

USP – UNIVERSIDADE DE SÃO PAULO  
EESC – ESCOLA DE ENGENHARIA DE SÃO CARLOS

GABRIEL JOSÉ NEGRELLI GOMES

**IDENTIFICAÇÃO DE MODELO DE CARGA  
UTILIZANDO MEDIDAS DE PERTURBAÇÕES  
PROVOCADAS POR TAPS DE TRANSFORMADORES**

SÃO CARLOS  
DEZEMBRO DE 2017



USP – UNIVERSIDADE DE SÃO PAULO  
EESC – ESCOLA DE ENGENHARIA DE SÃO CARLOS

GABRIEL JOSÉ NEGRELLI GOMES

**IDENTIFICAÇÃO DE MODELO DE CARGA  
UTILIZANDO MEDIDAS DE PERTURBAÇÕES  
PROVOCADAS POR TAPS DE TRANSFORMADORES**

Trabalho apresentado ao Curso de Engenharia Elétrica com Ênfase em Automação e Sistemas de Energia da Escola de Engenharia de São Carlos da Universidade de São Paulo como parte dos requisitos para obtenção do título de Engenheiro Eletricista

Orientador: Prof. Dr. Elmer P. T. Cari

SÃO CARLOS  
DEZEMBRO DE 2017



AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes e  
Seção Técnica de Informática, EESC/USP com os dados fornecidos pelo(a) autor(a).

G633i      Gomes, Gabriel José Negrelli  
Identificação de Modelo de Carga Utilizando Medidas  
de Perturbações Provocadas por Taps de Transformadores /  
Gabriel José Negrelli Gomes; orientador Elmer Pablo  
Tito Cari. São Carlos, 2017.

Monografia (Graduação em Engenharia Elétrica com  
ênfase em Sistemas de Energia e Automação) -- Escola de  
Engenharia de São Carlos da Universidade de São Paulo,  
2017.

1. Modelo de Carga. 2. Z-IM. 3. Parâmetros. 4. Tap.  
5. Transformador. 6. Python. I. Título.

Bibliotecário responsável pela estrutura de catalogação da publicação:

Eduardo Graziosi Silva - CRB - 8/8907



# FOLHA DE APROVAÇÃO

Nome: Gabriel José Negrelli Gomes

Título: “Identificação de modelo de carga utilizando medidas de perturbações provocadas por TAPS de transformadores”

Trabalho de Conclusão de Curso defendido e aprovado  
em 01 / 12 / 2017,

com NOTA 9,0 (nove , zero), pela Comissão Julgadora:

*Prof. Dr. Elmer Pablo Tito Cari - Orientador - SEL/EESC/USP*

*Mestre Taylon Gomes Landgraf - Doutorando - SEL/EESC/USP*

*Mestre Carlos Matheus Rodrigues de Oliveira - Doutorando - SEL/EESC/USP*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Rogério Andrade Flauzino





## AGRADECIMENTOS

Ao Dr. Elmer Pablo, pelo conhecimento e oportunidade de realizar essa pesquisa.

Aos amigos Kae Augusto, Eduardo Tanizaka, Augusto Mussi, Bruno de Oliveira e Guilherme Chiqueti, pela amizade desde o começo dessa jornada.

Aos companheiros da República Mói-Cana, pelas risadas e momentos de descontração, em especial a Agostinho Junior e Pedro Paulo Frizzo, pelas ótimas e péssimas ideias.

À minha família, por todo o apoio e amor.

À Jéssica Freire Feitor, por todo o carinho, compreensão e, principalmente, por toda paciência.



# SUMÁRIO

ÍNDICE DE FIGURAS .....	iii
LISTA DE ABREVIATURAS E SIGLAS .....	v
RESUMO .....	vii
ABSTRACT.....	ix
1. INTRODUÇÃO .....	1
1.1. Objetivos .....	2
1.2. Organização do trabalho .....	3
2. CONCEITOS TEÓRICOS .....	5
2.1. Perturbações de Transformadores .....	5
2.2. Modelo de Carga .....	5
2.3. Modelo de Carga Linearizado .....	8
2.4. Método de Sensibilidade de Trajetória.....	9
2.5. Estimação em Duas Etapas .....	11
3. MATERIAIS E MÉTODOS .....	13
3.1. Python.....	13
3.2. Interface Gráfica.....	14
4. RESULTADOS .....	19
5. CONCLUSÕES .....	23
6. TRABALHOS FUTUROS .....	25
7. ARTIGOS PUBLICADOS .....	27
8. REFERÊNCIAS .....	29



## ÍNDICE DE FIGURAS

Figura 1 - Potência observada e simulada por operadores americanos .....	1
Figura 2 - Esquemático de modelo Z-IM.....	6
Figura 3 - Fluxograma do método original. ....	9
Figura 4 - Página inicial do software .....	14
Figura 5 - Menu "Input Value" .....	15
Figura 6 - Menu "Settings" .....	15
Figura 7 - Página de valores iniciais .....	16
Figura 8 - Flags .....	16
Figura 9 - Comportamento das saídas anterior à estimação .....	17
Figura 10 - Comportamento das saídas após a convergência .....	17
Figura 11 - Saídas reais e simuladas antes da estimação .....	20
Figura 12 - Saídas reais e simuladas após da estimação .....	21
Figura 13 - Condicionamento dos parâmetros .....	21
Figura 14 - Faixa de incerteza dos parâmetros .....	22



## LISTA DE ABREVIATURAS E SIGLAS

GUI	-	<i>Graphical User Interface</i> (Interface Gráfica de Usuário)
ONS	-	Operador Nacional do Sistema Elétrico
SEP	-	Sistema Elétrico de Potência





## RESUMO

GOMES, G. J. N. **Identificação de Modelo de Carga Utilizando Medidas de Perturbações Provocadas por Taps de Transformadores.** 2017. 46f. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017.

Modelos de carga representam o comportamento de uma cidade (unidade consumidora), e podem ser modelados como um motor de indução em paralelo a uma impedância, que representam, respectivamente, as cargas industriais e domésticas. O objetivo desta pesquisa é a estimação correta dos parâmetros (resistências, reatâncias e constantes de tempo) deste modelo, sendo importante em simulações para estudos de estabilidade transitória e de tensão.

Um dos problemas para estimar os parâmetros do modelo de carga é a necessidade de medidas durante uma perturbação na rede (curto-circuitos, contingências, etc.), que não são frequentes. Nesta pesquisa, as medidas são obtidas durante a atuação do tap de transformador, que é mais comum e facilita a aplicação na prática. Entretanto, é necessário linearizar as equações do modelo. As saídas do modelo computacional são comparadas com as medidas reais e os parâmetros do modelo são ajustados a fim de aproximá-las. O método de estimação chama-se sensibilidade de trajetória.

O processo de estimação foi aplicado no modelo de carga Z-IM a partir de uma estimativa inicial dos parâmetros, convergindo em 5 iterações, com tempo de processamento de 2 segundos em um PC CORE i7, mostrando-se rápido e eficaz. O algoritmo, criado em linguagem Python, se mostrou eficiente em identificar corretamente os parâmetros do modelo, resultando em saídas próximas às reais. Um pequeno exemplo do código utilizado é disponibilizado no link: <http://redmine2.lacosep.sel.eesc.usp.br/redmine/documents/69>

Palavras-chave: Modelo de carga, Z-IM, parâmetros, tap, transformador, python.



## ABSTRACT

GOMES, G. J. N. **Load Model Identification Using OLTC Created Disturbances.** 2017. 46f. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017.

Load models represent the behaviour of a city and can be modelled as an induction motor (IM) and an impedance (Z), representing, respectively, industrial and residential consumers. The project's goal is to correctly estimate the model's parameters (resistances, reactances and time constants), impacting on simulations of voltage and transitory stability.

