimum variance unbiased* (MVU) estimator. That is, the estimator with a zero bias and the minimum estimation variance among all possible estimators. Unbiasedness means that the estimator should yield the true value on average, i.e., that:

$$\mathbb{E}[\hat{\mathbf{S}}_i] = \mathbf{S}_i, \quad (4.16)$$

where the expectation  $\mathbb{E}$  is taken over multiple random noise realisations. The MVU estimators of our problem can therefore be expressed as:

$$\arg \min_{\hat{\mathbf{S}}_i} \quad \text{Var}(\hat{\mathbf{S}}_i) \quad (4.17a)$$

$$\text{subject to} \quad \mathbb{E}[\hat{\mathbf{S}}_i] = \mathbf{S}_i. \quad (4.17b)$$

### 4.6.1 Lower bound derivation

According to the Gauss-Markov theorem [58], given a data distribution that follows exactly the linear model:

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\omega}, \quad (4.18)$$

where  $\boldsymbol{\omega}$  is a vector of zero-mean Gaussian noise with covariance matrix  $\mathbf{C}$ , i.e.,  $\boldsymbol{\omega} \sim \mathcal{N}(0, \mathbf{C})$ , the MVU estimator of  $\boldsymbol{\theta}$  is the generalised least squares solution:

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \mathbf{x}. \quad (4.19)$$

In addition, the minimum variance of the  $i^{\text{th}}$  component of  $\hat{\boldsymbol{\theta}}$ ,  $\hat{\theta}_i$ , is given by the Cramer-Rao lower bound associated with the unbiased least squares estimator [58]:

$$\text{CRLB}(\hat{\theta}_i) = [(\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1}]_{ii}. \quad (4.20)$$

In the linear models introduced in Sections 4.2 (Model 1) and 4.3 (Model 2), we assumed that the measurement data followed the ideal distribution given by the general form (4.18) with a covariance matrix  $\mathbf{C} = \sigma^2 \boldsymbol{\Sigma}$ , where  $\sigma$  is the measurement noise standard deviation specified by the PMU manufacturer and  $\boldsymbol{\Sigma}$  is given by (4.5) for Model 1 and by  $\mathbf{I}_m$  for Model 2. For each estimation (i.e., time window  $\tau$ ), The Cramer-Rao lower bound of Model 1 is therefore given by:

$$\text{CRLB}(\hat{\theta}_i^{M1}) = \sigma^2 [(\boldsymbol{\Delta} \mathbf{u}_\tau^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Delta} \mathbf{u}_\tau)^{-1}]_{ii} \quad (4.21)$$

and that of Model 2 is given by:

$$CRLB(\hat{\theta}_i^{M2}) = \sigma^2 [(\mathbf{v}_\tau^T \mathbf{v}_\tau)^{-1}]_{ii}. \quad (4.22)$$

#### 4.6.2 Estimation variance: numerical results

One way to further assess the performance of the least squares estimation methods described in this chapter is to compare their respective Cramer-Rao lower bounds and empirical estimation variances. The estimation variance associated with the estimation of a single sensitivity coefficient  $S_{ij}$  is computed as follows:

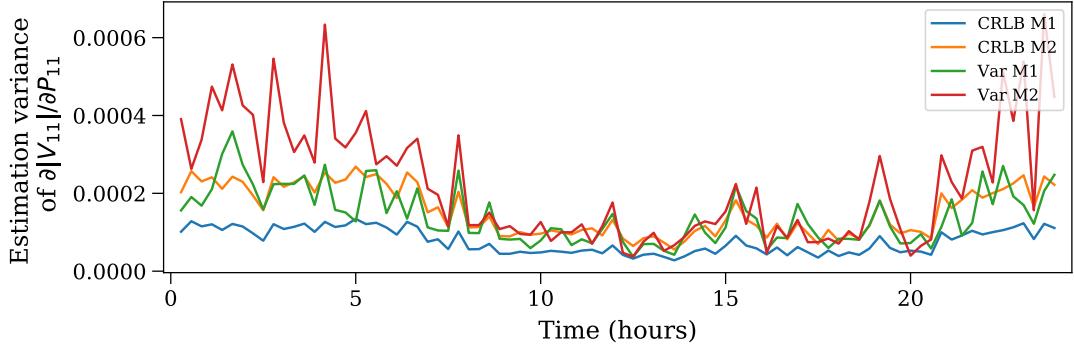
1. Generate  $N_{exp}$  noisy phasor measurement simulations by adding  $N_{exp}$  different random noise realisations (as described in Section 2.3.2) to the original load flow time series.
2. Estimate the sensitivity coefficient of interest using the different least squares methods on each of the  $N_{exp}$  experiments, resulting in  $N_{exp}$  independent estimations.
3. Approximate the estimation variance of each method for that particular sensitivity coefficient  $S_{ij}$  as the empirical variance observed over the  $N_{exp}$  estimations, defined as:

$$Var(\hat{S}_{ij}) = \frac{1}{N_{exp}} \sum_{n=1}^{N_{exp}} \left| \hat{S}_{ij} - \bar{S}_{ij} \right|^2, \quad (4.23)$$

where  $\bar{S}_{ij} = \frac{1}{N_{exp}} \sum_{n=1}^{N_{exp}} \hat{S}_{ij}$  is the mean of the estimations.

Figure 4.5 shows the CRLB and the empirical variance of the estimation of sensitivity coefficient  $\partial|V_{11}|/\partial P_{11}$  for the CIGRE-13 network, obtained from  $N_{exp} = 30$  experiments. Each time series contains 100 data points, each corresponding to the CRLB/variance of the estimation at different times of the day and for measurement error sensor class 0.5. Table 4.1 then reports the average value of each of the curves in Figure 4.5 for 6 different sensitivity coefficients. The estimation variances are empirically approximated through 30 experiments, i.e., 30 random noise realisations.

There are a number of conclusions that can be drawn from Figure 4.5 and Table 4.1. First, we observe that Model 1 consistently has a lower CRLB than Model 2. This gives a clear theoretical advantage to Model 1. Secondly, the actual estimation variance of Model 1 is regularly smaller than the CRLB of Model 2. This is an additional reason to favour Model 1 over Model 2, since Model 2 would not be able to achieve lower estimation variance than its CRLB. Finally, we note that both the CRLB and estimation variances are lower during the middle of the day. This is consistent with the quality of the estimated sensitivity coefficients from Figure 4.1 and is due to the fact that there is more variation in the measurements during those hours, which results in less numerical instability.



**Figure 4.5:** A comparison between the estimation variance of the least squares (LS) Model 1 and Model 2 (M2) methods and their respective Cramer-Rao lower bounds for the measurement error sensor class 0.5 (i.e.,  $\sigma = 1.667 \times 10^{-3}$ ). The estimation variances are approximated from 30 random noise realisations.

Sensitivity Coefficient	CRLB(M1)	Var(M1)	CRLB(M2)	Var(M2)
$\partial V_{11} /\partial P_4$	7.7e-5	1.4e-4	1.5e-4	2.2e-4
$\partial V_{11} /\partial P_8$	7.9e-5	1.4e-4	1.6e-4	2.3e-4
$\partial V_{11} /\partial P_{11}$	7.7e-5	1.4e-4	1.5e-4	2.2e-4
$\partial V_{11} /\partial Q_4$	4.9e-4	9.1e-4	7.4e-4	9.9e-4
$\partial V_{11} /\partial Q_8$	3.3e-4	6.3e-4	6.6e-4	9.1e-4
$\partial V_{11} /\partial Q_{11}$	4.6e-4	8.2e-4	9.2e-4	1.5e-3

**Table 4.1:** Average Cramer-Rao lower bound and estimation variance of the least squares Model 1 (M1) and Model 2 (M2) estimators for 6 sensitivity coefficients of  $|V_{11}|$ . Each value reported is the average over 100 estimations (time windows). For example, values reported for  $\partial|V_{11}|/\partial P_{11}$  correspond to the average value of each curve in Figure 4.5.

## 4.7 Summary

In this chapter, we introduced least squares-based sensitivity coefficient estimation methods and considered two variants, Model 1 and Model 2, which differ in terms of how they express the system of linear equations to be solved. We then discussed scenarios in which these methods suffer from numerical stability issues and proposed two solutions: (a) a measurement pre-filtering step and (b) a Principal Component Regression approach. Results of numerical experiments were next presented on the CIGRE-13 benchmark network and we concluded that (a) Model 1 and Model 2 yield highly similar estimations and that (b) both models are highly sensitive to the magnitude of the measurement errors. Finally, we compared the estimation variance associated with both models against each other and against their respective Cramer-Rao lower bound. Through this comparison, we observed that Model 1 presented a lower estimation variance, both in terms of CRLB and empirical variance.

In the next chapter, we will introduce the other model-less estimation approach considered in this work, which uses neural networks to predict sensitivity coefficients. We will then empirically compare their performance against the least squares methods from this chapter.

## Chapter 5

# Neural Network-based Sensitivity Coefficient Estimation

### 5.1 Introduction

In this chapter, we now introduce two additional measurement-based procedures that use neural networks. Both methods were designed over the course of this project and are investigated in Sections 5.2 and 5.3, respectively. Both sections follow the same structure, each with three subsections. The first one provides a brief overview of the type of neural network used in each method: feedforward neural networks (Section 5.2.1) and Long-Short Term Memory neural networks (Section 5.3.1). The second subsection then introduces the estimation model designed in the context of this work. Finally, the last subsection presents numerical comparisons with the previously discussed estimation methods on the CIGRE-13 benchmark.

### 5.2 Feedforward neural network (FNN) model

#### 5.2.1 Feedforward neural networks

For the reader unfamiliar with feedforward neural networks (FNNs), this section provides a short overview in the context of this work.

##### Multilayer perceptrons

Also referred to as *multilayer perceptrons*, FNNs are function approximators that aim to learn a mapping between an input  $\mathbf{x}$  and an output  $\mathbf{y}$ . For example, if the input-output relationship in the data can be explained by a function  $f^*$ , such that  $\mathbf{y} = f^*(\mathbf{x})$ , the objective of a FNN would be to estimate  $f^*$  with a parametric function  $f(\mathbf{x}; \boldsymbol{\theta})$ . The process of finding the parameters  $\boldsymbol{\theta}$  that result in the best function approximation is referred to as *training* the neural network [59].

Feedforward neural networks are described by a series of so-called *layers*, each consisting of a number of cells, or neurons. Each layer takes as input the output from the previous layer, applies a function to that input, and passes the output to the next layer. Eventually, the output of the last layer, known as the *output layer*, is presented as the output of the neural network. Similarly, the first layer of the network is known as the *input layer*. Any layer in between the input and output layers are referred to as a *hidden layer*.

Each layer  $l$  is characterised by a weight matrix  $\mathbf{W}_l$  and a bias vector  $\mathbf{b}_l$ , which together represent the neural network parameters  $\boldsymbol{\theta}$ . Each layer may also have a nonlinear function  $f_l$  associated with it, known as the *activation function* of  $l$ . The output of an arbitrary layer  $l$  is then given by:

$$\mathbf{y}_l = f_l(\mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l), \quad (5.1)$$

where  $f_l$  is applied in an elementwise manner,  $\mathbf{x}_l \in \mathbb{R}^n$  is the input vector,  $\mathbf{W}_l \in \mathbb{R}^{m \times n}$  is the weight matrix,  $\mathbf{b}_l \in \mathbb{R}^m$  is the bias vector, and  $\mathbf{y}_l \in \mathbb{R}^m$  is the output vector. Unless  $l$  is the output layer,  $\mathbf{y}_l$  becomes the input to the next layer  $\mathbf{x}_{l+1}$ . The term *feedforward* comes from the fact that data is always propagated through the network from the input layer to the output layer, i.e., there is no feedback loop. For a network with three layers, the final function approximation can therefore be expressed as:

$$f(\mathbf{x}; \boldsymbol{\theta}) = f_3(\mathbf{W}_3 f_2(\mathbf{W}_2 f_1(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3). \quad (5.2)$$

According to the universal approximation theorem, any function can be represented by a sufficiently large FNN [59]. Note, however, that this does not mean that it will be straightforward, or even possible, to find the corresponding optimal set of weights  $\boldsymbol{\theta}$ . There are numerous reasons for which a neural network may not be able to find the optimal  $\boldsymbol{\theta}$  which include, among other reasons, (a) the training procedure, (b) the amount and quality of the data available, and (c) the initial assumptions made.

### Training feedforward neural networks in supervised learning

In supervised learning, one of the three main paradigms of Machine Learning among unsupervised learning and reinforcement learning, the task is to learn a function that maps each input  $\mathbf{x}$  to an output  $\mathbf{y}$ , given a dataset of  $D$  input-output pairs  $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^D$ . With the recent success of deep learning, FNNs have become one of the most widely used supervised learning algorithms. The training of FNN for supervised learning tasks can be summarised as follows:

1. Define a loss function  $L(\mathbf{y}, \hat{\mathbf{y}})$  to measure the error between the output predicted by the neural network  $\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta})$  and the true target output  $\mathbf{y}$ .
2. Sample an example pair  $(\mathbf{x}_i, \mathbf{y}_i)$  from the dataset and pass it through the neural network to obtain the output  $\hat{\mathbf{y}}_i = f(\mathbf{x}_i)$ .
3. Compute the loss  $L(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ .
4. Update the parameters of the neural network based on the computed loss.

5. Repeat steps 2-4 until convergence.

The process outlined above is usually carried out in terms of *batches* and *epochs*. A batch of training samples  $\mathcal{B}$  is a uniformly sampled subset of  $\mathcal{D}$  that gets fed into the neural network (steps 2-3) before updating the parameters (step 4). Updating the parameters every  $|\mathcal{B}|$  steps has been shown to drastically reduce training time and lead to better convergence [59]. An epoch, on the other hand, refers to a complete pass through the dataset. In other words, if a FNN is trained for 10 epochs using a dataset of  $D$  input-output pairs and a batch size of  $B$ , the neural network parameters will be updated  $10 \times \lceil D/B \rceil$  times. The neural network parameters are usually updated through the *backpropagation* algorithm, a gradient-based optimisation procedure for which a wide range of software is available [60].

