

CONTENTS

1	Introduction	2
2	Formulation	4
2.1	States, parameters and dimensions	4
2.2	Principal Component Analysis	5
2.3	Dimension reduction	7
2.4	Polynomial function	7
2.5	Procedure	9
3	Case study	10
4	Potential adaptation to power electronics	11
5	Conclusions	12

1. INTRODUCTION

It is a well-known fact that the power flow in power systems is represented by non-linear equations. Multiple solvers have been applied during these past decades. Some of them have been intentionally created for such purpose, while others are based on common numerical methods, being the Newton-Raphson the most popular. At its core, all these solvers are intended to be used with deterministic input data, that is, they leave no room for variation. In case an input changes, the power flow has to be recalculated.

With this conventional approach, given a power system where there are m uncertain parameters and each of them takes n values, the solver would have to be called n^m times in order to extract all solutions. Thus, an increase in the number of parameters involves a meteoric rise in the computational effort, which is most likely unjustified.

The probabilistic power flow addresses the uncertainty in the parameters' values. It aims at generating the probability distribution of a grid state (e.g., the voltage magnitude at a given bus) as a function of the input varying data (such as the active power provided by a generator). It was first formulated by Borkowska back in 1974 [1]. Nowadays it has gained attraction in the realm of renewable energy sources due to their relative unpredictability. In particular, it has been employed for photovoltaics and wind power, which rely on Gaussian and Weibull distributions respectively [2]–[4]. Moreover, the impact of the electric vehicle charging demand on the grid has been assessed with this technique [5]. However, using the probabilistic power flow does not allow to know with certainty a grid state considering deterministic input data.

The parametric power flow ideally provides a solution to the aforementioned challenge. Instead of dealing with distributions, it generates a closed-form expression that relates the input data (called parameters from now on) with the state of interest. It sacrifices a bit of precision in order to obtain a satisfactory approximation, much more computationally efficient than solving the n^m cases. It has been implemented in conjunction with an optimal power flow formulation [6], yet for the most part it has remained an unexplored methodology. García-Blanco et al. have formulated the so-called Proper Generalized Decomposition, which manages to obtain all n^m solutions reasonably fast and accurately [7], [8]. Nevertheless, the methodology is not scalable. It is tied to a traditional power system formed by PQ buses and a single slack bus. There are workarounds to add PV buses but they are prone to cause divergence. Thus, as power systems

tend to incorporate controllable power electronic devices, there is little relevance in such technique. A much more appealing approach, formulated by Shen et al., deals with expressing a state as a function of the parameters [9]. This is the central reference of this work.

This monograph presents a detailed formulation of the parametric power flow and justifies how the dimensions can be reduced. Then, a basic 5-bus system is tackled to exemplify the methodology. Some thoughts on whether or not this technique can be combined with power electronics are also verbalized. Finally, conclusions are extracted along with the conceptualization of potential research ideas.

2. FORMULATION

2.1. States, parameters and dimensions

States are denoted by \mathbf{x} and stand for the unknowns of the power flow, such as the voltage magnitude. The traditional power flow looks to obtain a solution for all states; in the parametric power flow, this is not necessarily the case, as there might be only one bus under study.

Parameters are defined as the input data which are meant to vary. Powers are part of this category, since loads change over time and most generation sources also experience variations. Considering a total of m parameters, grouped under $\mathbf{p} = [p_1, p_2, \dots, p_m]$, each parameter $p_k \forall k \in \{1, \dots, m\}$ takes values inside the $[a_k, b_k]$ range.

This way, the power flow problem can be written as:

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = 0, \quad (1)$$

where \mathbf{f} symbolizes all implicit functions involved in the solution of the problem. The formulation that follows is generic, in the sense that the methodology would be equally valid for other problems. Hence, this technique is not limited to the traditional power flow.

The parametric approach looks to express a certain state as a function of the parameters:

$$x = g(\mathbf{p}), \quad (2)$$

where g is a function pending to be found.

From (2) it becomes clear that all parameters will potentially affect a given state. Visually speaking, Fig. 1.a shows a representation of a state as a function of a single parameter, that is, assuming that only one input changes. Fig. 1.b presents a similar visualization with two parameters involved. This justifies why, when viewed as a spatial representation, parameters are also denoted as dimensions. Each one of them stands for a new axis independent axis, and thus, orthogonal to the rest.