The problem in estimating load model's parameters is the need of measurements taken during disturbances (short circuit, faults, etc.), which are not frequent. For this paper, the measurements were obtained during a tap change, which is more frequent and makes the application easier. However, the model's linearized equations are needed. The model's outputs and real measurements are compared and the parameters are adjusted until the behaviours are close enough. This estimation method is called trajectory sensitivity.

The estimation process was applied to an initial set of parameters, converging in 5 iterations. The whole process took 2 seconds on a PC Core i7. The algorithm, created in Python, was efficient identifying parameters of a Z-IM load model, resulting in an output's behaviour close to the observed in real systems. A small example of the code developed can be found at: <http://redmine2.lacosep.sel.eesc.usp.br/redmine/documents/69>

Keywords: Load model, Z-IM, parameters, tap, OLTC, python.



## 1. INTRODUÇÃO

A operação e o planejamento da rede elétrica requerem que o operador tenha ciência do estado atual do Sistema Elétrico de Potência (SEP) e de como esse se comportará frente a mudanças. Para tanto, são necessários modelos matemáticos que representem de forma fiel o comportamento dos elementos que compõem a rede. Geradores e linhas de transmissão tiveram grande enfoque nas últimas décadas, apresentando hoje modelos confiáveis e de precisão elevada. Entretanto, cargas ainda são representadas por modelos simplificados, que combinam impedância ( $Z$ ), corrente ( $I$ ) e potência ( $P$ ) constantes. Tais modelos funcionam muito bem para representar comportamento em regime permanente, mas destoam do comportamento real durante análises de transitórios (Guo, Rudion, Abildgaard, Komarnicki, & Styczynski, 2012).

Análises realizadas com base em modelos obsoletos acarretam em resultados não condizentes com a realidade, possivelmente ocasionando em um colapso do sistema sem previsão. Tais discrepâncias resultaram em blecautes pelos Estados americanos da Califórnia e Oregon, em 1996, e do Rio de Janeiro, em 1997, devido a inequações do modelo de carga ZIP utilizado nos programas padrões da época (Hauer & Dagle, 1999), (Henriques, et al., 2002). A Figura 1 abaixo mostra como modelos incorretos dificultam o trabalho da equipe de operação em prever o comportamento do sistema frente a transitórios e, dessa forma, ameaçam a estabilidade do SEP. No gráfico superior é apresentado o comportamento oscilatório da potência na linha Califórnia-Oregon enquanto no inferior é mostrado o comportamento simulado na época pelos operadores.

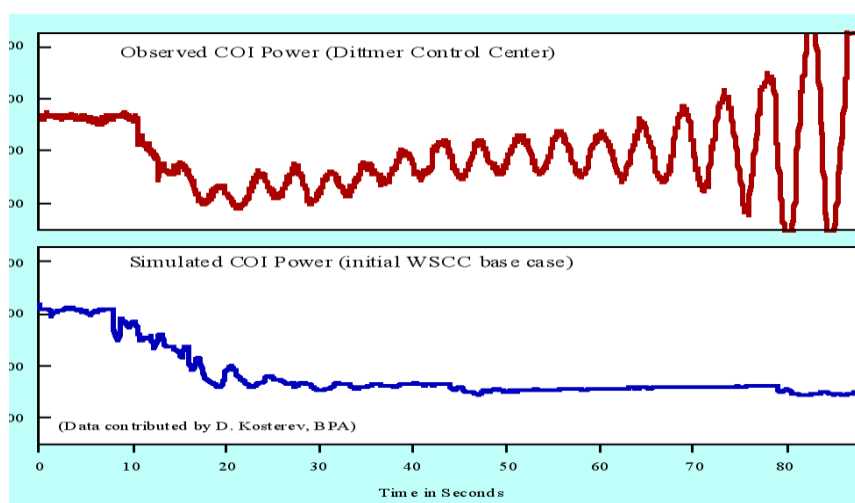


Figura 1 - Potência observada e simulada por operadores americanos. Fonte: (Hauer and Dagle 1999)

No decorrer das últimas décadas, dois métodos foram propostos para obtenção de um modelo de carga fiel, tanto para regime permanente quanto para transitórios, sendo eles baseados na composição dos agregados e em medidas reais durante perturbações. O primeiro consiste em obter o modelo de cada um dos componentes que compõem a carga e agrupá-los. Esse método resulta em um modelo mais próximo do real, porém muito mais difícil de ser implementado devido à grande extensão do sistema e à diversidade de elementos a serem modelados. O segundo, baseado em medidas reais obtidas diretamente do sistema, exige a ocorrência de alguma perturbação na tensão da rede, seja ela natural, como faltas monofásicas, desligamentos de carga não programados e curto-circuitos, que são pouco frequentes e imprevisíveis, ou intencional, por meio de manobras, retirada de grandes cargas, entre outros (Arefifar & Xu, 2013). Por representarem um risco para a estabilidade do sistema, perturbações intencionais na rede devem ser aprovadas pelo Operador Nacional do Sistema (ONS), e geralmente só ocorrem em casos de extrema necessidade.

Dessa forma, se faz necessário encontrar uma forma de estimar os parâmetros do modelo de carga utilizando perturbações que sejam seguras e naturais da rede, como é o caso das variações de *tap* de transformadores. Tais variações ocorrem para que seja mantido o valor nominal de tensão na rede, independente da demanda da carga. Apesar de serem variações de pequena magnitude, em torno de 0,625% da tensão da rede, as mudanças de *tap* podem ser suficientes para a identificação correta dos parâmetros da carga. Outras vantagens desse tipo de perturbação são a frequência com que acontecem, chegando a ocorrer dezenas de vezes durante o dia com previsibilidade, resultando em modelos de carga constantemente atualizados e de maior relevância para a operação (Guo, Rudion, Abildgaard, Komarnicki, & Styczynski, 2012), (Xu, Vaahedi, Mansour, & Tamby, 1997). Entretanto, um problema ao usar este tipo de perturbação é o fato de os dados obtidos por meio das medidas realizadas não conterem informação suficiente para um modelo de carga com características não lineares.

## **1.1. Objetivos**

O projeto surgiu com o objetivo de identificar um modelo de carga por meio de perturbações provocadas na rede por mudanças de taps de transformadores. A identificação ficará a cargo de um algoritmo próprio baseado no método de sensibilidade de trajetória. Os

códigos foram desenvolvidos de tal forma que eles possam ser reaproveitados por outros alunos do grupo de pesquisa que se utilizem do método de sensibilidade de trajetória para identificar parâmetros de qualquer modelo físico. Também foi criada uma interface gráfica amigável ao usuário para facilitar a utilização e visando uma futura comercialização do software. Todos os programas foram desenvolvidos em linguagem Python, devido especialmente ao fato de ser uma linguagem de código aberto e com grande colaboração entre os usuários. Os resultados obtidos nesse projeto suprirão a necessidade atual de modelos realistas e confiáveis, tanto para comportamentos estáticos quanto dinâmicos.

## **1.2. Organização do trabalho**

Este trabalho é apresentado em 5 seções, sendo o capítulo 2 uma recapitulação dos principais conceitos teóricos empregados durante a pesquisa. Em seguida, no capítulo 3, são apresentadas a linguagem de programação e as bibliotecas empregadas na construção do algoritmo, assim como a interface gráfica criada. O capítulo 4 traz os resultados obtidos utilizando os métodos propostos. No capítulo 5 são apresentadas as conclusões tiradas a partir desses resultados e, por fim, o capítulo 6 traz as perspectivas futuras para o trabalho.





## 2. CONCEITOS TEÓRICOS

O algoritmo criado se utiliza da variação das potências ativa e reativa durante perturbações provocadas por *taps* de transformadores para estimar os parâmetros do modelo de carga Z-IM. Com parâmetros atualizados, esse modelo é capaz de simular o comportamento da carga conectada ao transformador. Esses e outros conceitos essenciais para a realização deste trabalho são explicados neste capítulo.

