

## Simulação de Transitórios Eletromecânicos e Análise da Estabilidade a Pequenas Perturbações através de ferramentas desenvolvidas em Python

Jonathan V. França\* Marley F. Tavares\*\*  
Murilo E. C. Bento\*\*\* José Gutemberg do Nascimento Sousa\*\*\*\*

\**Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Paulo Afonso, Brasil, (e-mail: jyf1996@hotmail.com).*

\*\**Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Paulo Afonso, Brasil, (e-mail: marley.tavares@ifba.edu.br).*

\*\*\**Departamento de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, (e-mail: murilobento@poli.ufrj.br).*

\*\*\*\**Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Paulo Afonso, Brasil, (e-mail: gutemberguenascimento@yahoo.com.br).*

---

**Abstract:** Computer simulations are an important tool for the expansion and operation of electrical systems. Within the simulations carried out for the study of electrical systems, there is the simulation of electromechanical transients, and this type of simulation is extremely important for the study of stability in electrical systems. This work aims to show the development of two electrical systems simulation codes written in Python language, one dedicated to time simulation of electromechanical transients and the other to the analysis of angular stability at small disturbances. These codes were applied in the simulation of a 14-bus system, presenting results that converge with traditional software.

**Resumo:** As simulações computacionais são uma importante ferramenta para a expansão e operação de sistemas elétricos. Dentro das simulações realizadas para o estudo de sistemas elétricos, existe a simulação de transitórios eletromecânicos, sendo este tipo de simulação de extrema importância para o estudo da estabilidade em sistemas elétricos. Este trabalho tem como objetivo mostrar o desenvolvimento de dois códigos de simulação de sistemas elétricos escritos na linguagem Python, um dedicado a simulação no tempo de transitórios eletromecânicos e outro a análise da estabilidade angular a pequenas perturbações. Estes códigos foram aplicados na simulação de um sistema de 14 barras, apresentando resultados convergentes com softwares tradicionais.

**Keywords:** Electromechanical Transients; Angular Stability; Small Disturbances; Computational Simulations; Python.

**Palavras-chaves:** Transitórios Eletromecânicos; Estabilidade Angular; Pequenas Perturbações; Simulações Computacionais; Python.

---

### 1. INTRODUÇÃO

As simulações computacionais de sistemas elétricos de potência são uma importante ferramenta para a sua expansão e operação. As simulações desenvolvidas variam em complexidade de acordo com as escalas de tempo observadas: enquanto os estudos de regime permanente são realizados com a resolução numérica de equações algébricas, os de regime transitório são mais complexos, exigindo a solução de equações algébricas e diferenciais.

Entre as simulações realizadas com frequência em sistemas elétricos, existe a de transitórios eletromecânicos. A partir desse tipo de estudo, é observada como eventos nos geradores e na rede de transmissão do sistema afetam o ângulo de carga, a frequência o valor eficaz das tensões e correntes e diversos outros parâmetros do sistema. Uma das principais aplicações das simulações de transitórios eletromecânicos em sistemas elétricos de potência é o estudo da estabilidade.

Segundo Kundur (1994), a estabilidade de um sistema elétrico é a capacidade desse sistema em permanecer em equilíbrio em regime de operação normal e de atingir um novo estado de equilíbrio após uma perturbação. Tal problema é subdividido em três grandes áreas: estabilidade de tensão, estabilidade de frequência e estabilidade angular, sendo esta última alvo de estudo no presente trabalho.

O problema da estabilidade angular se separa em dois, a estabilidade a pequenas perturbações e a estabilidade a grandes perturbações (também chamada de estabilidade transitória). A estabilidade a pequenas perturbações reflete a estabilidade ao redor do ponto de operação do sistema, dessa forma, o estudo da estabilidade a pequenas perturbações é feito linearizando as equações dinâmicas do sistema e aplicando as técnicas de análise de estabilidade de sistemas lineares. A estabilidade transitória, por outro lado é estudada através de simulações no tempo considerando-se as não linearidades inerentes aos modelos do sistema.