More than two dimensions are expected to be present in a typical analysis of the parametric power flow. In a mid-size system, assuming all powers are treated as parameters, there could be hundreds of dimensions. If $m \approx 100$, opting for

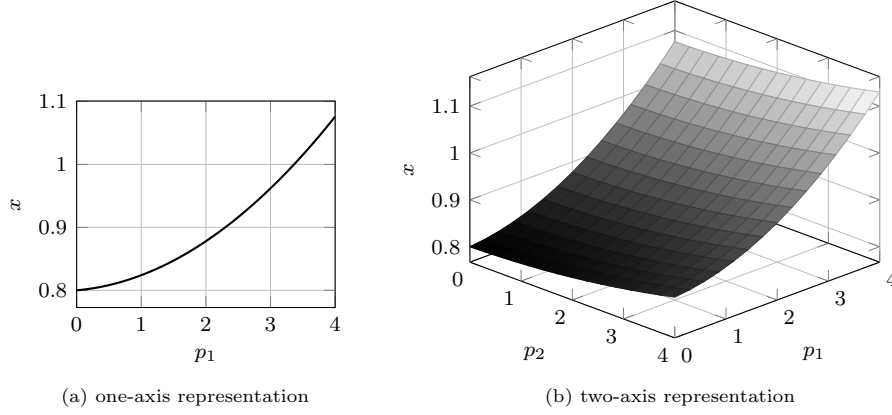


Figure 1. Representation of a state with one or two parameters

the naive approach with n^m cases would be painful to compute. This is often called the curse of dimensionality.

One fundamental step to circumvent this challenge is to reduce the dimensions. Note that in Fig. 1 the value of x is more or less the same in both Fig. 1.a and Fig. 1.b. That is, p_1 has a larger influence than p_2 . The essence of dimensionality reduction lies in compacting all dimensions into a few. A prerequisite is to determine the impact all parameters have on the state.

2.2. Principal Component Analysis

The Principal Component Analysis (PCA) weights the parameters according to their influence. To do so, the first step is concerned with computing the gradient:

$$\nabla_{\mathbf{p}} g = \left[\frac{\partial x}{\partial p_1}, \dots, \frac{\partial x}{\partial p_k}, \dots, \frac{\partial x}{\partial p_m} \right]^T. \quad (3)$$

Calculating the gradient numerically is probably the most beneficial option because it allows to operate with implicit functions, just like the power flow is commonly presented. Hence, for all k in $[1, \dots, m]$:

$$\frac{\partial x}{\partial p_k} = \frac{g(p_1, \dots, p_k + \delta, \dots, p_m) - g(p_1, \dots, p_k, \dots, p_m)}{\delta}, \quad (4)$$

where δ is an arbitrary small number, such as $\delta = 1 \cdot 10^{-10}$ for instance. Computing the gradient with this approach means that the power flow solver has to be called $m + 1$ times.

Let $\mathbf{C} \in \mathbb{R}^{m \times m}$ be the covariance matrix. Its elements identify the influence of the parameters on the states. Large elements represent that the two involved

parameters have a significant impact on the state, whereas small elements that these two parameters are potentially irrelevant. Since a parameter k can take any value inside its corresponding range $[a_k, b_k]$, several samples have to be extracted to have a representative \mathbf{C} matrix:

$$\mathbf{C} \approx \frac{1}{M} \sum_{i=1}^M \nabla_{\mathbf{p}} g(\nabla_{\mathbf{p}} g)^T, \quad (5)$$

where M is the total number of samples and takes a relatively arbitrary value.

The parameters take random values in each sample. The power flow solver would have to be called $M(m+1)$ times in total. Although it largely depends on the number of initial dimensions, M can take values in the order of a hundred. As a consequence, it can be foreseen that calculating $M(m+1)$ power flows is generally way faster than computing n^m cases. The reason is that there is no dependence on the discretized n values and this methodology provides a continuous solution (in the form of a polynomial).

Matrix \mathbf{C} is symmetrical by definition, which means it can be diagonalized by orthogonal matrices:

$$\mathbf{C} = \mathbf{W} \mathbf{A} \mathbf{W}^T, \quad (6)$$

where \mathbf{A} is a diagonal $m \times m$ matrix formed by eigenvalues $[\lambda_1, \dots, \lambda_k, \dots, \lambda_m]$ sorted in order of relevance, and \mathbf{W} contains their respective eigenvectors, denoted as $[\mathbf{w}_1, \dots, \mathbf{w}_k, \dots, \mathbf{w}_m]$, in the form of columns.

The eigenvectors in \mathbf{W} can be understood as dimensions. The idea behind the PCA is to determine the impact of a given dimension, which ends up being proportional to its eigenvalue [9]. Since eigenvalues are sorted by order of importance, \mathbf{w}_1 is more representative than \mathbf{w}_2 , \mathbf{w}_2 is more important than \mathbf{w}_3 , and so on. This process of determining the relevant dimensions is the core of the Proper Orthogonal Decomposition (POD), which has been widely employed in the field of fluid dynamics [10], [11].