### 2.1. Perturbações de Transformadores

Para que seja possível estimar os parâmetros do modelo de carga a partir de uma mudança de *tap* de transformador, primeiramente é necessário certificar-se de que o sistema elétrico enxerga essa manobra como uma perturbação. Para tal, análises acerca da magnitude da perturbação e da velocidade de atuação são necessárias.

A velocidade de atuação do *tap* é de 2 a 3 ciclos, sendo rápido o bastante para ser vista como um degrau de tensão pela carga. Dessa forma, as constantes transitórias influenciam no comportamento das saídas e se torna possível estimá-las. Quanto à magnitude, estudos mostraram que os valores estimados dos parâmetros variam muito pouco para diferentes valores de variação de tensão (Arefifar & Xu, 2013). De fato, variações entre 1% e 15% da tensão nominal da rede resultam nos mesmos valores de parâmetros (Xu, Vaahedi, Mansour, & Tamby, 1997).

Outra vantagem em utilizar perturbações de transformadores é o fato de este equipamento atuar diversas vezes durante o dia e com certa regularidade. Isso acarreta em modelos constantemente atualizados e que podem ser comparados para estudo de planejamento e expansão da rede. Além disso, este equipamento já se encontra instalado na rede, sendo apenas necessária a aquisição dos dados de potência da carga que está ligada a ele.

### 2.2. Modelo de Carga

Modelos de carga são modelos matemáticos que representam o agregado de diversos equipamentos e que descrevem, por meio de um sistema de equações, seu comportamento visto do barramento dos geradores (IEEE Task Force on Load Representation for Dynamic Performance, 1995). Existem diversos modelos e a escolha de qual será empregado depende

da aplicação. Alguns modelos mais simples levam em consideração apenas o comportamento em regime permanente da carga, como é o exemplo do modelo ZIP (carga de impedância (Z), corrente (I) ou potência (P) constante), enquanto outros descrevem comportamentos durante transitórios, como os modelos de recuperação exponencial.

Nesse projeto, foi considerado o modelo Z-IM, que representa cargas por meio de uma admitância conectada em paralelo a um motor de indução de terceira ordem, como pode ser visto na Figura 2.

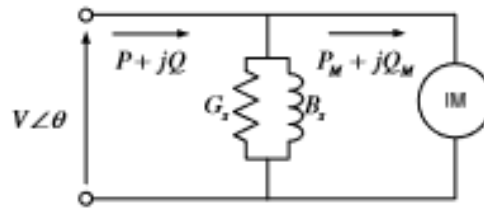


Figura 2 - Esquemático de modelo Z-IM. Fonte: (Choi, et al. 2006)

O sistema de equações que rege esse modelo é dado pela equação 1, com alguns de seus termos obtidos conforme as equações 2, 3 e 4.

$$\begin{cases} T_0' \frac{dE'}{dt} = -\frac{X}{X'} E' + \frac{X-X'}{X'} V \cos \delta \\ \frac{d\delta}{dt} = \omega - \omega_s - \frac{X-X'}{X'} \cdot \frac{V \sin \delta}{T_0' E'} \\ M \frac{d\omega}{dt} = -\frac{V \cdot E' \sin \delta}{X'} - T_m \end{cases} \quad 1$$

$$X = X_s + X_m \quad 2$$

$$X' = X_s + \frac{X_m X_r}{X_m + X_r} \quad 3$$

$$T_0' = \frac{X_r + X_m}{\omega_s R_r} \quad 4$$

onde  $E'$  e  $\delta$  são a magnitude e o ângulo da tensão nos terminais do motor,  $\omega_s$  e  $\omega$  são, respectivamente, as velocidades do estator e rotor, em rad/s,  $X_m$ ,  $X_s$  e  $X_r$  são as reatâncias de magnetização, de estator e rotor,  $R_r$ , a resistência do rotor,  $T_0'$ , a constante de tempo transitória em circuito aberto,  $M$ , a inércia do motor e  $T_m$ , o torque da carga. As equações das

potências podem ser vistas em 5 e 6, onde  $G_s$  e  $B_s$  são, respectivamente, a condutância e a susceptância da parcela constante.

$$P = G_s V^2 - \frac{VE'}{X'} \sin \delta \quad 5$$

$$Q = B_s V^2 + \frac{V}{X'} (V - E' \cos \delta) \quad 6$$

Esse modelo se assemelha ao comportamento de cargas devido às suas parcelas constante (Z) e transitória (IM). Além disso, tem como vantagem o baixo número de parâmetros a serem calculados, o que torna mais simples e rápida a convergência do método numérico. A Tabela 1 compara o modelo Z-IM a outros 2 tipos de modelos de carga de maior complexidade, mostrando que pouco se perde em exatidão utilizando o modelo escolhido. O vetor de parâmetros é definido por  $p = [M, T'_0, X, X', T_m, G_s, B_s, E'_0, \delta_0, \omega_0]$ , onde  $E'_0$ ,  $\delta_0$  e  $\omega_0$  são os valores iniciais da magnitude e ângulo da tensão e da velocidade do rotor (Choi, et al., 2006).

Tabela 1 - Comparação entre diferentes modelos de carga. Fonte: (Choi, et. al., 2006)

$p^*$	GZIP-IM	$p$	EXP-IM	$p$	Z-IM
$R_s$	0.1247	$R_s$	0.1914	$M$	0.0139
$R_r$	0.0210	$R_r$	0.0220	$T'_0$	0.0963
$X_s$	0.1056	$X_s$	0.1682	$X$	0.2089
$X_m$	1.0460	$X_m$	3.0041	$X'$	0.0446
$X_r$	0.1423	$X_r$	0.1194	$T_m$	8.6157
$H$	0.0043	$H$	0.0033	$E'_0$	1.0750
$T_0$	1.1120	$T_0$	0.9068	$\delta_0$	-0.3689
$\beta$	2.8888	$\beta$	2.9614	$\omega_{r0}$	364.381
$\omega_{r0}$	359.935	$\omega_{r0}$	361.734	$G_s$	4.1358
$P_{zro}$	0.8650	$P_{exp0}$	0.7550	$B_s$	2.8004
$p_1$	0.3631	$K_{pr}$	1.8444		N/A
$p_2$	0.4963	$Q_{exp0}$	0.3583		N/A
$p_3$	0.1406	$K_{qr}$	3.2623		N/A
$Q_{zro}$	0.0875		N/A		N/A
$q_1$	0.4313		N/A		N/A
$q_2$	0.7082		N/A		N/A
$q_3$	-0.1395		N/A		N/A
$\varepsilon_p(\%)$	0.3241	$\varepsilon_r(\%)$	0.3627	$\varepsilon_r(\%)$	1.1504
$\varepsilon_Q(\%)$	4.1538	$\varepsilon_Q(\%)$	4.4273	$\varepsilon_Q(\%)$	4.4207

$p^*$ : parameter vector

### 2.3. Modelo de Carga Linearizado

Como a perturbação provocada pela mudança de *tap* é pequena, pode-se lançar mão das equações linearizadas do modelo Z-IM. Equações linearizadas potencializam as variações nas saídas do sistema, mesmo aquelas provocadas por perturbações mínimas, possibilitando a estimação dos parâmetros.