Diversos programas proprietários existem com o propósito de estudar a resposta dinâmica de sistemas elétricos, esses programas possuem suas rotinas inacessíveis ao usuário. Em contrapartida, o uso de softwares e bibliotecas de linguagens de programação com licença livre em diversas áreas tem aumentado com o passar do tempo, inclusive o estudo de sistemas elétricos, como o PyPSA (Brown et al., 2018) e o pandapower (Thurner et al., 2018).

Uma das linguagens de programação com uma grande comunidade de usuários é o Python (Python, n.d.). A linguagem possui diversas bibliotecas de código livre para computação científica e análise e visualização de dados. Por ser uma linguagem com uma ampla rede de usuários, e por ter uma sintaxe simples, o aprendizado da linguagem é bastante simples se comparado a linguagens tradicionais.

Este artigo propõe mostrar os resultados da utilização de bibliotecas *open-source* da linguagem Python para a criação de ferramentas de simulação no tempo de transitórios eletromecânicos e análise da estabilidade a pequenas perturbações em sistemas elétricos de potência.

## 2. MODELOS MATEMÁTICOS EMPREGADOS

Esta seção dedica-se a expor os modelos matemáticos empregados nos códigos desenvolvidos. Os códigos desenvolvidos são utilizados para simulação de sistemas com geradores síncronos interligados por uma rede de transmissão em corrente alternada e cargas. A modelagem de cada um desses elementos será exposta a seguir.

### 2.1 Modelagem da máquina síncrona

A modelagem da máquina síncrona se separa em dois conjuntos de equações diferenciais: um relacionado ao movimento do rotor da máquina, e outro relacionado a variação de fluxo em seus enrolamentos. O desenvolvimento das equações relativas ao movimento do rotor da máquina síncrona é feito através da aplicação da segunda lei de Newton para rotações. Uma das possíveis formas de descrever o comportamento mecânico de uma máquina síncrona é através de (1) e (2) (Anderson & Fouad, 2003).

$$2H \frac{d\omega_{r(p.u.)}}{dt} = P_{m(p.u.)} - P_{e(p.u.)} \quad (1)$$

$$\frac{d\delta}{dt} = (\omega_{r(p.u.)} - 1)\omega_s \quad (2)$$

Em que  $\delta$ ,  $\omega_s$ ,  $\omega_{r(p.u.)}$ ,  $P_{m(p.u.)}$  e  $P_{e(p.u.)}$  são, respectivamente, o ângulo entre o rotor e a referência síncrona girante, em radianos elétricos, a velocidade da referência síncrona, em radianos elétricos por segundo, a velocidade do rotor, a potência mecânica e a potência elétrica em valores por unidade. A constante  $H$  é chamada de constante de inércia, e é a relação entre a energia cinética armazenada no conjunto girante e a potência base do gerador.

Para a modelagem das equações referentes a variação de fluxo no interior da máquina o modelo adotado foi o da tensão atrás da reatância subtransitória (Anderson & Fouad, 2003). Este modelo considera que a velocidade do rotor permanece próxima da velocidade síncrona, desconsidera a saliência subtransitória ( $X'_d = X''_d$ ) e a variação de fluxo nos enrolamentos do estator.

Para as condições de regime permanente, as equações elétricas de uma máquina síncrona podem ser escritas considerando o diagrama fasorial apresentado pela Figura 1. As Equações 3 e 4 relacionam as tensões e correntes em eixo direto e em quadratura em regime permanente.