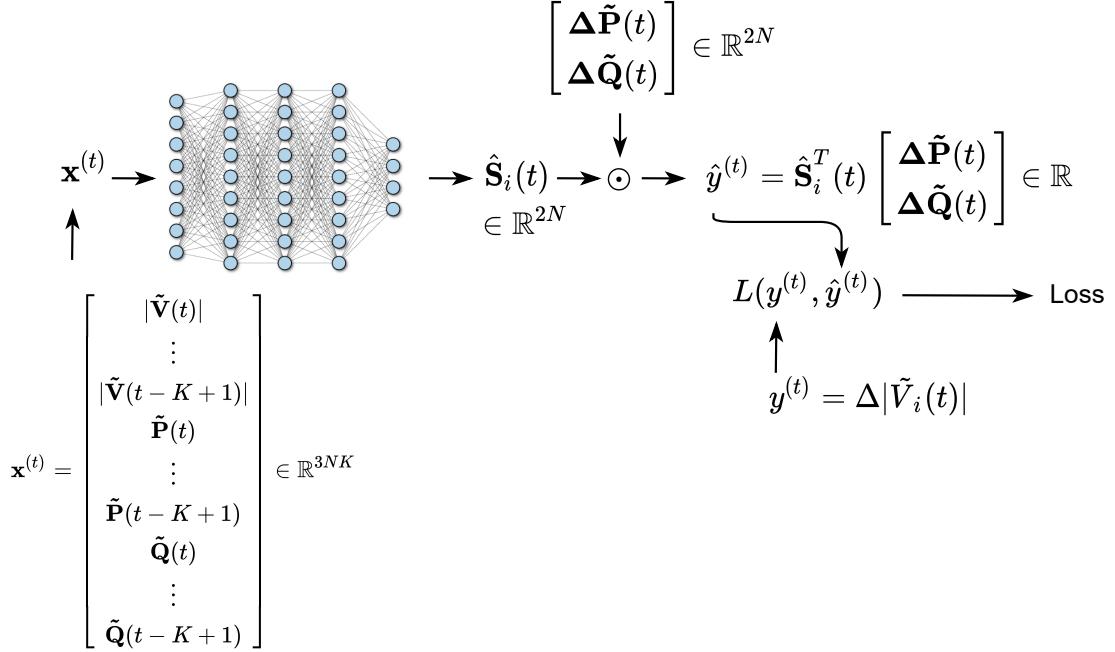
The final objective in training a neural network is to ensure that it is able to perform well on previously unseen data points, i.e., that is *generalises* well. To that end, the available dataset is often divided into a training set and a validation set. The network is trained on the training set and its performance is regularly evaluated on the held-out validation set. The assumption here is that the validation accuracy is a good surrogate for the generalisation accuracy. Often, the training procedure is adapted based on the resulting validation accuracy in order to minimise *overfitting*, hence ensuring a good generalisation performance [61].

### 5.2.2 FNN-based sensitivity coefficient estimation

Ideally, we would like to formulate the problem of estimating sensitivity coefficients as a supervised learning task. The idea would be that, given a set of measurements as input, the neural network would learn to approximate the vector of sensitivity coefficients. The challenge here, however, is that we do not have access to the true sensitivity coefficients in practice during the training phase. If we did, there would be no point in designing model-less estimation methods. To refer back to the previous section, we do not have access to the target outputs  $\mathbf{y}_i$  to form the training dataset  $\mathcal{D}$ . As a result, we cannot simply update the neural network parameters based on the error in sensitivity coefficient estimation.

Instead, we designed a surrogate loss function that only requires access to phasor measurements and does not rely on the availability of the true sensitivity coefficients during training. The final training procedure is presented in Figure 5.1 and described in the remainder of this section.

Let  $\mathbf{x}^{(t)} \in \mathbb{R}^{3KN}$  be the column vector that gathers the history of the last  $K$  PMU measurements of both independent and dependent variables. In the case of voltage sensitivities with respect to power



**Figure 5.1:** Overview of the neural network training procedure.

injections,  $\mathbf{x}^{(t)} \in \mathbb{R}^{3NK}$  is defined as:

$$\mathbf{x}^{(t)} = \begin{bmatrix} |\tilde{\mathbf{V}}(t)| \\ \vdots \\ |\tilde{\mathbf{V}}(t-K+1)| \\ \tilde{\mathbf{P}}(t) \\ \vdots \\ \tilde{\mathbf{P}}(t-K+1) \\ \tilde{\mathbf{Q}}(t) \\ \vdots \\ \tilde{\mathbf{Q}}(t-K+1) \end{bmatrix}. \quad (5.3)$$

Then let  $\hat{\mathbf{S}}_i(t) \in \mathbb{R}^{2N}$  be the neural network output when fed  $\mathbf{x}^{(t)}$  as input for some node  $i$ . Each neural network is thus trained to estimate the vector of sensitivity coefficients corresponding to a particular node  $i$  and a total of  $N$  neural networks should be used to estimate the full sensitivity coefficient matrix  $\mathbf{S}$ .

Concretely, the vector  $\mathbf{x}^{(t)}$  regroups all measurements obtained during the last  $K$  timesteps  $\{t-K+1, \dots, t\}$  up to  $t$  and can be thought of as a parameter analogous to the length of the time windows  $|\tau|$  used in the least squares approach of Chapter 4. This parameter  $K$  must be chosen large enough so as

to provide the neural network with enough information to make accurate predictions, but not too large as high-dimensional input vectors may lead to poor training performance.

Our objective is to train the neural network to output vectors  $\hat{\mathbf{S}}_i(t)$  that approximate the true sensitivity coefficient vector  $\mathbf{S}_i(t)$  at time  $t$ . Because we do not have access to the true  $\mathbf{S}_i(t)$ , however, we instead estimate the quality of  $\hat{\mathbf{S}}_i(t)$  by computing the error in the predicted linear approximation of  $\Delta|V_i(t)|$ , denoted  $\hat{y}^{(t)}$  and defined as:

$$\hat{y}^{(t)} = \hat{\mathbf{S}}_i(t)^T \begin{bmatrix} \Delta\tilde{\mathbf{P}}(t) \\ \Delta\tilde{\mathbf{Q}}(t) \end{bmatrix}. \quad (5.4)$$

Expression (5.4) thus represents the voltage magnitude deviation that we would expect to observe at bus  $i$  given the vector of sensitivity coefficients estimated by the neural network  $\hat{\mathbf{S}}_i(t)$ .

Contrarily to the vector of true sensitivity coefficients, we do have access to the PMU  $\Delta|\tilde{V}_i(t)|$  measurements and we can use them as target outputs in the neural network training procedure. Note, however, that these measurements contain random PMU measurement errors which can impact the quality of the mapping learned by the FNN.

The final step in the training procedure is to compute the loss associated with a particular input-output pair. We used the mean squared error (MSE) loss function in all our experiments, which is given by:

$$L(y^{(t)}, \hat{y}^{(t)}) = MSE(y^{(t)}, \hat{y}^{(t)}) = (y^{(t)} - \hat{y}^{(t)})^2, \quad (5.5)$$

and returns a scalar value that can be used by the FNN parameter update procedure (e.g., gradient descent).

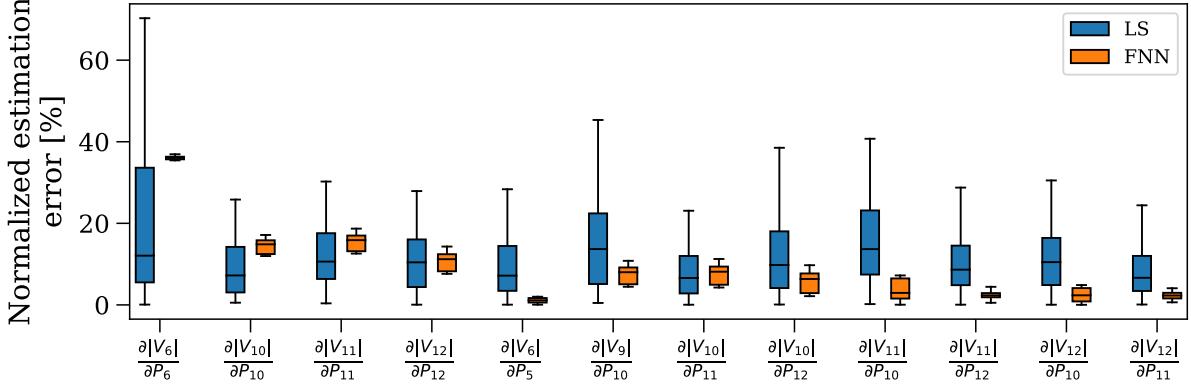
### 5.2.3 Numerical results

In this section, we compare the performance of the FNN method against the least squares approach. For the latter, we only report results for Model 1, since Model 2 has previously been shown to yield similar sensitivity coefficient estimations (Section 4.5) and is also characterised by a higher estimation variance (Section 4.6.2).

#### Training procedure

All neural networks were trained using a second 24-hour load flow simulation on CIGRE-13, similar to the one presented in Section 2.3.2. This was done to simulate a real-world scenario in which we would have to train the neural networks on historical data from previous days. The 24 hours of data were divided into a training set (12 hours) and a validation set (12 hours).

After each epoch of training, the validation loss was computed as the average loss on the validation set. Every time a new lowest validation loss was encountered, the parameters of the neural network were saved and after 20 epochs, the best model was kept. This procedure was repeated for different



**Figure 5.2:** Normalised error (given by (4.14) in %) of the estimation of a number of voltage sensitivity coefficients using the least squares (Model 1) and FNN approaches for measurement error sensor class 0.5.

hyperparameters  $K$  in the range [10, 600] and the value which led to the best validation accuracy was kept. A total of  $N$  FNNs were trained this way, one for each non-slack bus  $i$ .

The neural network architecture was fixed to two hidden layers of size 128 and 64 units, respectively, and all layers except the output one were followed by a ReLU nonlinear activation function. During training, the Adam optimiser [62] was used with a learning rate of  $1 \times 10^{-3}$ .

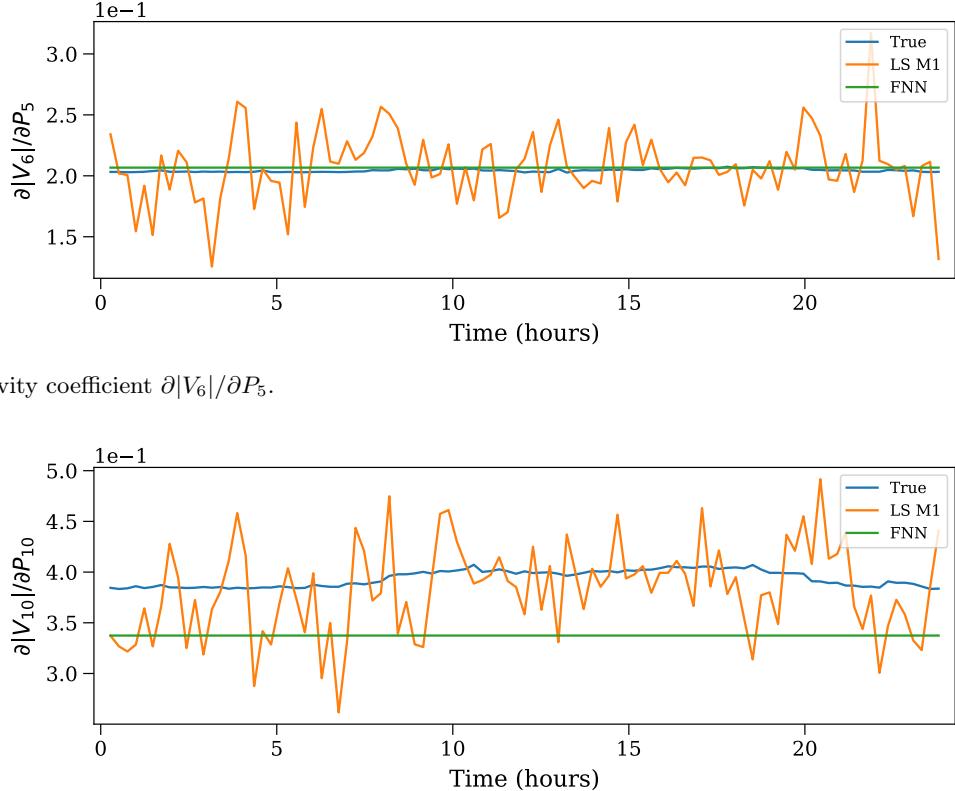
## Results

Figure 5.2 shows the distribution of the normalised estimation errors (in %) between true (model-based) sensitivity coefficients  $S_{ij}$  and their estimated values  $\hat{S}_{ij}$  from the least squares (Model 1) or the FNN methods (using (4.14)). These sensitivity coefficients were chosen in particular because they have non-negligible values ( $\geq 0.1$ ) and the corresponding power injections are non-zero throughout the whole 24-hour simulation – too small power deviations lead to large numerical errors in the case of least squares. The results presented were obtained for measurement error sensor class 0.5 and the coefficients were estimated using time windows of size  $|\tau| = 1000$  for least squares and  $K = 50$  for the FNN model.

Overall, there is no estimation method that clearly outperforms the other. Nevertheless, there are at least two observations that can be made. The first one is that the median of the error distribution is smaller in the case of FNN for 7 out of 12 sensitivity coefficients. This observation, although not sufficient to claim that the FNN method is more suited to the task, indicates that it is at least competitive with the least squares approach.