As equações são linearizadas em torno do ponto  $x_0 = [E_0, \delta_0, \omega_0]^T$ , com os vetores de entrada, saída e espaço de estados passando a ser dados conforme as equações 7, 8 e 9.

$$\Delta u = [\Delta V_L] \quad 7$$

$$\Delta y = \begin{bmatrix} \Delta P_e \\ \Delta Q_e \end{bmatrix} \quad 8$$

$$\Delta x = \begin{bmatrix} \Delta E' \\ \Delta \delta \\ \Delta \omega \end{bmatrix} \quad 9$$

Dessa forma, as equações de estados passam a ser descritas pelo sistema exposto em 10.

$$\begin{cases} \Delta \dot{x} = A * \Delta x + B * \Delta u \\ \Delta y = C * \Delta x + D * \Delta u \end{cases} \quad 10$$

As matrizes A, B, C e D são obtidas derivando as equações de estado do modelo de carga em relação a cada variável de estado. Para este sistema, foram encontradas as matrizes apresentadas em 11, 12, 13 e 14.

$$A = \begin{bmatrix} \frac{-X}{T_0 X'} & \frac{-(X-X')V_{L0} \sin \delta_0}{T_0 X'} & 0 \\ \frac{(X-X')V_{L0} \sin \delta_0}{T_0 X' (E'_0)^2} & \frac{-(X-X')V_{L0} \cos \delta_0}{T_0 X' E'_0} & 1 \\ \frac{-V_{L0} \sin \delta_0}{MX'} & \frac{-V_{L0} E'_0 \cos \delta_0}{MX'} & 0 \end{bmatrix} \quad 11$$

$$B = \begin{bmatrix} \frac{(X-X') \cos \delta_0}{T_0 X'} \\ \frac{-(X-X') \sin \delta_0}{T_0 X' E'_0} \\ \frac{-E'_0 \sin \delta_0}{M X'} \end{bmatrix} \quad 12$$

$$C = \begin{bmatrix} \frac{-V_{L0} \sin \delta_0}{X'} & \frac{-V_{L0} E'_0 \cos \delta_0}{X'} & 0 \\ \frac{-V_{L0} \cos \delta_0}{X'} & \frac{V_{L0} E'_0 \sin \delta_0}{X'} & 0 \end{bmatrix} \quad 13$$

$$D = \begin{bmatrix} 2G_s V_{L0} - \frac{E'_0 \sin \delta_0}{X'} \\ 2B_s V_{L0} + \frac{2V_{L0} - E'_0 \cos \delta_0}{X'} \end{bmatrix} \quad 14$$

Com isso, o vetor de parâmetros passa a ser dado por  $p = [M, T'_0, X, X', G_s, B_s, E'_0, \delta_0, V_{L0}]$ , onde  $E'_0$ ,  $\delta_0$  e  $V_{L0}$  são, respectivamente, os valores iniciais da magnitude e ângulo da tensão e da tensão de linha.

## 2.4. Método de Sensibilidade de Trajetória

O método de estimação utilizado nesse trabalho identifica os valores dos parâmetros por meio do método de sensibilidade de trajetória. O algoritmo, desenvolvido em linguagem Python, funciona de acordo com o fluxograma mostrado na Figura 3.

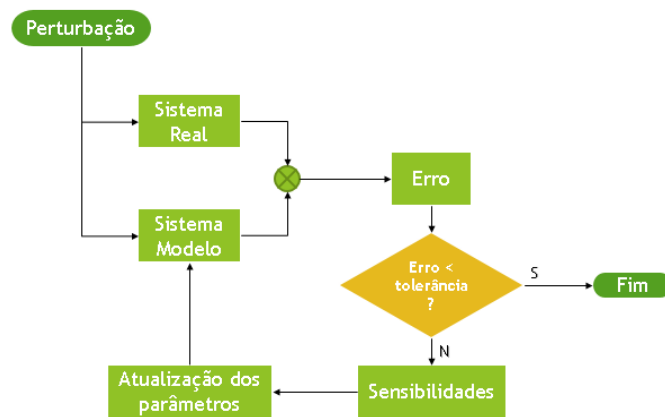


Figura 3 - Fluxograma do método original.

A cada iteração, as saídas do sistema simulado são comparadas com os valores reais e, caso o erro seja maior do que a tolerância, os parâmetros do sistema são atualizados. Esse processo ocorre até que a tolerância seja atingida, indicando que os valores dos parâmetros obtidos correspondem ao real. O erro ( $J_p$ ) é obtido por meio do método dos mínimos quadrados, e é dado pela equação 15 abaixo, onde  $y_r$  e  $y_m$  são as saídas dos sistemas real e modelado, respectivamente (Cari, 2009).

$$J_p = \frac{1}{2} \int_0^{T_0} (y_r(t) - y_m(t))^T (y_r(t) - y_m(t)) dt \quad 15$$

Como o objetivo é minimizar o erro, deve-se encontrar o vetor de parâmetro,  $p^*$ , tal que se obtenha:

$$G(p^*) = \frac{\partial J(p^*)}{\partial p} = 0 \quad 16$$

Ao expandir a função  $G(p)$  e truncando no termo de primeira ordem, obtém-se a equação 17, onde  $\Gamma(p)$  é a matriz Hessiana da função, expressa por 18.

$$G(p^*) = G(p) + \Gamma(p)(p^* - p) \quad 17$$

$$\Gamma(p) = \frac{\partial G}{\partial p} \approx \int_0^{T_0} \left( \frac{\partial y}{\partial p} \right)^T \left( \frac{\partial y}{\partial p} \right) dt \Big|_{p=p(k)} \quad 18$$

A fim de facilitar o cálculo, pode-se aproximar as derivadas pela diferença das saídas do sistema dividido pela variação dos parâmetros, como é mostrado na equação, onde  $y^1(t)$  é o vetor de saída do sistema para uma matriz de parâmetros  $p_i^1$ ,  $y^0(t)$ , as saídas para  $p_i^0$ , e  $p_i^1 = p_i^0 + \Delta p_i$  (Farias, Cari, & Alberto, 2016).

$$\frac{\partial y}{\partial p_i} \approx \frac{y^1(t) - y^0(t)}{\Delta p_i} \quad 19$$

Dessa forma, rearranjando os fatores em 17 e aplicando em 16, obtém-se a equação 20, que resulta nos novos valores dos parâmetros após a  $n$ -ésima iteração.

$$p^{(n+1)} = p^n - \Gamma^{-1}(p)G(p)|_{p=p(k)} \quad 20$$

O algoritmo implementado utiliza-se de um método iterativo baseado em Newton-Raphson, o que torna a convergência rápida, porém dependente um bom “chute inicial” para os valores dos parâmetros. Por conta disso, sua região de convergência fica restrita a uma pequena área em torno do ponto real.

## 2.5. Estimação em Duas Etapas

Uma maneira encontrada para contornar o problema da região convergência foi a estimação dos parâmetros em duas partes. Esse método consiste em classificar os parâmetros do sistema em bem e mal condicionados e estimá-los em duas etapas. Primeiramente, são estimados os parâmetros mal condicionados, que mais influenciam no comportamento das saídas, mantendo o restante com valor fixo. Em seguida, todos os parâmetros são estimados juntos até que se obtenha um erro menor do que a tolerância pré-estabelecida.

A classificação dos parâmetros ficou a cargo de um algoritmo que se utiliza da matriz Hessiana,  $\Gamma(p)$ , e de sua decomposição em autovalores e autovetores (Cari, 2009), (Vélez-Reyes & Verghese, 1995), e segue os seguintes passos:

1. Dada a matriz  $\Gamma(p)$ , é feita sua decomposição em autovalores e autovetores, dada por 21.

$$\Gamma(p) = V\lambda V^{-1} \quad 21$$

2. Em seguida, é estabelecido um valor  $\delta$  que servirá de critério para classificar os autovalores. A matriz  $V$  é então reordenada conforme essa classificação e particionada e  $V = [V_1 \ V_2]$ , onde  $V_1$  corresponde aos  $p$  autovalores maiores que  $\delta$  e  $V_2$  aos  $n-p$  autovalores menores que  $\delta$ .
3. Então, calcula-se a matriz de permutação,  $P$ , por meio da fatoração QR da matriz  $V_1^T$ , conforme a equação 22.

$$V_1^T P = QR \quad 22$$

4. Por fim, os parâmetros são reordenados de acordo com seu condicionamento seguindo a equação  $p = P^T \tilde{p}$ .



### 3. MATERIAIS E MÉTODOS

Todos os algoritmos empregados no cálculo e resolução do problema de identificação dos parâmetros do modelo de carga foram desenvolvidos em linguagem Python, utilizando para tal a plataforma Python(x,y). Além disso, foi criada uma interface gráfica em mesma linguagem, a fim de facilitar a experiência do usuário e, futuramente, comercializar o software. Esse capítulo foca no uso da linguagem Python na construção das rotinas de cálculo e da GUI (*Graphical User Interface*).