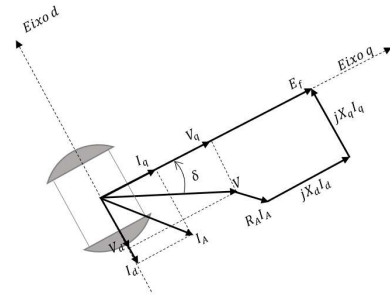


Fig. 1: Diagrama fasorial em regime permanente de uma máquina síncrona.

$$E_f - V_q = R_A I_q - X_d I_d \quad (3)$$

$$-V_d = R_A I_d + X_q I_q \quad (4)$$

Em regime dinâmico, a tensão e a corrente das máquinas síncronas variam de acordo com as forças eletromotrizes transitórias ( $E'_q$  e  $E'_d$ ) e subtransitórias ( $E''_q$  e  $E''_d$ ) em seus eixos direto e em quadratura. As equações que relacionam as fem's transitórias e sub-transitórias com as tensões, correntes, reatâncias transitórias ( $X'_d$  e  $X'_q$ ), subtransitórias ( $X''_d$  e  $X''_q$ ) e resistência de armadura ( $R_A$ ) podem ser vistas de (5) a (8).

$$E'_d - V_d = R_A I_d + X'_q I_q \quad (5)$$

$$E'_q - V_q = R_A I_q - X'_d I_d \quad (6)$$

$$E''_d - V_d = R_A I_d + X''_q I_q \quad (7)$$

$$E''_q - V_q = R_A I_q - X''_d I_d \quad (8)$$

Simplificações podem ser feitas considerando  $X'_d = X''_d$  (desprezando a saliência subtransitória),  $X'_q = X_q$  e  $E'_d = 0$  (desprezando os efeitos das correntes induzidas no núcleo magnético do rotor). Tal modelo, adotado neste trabalho, mostra-se aceitável na modelagem de máquinas de polos

salientes com dois enrolamentos amortecedores, um no eixo direto e outro no eixo em quadratura.

Para este caso, as forças eletromotrizes transitórias e subtransitórias variam de acordo com as equações diferenciais mostradas de (9) a (11). As constantes  $T'_{d0}$ ,  $T''_{d0}$  e  $T''_{q0}$  são definidas como constantes de tempo transitórias e subtransitórias em circuito aberto para os eixos direto e em quadratura.

$$\frac{dE'_q}{dt} = \frac{1}{T'_{d0}} \left( E_f + \frac{(X_d - X'_d)}{(X'_d - X_l)} E''_q - \frac{(X_d - X_l)}{(X'_d - X_l)} E'_q + \frac{(X_d - X'_d)(X'_d - X_l)}{(X'_d - X_l)} I_d \right) \quad (9)$$

$$\frac{dE''_q}{dt} = \frac{1}{T''_{d0}} (E'_q - E''_q + (X'_d - X''_d) I_d) + \frac{(X'_d - X_l)}{(X'_d - X_l)} \frac{dE'_q}{dt} \quad (10)$$

$$\frac{dE''_d}{dt} = \frac{1}{T''_{q0}} (-(X'_q - X''_q) I_q - E''_d) \quad (11)$$

As Equações (9) a (11) junto com as equações (1) e (2) formam as equações de estado da máquina síncrona de polos salientes. O valor da potência elétrica  $P_e$  na Equação (1) pode ser calculada como mostra (12).

$$P_e = E''_d I_d + E''_q I_q \quad (12)$$

Como as equações desenvolvidas possuem suas variáveis refletidas ao referencial síncrono (eixo direto e em quadratura), faz-se necessário transformar tais variáveis ao referencial do sistema. Tal transformação é descrita pela Equação (13) e ilustrada pela Figura 2. Como  $T^{-1} = T$ , a transformação inversa é idêntica a definida em (19).