The second observation is that the error distributions are significantly more spread for least squares than for the FNN method. As shown in Figure 5.3, which presents the estimation time series of  $\partial|V_6|/\partial P_5$  and  $\partial|V_{10}|/\partial P_{10}$ , this is a result of the fact that both the true and FNN estimated values change little over time. On the other hand, however, the least squares estimation tend to fluctuate around the true values with large jumps between subsequent estimations.

The sensitivity coefficients shown in Figure 5.3 were chosen due to the fact that they represent

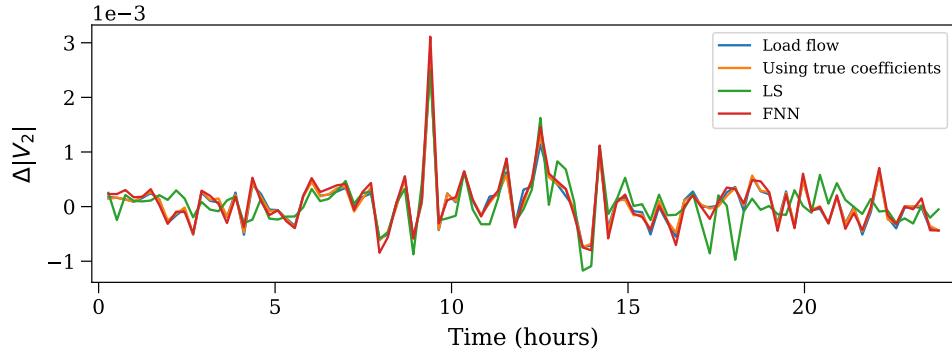
(a) Sensitivity coefficient  $\partial|V_6|/\partial P_5$ .(b) Sensitivity coefficient  $\partial|V_{10}|/\partial P_{10}$ .

**Figure 5.3:** Comparison of voltage sensitivity coefficients estimated using the least squares (Model 1) and the FNN approaches for measurement error sensor class 0.5, plotted against the true value computed using the model-based method of Section 3.4.2.

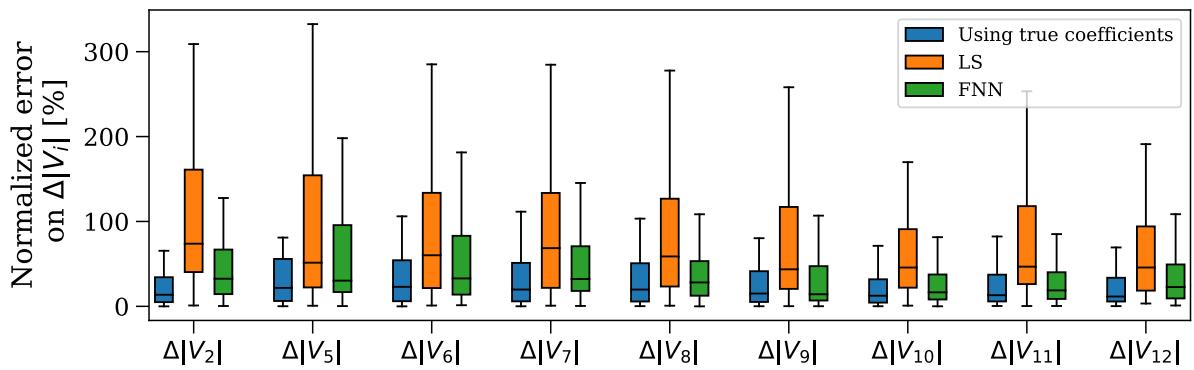
two characteristic scenarios: one in which the FNN estimations are almost identical to the true values ( $\partial|V_6|/\partial P_5$ ) and one in which least squares lead to better estimations ( $\partial|V_{10}|/\partial P_{10}$ ). This observation is consistent across other sensitivity coefficients, too. For some, the FNN estimations are highly accurate, while for some others are consistently off by a near-constant term.

Then, Figure 5.4 shows the estimated  $\Delta|V_2|$  obtained from substituting the estimated sensitivity coefficients into expression (4.15). This Figure is mainly provided for a visual confirmation that the neural networks have indeed learned to estimate vectors  $\mathbf{S}_i$  that lead to good linear approximations of  $\Delta|V_i|$ , as they were trained to do.

Finally, Figure 5.5 reports the distribution of the normalised errors (in %) of  $\Delta|V_i|$  at eight different buses for the measurement error sensor class 0.5. Here, we see that the FNN method leads to smaller errors at all buses compared to the least squares approach. In addition, the error distributions of the FNN-based estimations are significantly closer to the error distribution obtained with sensitivity coefficients computed in a model-based manner.



**Figure 5.4:** Linear approximation of  $\Delta|V_2|$  using the true (model-based), the least squares (Model 1), or the FNN approaches, for measurement error sensor class 0.5.



**Figure 5.5:** Distribution of the normalised error  $100 \times \left| \frac{\Delta|V_i| - \Delta|\hat{V}_i|}{\Delta|V_i|} \right|$  (in %) between the true deviation  $\Delta|V_i|$  obtained from the noiseless load flow and the linear estimations  $\Delta|\hat{V}_i|$  obtained using sensitivity coefficients. The errors presented are for measurement error sensor class 0.5.

Overall, we have observed that the least squares and the FNN methods often lead to the estimation of individual sensitivity coefficients with similar accuracy, each performing better on some coefficients but not on others. Nevertheless, Figure 5.5 suggests that the neural network approach would be more suited to the task of estimating the linearised voltage deviations  $\Delta|V_i|$ .

### 5.3 Long-Short Term Memory (LSTM) neural network model

When modelling time-dependent processes, feedforward neural networks are not the best [63, 64]. They lack the capability to represent time dependency in the available data; there is no memory component or feedback loop. Every estimation is independent of both previous and future estimations. Recurrent neural networks (RNNs), on the other hand, have that capability built in. Similarly to hidden Markov models, recurrent neural networks keep an internal hidden state, which gets updated after each estimation. In other words, the function approximator is, in theory, able to remember past inputs and estimations. Motivated by this observation, we decided to adapt the FNN method established in Section 5.2 to RNNs, as detailed in this section.

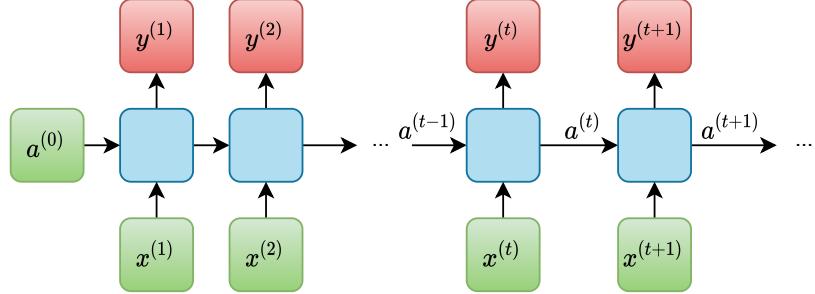
Because Long-Short Term Memory (LSTM) networks, a type of RNN, have shown to perform extremely well at a wide range of tasks [65], the method introduced here is described for LSTMs. Note, however, that the framework is general and that any other type of RNN could be used instead.

#### 5.3.1 LSTM neural networks

This section provides a short overview of LSTM neural networks for the unfamiliar reader.

Recurrent neural networks are a type of neural networks that allow passing output values at a given timestep to the input of the network at the next timestep, while also keeping an internal hidden state [66]. The fact that previous outputs can be used as later inputs provides a big advantage over feedforward neural networks, as it allows RNNs to process input sequences of any length. This is the reason for which RNNs have been mostly used to deal with tasks in which the size of the input changes over time (e.g., speech or written sentences). The fact that it keeps an internal hidden state that gets updated at each timestep also makes it ideal for tasks that require contextual information or a memory component. Figure 5.6 shows the structure of a typical RNN, where green and red elements represent inputs and outputs over multiple timesteps, respectively, and  $a^{(t)}$  is the internal hidden state produced after timestep  $t$ . As seen in this diagram, RNNs can also be thought of as a series of feedforward neural networks, one used for each timestep, in which all FNNs share the same parameters.

There is one main problem often encountered with traditional RNNs, however, and that is the so-called *vanishing gradient* problem. The vanishing gradient problem is related to the fact that the influence of a given input on the hidden state, and therefore on the final output, either decays or blows up exponentially after only a few discrete timesteps (i.e., in the range 5-10) [68, 69]. The consequence of such vanishing/exploding gradients is that traditional RNNs tend to fail at learning time dependencies over more than 5-10 discrete timesteps. In the context of this work, it means using a RNN would only take into account the most recent 5-10 timesteps of measurements.



**Figure 5.6:** Recurrent neural network, inspired from [67].

LSTMs were then proposed as a solution to the vanishing gradient problem, making it possible to take into account long-term dependencies in the input data [70]. LSTMs are a type of RNNs and thus also follow the functional diagram of Figure 5.6. Their particularity lies in the internal structure of each network and especially in their use of so-called *gates*. Each gate is made of a single hidden layer and an associated activation function and each LSTM contains 4 gates. First we have the *forget gate*, which impacts how much of the previous hidden state (also known as the *cell state*) will be kept in the next cell state. Then we have the *input gate*, which controls how much of the new input signal will be added to the cell state. Finally, the *output gate* will use the current cell state in order to produce the final output.

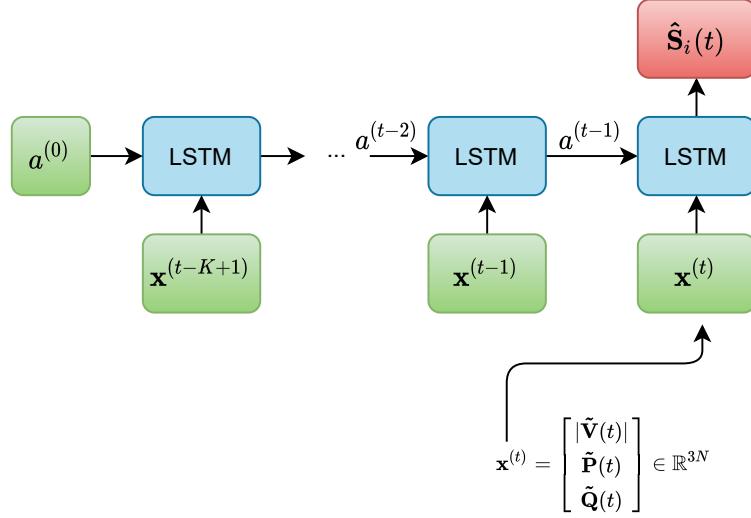
For the interested reader, a more in-depth description of LSTM neural networks can be found in [70, 71].

### 5.3.2 LSTM-based sensitivity coefficient estimation

The training procedure followed to estimate sensitivity coefficients through LSTM neural networks is similar to the procedure followed for FNNs and previously described in Section 5.2.2. In particular, the LSTM model is also trained to output an estimate  $\hat{\mathbf{S}}_i$  of the vector of sensitivity coefficients  $\mathbf{S}_i$  associated with bus  $i$ , for all buses  $i \in \mathcal{N} - \{0\}$ . Because the output of the network is the same, we are also able to use the same surrogate loss function that uses the  $\Delta|\hat{V}_i(t)|$  approximated using  $\hat{\mathbf{S}}_i$  (given by (5.4) and (5.5)).

The difference with the FNN approach presented in Figure 5.1 is the way input vectors are fed into the LSTM. This is shown in Figure 5.7. Instead of feeding vectors of size  $3NK$  that contain all measurements over the time window  $[t - K + 1, t]$ , we now feed single-timestep vectors of size  $3N$ , one at a time. The parameter  $K$  thus now represent the length of the time series fed into the LSTM before a sensitivity coefficient estimation is performed.

Concretely, in order to estimate the sensitivity coefficients at time  $t$ , we sequentially feed the LSTM



**Figure 5.7:** Overview of the LSTM-based method for estimating sensitivity coefficients.

with the last  $K$  measurement vectors, each defined as:

$$\mathbf{x}^{(l)} = \begin{bmatrix} |\tilde{\mathbf{V}}(l)| \\ \tilde{\mathbf{P}}(l) \\ \tilde{\mathbf{Q}}(l) \end{bmatrix}, \quad (5.6)$$

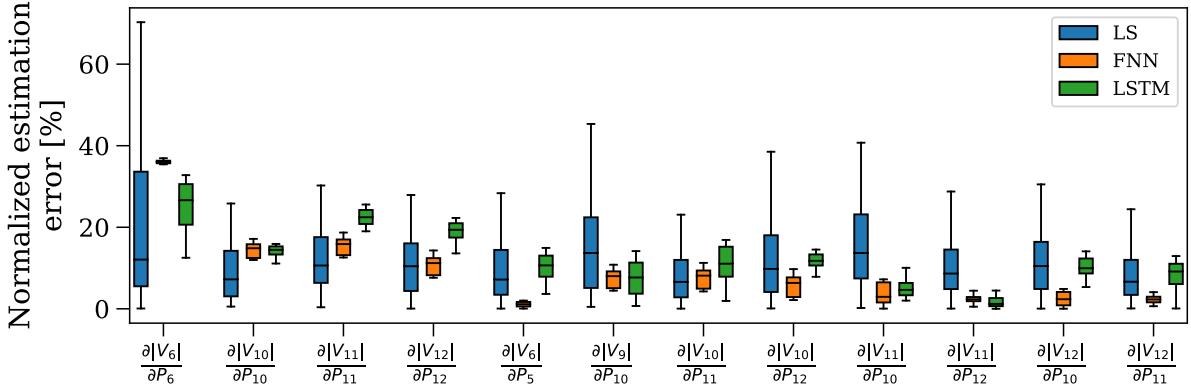
as represented in Figure 5.7. This allows the internal cell state to incrementally incorporate novel information over the time window before being used in the computation of the sensitivity coefficient vector at the last timestep  $t$ .