If some dimensions are more influential than others, it makes sense to eliminate the insignificant ones. Although this is a subjective step, the truncation error can be defined as:

$$e_t = \frac{\lambda_{k+1} + \dots + \lambda_m}{\lambda_1 + \dots + \lambda_m}, \quad (7)$$

which implies that only eigenvalues $[\lambda_1, \dots, \lambda_k]$ have been selected. The user ought to select a desired error (e.g., $e_t = 10\%$) and identify the eigenvalues that

allow obtaining a similar truncation error. The lower the truncation error, the more accurate and complex the model will be, and vice versa.

2.3. Dimension reduction

Up to this point, the most relevant dimensions have been identified. The next step is to reduce the dimensions:

$$\mathbf{y} = \mathbf{W}_{\mathbf{y}}^T \mathbf{p}, \quad (8)$$

where $\mathbf{W}_{\mathbf{y}}^T$ is formed by the rows of eigenvectors $[\mathbf{w}_1, \dots, \mathbf{w}_k]$. As $\mathbf{W}_{\mathbf{y}}^T \in \mathbb{R}^{k \times m}$, it transforms all m directions into k impactful directions.

Consequently, the initial parametric power flow of the form $x = g(\mathbf{p})$ is transformed into:

$$x \approx h(\mathbf{y}), \quad (9)$$

where h is a function which has yet to be found. In more detail, it will consist of a polynomial. Working with \mathbf{y} , which symbolizes k directions, instead of \mathbf{p} where there are m directions involved, is not mandatory. The state x could still be approximated by a polynomial function $h(\mathbf{p})$. However, if m tends to a large number, the method would suffer from the curse of dimensionality.

More details are provided in the following section, but for now it will be assumed that the number of terms N_t in the function h is given by [9], [12]:

$$N_t = \binom{l+m}{m} = \frac{(l+m)!}{m! (l+m-m)!} = \frac{(l+m)!}{m! l!}, \quad (10)$$

where l denotes the expansion order and it usually does not have to exceed 3. From (10), if $l = 3$ and $m = 5$, $N_t = 56$; if $m = 10$ while $l = 3$, $N_t = 286$. The computational effort would have increased by a factor of approximately 5 even though the number of dimensions has only doubled. This numerical example indicates why it is appropriate to reduce the number of dimensions in spite of losing a bit of accuracy.

2.4. Polynomial function

Function h is a polynomial function of degree l at most. It is formed by a succession of terms such as:

$$x \approx h(\mathbf{y}) = \sum_{j=0}^l \mathbf{c}_j^T \phi_j(\mathbf{y}), \quad (11)$$

where \mathbf{c}_j is a vector of coefficients with the same length as ϕ_j , which contains all basis functions of degree j . All coefficients have to be found in order to determine h . Once this is achieved, h will be fully defined and ready to be used to compute the state for whatever values the parameters take.

As (11) might be too compact, it is exemplified in the case where $k = 3$ (that is, there are 3 meaningful directions) and $l = 2$. The three significant directions are denoted by y_1, y_2, y_3 . The rule is that all three parameters have to be multiplied between them, and the degree of the resulting function has to be equal to j . Thus, when $j = 0$:

$$\begin{aligned} f_0(\mathbf{y}) &= [c_{0,1}] \cdot [y_1^0 y_2^0 y_3^0]^T \\ &= c_{0,1} \cdot 1, \end{aligned} \tag{12}$$

where $c_{0,1}$ stands for the first (and only) coefficient of the basis function with degree 0, and it is unknown. The values of y_1, y_2, y_3 are known since they are treated as inputs.

When $j = 1$:

$$\begin{aligned} f_1(\mathbf{y}) &= [c_{1,1}, c_{1,2}, c_{1,3}] \cdot [y_1^1 y_2^0 y_3^0, y_1^0 y_2^1 y_3^0, y_1^0 y_2^0 y_3^1]^T \\ &= c_{1,1} \cdot y_1 + c_{1,2} \cdot y_2 + c_{1,3} \cdot y_3. \end{aligned} \tag{13}$$

When $j = 2$:

$$\begin{aligned} f_2(\mathbf{y}) &= [c_{2,1}, c_{2,2}, c_{2,3}, c_{2,4}, c_{2,5}, c_{2,6}] \cdot [y_1^2 y_2^0 y_3^0, y_1^1 y_2^1 y_3^0, y_1^1 y_2^0 y_3^1, \\ &\quad y_1^0 y_2^2 y_3^0, y_1^0 y_2^1 y_3^1, y_1^0 y_2^0 y_3^2]^T \\ &= c_{2,1} \cdot y_1 y_2 + c_{2,2} \cdot y_1 y_3 + c_{2,3} \cdot y_2 y_3 + c_{2,4} \cdot y_1^2 + c_{2,5} \cdot y_2^2 + c_{2,6} \cdot y_3^2. \end{aligned} \tag{14}$$

Finally, all f are added:

$$x \approx h(\mathbf{y}) = f_0(\mathbf{y}) + f_1(\mathbf{y}) + f_2(\mathbf{y}). \tag{15}$$

Note that taking into account (12), (13) and (14), there are a total of 10 terms, just like (10) indicates when $l = 2$ and $m = 3$.