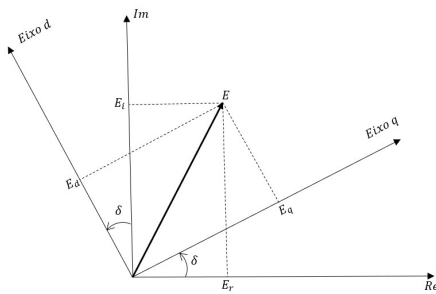


Fig. 2. Transformação da referência síncrona para a referência do sistema.

$$\begin{bmatrix} E_d \\ E_q \end{bmatrix} = \begin{bmatrix} -\sin \delta & \cos \delta \\ \cos \delta & \sin \delta \end{bmatrix} \begin{bmatrix} E_r \\ E_i \end{bmatrix} \quad (13)$$

Cada máquina síncrona modelada através das equações (1), (2), (9), (10) e (11) acrescenta cinco equações diferenciais ao sistema dinâmico a ser solucionado numericamente. Além das equações diferenciais referentes a dinâmica da máquina síncrona, existe também as equações referentes aos sistemas de controle, responsáveis pela regulação e estabilização da máquina.

## 2.2 Modelagem dos sistemas de controle da máquina síncrona

Os sistemas de controle são essenciais para a operação das máquinas síncronas do sistema. Entre os sistemas de controle utilizados em máquinas síncronas três são principais: O regulador de velocidade (RV), o regulador de tensão (AVR) e o estabilizador de sistemas de potência (PSS). A Figura 3 ilustra o fluxo de sinais entre os controladores e a máquina síncrona.

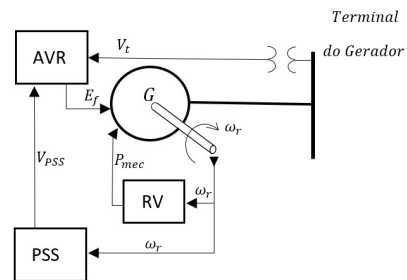


Fig. 3. Fluxo de sinais entre controladores e máquina síncrona.

O regulador de velocidade possui como finalidade controlar, por meio de distribuidores ou válvulas, a potência advinda da fonte primária, de forma a manter constante a velocidade angular do gerador. O estudo desse controle não é tão abordado no contexto da estabilidade angular, e por isso foi desconsiderado no estudo.

O regulador de tensão tem como função manter constante a tensão do barramento da máquina, para isso, ele atua diretamente na corrente de campo, e por consequência na tensão  $E_f$ . Diversos tipos de reguladores de tensão são aplicados no controle de tensão das máquinas síncronas, para as simulações desenvolvidas, o regulador de tensão empregado é mostrado na Figura 4 na forma de diagrama de blocos, e representa um regulador de tensão estático de primeira ordem.

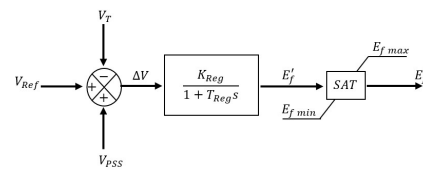


Fig. 4. Modelo do regulador de tensão empregado no trabalho.

A influência do regulador de tensão na estabilidade dinâmica de sistemas elétricos é mostrada por DeMelo e Concordia (1969). Para situações como alta impedância vista da barra do gerador e maior despacho de geração, o regulador de tensão pode aumentar o torque sincronizante, porém para essas mesmas situações, diminuir o torque de amortecimento. A diminuição do torque de amortecimento pode causar instabilidade oscilatória. Um sinal adicional pode ser implementado na entrada do regulador de tensão com o propósito de melhorar o desempenho dinâmico do sistema.

Esse sinal adicional é advindo do estabilizador de sistema de potência (PSS), tendo como função induzir torque em fase com o desvio de velocidade através do regulador de tensão (Kundur et al., 1981). Diversos sinais de entrada podem ser adotados

para o PSS, entre eles a frequência da tensão do barramento do gerador, a potência despachada e a velocidade do rotor (Larsen & Swann, 1981). O modelo do PSS utilizado é mostrado na Figura 5 e possui como entrada a velocidade do rotor da máquina.