### 5.3.3 Numerical results

In this section, we now present a comparison of the quality of the estimated sensitivity coefficients using the LSTM approach against results previously presented for the least squares and FNN methods. The results are presented following the same structure as Numerical Results sections 4.5 and 5.2.3.

#### Training procedure

The LSTM neural network used in these experiments was trained in a similar manner to the FNN (see Section 5.2.3). We also experimented with a number of values for the hyperparameter  $K$  in  $[10, 100]$  and found that a value of  $K = 25$  led to the best performance, although performance differences were very small. For the internal hidden layer of the LSTM, which keeps track of the cell state, 100 units were used.



**Figure 5.8:** Normalised error (given by (4.14) in %) of the estimation of voltage sensitivity using the least squares (Model 1), the FNN, and the LSTM approaches for measurement error sensor class 0.5.

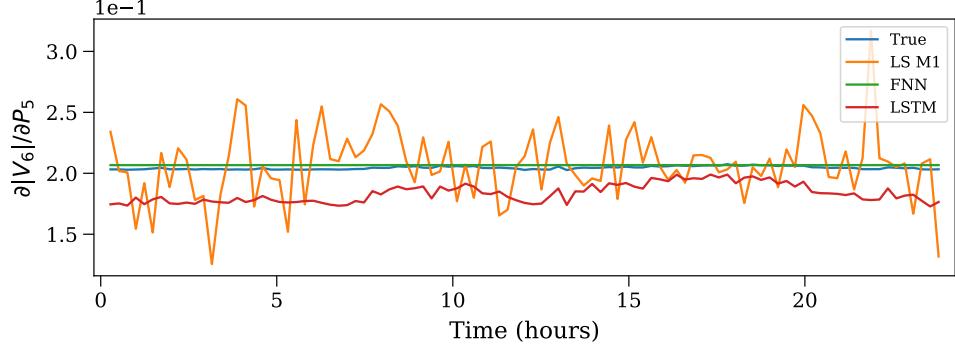
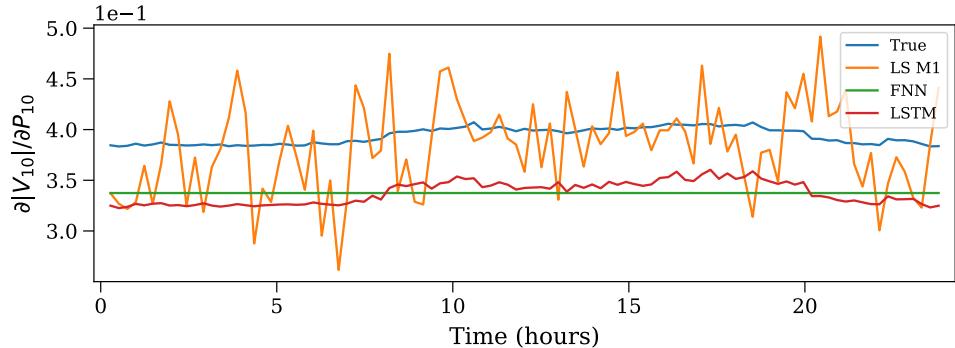
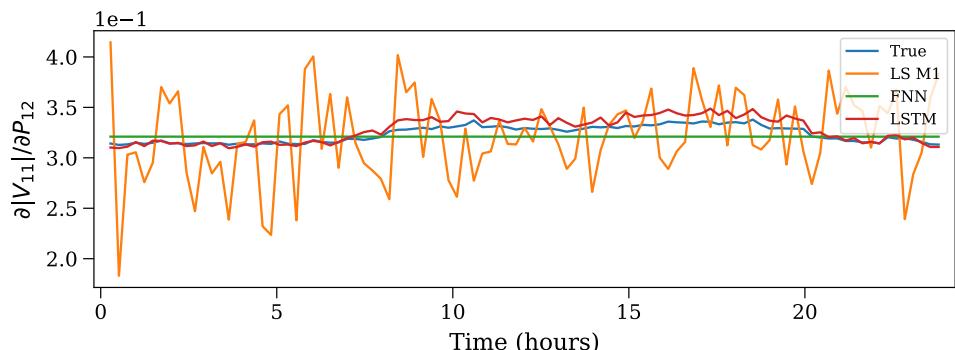
## Results

Figure 5.8 shows the distribution of the normalised errors (in %) between true (model-based) sensitivity coefficients  $S_{ij}$  and their estimated values  $\hat{S}_{ij}$  for all three measurement-based methods considered. This is the same figure as Figure 5.2, with the addition of the LSTM errors. The results were obtained for measurement error sensor class 0.5 and the coefficients were estimated using time windows of size  $|\tau| = 1000$  for least squares,  $K = 50$  for FNN, and  $K = 25$  for LSTM.

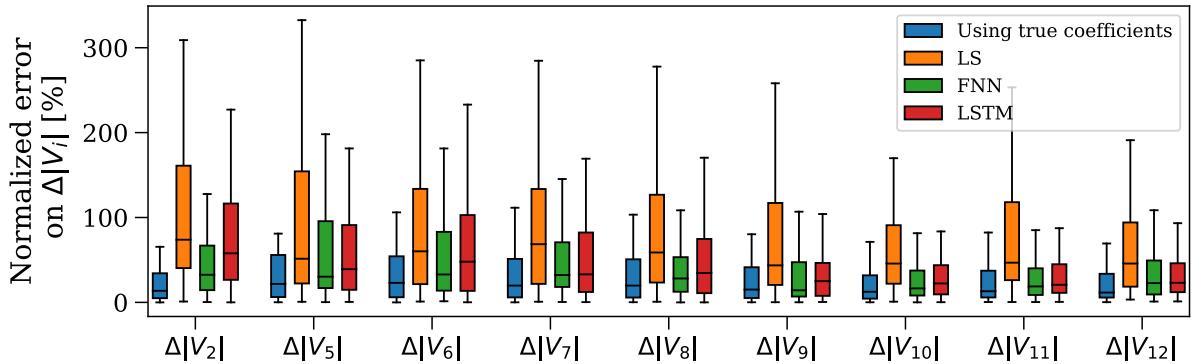
Overall, the accuracy of the estimated sensitivity coefficients using the LSTM model is similar to that of the FNN approach. In some cases, such as  $\partial|V_6|/\partial P_6$  and  $\partial|V_{11}|/\partial P_{12}$ , the LSTM errors are lower than the FNN ones. In other cases, they are worse, e.g.,  $\partial|V_{11}|/\partial P_{11}$  and  $\partial|V_{12}|/\partial P_{10}$ . In most cases, however, they estimation error distributions are highly similar, such as for  $\partial|V_{10}|/\partial P_{10}$ ,  $\partial|V_{11}|/\partial P_{10}$ , and  $\partial|V_{10}|/\partial P_{10}$ . Similar remarks can be made about the comparison with the least squares approach.

Figure 5.9 then displays three estimated sensitivity coefficients using all three methods against the true values. The first two coefficients,  $\partial|V_6|/\partial P_5$  and  $\partial|V_{10}|/\partial P_{10}$ , are the same ones as those presented in Figures 5.3 and 4.1. Again, these three coefficients were chosen to highlight cases in which each of the three estimation methods performs best.

In Figure 5.9a, we see that the FNN method remains best at estimating coefficient  $|\partial V_6|/\partial P_5$  and that the LSTM approach tends to under-estimate the sensitivity coefficient throughout the day. Figure 5.9b then represents the case in which least squares yields accurate estimations the most often compared to the other methods. In this case, however, we see that, as opposed to the FNN, the LSTM estimation follows the trend of the true values, although off by a near-constant value throughout the day. Finally, Figure 5.9c shows a case in which the LSTM approach yield the best sensitivity coefficients. Although the FNN method already produced good estimations, the LSTM approach is able to track the changes in the value of  $\partial|V_{11}|/\partial P_{12}$  throughout the day. This results is particularly interesting, as it suggests that the LSTM method would be more suited than the FNN one in the case of sensitivity coefficients that vary a lot over time. As discussed further in Section 5.4.2, this is likely to be a consequence of the higher

(a) Sensitivity coefficient  $\partial|V_6|/\partial P_5$ .(b) Sensitivity coefficient  $\partial|V_{10}|/\partial P_{10}$ .(c) Sensitivity coefficient  $\partial|V_{11}|/\partial P_{12}$ .

**Figure 5.9:** Comparison of voltage sensitivity coefficients estimated using the least squares (Model 1), the FNN, and the LSTM approaches for measurement error sensor class 0.5, plotted against the true value computed using the model-based method of Section 3.4.2.



**Figure 5.10:** Distribution of the normalised error  $100 \times \left| \frac{\Delta|V_i| - \Delta|\hat{V}_i|}{\Delta|V_i|} \right|$  (in %) between the true deviation  $\Delta|V_i|$  obtained from the noiseless load flow and the linear estimations  $\Delta|\hat{V}_i|$  obtained using sensitivity coefficients. The errors presented are for measurement error sensor class 0.5.

bias associated with the FNN estimations.

Finally, Figure 5.10 shows the normalised estimation errors on  $\Delta|V_i|$  for a number of nodes  $i$ . In most cases, the LSTM approach leads to estimation errors between those of the LS and the FNN methods. In many cases, however, such as those with  $i \in \{9, 10, 11, 12\}$ , the distribution of errors are highly similar between the FNN and LSTM methods. Note that in those cases, the error distributions of FNN and LSTM are also very close to the approximation errors obtained with the true (model-based) sensitivity coefficients.

## 5.4 Discussion

With the aim of providing a broader comparison between the different data-driven estimation approaches considered in this work, this section provides comparisons on two additional bases. Those bases are (a) the computational costs (Section 5.4.1) and (b) the empirical estimation variance (Section 5.4.2) associated with each method.

### 5.4.1 Computational considerations

#### Real-time single point estimate

The first and most important cost associated with the computation of sensitivity coefficients is the amount of time required to compute a single point estimate. This is particularly important if the coefficients are to be used in a real-time control scheme acting on a small time scale in some distribution network. With the constant addition of highly intermittent DERs, DNOs are transitioning towards smaller and smaller control time scales (e.g., 1-5 minutes). This makes any real-time computational steps required a potential bottleneck for the system.

Table 5.1 reports the mean and standard deviation of the computational time (in seconds) required to

Least squares M1 ( $ \tau  = 1000$ )	FNN ( $K = 50$ )	LSTM ( $K = 50$ )
$0.233 \pm 0.041$	$0.015 \pm 0.002$	$0.033 \pm 0.006$

**Table 5.1:** Average CPU time required to estimate the full matrix of voltage magnitude sensitivity coefficients  $\mathbf{S} \in \mathbb{R}^{N \times 2N}$  w.r.t. nodal power injections on a MacBook Pro 2.3 Ghz Dual-Core Intel Core i5 computer with 8GB of RAM (in seconds).

estimate the full voltage sensitivity coefficient matrix  $\mathbf{S} \in \mathbb{R}^{N \times 2N}$  using each of the proposed model-less methods for the CIGRE-13 network. These results were obtained on a MacBook Pro 2.3 Ghz Dual-Core Intel Core i5 computer with 8GB of RAM. In the case of least squares, it entails inverting a  $(|\tau|, |\tau|)$  matrix of measurements, where a value of  $|\tau| = 1000$  is used here to match all other results previously presented. In the case of both neural network approaches, it accounts for a single forward pass through  $N$  neural networks (one for each node  $i$ ).

The least squares approach takes, on average, 15 times longer than the FNN method and 7 times longer than the LSTM model. In addition, note that the neural network estimations for different nodes could be parallelised, potentially speeding up the estimations further. Nevertheless, the computational time of all methods considered are well below practical requirements with estimations computed in less than 0.3 seconds on average. All three methods would therefore be adequate for any real-time control scheme in terms of estimation speed.

### Required training time

Figure 5.2 then reports the average computational time required to pre-train the neural networks. This cost is obviously null for the least squares approach, since no preparation steps are required in order to employ it. On the other hand, neural networks must be trained in advance on historical data. The results presented here correspond to the time required to train the models used in previous numerical results, i.e., train for 20 epochs on a 24-hour ( $T = 86400$ ) historical simulation.

As shown in the table, the LSTM network takes significantly longer to train than the FNN network, with 55.3 and 9.1 minutes required, respectively. This is a consequence of the fact that, although the input vectors to the LSTM network are smaller ( $3N$  instead of  $3KN$  for FNN), each estimation must be preceded by  $K$  other forward passes so as to build the corresponding hidden state. This means that a single epoch corresponds to  $T$  forward passes in the case of FNN, where it requires  $KT$  forward passes for the LSTM network. It is worth noting, however, that we observed the LSTM networks converging in less epochs than the FNN networks. Whereas validation accuracy often reached a minimum value around epoch 7-8 for the LSTM, the FNN usually required a minimum of 15 epochs.

Although these training times are non-negligible, they should have no practical negative consequences since they are a one-time cost. In addition, both types of neural networks could be continuously trained with the most recent measurements available, in a close-to real-time manner. This way, any change in the network topology or parameters would be, in theory, quickly taken into consideration through a change

Least squares (M1)	FNN ( $K = 50$ )	LSTM ( $K = 50$ )
0	$9.1 \pm 1.3$	$55.3 \pm 6.1$

**Table 5.2:** Average CPU time required to pre-train  $N$  neural networks (one for each bus  $i$ ) on a MacBook Pro 2.3 Ghz Dual-Core Intel Core i5 computer with 8GB of RAM (in minutes).

in the training data distribution. This, however, is a direction that was not explored in this work.