#### 3.1. Python

Criado em 1991, Python é uma linguagem de alto nível orientada a objetos de código aberto e, por essas e outras características, tem sido amplamente utilizada em empresas e instituições de pesquisa. Além disso, a linguagem apresenta uma comunidade vasta e cooperativa, o que facilita o aprendizado e o desenvolvimento de projetos, visto que muitas das dúvidas que surgem durante a programação podem ser sanadas pelos fóruns e tutoriais, disponíveis de forma gratuita. Outra vantagem é a existência de diversas bibliotecas, cada uma voltada para um fim diferente, permitindo ao desenvolvedor baixar e usar apenas as bibliotecas necessárias para o funcionamento de seu programa. Isso torna a aplicação desenvolvida leve e eficaz, diferente dos *softwares* comerciais atuais, que, por serem robustos, se tornam extremamente pesados.

Apesar da linguagem já estar disponível na versão 3.6.x, neste trabalho foi utilizada a versão 2.7.x, por já ter sido mais trabalhada pela comunidade, apresentando mais soluções em fóruns, e ser compatível com todos os sistemas operacionais atuais. As principais bibliotecas utilizadas foram:

- NumPy – responsável por cálculos de álgebra linear e operações com matrizes n-dimensionais;
- Matplotlib – voltada para a criação de gráficos 2D e 3D; e
- Tkinter – biblioteca de criação de GUI's.

As funções desenvolvidas durante este trabalho foram escritas de forma genérica, podendo ser aplicadas a outros problemas de estimação de parâmetros, necessitando apenas de pequenas adaptações. Dessa forma, outros projetos podem se utilizar das rotinas criadas. Um código-exemplo, utilizado para calcular os parâmetros de um sistema massa-mola, pode ser encontrado em <http://redmine2.lacosep.sel.eesc.usp.br/redmine/documents/69>

### 3.2. Interface Gráfica

A interface gráfica criada é voltada especificamente para este trabalho, mas pode ser facilmente adaptada para outras aplicações, caso seja necessário. A GUI é iniciada em uma página inicial com apenas um botão, como pode ser visto na figura 4. Essa página foi criada dessa forma para que, futuramente, sejam colocadas nela informações sobre o *software*. Além disso, foram criados dois menus no topo da janela. No menu “*Input Value*” pode-se alterar o valor inicial da entrada,  $u_0$ , e sua variação,  $\Delta u$ . Já na opção “*Settings*” possibilita o usuário a modificar algumas configurações do programa como a variação dos parâmetros,  $\Delta p$ , utilizada no cálculo das sensibilidades, os instantes inicial,  $t_0$ , e final,  $t_f$ , e o passo (*step*) de simulação e a tolerância ao erro (*tolerance*). Os dois menus com suas respectivas opções são mostrados nas Figura 5 e Figura 6.

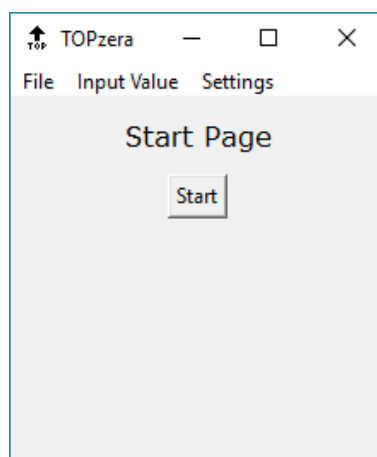


Figura 4 - Página inicial do software

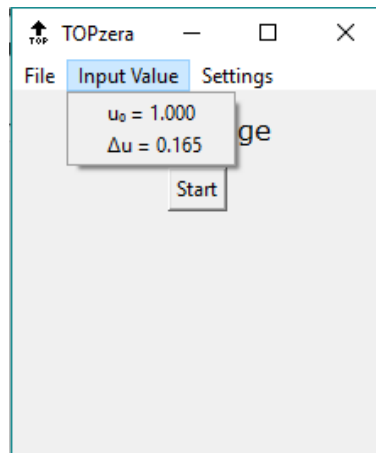


Figura 5 - Menu "Input Value"

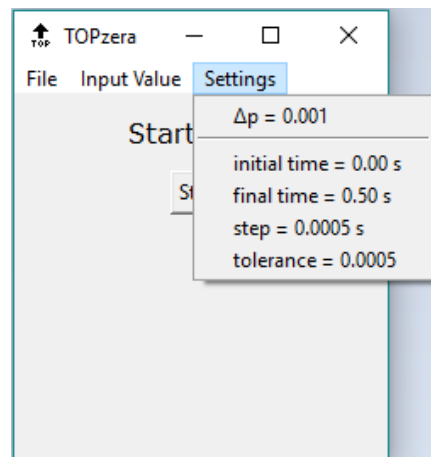


Figura 6 - Menu "Settings"

Após clicar em “Start”, o usuário é levado à página apresentada na Figura 7, onde ele poderá inserir os valores iniciais dos parâmetros do modelo de carga. Em seguida, é exibida a página de *Flags*, mostrada na Figura 8, onde pode-se escolher se a simulação será feita pausadamente, mostrando o resultado de cada iteração (opção “*Step by Step*”) ou apenas mostrando o resultado final, se os gráficos serão ou não exibidos (opção “*Print Figures*”) e se a identificação dos parâmetros será feita em duas etapas, conforme explicado na seção 2.5 (opção “*2-Step Ident.*”).

TOPzera

File Input Value Settings

Parameters

X	0.21
X'	0.05
T <sub>0</sub>	0.1
M	0.01
G <sub>s</sub>	4
B <sub>s</sub>	3
E <sub>0</sub>	1.1
δ <sub>0</sub>	-0.37

Back to Home Next

Figura 7 - Página de valores iniciais

TOPzera

File Input Value Settings

Flags

☐ Step by Step

☐ Print Figures

☐ 2-Step Ident.

Back

Next

Figura 8 - Flags

Ao clicar em “Next” é iniciada a estimação dos parâmetros e, caso as opções “Step by Step” e “Print Figures” tenham sido selecionadas, será exibido os gráficos sobrepostos dos comportamentos real e modelado das saídas do sistema, como pode ser visto na Figura 9. Ao clicar no botão “Iterate”, os parâmetros são recalculados e o novo comportamento do sistema modelado é exibido até que o erro entre as saídas seja menor que a tolerância. Quando isso ocorre o botão “Iterate” é desabilitado, como mostrado na Figura 10. É importante salientar que as opções dos menus “Input Value” e “Setting” ficam desabilitadas durante o processo de estimação, impedindo que esses parâmetros sejam modificados entre uma iteração e outra, o que acarretaria em um erro na estimação.