Recall that a total of M samples, where the parameters take random values, are needed to build the covariance matrix. Now these samples have to be used in order to compute the coefficients c . Let $\mathbf{Q} \in \mathbb{R}^{M \times N_t}$ denote the matrix where the terms that multiply the coefficients c are stored, and \mathbf{h}_x the vector that contains the value of the state x at each sample. Then, the goal is to minimize the error ϵ in the generic expression:

$$\mathbf{h}_x = \mathbf{Q}\mathbf{c} + \epsilon, \tag{16}$$

where \mathbf{c} is constituted by the N_t coefficients.

Several ways to calculate the coefficients in \mathbf{c} exist [9]. If the least squares regression method is chosen [13], the best estimate of \mathbf{c} is:

$$\mathbf{c} = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{h}_{\mathbf{x}}. \quad (17)$$

To apply this least squares regression, it is necessary to have $M \geq N_t$, and as a rule of thumb, M should be around 1.5 to 3 times N_t [13]. This concludes the calculation process, as $h(\mathbf{y})$ is fully known.

2.5. Procedure

The steps to take in order to obtain the reduced dimension model are summarized below:

1. Compute the gradients for the M samples to calculate the covariance matrix \mathbf{C} with (5).
2. Perform the orthogonal decomposition of \mathbf{C} and select the most influential dimensions according to the desired truncation error.
3. With the random samples, transform \mathbf{p} into \mathbf{y} with (8).
4. Build the \mathbf{Q} matrix, the $\mathbf{h}_{\mathbf{x}}$ vector, and find the coefficients with (17).

Once the coefficients \mathbf{c} are calculated, the procedure yields an explicit function which relates the state x with any value of the parameters.

3. CASE STUDY

4. POTENTIAL ADAPTATION TO POWER ELECTRONICS

5. CONCLUSIONS

BIBLIOGRAPHY

- [1] B. Borkowska, “Probabilistic load flow,” *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 752–759, 1974.
- [2] G. Li, W. Lu, J. Bian, F. Qin, and J. Wu, “Probabilistic optimal power flow calculation method based on adaptive diffusion kernel density estimation,” *Frontiers in Energy Research*, vol. 7, p. 128, 2019.
- [3] J. M. Morales, L. Baringo, A. J. Conejo, and R. Mínguez, “Probabilistic power flow with correlated wind sources,” *IET generation, transmission & distribution*, vol. 4, no. 5, pp. 641–651, 2010.
- [4] M. Fan, V. Vittal, G. T. Heydt, and R. Ayyanar, “Probabilistic power flow studies for transmission systems with photovoltaic generation using cumulants,” *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2251–2261, 2012.
- [5] G. Li and X.-P. Zhang, “Modeling of plug-in hybrid electric vehicle charging demand in probabilistic power flow calculations,” *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 492–499, 2012.
- [6] K. d. Almeida, F. Galiana, and S. Soares, “A general parametric optimal power flow,” *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 540–547, 1994.
- [7] R. García-Blanco, “Efficient solvers for power flow equations: Parametric solutions with accuracy control assessment,” 2017.
- [8] R. G. Blanco, “Efficient solvers for power flow equations: Parametric solutions with accuracy control assessment,” Ph.D. dissertation, Universitat Politècnica de Catalunya (UPC), 2017.
- [9] D. Shen, H. Wu, B. Xia, and D. Gan, “A principal component analysis-based dimension reduction method for parametric power flow,” in *2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2020, pp. 1–5. DOI: 10.1109/APPEEC48164.2020.9220420.
- [10] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [11] J. Weiss, “A tutorial on the proper orthogonal decomposition,” in *AIAA Aviation 2019 Forum*, 2019, p. 3333.
- [12] Y. Zhou, H. Wu, C. Gu, and Y. Song, “A novel method of polynomial approximation for parametric problems in power systems,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3298–3307, 2016.

- [13] Y. Shin and D. Xiu, “Nonadaptive quasi-optimal points selection for least squares linear regression,” *SIAM Journal on Scientific Computing*, vol. 38, no. 1, A385–A411, 2016.