### Impact of training dataset size

Finally, the last point of interest in terms of computational requirements is the impact of the amount of historical data available during the training phase of the neural network methods on the quality of the estimated sensitivity coefficients. To investigate this, we designed the following experiment. Starting from a 24-hour load flow simulation, we kept the final 6 hours for testing the performance of the neural networks. Then, we trained a number of neural networks with various amount of historical data available for training. The results are presented in Figure 5.11 for the CIGRE-13 network with measurement error sensor class 0.5, where the x-axis represents the number of hours of data used for training, 50% of which was used for actual training and the remainder 50% for choosing the best model via validation (same procedure as previous experiments).

The results in Figure 5.11 are presented for node  $i = 11$  and correspond to estimation errors on the held-out test dataset. In Figure 5.11a, the average estimation error associated with the sensitivity coefficient vector  $\mathbf{S}_i$ , which we define as:

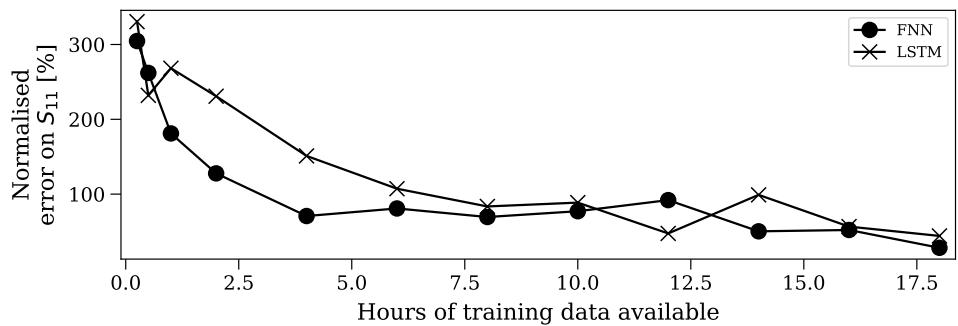
$$100 \times \frac{1}{2NT} \sum_{t=1}^T \sum_{j=1}^{2N} \left| \frac{S_{ij}(t) - \hat{S}_{ij}(t)}{S_{ij}(t)} \right|, \quad (5.7)$$

for  $i = 11$  is reported (in %). Figure 5.11b then shows the average estimation error on  $\Delta|V_i|$ , computed as:

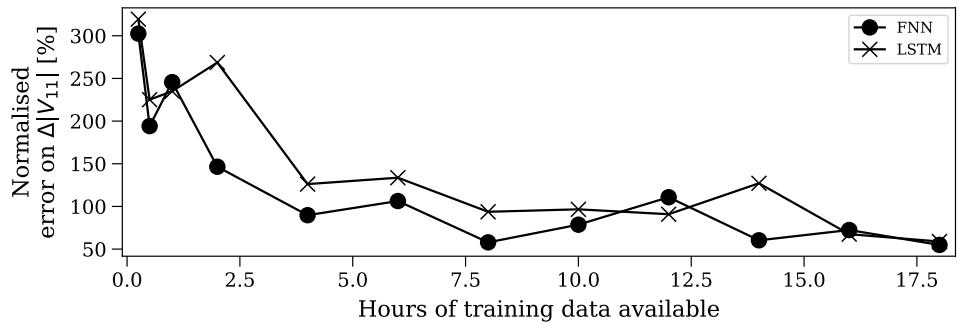
$$100 \times \frac{1}{T} \sum_{t=1}^T \left| \frac{\Delta|V_i| - \Delta|\hat{V}_i|}{\Delta|V_i|} \right|, \quad (5.8)$$

for  $i = 11$  (in %).

As would be expected, both figures suggest that an increase in the amount of historical data available for training leads to better estimations. In addition, we see that estimation errors decrease exponentially with the first hours of training data available. This would suggest that in the case of topology or network parameter change, for example, such neural networks could be trained and made ready for deployment within as little as 4 hours. This training time would likely be even smaller if the neural networks are trained continuously, as proposed in the last section, since any network model change would likely have little effect on individual sensitivity coefficients.



(a) Average normalised estimation error of voltage sensitivity coefficients of node  $i = 11$ .



(b) Average normalised error of the linear estimation of  $\Delta|V_{11}|$  using sensitivity coefficients.

**Figure 5.11:** A comparison of the quality of estimated sensitivity coefficients using the FNN and LSTM approaches against the number of hours of measurement data available for training, for measurement error sensor class 0.5.

Sensitivity coefficient	Least squares (M1)	FNN	LSTM
$\partial V_{11} /\partial P_4$	1.4e-4	3.2e-8	2.4e-6
$\partial V_{11} /\partial P_8$	1.4e-4	2.8e-7	1.6e-6
$\partial V_{11} /\partial P_{11}$	1.4e-4	3.1e-8	3.2e-7
$\partial V_{11} /\partial Q_4$	9.1e-4	2.6e-8	1.6e-7
$\partial V_{11} /\partial Q_8$	6.3e-4	6.3e-9	2.0e-7
$\partial V_{11} /\partial Q_{11}$	8.2e-4	4.2e-8	4.1e-7

**Table 5.3:** Average estimation variance for different sensitivity coefficients, approximated using 30 random noise realisations. Each value reported is the average over 100 estimations (time windows), for measurement sensor class 0.5.

#### 5.4.2 Estimation variance

When using neural networks as function approximators, one must keep in mind the associated bias-variance dilemma, which states that there will always be a tradeoff between estimation bias and variance [72]. This observation is at the core of the under/over-fitting dilemma: in general, a neural network with a low estimation bias will have a large estimation variance (overfitting), whereas one with a large estimation bias will have a low estimation variance (underfitting). The hope is that, through techniques such as validation and early-stopping, which is the approach taken in all our experiments, the bias-variance sweet spot may be achieved.

In the Cramer-Rao lower bound section of Chapter 4, we discussed the fact that the least squares estimator is the MVU estimator for the linear data model with white Gaussian noise assumed throughout this work. Although the FNN and LSTM models are not unbiased estimators, we can still compare their estimation variance against that of the least squares method, as reported in Table 5.3. The reported variances were estimated over 30 random noise realisations for the CIGRE-13 network with measurement error sensor class 0.5, following the procedure used to obtain the results from Table 4.1.

Table 5.3 clearly shows that both the FNN and LSTM models have a significantly lower estimation variance than the least squares approach, with variances in the range of  $10^{-8}$ ,  $10^{-7}$ , and  $10^{-4}$ , respectively. This is an observation in favour of the neural network-based methods, since a lower estimation variance makes the estimation more robust to random noise perturbations. Due to the bias-estimation tradeoff, however, the observed low estimation variance comes with the cost of a higher bias. This can be visually observed in Figure 5.9, for example, where the FNN approach leads to estimated sensitivity coefficients that are almost constant throughout the day.

## 5.5 Summary

In this chapter, we introduced two neural network-based model-less sensitivity coefficient estimation methods; one that uses a feedforward neural network and the other a type of recurrent neural network

to predict the values of the coefficients in real-time. Along with these models, we proposed a procedure allowing the training of these neural networks as a classical supervised learning task, even in the absence of the knowledge of the values of the true sensitivity coefficients to be estimated. This was done through the approximation of voltage deviations  $\Delta|V_i|$  using the available power injection measurements.

We then numerically showed that (a) both neural networks lead to estimation performance similar to least squares in terms of individual sensitivity coefficients and often outperformed least squares in  $\Delta|V_i|$  estimations, (b) the computational costs of all three methods are low enough for any practice usage, and (c) both FNN and LSTM methods have lower estimation variance than least squares, making them more predictable in practice.

In the next chapter, we will discuss a practical usage of sensitivity coefficients in a real-time voltage control application and compare the performance of each of the estimation methods presented in **Chapters 3, 4, and 5**

## Chapter 6

# Application: Real-Time Voltage Control

### 6.1 Introduction

In the Motivation section of **Chapter 1**, we mentioned the usefulness of sensitivity coefficients as a tool to linearise the power flow equations and, in doing so, cast potential OPF problems into convex or even linear optimisation problems that can be solved efficiently. **Chapter 3** further discussed this idea in the particular context of real-time voltage control problems. In particular, Section **3.3** introduced the 1st-order Taylor-series expansion of voltage magnitude in terms of nodal power injections and Section **3.5** discussed the formulation of a practical real-time voltage control problem in the presence of DERs.

In this chapter, we now apply the different sensitivity coefficient estimation methods investigated so far in this work to such a real-time voltage control problem. We do so with the aim of comparing the performance of the control algorithm in the case where it uses sensitivity coefficients computed in a model-based approach versus model-less/measurement-based approaches.

### 6.2 The voltage-control problem

We consider the PV curtailment problem that was introduced and formulated as a real-time model predictive control (MPC) problem by Gupta et al. in **[32]**. Through the addition of restrictive voltage constraints, we create a voltage-control scenario in which the PV plant connected to the grid must be adequately curtailed so as to keep nodal voltages within the allowed range of operation.

Assuming a single PV plant  $g$  and perfect generation and load forecasts, the full MPC problem that

we consider can be formalised as:

$$\underset{\mathbf{x}_g}{\text{minimize}} \quad \sum_{t=\underline{t}+1}^T \{(p_{g,t} - \hat{p}_{g,t})^2 + q_{g,t}^2\} \quad (6.1a)$$

$$\text{subject to} \quad \mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{S}_{\underline{t}}[\Delta \mathbf{P}_t, \Delta \mathbf{Q}_t]^T \quad t = \underline{t} + 1, \dots, T \quad (6.1b)$$

$$0 \leq (p_{g,t})^2 + (q_{g,t})^2 \leq (S_{max}^g)^2 \quad t = \underline{t} + 1, \dots, T \quad (6.1c)$$

$$0 \leq p_{g,t} \leq \hat{p}_{g,t} \quad t = \underline{t} + 1, \dots, T \quad (6.1d)$$

$$v_{min} \leq \mathbf{v}_t \leq v_{max} \quad t = \underline{t} + 1, \dots, T \quad (6.1e)$$

where:

- $\underline{t}$  is the operating timestep for which the MPC optimisation problem is solved and  $T$  is the last timestep of the optimisation horizon, leading to a look-ahead time of  $\Delta t(T - \underline{t})$  for a  $\Delta t$  control discretisation,
- $x_{g,t} = [p_{g,t}, q_{g,t}]$  are the decision variables for the curtailed active and reactive power injections of the PV plant  $g$ ,
- $\mathbf{x}_g$  contains the  $x_{g,t}$  decision variables over the optimization horizon considered  $\{\underline{t} + 1, T\}$ ,
- $\hat{p}_{g,t}$  is the maximum power point (MPP) forecast of PV generation,
- $\mathbf{v}_t$  is the vector of nodal voltage magnitudes  $\mathbf{v}_t = [|V_1(t)|, \dots, |V_N(t)|]$ ,
- $\mathbf{S}_{\underline{t}}$  is the sensitivity coefficient matrix estimated at the operating timestep  $\underline{t}$ ,
- $\Delta \mathbf{P}_t$  and  $\Delta \mathbf{Q}_t$  are the vectors of total nodal active and reactive power deviations between timestep  $t - 1$  and  $t$ , which includes both uncontrollable loads and the PV generation,
- $S_{max}^g$  is the rated power of the PV power converter, and
- $[v_{min}, v_{max}]$  is the allowed range of voltage magnitude at all buses.

In the above formulation, the objective function (6.1a) is chosen with the goal of minimising the total PV curtailment (i.e., the difference between the curtailed injections and the MPP forecast) while operating at near-unity power factor (i.e., minimising the reactive power generated). Expression (6.1b) represents the linearised model of the grid through sensitivity coefficients that we have been considering throughout this work. Next, inequality constraints (6.1c) and (6.1d) model the operating constraint of the PV plant and, finally, (6.1e) models the voltage magnitude constraints to be satisfied at all times.

The matrix of sensitivity coefficients  $\mathbf{S}_{\underline{t}}$  is computed at time  $\underline{t}$  and is assumed constant throughout the optimisation horizon. The  $i^{\text{th}}$  element of  $\Delta \mathbf{P}_t$ ,  $\Delta P_{i,t}$ , which corresponds to the active power injection deviation between timesteps  $t + 1$  and  $t$  at bus  $i$  is constructed as:

$$\Delta P_{i,t} = \begin{cases} p_{g,t} - \hat{p}_{i,t}^l - \tilde{P}_i(\underline{t}), & \text{if } i = g, \\ -\hat{p}_{i,t}^l - \tilde{P}_i(\underline{t}), & \text{if } i \neq g, \end{cases} \quad (6.2)$$

where  $\hat{p}_{i,t}^l$  is the forecasted (assumed perfect) demand and  $\tilde{P}_i(t)$  is the last noisy PMU measurement available, which was recorded at the operating point  $t$ . We thus assume that the observations (i.e., noisy measurements  $\tilde{P}_i(t)$ ) are constant throughout the optimisation horizon. A similar expression to (6.2) is used for  $\Delta Q_{i,t}$ .