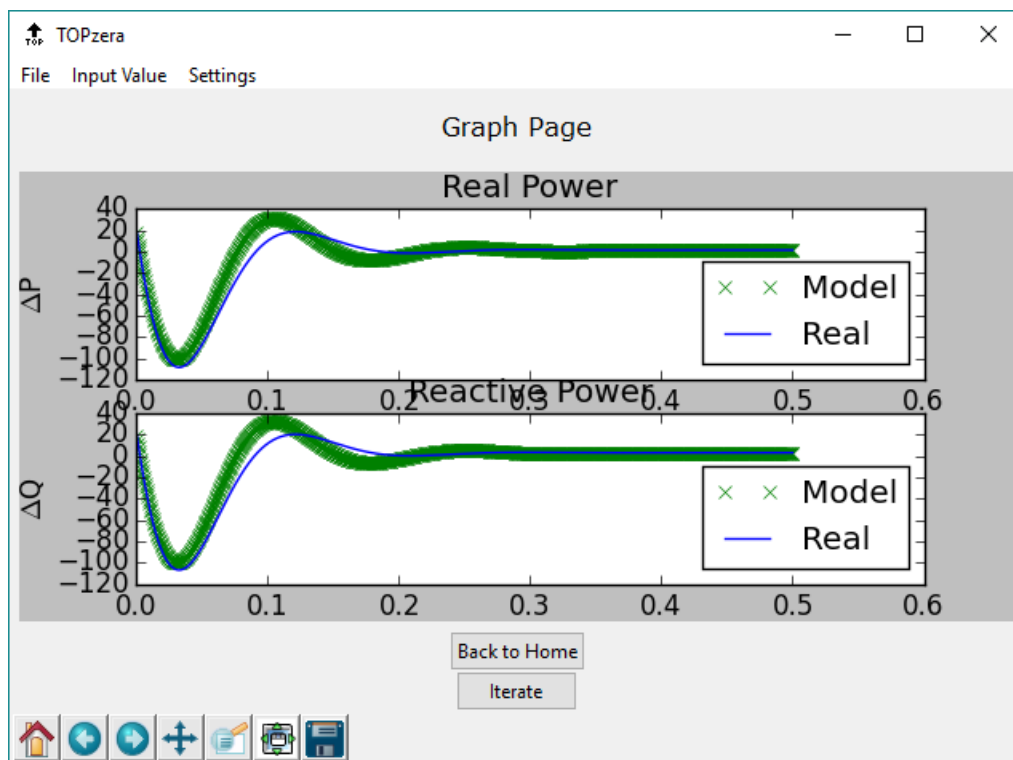


Figura 9 - Comportamento das saídas anterior à estimação

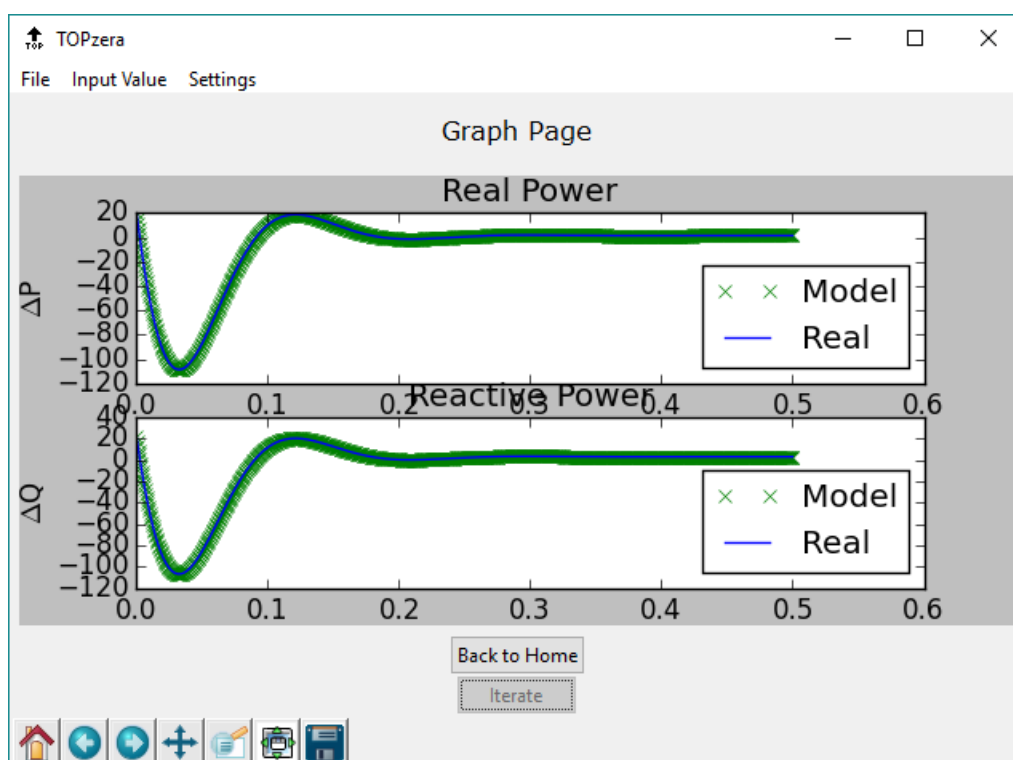


Figura 10 - Comportamento das saídas após a convergência



## 4. RESULTADOS

Nesta seção são apresentados os resultados obtidos na estimação dos parâmetros do modelo de carga Z-IM. Também é exposta a classificação dos parâmetros em relação ao condicionamento destes, obtida por meio do algoritmo descrito em 2.5. Por fim, é apresentada a faixa de valores de cada parâmetros que garante a convergência.

Os comportamentos das saídas do sistema foram calculados utilizando o Método de Runge-Kutta de 4ª Ordem. Para simular o sistema real, os valores de parâmetros utilizados ( $p_{real}$ ) foram retirados de (Choi, et al. 2006). Os valores escolhidos como “chute inicial” dos parâmetros são apresentados na matriz  $p$  e são 20% maiores do que os valores reais.

$$p_{real} = \begin{bmatrix} X_{real} \\ X'_{real} \\ T_{oreal} \\ M_{real} \\ G_{sreal} \\ B_{sreal} \\ E'_{0real} \\ \delta_{0real} \end{bmatrix} = \begin{bmatrix} 0,2089 \\ 0,0446 \\ 0,0963 \\ 0,0139 \\ 4,1358 \\ 2,8004 \\ 1,0750 \\ -0,3689 \end{bmatrix} \quad 23$$

$$p = \begin{bmatrix} X \\ X' \\ T_o \\ M \\ G_s \\ B_s \\ E'_0 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0,25068 \\ 0,05352 \\ 0,11556 \\ 0,01668 \\ 4,96296 \\ 3,36048 \\ 1,29 \\ -0,44268 \end{bmatrix} \quad 24$$

O valor da tensão de linha pré-perturbação,  $V_{L0}$  nas equações linearizadas do modelo, foi adotado como sendo de 1 pu e o valor de  $\Delta V_L$  escolhido foi de 0,0165 pu, variação típica para a movimentação de *tap* de transformador. O valor da tolerância foi de  $5,0 \times 10^{-4}$  e o tempo total simulado foi de 0,5 segundos.

Para os valores iniciais escolhidos para os parâmetros, o método convergiu em 5 iterações, resultando nos valores de parâmetros e erros em cada iteração apresentados na Tabela 2. O comportamento das saídas ao início e ao fim do processo de estimação são ilustrados nas Figura 11 e Figura 12.