### 6.3 Simulation and results

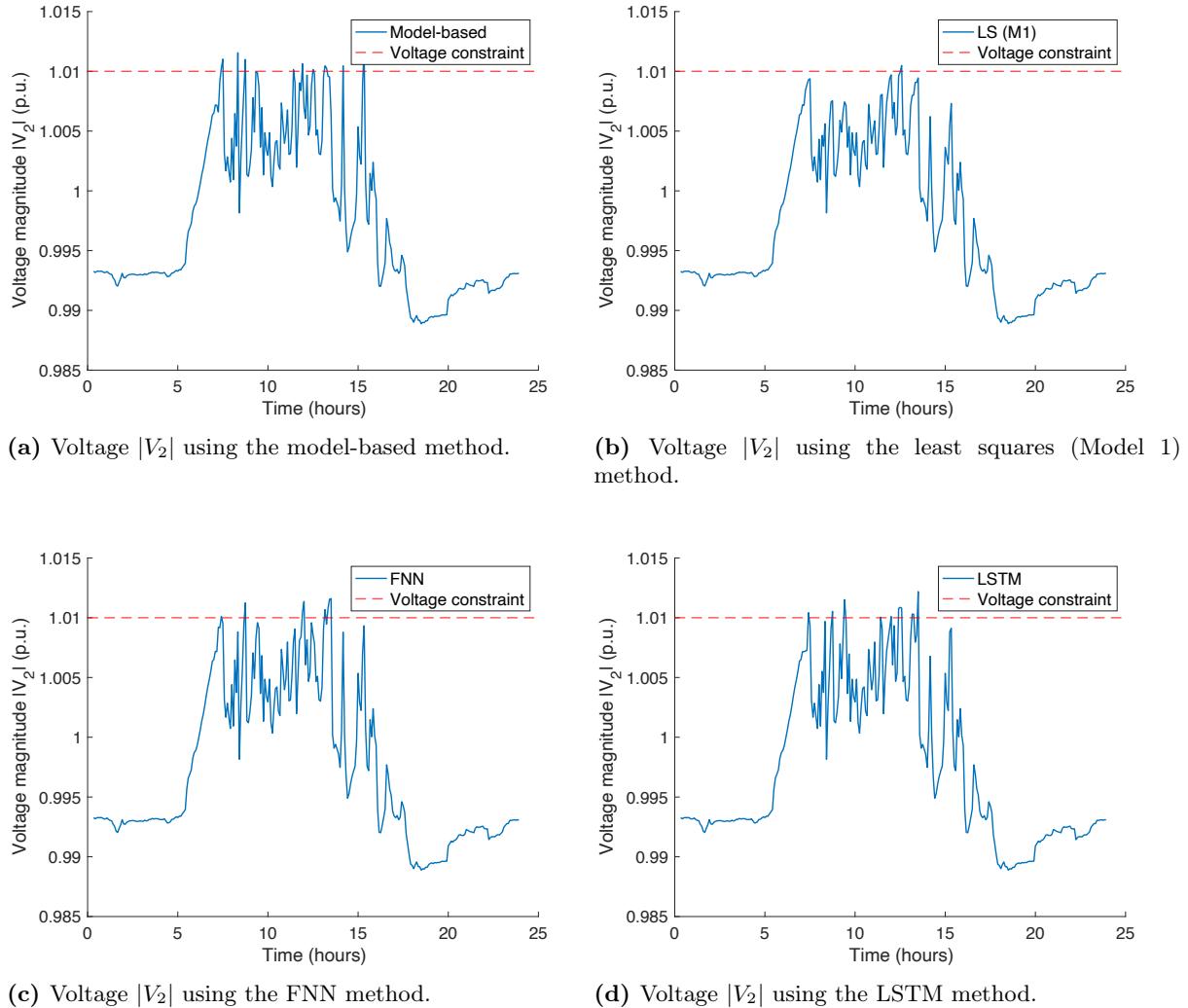
In our simulations, we used the same 24-hour load flow simulation on the CIGRE-13 low voltage distribution network as was introduced in Section 2.3.1. The original voltage and power injection profiles are available in Figure 2.3. In particular, note the voltage magnitude at the PV node,  $|V_2|$ , which rises above 1.02 p.u. in the original load flow as a result of the PV generation in the middle of the day. We now consider the problem of curtailing the PV injection so as to keep  $|V_2|$  below 1.01 p.u. with a control resolution of  $\Delta t = 300$  seconds (i.e., 5 minutes).

The same voltage control problem was repeated 4 times, once for each type of sensitivity coefficient computation method, namely (a) the model-based analytical computation method of Section 3.4.2, (b) the least squares Model 1 approach of Section 4.2, (c) the feedforward neural network method of Section 5.2, and (d) the LSTM-based method of Section 5.3. Measurement time windows of size  $|\tau| = 1000$  were used for least squares,  $K = 50$  for the FNN, and  $K = 25$  for the LSTM models. The same neural network architecture and training procedures as used in previous experiments were also used.

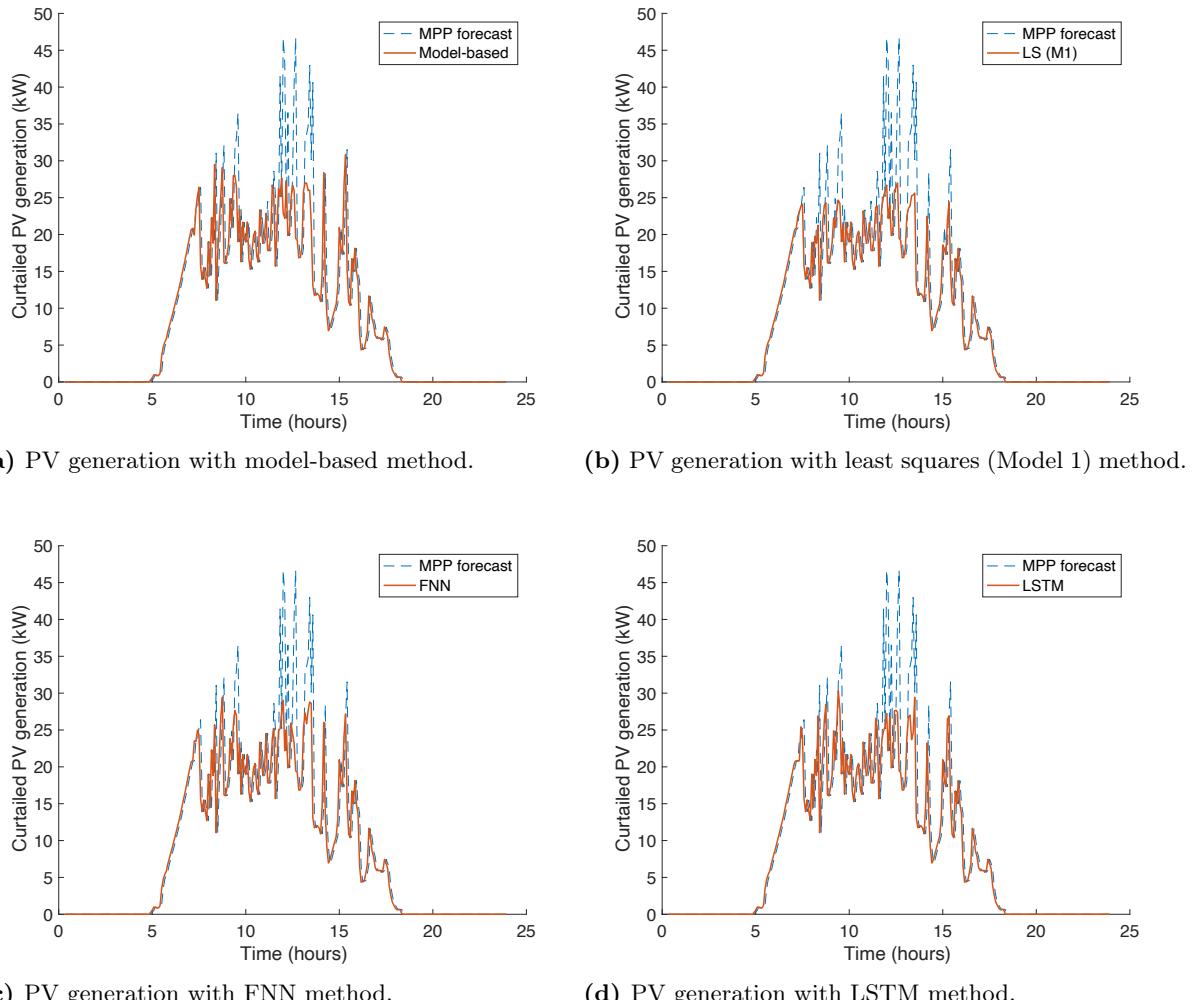
The results are presented in Figures 6.1, 6.2, and 6.3. Figure 6.1 shows the profiles of the voltage magnitude at the PV bus,  $|V_2|$ , for all sensitivity coefficient estimation methods. These profiles were obtained after re-computing a load flow with the final curtailed PV injections, which are shown in Figure 6.2 against the PV MPP forecast. Finally, Figure 6.3 shows the empirical cumulative distribution functions (CDFs) of  $|V_2|$  (left) and the curtailed PV generation (right).

From Figures 6.1 and 6.2, we observe very similar voltage and PV generation profiles for all four estimation methods, in which  $|V_2|$  remains below or very close to the constraint boundary at 1.01. In particular, we note that even the method using model-based estimated sensitivity coefficients fails to respect the constraint at all times. This is a consequence of the fact that noisy observations  $\tilde{P}_i(t)$  are used in the formulation of  $\Delta P_{i,t}$  in (6.2). Although obviously not ideal, it adequately represents a real-world scenario in which only noisy measurements are available. In addition, note that the voltage constraint violations are negligible. As reported in Table 6.2, the maximum constraint violation is 1.0122 for both the model-based and LSTM methods, which corresponds to a 0.11% violation. Table 6.2 also reports the maximum curtailed PV generation of each scenarios, which also corresponds to the largest voltage magnitude recorded.

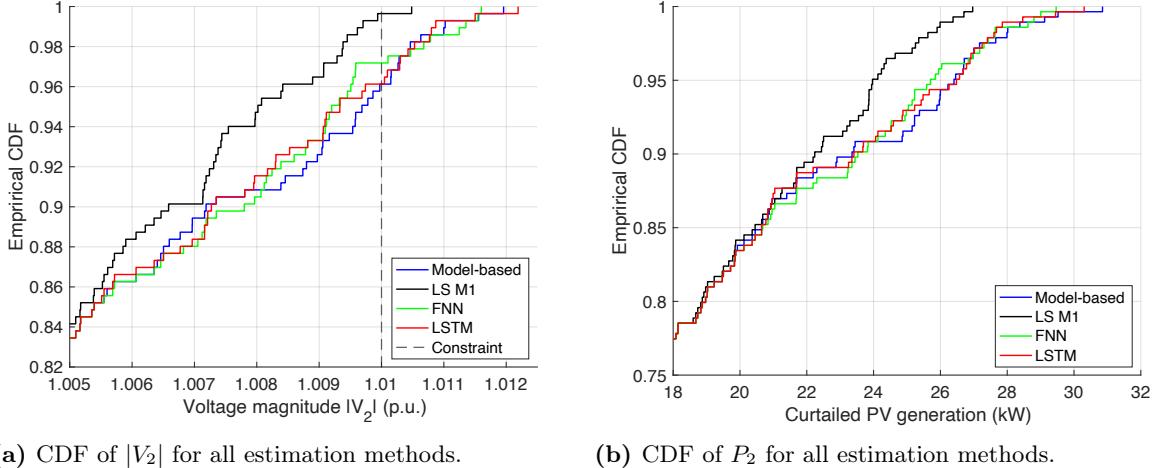
Finally, Figures 6.3a and 6.3b show the proportion of timesteps during which  $|V_2|$  and  $P_2$  are below certain values, respectively. On the one hand, we see that the model-based, FNN, and LSTM approaches lead to similar behaviours at the PV bus. The least squares approach, on the other hand, leads to slightly less voltage constraint violations, but at the cost of more PV curtailment. This is further illustrated in Table 6.1, which shows the total energy produced (in kWh) by the PV plant over the course of the day. This total energy was computed by assuming constant power injection during the  $\Delta t = 5$  minutes



**Figure 6.1:** A comparison between voltage profiles  $|V_2|$  at the PV node after curtailment for each sensitivity coefficient computation/estimation method. One way to justify that voltage goes behind the imposed voltage constraint at 1.01 p.u. in all cases is the fact that measurement errors are present in the formulation of the MPC optimisation step.



**Figure 6.2:** Profiles of active power generation from the PV plant after curtailment, plotted against the MPP forecast (assumed perfect).

(a) CDF of  $|V_2|$  for all estimation methods.(b) CDF of  $P_2$  for all estimation methods.**Figure 6.3:** Empirical cumulative distributions of  $|V_2|$  (left) and  $P_2$  (right).

	Model-based	LS (M1)	FNN	LSTM
Total generation (kWh)	202.4	196.7	202.0	201.8
Proportion of theoretical MMP generation (%)	92.5	89.9	92.3	92.2

**Table 6.1:** Total PV active power generation (in kWh) and the % of the forecasted MPP it represents, for all sensitivity computation/estimation methods.

following each control timestep.

## 6.4 Summary

In this chapter, we applied the four sensitivity coefficient estimation methods studied throughout this thesis to a real-time voltage control/PV curtailment problem, simulating at best a real-world scenario. Our results showed that all sensitivity estimation methods yield similar voltage and curtailed PV generation profiles. In particular, we showed that the FNN and LSTM approaches led to behaviours highly similar to the model-based method. We also noted that all methods led to some constraint violations, but that the violation was negligible at 0.11% in the worst case. One way to avoid such constraint violations in practice would be to take such measurement errors into account while setting the constraint (i.e., by choosing a slightly tighter limit).

These numerical results suggest that all three model-less sensitivity coefficient estimation methods are suitable to be used instead of a model-based approach in distribution networks for which the network model is unknown. They also suggest that the FNN and LSTM approaches could provide slight improvements on the least squares method, although more experiments are required to confidently draw

	Uncurtailed	Model-based	LS (M1)	FNN	LSTM
Maximum nodal voltage $ V_2 $ (p.u.)	1.034	1.0122	1.0105	1.0116	1.0122
Maximum voltage constraint violation (%)	2.38	0.22	0.05	0.16	0.22
Maximum curtailed power PV generation $P_2$ (kW)	46.7	30.85	29.97	29.47	30.31

**Table 6.2:** Maximum voltage magnitude  $|V_2|$  (row 1), voltage constraint violation (i.e.,  $100 \times \frac{\max(|V_2|)}{1.01} - 1$ ) (row 2), and curtailed PV generation  $P_2$  (row 3).

this conclusion.

# Chapter 7

## Impact and Exploitation

### 7.1 Introduction

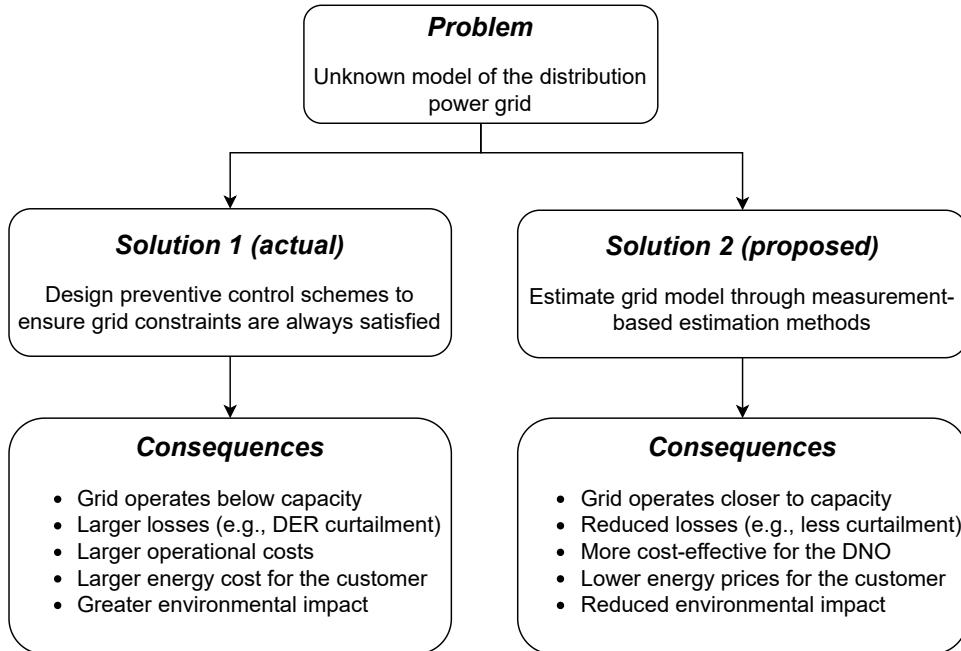
In this chapter, we discuss the place of this project within the larger societal context. Section 7.2 starts by re-stating the core motivations at the root of the proposed research while placing the work done in the bigger picture of the management of power systems. The ideal research outcomes are also advanced. Section 7.3 then discusses both the potential economical, environmental, and societal impact that continuing the proposed research could have.

### 7.2 The big picture

As discussed in the **Introduction Chapter**, the principal motivation for this project, as for other similar ones, is that DNOs sometimes do not have access to an accurate and up-to-date model of the low voltage distribution network they operate. This is a result of the fact that, for example, companies in charge of those networks have erroneous or missing information about the status of breakers and line parameters [35, 36]. DNOs must therefore design control schemes that adequately take into account that knowledge gap while still providing a safe and reliable operation of the grid to the end customers.

The traditional approach to dealing with this lack of information is to design preventive control schemes. This line of action is represented by the left side of Figure 7.1. In this approach, preventive actions are constantly taken so as to minimise the risk of transmission line overheating, over-voltages, power outages, and other potential network failures. In a sense, this can be thought of as "playing it safe" by always ensuring that the state of the network is always well within the grid constraints. This is, of course, a very sensitive approach, since failing to satisfy these constraints can lead to material damage to the network infrastructure or failure to meet the energy demand, both of which can have disastrous economic consequences for both the DNO and the customers [73].

The recent and ongoing addition of PMUs and other measuring infrastructures throughout low voltage networks, however, now offers DNOs the opportunity to explore a much wider range of control schemes. By efficiently processing the ever-growing quantity of available measurements, they can start to consider



**Figure 7.1:** A comparison between classical control schemes in low voltage distribution networks and more recent approaches of model-less/measurement-based control proposed in the literature.

more idealistic strategies that depend on the availability of a more accurate model of the grid. This approach requires the development of data-driven algorithms and procedures that can extract the required information from the measurements and turn it into a useful representation of the grid model and state. This line of action is represented by the right side of Figure 7.1.

The data-driven sensitivity coefficient estimation methods explored in this work fall into that categories of algorithms that could help DNOs design more accurate control schemes for distribution networks. Ideally, this area of research would eventually lead to the design of automatic control frameworks for distribution grids that no longer rely on wrong or outdated network models. These systems would become increasingly autonomous, requiring less and less human supervision.

Of course, the methods investigated during this project are at a very early stage in the typical development chain for a final commercial process and are still far from being implemented in the management of real-world networks. Nevertheless, this initial research step is of crucial importance, as it can help to orientate future research directions taken by both the academic and industrial communities.

In particular, the different analyses and empirical results presented in this work suggest that all three methods could potentially be used as efficient model-less sensitivity coefficient strategies. The following steps would, however, need to be completed before these approaches are applied to real-world grids:

1. Validate the proposed methods on a wider range of increasingly complex scenarios than those investigated in this work (e.g., larger networks, multiple PV installations, etc.). Although we

understand the least squares approach rather well and can characterise its expected performance through theoretical tools (e.g., Cramer-Rao lower bound), the same cannot be said about the neural network-based methods, which are often seen as "black-box" models. Considering any of these methods for practical applications would first require more in-depth performance analyses than those presented here.

2. Once the methods have been experimentally validated on a wide range of computer-based simulations, it is then likely to be used in a pilot implementation in one or a few distribution networks.
3. Finally, after the estimation methods have been successfully applied to a few test cases, it would potentially be deployed to a wider number of larger low voltage grids.

### 7.3 Potential impact

In 2019, the total annual electricity consumption of the UK was 305 TWh, with a total transmission and distribution loss of 7.6% [74] and an average electricity price for UK households of 0.22€ per kWh [75]. This accounts for an annual total electricity loss worth 5.1 billion euros; a non-negligible cost for the UK economy. As a result, developing tools, techniques, and processes to render the daily operation of network operators more efficient could have significant economic advantages for the UK taxpayer. Our research in data-driven estimation methods for sensitivity coefficients in distribution networks fall in that category.

As summarised in the right hand side of Figure 7.1, a decrease in energy distribution losses would mean a grid operating closer to capacity. This would, as previously mentioned, reduce the operation costs for DNOs, which would in turn lower the energy prices for the average UK household. Indirectly, developing model-less methods for sensitivity coefficient estimation in distribution networks could thus drive cheaper electricity in the UK, benefiting the economy, the power distribution industry, and the final customers.

Finally, an important additional consequence of being able to operate distribution grids closer to capacity is the increase in the amount of intermittent renewable energy it could accommodate. As illustrated in Chapter 6, DNOs must regularly curtail the production of DERs to ensure grid constraints are satisfied. This is particularly true in the case where the grid model is unknown or inaccurate and preventive actions are taken. With better sensitivity coefficient estimation methods, DNOs would be able to use more adequate (i.e., less preventive) DER curtailment policies. This would both have important economic (e.g., less energy loss) and environmental (e.g., the ability to rely more on DERs) impacts.

### 7.4 Summary

In this chapter, we discussed the place of the proposed research (model-less/measurement-based sensitivity coefficient estimation methods) in the typical development chain for a final commercial process and the impact it could have on the larger UK society. We argued that, although the research is still at an

early development stage, it represents a critical step as it can help to (a) orient future research interests from both the academic and industrial communities and (b) reduce energy losses in distribution power grids. The latter would have important economical (e.g., lower electricity prices) and environmental (e.g., less DER curtailment) impacts in the UK.

# Chapter 8

## Conclusion

### 8.1 Conclusions

This thesis considered the problem of estimating voltage sensitivity coefficients in low voltage distribution power grids when the network model is unknown or inaccurate, through the use of data-driven estimation methods. We designed two neural network-based estimation methods; one that uses feedforward neural networks and the other recurrent neural networks. Both methods were empirically compared against the regression (least squares) approach previously proposed in the literature and an analytical model-based approach in terms of (a) the estimation accuracy of different coefficients  $S_{ij}$ , (b) the resulting approximations of  $\Delta|V_i|$ , (c) the estimation variance, and (d) the computational requirements. In addition, all four (i.e., model-based, least squares, FNN, and LSTM) computation/estimation methods were applied to a real-time voltage control in which voltage sensitivity coefficients were used to control the curtailment of a PV plant.

Our key findings can be summarised as follows:

1. Least squares sensitivity coefficient estimation methods are often prone to numerical stability issues, have a high estimation variance, and are highly sensitive to measurement errors (**Chapter 4**).
2. Both neural network-based estimation methods proposed in this work yield individual sensitivity coefficients  $S_{ij}$  of similar accuracy to those estimated by least squares, regularly lead to better  $\Delta|V_i|$  approximations, have lower estimation variance, and are more numerically stable (**Chapter 5**).
3. All three data-driven sensitivity coefficient estimation methods (i.e., least squares, FNN, and LSTM) are suitable to replace model-based approaches when the network model is unavailable. This was illustrated with the voltage control application of **Chapter 6**, in which all three methods led to curtailment policies highly similar to their model-based equivalent.

## 8.2 Future work

In terms of future work, we identified three main research directions that could be taken following this work. They are:

1. Conducting numerical experiments with the three data-driven estimation methods on a wider range of distribution power grids and spanning across broader simulation scenarios. This could include larger networks, grids with more DERs (e.g., several PV plants and/or wind farms), meshed networks, and with the addition of storage units, for example. An additional interesting scenario would include highly correlated power injections from different nodes – a case in which least squares is known to lead to poor performance.
2. Extending our experiments and the previous bullet point's in the case of unbalanced three phase grids. Although we do not expect significant changes in performance, experimental validation of this hypothesis would be valuable.
3. Extending the estimation of sensitivity coefficients to branch currents. This would lead to the possibility of doing model-less transmission congestion management, in addition to the model-less voltage control presented here.

# Acknowledgements

I would like to extend my deepest gratitude to my external supervisor, Professor Mario Paolone, for giving me the opportunity to carry out this project with his lab at EPFL and for providing guidance and feedback throughout its completion. I would also like to sincerely thank my internal supervisor, Dr. Sasa Djokic, for the valuable advice and suggestions he gave me whenever I asked for it, and Dr. Jonathan Shek, for taking the time to read my work. I am also extremely grateful to Rahul Gupta, whose support as part of his PhD has been invaluable and helpful in so many ways (and sorry for all the extra work!). I hope we will get to meet in person in a post-pandemic world.

To both of my parents, without whom studying in three different countries over the past five years would not have been possible, I love you. To my mother, Isabelle, and little brother, Arthur, in particular, you have kept me sane over these last pandemic months; *merci*. Finally, to my best friend and partner in life, Isabella Campilisso, thank you; you inspire and bring out the best in me every day.

# References

- [1] Fang, X., Misra, S., Xue, G. and Yang, D.: ‘Smart grid—The new and improved power grid: A survey’, in: *IEEE communications surveys & tutorials* 14.4 (2011), pp. 944–980.
- [2] Joskow, P. L. et al.: *Lessons learned from the electricity market liberalization*, 2008.
- [3] Lasseter, R. H.: ‘Microgrids’, in: *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 02CH37309)*, vol. 1, IEEE, pp. 305–308.
- [4] Carpentier, J. and Siroux, J.: ‘L’optimisation de la production a l’Electricite de France’, in: *Bulletin de la Societe Francaise des Electriciens* (1963).
- [5] Peschon, J., Piercy, D. S., Tinney, W. F., Tveit, O. J. and Cuenod, M.: ‘Optimum control of reactive power flow’, in: *IEEE Transactions on Power Apparatus and Systems* 1 (1968), pp. 40–48.
- [6] Domme, H. W. and Tinney, W. F.: ‘Optimal power flow solutions’, in: *IEEE Transactions on power apparatus and systems* 10 (1968), pp. 1866–1876.
- [7] Sun, D. I., Ashley, B., Brewer, B., Hughes, A. and Tinney, W. F.: ‘Optimal power flow by Newton approach’, in: *IEEE Transactions on Power Apparatus and systems* 10 (1984), pp. 2864–2880.
- [8] Frank, S., Steponavice, I. and Rebennack, S.: ‘Optimal power flow: a bibliographic survey I’, in: *Energy Systems* 3.3 (2012), pp. 221–258.
- [9] Frank, S., Steponavice, I. and Rebennack, S.: ‘Optimal power flow: a bibliographic survey II’, in: *Energy Systems* 3.3 (2012), pp. 259–289.
- [10] Zhu, J.: ‘Optimization of power system operation’, (John Wiley & Sons, 2015).
- [11] Frank, S. and Rebennack, S.: ‘An introduction to optimal power flow: Theory, formulation, and examples’, in: *IIE Transactions* 48.12 (2016), pp. 1172–1197.
- [12] Padhy, N. P.: ‘Unit commitment – a bibliographical survey’, in: *IEEE Transactions on power systems* 19.2 (2004), pp. 1196–1205.
- [13] Alsac, O. and Stott, B.: ‘Optimal load flow with steady-state security’, in: *IEEE transactions on power apparatus and systems* 3 (1974), pp. 745–751.
- [14] Zhang, W., Li, F. and Tolbert, L. M.: ‘Review of reactive power planning: objectives, constraints, and algorithms’, in: *IEEE transactions on power systems* 22.4 (2007), pp. 2177–2186.