Tabela 2 - Evolução dos parâmetros e erro

Iter.	X	X'	T <sub>o</sub>	M	G <sub>s</sub>	B <sub>s</sub>	E' <sub>o</sub>	δ <sub>o</sub>	Erro
0	0,25068	0,05352	0,11556	0,01668	4,96296	3,36048	1,29	-0,44268	59,14
1	0,30006	0,05425	0,11160	0,01402	6,67550	-1,17954	1,23826	-0,39143	1,20
2	0,15499	0,04313	0,08093	0,01392	4,65065	2,13598	1,04980	-0,37014	4,98
3	0,19938	0,04461	0,09603	0,01381	2,10683	2,48289	1,07074	-0,36748	0,42
4	0,20980	0,04463	0,09709	0,01389	3,88913	2,72784	1,07489	-0,36870	1,24x10 <sup>-3</sup>
5	0,20893	0,04460	0,09631	0,013900	4,12666	2,79303	1,07503	-0,36890	5,68x10 <sup>-7</sup>
Real	0,2089	0,0446	0,0963	0,0139	4,1358	2,8004	1,0750	-0,3689	-

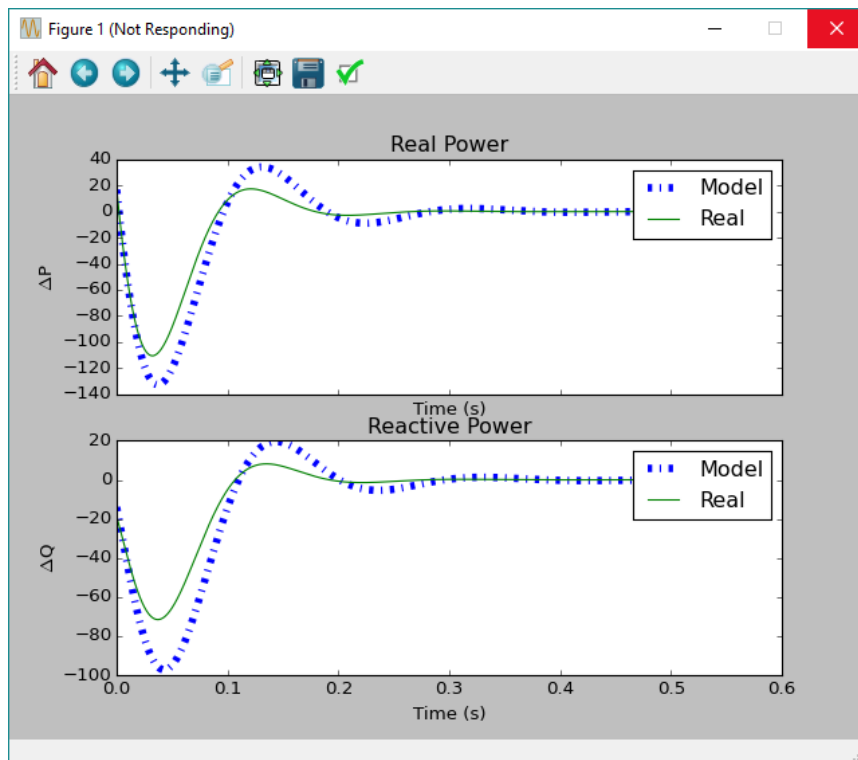


Figura 11 - Saídas reais e simuladas antes da estimação



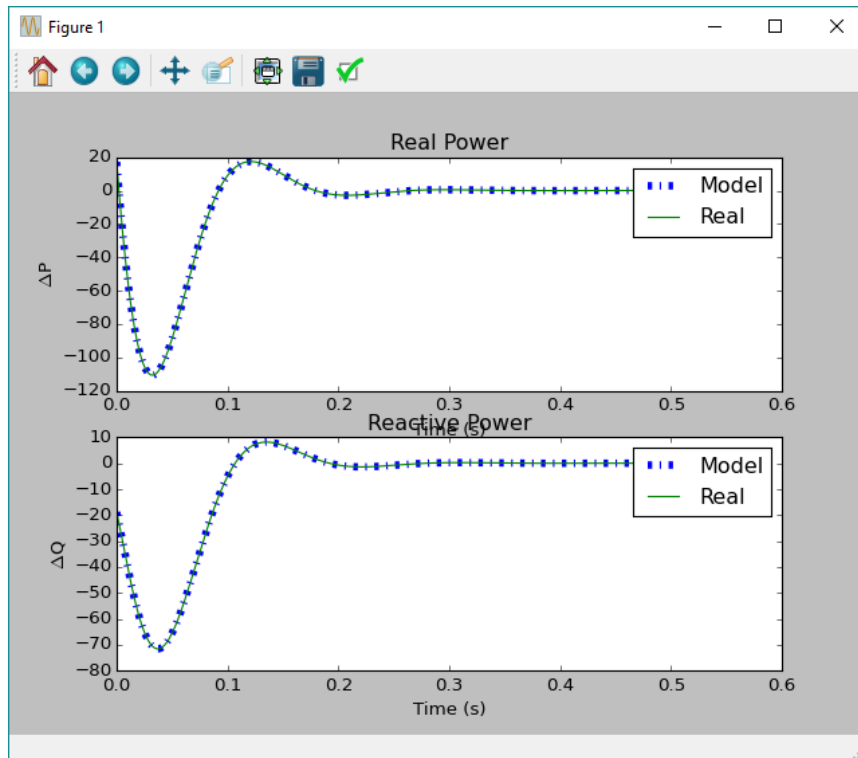


Figura 12 - Saídas reais e simuladas após da estimação

O condicionamento dos parâmetros, obtido pelo algoritmo descrito em 2.5, pode ser observado na Figura 13, abaixo. O resultado das classificações aponta o parâmetro “M” como sendo o de pior condicionamento, seguido por  $\delta_0$ ,  $X'$ ,  $T_0$ ,  $X$ ,  $B_s$ ,  $E'_0$  e, por fim,  $G_s$  como o melhor condicionado.

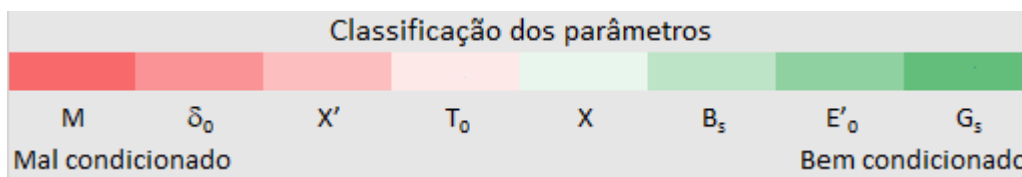


Figura 13 - Condicionamento dos parâmetros

A faixa de incerteza de cada parâmetro foi obtida por meio de sucessivas simulações e o resultado obtido pode ser visto na Figura 14. Nela, cada barra corresponde aos valores que podem ser dados como “chute inicial” para cada parâmetro em que ainda ocorre a convergência do método, sendo 100% o valor real do parâmetro. A primeira barra corresponde ao teste variando todos os parâmetros juntos e em mesma proporção, enquanto as demais barras correspondem ao teste variando apenas o respectivo parâmetro e mantendo o restante igual a seus valores reais.

Para o primeiro caso descrito, a faixa de incerteza obtida foi próxima a 40%, indo de 80% a 124% do valor real dos parâmetros. Para o teste com cada parâmetro por vez, vale ressaltar o fato de os parâmetros  $G_s$  e  $B_s$  apresentarem uma grande faixa de incerteza, convergindo mesmo para valores 10000 vezes maior que o real. Outro resultado a ser levado em conta é a incerteza do parâmetro  $E'_0$ , sendo a menor entre os parâmetros, com seu valor podendo variar de 83% a 124% do valor real.