- [15] Peschon, J., Piercy, D. S., Tinney, W. F. and Tveit, O. J.: ‘Sensitivity in power systems’, in: *IEEE Transactions on Power Apparatus and Systems* 8 (1968), pp. 1687–1696.
- [16] Hano, I., Tamura, Y., Narita, S. and Matsumoto, K.: ‘Real time control of system voltage and reactive power’, in: *IEEE Transactions on Power Apparatus and Systems* 10 (1969), pp. 1544–1559.
- [17] Kishore, A. and Hill, E.: ‘Static optimization of reactive power sources by use of sensitivity parameters’, in: *IEEE Transactions on Power Apparatus and Systems* 3 (1971), pp. 1166–1173.
- [18] Gembicki, F. and Haimes, Y.: ‘Approach to performance and sensitivity multiobjective optimization: The goal attainment method’, in: *IEEE Transactions on Automatic control* 20.6 (1975), pp. 769–771.
- [19] Hcbsone, E.: ‘Network constrained reactive power control using linear programming’, in: *IEEE Transactions on Power Apparatus and Systems* 3 (1980), pp. 868–877.
- [20] Bandler, J. and El-Kady, M.: ‘A unified approach to power system sensitivity analysis and planning, Part I: Family of adjoint systems’, in: *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 681–687.
- [21] Mamandur, K. and Chenoweth, R.: ‘Optimal control of reactive power flow for improvements in voltage profiles and for real power loss minimization’, in: *IEEE transactions on power apparatus and systems* 7 (1981), pp. 3185–3194.
- [22] Shirmohammadi, D., Hong, H. W., Semlyen, A. and Luo, G.: ‘A compensation-based power flow method for weakly meshed distribution and transmission networks’, in: *IEEE Transactions on power systems* 3.2 (1988), pp. 753–762.
- [23] Ferreira, L. A. F.: ‘Tellegen’s theorem and power systems-new load flow equations, new solution methods’, in: *IEEE transactions on circuits and systems* 37.4 (1990), pp. 519–526.
- [24] Begovic, M. M. and Phadke, A. G.: ‘Control of voltage stability using sensitivity analysis’, in: *IEEE Transactions on Power Systems* 7.1 (1992), pp. 114–123.
- [25] Noroozian, M. and Andersson, G.: ‘Power flow control by use of controllable series components’, in: *IEEE transactions on power delivery* 8.3 (1993), pp. 1420–1429.
- [26] Gurram, R. and Subramanyam, B.: ‘Sensitivity analysis of radial distribution network–adjoint network method’, in: *International Journal of Electrical Power & Energy Systems* 21.5 (1999), pp. 323–326.
- [27] Greene, S., Dobson, I. and Alvarado, F. L.: ‘Contingency ranking for voltage collapse via sensitivities from a single nose curve’, in: *IEEE Transactions on Power Systems* 14.1 (1999), pp. 232–240.
- [28] Greene, S., Dobson, I. and Alvarado, F. L.: ‘Sensitivity of transfer capability margins with a fast formula’, in: *IEEE Transactions on Power Systems* 17.1 (2002), pp. 34–40.
- [29] Capitanescu, F. and Van Cutsem, T.: ‘Unified sensitivity analysis of unstable or low voltages caused by load increases or contingencies’, in: *IEEE Transactions on Power Systems* 20.1 (2005), pp. 321–329.

- [30] Christakou, K., LeBoudec, J.-Y., Paolone, M. and Tomozei, D.-C.: ‘Efficient computation of sensitivity coefficients of node voltages and line currents in unbalanced radial electrical distribution networks’, in: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 741–750.
- [31] Mugnier, C., Christakou, K., Jaton, J., De Vivo, M., Carpita, M. and Paolone, M.: ‘Model-less/measurement-based computation of voltage sensitivities in unbalanced electrical distribution networks’, in: *2016 Power Systems Computation Conference (PSCC)*, IEEE, pp. 1–7.
- [32] Guptal, R., Sossan, F. and Paolone, M.: ‘Performance Assessment of Linearized OPF-based Distributed Real-time Predictive Control’, in: *2019 IEEE Milan PowerTech*, IEEE, pp. 1–6.
- [33] Gupta, R. K., Sossan, F. and Paolone, M.: ‘Grid-aware Distributed Model Predictive Control of Heterogeneous Resources in a Distribution Network: Theory and Experimental Validation’, in: *arXiv preprint arXiv:2008.02848* (2020).
- [34] Bandler, J. W. and El-Kady, M.: ‘A new method for computerized solution of power flow equations’, in: *IEEE Transactions on Power Apparatus and Systems* 1 (1982), pp. 1–10.
- [35] Al-Othman, A. and Irving, M.: ‘Analysis of confidence bounds in power system state estimation with uncertainty in both measurements and parameters’, in: *Electric power systems research* 76.12 (2006), pp. 1011–1018.
- [36] Choi, D.-H. and Xie, L.: ‘Impact analysis of locational marginal price subject to power system topology errors’, in: *2013 IEEE International Conference on Smart Grid Communications (Smart-GridComm)*, IEEE, pp. 55–60.
- [37] Chen, Y. C., Dominguez-Garcia, A. D. and Sauer, P. W.: ‘Measurement-based estimation of linear sensitivity distribution factors and applications’, in: *IEEE Transactions on Power Systems* 29.3 (2013), pp. 1372–1382.
- [38] Chen, Y. C., Dominguez-Garcia, A. D. and Sauer, P. W.: ‘A sparse representation approach to online estimation of power system distribution factors’, in: *IEEE Transactions on Power Systems* 30.4 (2014), pp. 1727–1738.
- [39] Yu, J., Weng, Y. and Rajagopal, R.: ‘Robust mapping rule estimation for power flow analysis in distribution grids’, in: *2017 North American Power Symposium (NAPS)*, IEEE, pp. 1–6.
- [40] Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J. and Vapnik, V.: ‘Support vector regression machines’, in: *Advances in neural information processing systems*, pp. 155–161.
- [41] Pourjafari, E. and Reformat, M.: ‘A Support Vector Regression Based Model Predictive Control for Volt-Var Optimization of Distribution Systems’, in: *IEEE Access* 7 (2019), pp. 93352–93363.
- [42] Liu, Y., Zhang, N., Wang, Y., Yang, J. and Kang, C.: ‘Data-driven power flow linearization: A regression approach’, in: *IEEE Transactions on Smart Grid* 10.3 (2018), pp. 2569–2580.
- [43] Qin, C., Wang, L., Han, Z., Zhao, J. and Wang, W.: ‘A Modified Data-Driven Regression Model for Power Flow Analysis’, in: *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, pp. 794–799.

- [44] Wang, D., Zheng, K., Chen, Q., Zhang, X. and Luo, G.: ‘A data-driven probabilistic power flow method based on convolutional neural networks’, in: *International Transactions on Electrical Energy Systems* (2020), e12367.
- [45] Kabalci, E. and Kabalci, Y.: ‘Chapter 1 - Introduction to smart grid and internet of energy systems’, in: *From Smart Grid to Internet of Energy*, ed. by Kabalci, E. and Kabalci, Y., (Academic Press, 2019), pp. 1–62, ISBN: 978-0-12-819710-3, DOI: <https://doi.org/10.1016/B978-0-12-819710-3.00001-6>, URL: <http://www.sciencedirect.com/science/article/pii/B9780128197103000016>.
- [46] Low, S. H.: ‘Convex relaxation of optimal power flow—Part I: Formulations and equivalence’, in: *IEEE Transactions on Control of Network Systems* 1.1 (2014), pp. 15–27.
- [47] Papathanassiou, S., Hatzigyriou, N., Strunz, K. et al.: ‘A benchmark low voltage microgrid network’, in: *Proceedings of the CIGRE symposium: power systems with dispersed generation*, CIGRE, pp. 1–8.
- [48] Paolone, M., Le Boudec, J.-Y., Sarri, S. and Zanni, L.: ‘Static and recursive PMU-based state estimation processes for transmission and distribution power grids’, in: *Advanced Techniques for Power System Modelling, Control and Stability Analysis* (2015), pp. 189–239.
- [49] Wehenkel, A., Mukhopadhyay, A., Le Boudec, J.-Y. and Paolone, M.: ‘Parameter Estimation of Three-Phase Untransposed Short Transmission Lines from Synchrophasor Measurements’, in: *IEEE Transactions on Instrumentation and Measurement* (2020).
- [50] Instrument Transformers: ‘Additional requirements for electronic voltage transformers’, Standard IEC, 2011, pp. 61 869–8.
- [51] Instrument Transformers: ‘Additional requirements for electronic current transformers’, Standard IEC, 2014, pp. 61 869–7.
- [52] Borghetti, A., Bosetti, M., Grillo, S. et al.: ‘Short-term scheduling and control of active distribution systems with high penetration of renewable resources’, in: *IEEE Systems Journal* 4.3 (2010), pp. 313–322.
- [53] Penfield, P., Spence, R. and Duinker, S.: ‘A generalized form of Tellegen’s theorem’, in: *IEEE Transactions on Circuit Theory* 17.3 (1970), pp. 302–305.
- [54] Van Cutsem, T. and Vournas, C.: ‘Voltage stability of electric power systems’, vol. 441, (Springer Science & Business Media, 1998).
- [55] Christakou, K., Tomozei, D.-C., Bahramipanah, M., Le Boudec, J.-Y. and Paolone, M.: ‘Primary voltage control in active distribution networks via broadcast signals: The case of distributed storage’, in: *IEEE Transactions on Smart Grid* 5.5 (2014), pp. 2314–2325.
- [56] Valverde, G., Zufferey, T., Karagiannopoulos, S. and Hug, G.: ‘Estimation of voltage sensitivities to power injections using smart meter data’, in: *2018 IEEE International Energy Conference (ENERGYCON)*, IEEE, pp. 1–6.
- [57] Jackson, J. E.: ‘A user’s guide to principal components’, vol. 587, (John Wiley & Sons, 2005).

- [58] Kay, S. M.: ‘Fundamentals of statistical signal processing’, (Prentice Hall PTR, 1993).
- [59] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y.: ‘Deep learning’, vol. 1, 2, (MIT press Cambridge, 2016).
- [60] Rojas, R.: ‘The backpropagation algorithm’, in: *Neural networks*, (Springer, 1996), pp. 149–182.
- [61] Hawkins, D. M.: ‘The problem of overfitting’, in: *Journal of chemical information and computer sciences* 44.1 (2004), pp. 1–12.
- [62] Kingma, D. P. and Ba, J.: ‘Adam: A method for stochastic optimization’, in: *arXiv preprint arXiv:1412.6980* (2014).
- [63] Brezak, D., Bacek, T., Majetic, D., Kasac, J. and Novakovic, B.: ‘A comparison of feed-forward and recurrent neural networks in time series forecasting’, in: *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, IEEE, pp. 1–6.
- [64] Varshney, S. and Verma, T.: ‘Half hourly electricity load prediction using echo state network’, in: *International Journal of Science and Research* 3.6 (2014), pp. 885–888.
- [65] Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R. and Schmidhuber, J.: ‘LSTM: A search space odyssey’, in: *IEEE transactions on neural networks and learning systems* 28.10 (2016), pp. 2222–2232.
- [66] Lipton, Z. C., Berkowitz, J. and Elkan, C.: ‘A critical review of recurrent neural networks for sequence learning’, in: *arXiv preprint arXiv:1506.00019* (2015).
- [67] Amidi, A. and Amidi, S.: *CS 230 - Recurrent Neural Networks Cheatsheet*, <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>, (Accessed on 08/01/2021).
- [68] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. and Schmidhuber, J.: ‘A novel connectionist system for unconstrained handwriting recognition’, in: *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2008), pp. 855–868.
- [69] Gers, F. A., Schmidhuber, J. and Cummins, F.: ‘Learning to forget: Continual prediction with LSTM’, in: (1999).
- [70] Hochreiter, S. and Schmidhuber, J.: ‘Long short-term memory’, in: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [71] Olah, C.: *Understanding LSTM Networks – colah’s blog*, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, (Accessed on 08/01/2021), 2015.
- [72] Geman, S., Bienenstock, E. and Doursat, R.: ‘Neural networks and the bias/variance dilemma’, in: *Neural computation* 4.1 (1992), pp. 1–58.
- [73] Morrissey, K., Plater, A. and Dean, M.: ‘The cost of electric power outages in the residential sector: A willingness to pay approach’, in: *Applied energy* 212 (2018), pp. 141–150.
- [74] UK Government: *Market Information Report: Great Britain*, [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/761225/UK\\_Report\\_v3.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/761225/UK_Report_v3.pdf), (Accessed on 08/01/2021), 2018.

- [75] Eurostat: *UK: household electricity prices 2010-2020* / Statista, <https://www.statista.com/statistics/418126/electricity-prices-for-households-in-the-uk/>, (Accessed on 09/01/2021), 2020.

## Appendix A

# CIGRE-13 Network Parameters

The parameters of each transmission line for the CIGRE-13 benchmark distribution power grid used are summarised in Table [A.1](#).

Bus $i$	Bus $j$	Resistance $r_{ij}$ (p.u.)	Reactance $x_{ij}$ (p.u.)	Susceptance $b_{ij}$ (p.u.)
0	1	0.02380	0.01044	0.0005629
1	2	0.12375	0.00530	0.0001131
1	3	0.01190	0.00522	0.0002815
3	4	0.02925	0.00471	0.0001583
3	6	0.15882	0.01732	0.0006069
5	6	0.04538	0.00495	0.0001734
3	7	0.04848	0.01072	0.0004574
7	8	0.01020	0.00447	0.0002412
7	9	0.15881	0.01731	0.0006069
9	10	0.1238	0.00530	0.0001131
9	11	0.0530	0.00577	0.0002023
11	12	0.046	0.00494	0.0001734

**Table A.1:** Total series resistance  $r_{ij}$ , total series reactance  $x_{ij}$ , and total line charging susceptance  $b_{ij}$  of each transmission line  $(i, j) \in \mathcal{E}$ .