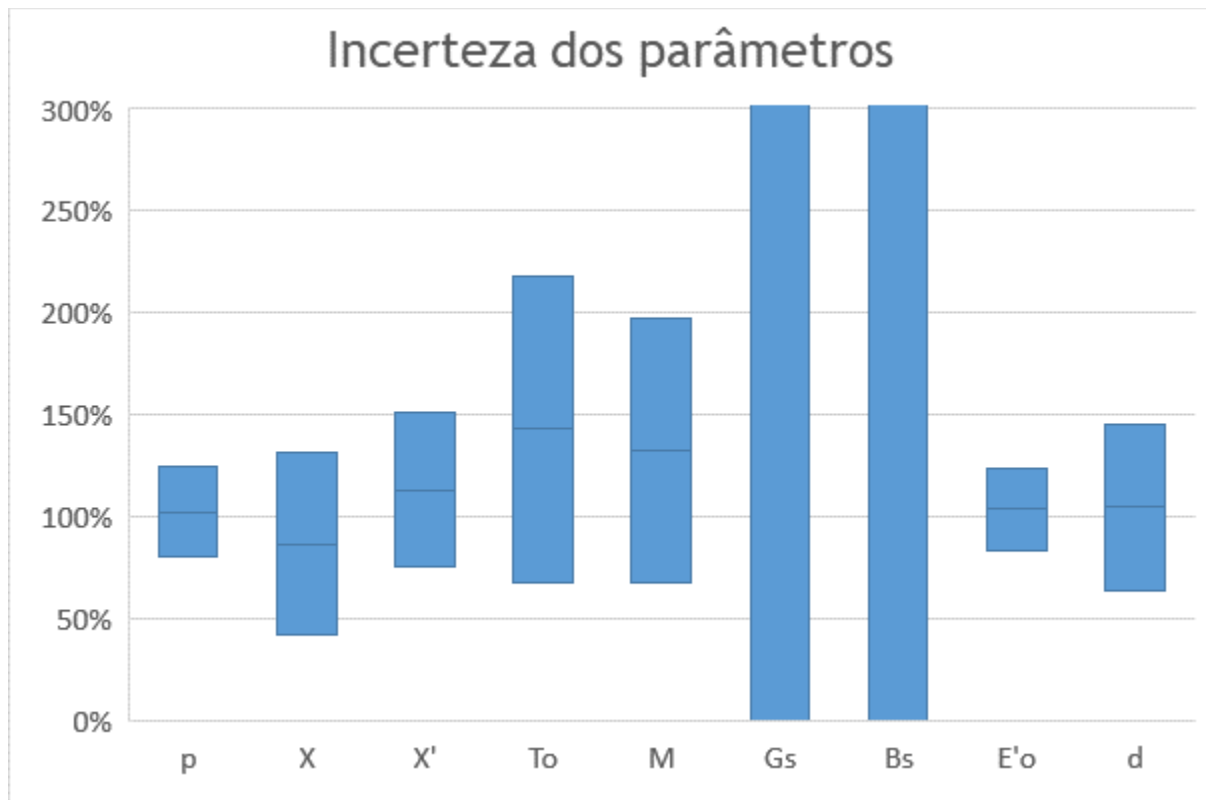


Figura 14 - Faixa de incerteza dos parâmetros

## 5. CONCLUSÕES

Neste trabalho foi utilizado um algoritmo capaz de estimar os parâmetros de um modelo de carga do tipo Z-IM a partir de medidas obtidas após a atuação de *taps* de transformadores. As vantagens de se utilizar essas variações é o fato de serem frequentes, acarretando em modelos sempre atuais, e de não apresentarem riscos à estabilidade da rede, já que elas ocorrem para manter o nível de tensão sempre próximo ao nominal. Para a correta identificação de parâmetros, também foi necessário linearizar o modelo de carga, visto que modelos não lineares se tornam muito complexos e apresentam sérios problemas de convergência.

A partir dos dados obtidos durante as mudanças de *tap*, o programa desenvolvido, baseado em sensibilidade de trajetória, é capaz de identificar os parâmetros do modelo Z-IM. O processo de estimação se mostrou extremamente rápido, convergindo em questão de segundos em um processador Core i7. O método se mostrou eficaz para valores iniciais até 20% maiores do que os reais, entretanto para valores maiores a convergência se torna incerta.

Todos os códigos criados para esse trabalho foram desenvolvidos em Python. Essa linguagem de alto nível se mostra interessante para o uso acadêmico devido ao fato de ser *open source*, com uma comunidade de usuários altamente colaborativa e com diversas bibliotecas voltadas para fins específicos. As rotinas elaboradas foram desenvolvidas de forma genérica, podendo todo o código ser reutilizado para outros modelos matemáticos, sendo necessário apenas o ajuste das matrizes que descrevem o modelo, do vetor de parâmetros e das entradas.

Além disso, foi desenvolvida uma interface gráfica para tornar a experiência do usuário mais fácil e intuitiva. A interface permite com que sejam modificadas configurações do método de estimação sem que seja necessário um grande conhecimento do usuário em Python. Dessa forma, o *software* acaba se tornando mais amigável e pode ser futuramente comercializado como uma solução para estudos de estabilidade e operação.

Um pequeno exemplo do algoritmo utilizado está disponível para acesso no seguinte link: <http://redmine2.lacosep.sel.eesc.usp.br/redmine/documents/69>

O estudo desenvolvido neste trabalho tem grande impacto no aumento da robustez e confiabilidade dos sistemas elétricos de potência atuais, visto que reduz a ocorrência de faltas ocasionadas por modelos desatualizados. Esses modelos poderão ser atualizados diversas

vezes ao dia, conforme ocorrem mudanças de tap em transformadores da rede, tornando as simulações mais precisas e auxiliando na operação do sistema.

## 6. TRABALHOS FUTUROS

Como perspectiva futura de continuidade deste trabalho destaca-se o estudo e implementação de métodos auxiliares a fim de expandir a região de convergência, reduzindo a influência dos valores iniciais escolhidos no resultado final do algoritmo. Além disso, serão utilizadas medidas reais de subestações para a validação do método e o *software* será adaptado para a estimação de parâmetros de outros modelos matemáticos.



## 7. ARTIGOS PUBLICADOS

CICTE 2017 – Congresso de Iniciação Científica e Tecnológica em Engenharia

SIICUSP 2017 – Simpósio Internacional de Iniciação Científica da USP

SBSE 2018 – Simpósio Brasileiro de Sistemas Elétricos (Em análise)





## 8. REFERÊNCIAS

- Arefifar, S. A., & Xu, W. (2013). Online Tracking of Voltage-Dependent Load Parameters Using ULTC Created Disturbances. *IEEE Transactions On Power Systems*, 28, 130-139.
- Cari, E. P. (2009). Metodologia de Estimação de Parâmetros de Sistemas Dinâmicos Não-Lineares com Aplicação em Geradores Síncronos. São Carlos: Tese (Doutorado em Sistemas Elétricos de Potência) - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- Choi, B.-K., Chiang, H.-D., Li, Y., Chen, Y.-T., Huang, D.-H., & Lauby, M. G. (2006). Development of Composite Load Models of Power Systems using On-line Measurement Data. *Power Engineering Society General Meeting*. Montreal.
- Farias, É. R., Cari, E. P., & Alberto, L. F. (2016). Método de Sensibilidade de Trajetória Aplicada à Estimação de Parâmetros de Sistemas Dinâmicos Não-Lineares. *CBA*.
- Guo, H., Rudion, K., Abildgaard, H., Komarnicki, P., & Styczynski, Z. A. (2012). Parameter Estimation of Dynamic Load Model Using Field Measurement Data Performed by OLTC Operation.
- Hauer, J. F., & Dagle, J. E. (1999). *Review of Recent Reliability Issues and Systems Events*. CERTS.
- Henriques, R. M., Martins, N., Ferraz, J. C., Martins, A. C., Pinto, H. C., & Carneiro Jr., S. (2002). Impact of Induction Motor Loads into Voltage Stability Margins of Large Systems - Rio Area Analysis. *VIII Simpósio de Especialistas em Planejamento da Operação e Expansão Elétrica*. Brasília.
- IEEE Task Force on Load Representation for Dynamic Performance. (1995, August). Standard Load Models for Power Flow and Dynamic Performance Simulation. *IEEE Transactions on Power Systems*, pp. 1302-13.

Vélez-Reyes, M., & Verghese, G. C. (1995). Subset Selection in Identification, and Application to Speed and Parameter Estimation for Induction Machines. *Proceedings of International Conference on Control Applications*. Albany.

Xu, W., Vaahedi, E., Mansour, Y., & Tamby, J. (1997). Voltage Stability Load Parameter Determination from Field Tests on B.C. Hydro's System. *IEEE Transactions on Power Systems*, 12.