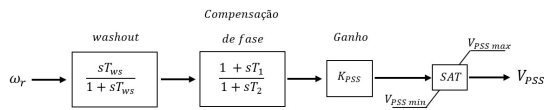


Fig. 5. Modelo do PSS empregado no trabalho.

### 2.3 Modelagem da rede.

Para os transitórios de baixa frequência, a rede é modelada através de equivalentes  $\pi$  para as linhas de transmissão e transformadores, sem variação de suas impedâncias com a frequência. Tal aproximação é válida, visto que os transitórios estudados a frequência varia pouco em relação a seu valor nominal (Arrillaga & Arnold, 1994).

Dessa forma, as correntes que entram em cada nó de uma rede de corrente alternada podem ser determinadas através da tensão desses nós e da matriz de admitância nodal, como mostra (14).

$$I_{Nós} = Y_{bus} V_{Nós} \quad (14)$$

Para o cálculo das correntes no regime transitório, os geradores podem ser integrados a rede por um equivalente Thévenin que varia de acordo com o modelo adotado para o gerador. Nesse trabalho, foi adotado o modelo da tensão atrás da reatância subtransitória, descrito na Subseção 2.1. Dessa forma, o equivalente Thévenin do gerador pode ser visto na Figura 6.

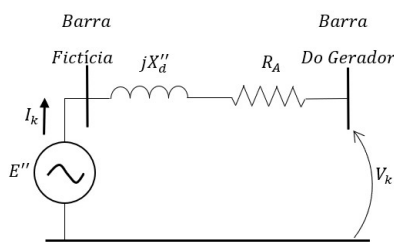


Fig. 6. Equivalente Thévenin do gerador para o modelo da tensão atrás da reatância subtransitória.

A barra fictícia inserida no modelo é considerada ao montar a matriz admitância, e dessa forma, a equação matricial que relaciona correntes e tensões da rede pode ser vista em (15), em que  $E_G$  representa o vetor de todas as fem's sub-transitórias de todos os geradores,  $V_N$  é a tensão nos nós em que não há injeção de corrente,  $I_G$  é a corrente injetada nos nós de geração e  $Y_{GG}$ ,  $Y_{GL}$ ,  $Y_{LG}$  e  $Y_{LL}$  são submatrizes da matriz admitância. Para as simulações desenvolvidas nesse trabalho, as cargas são transformadas em admitâncias constantes, dessa forma, não há injeção de corrente nas barras de carga (Machowski et al., 2008).

$$\begin{bmatrix} I_G \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{GG} & Y_{GL} \\ Y_{LG} & Y_{LL} \end{bmatrix} \begin{bmatrix} E_G \\ V_N \end{bmatrix} \quad (15)$$

Os nós em que não há injeção de corrente podem ser reduzidos aplicando-se a redução de Kron (Stevenson, 1986), resultando em (16).

$$I_G = (Y_{GG} - Y_{GL} Y_{LL}^{-1} Y_{LG}) E_G \quad (16)$$

Através da Equação (16), é possível calcular a corrente dos geradores do sistema, e através da transformação descrita em (13) essas correntes podem ser decompostas em seus eixos direto e em quadratura. Eventos como curtos-circuitos trifásicos, abertura de linhas e chaveamento de cargas podem ser simulados modificando a matriz admitância durante o processo de integração das equações diferenciais do sistema.

### 3. ANÁLISE LINEAR

A análise linear das equações dinâmicas de um sistema elétrico pode ser bastante útil na compreensão de algumas características do sistema. Esta seção mostrará como é feita a linearização de um sistema de equações diferenciais tais quais as definidas na Seção 2, e como essa análise pode prover informações úteis sobre o sistema.

Através do primeiro método de Lyapunov, a estabilidade de um sistema não linear nas redondezas de um ponto de operação pode ser avaliada através de sua linearização da seguinte forma (Machowski et al., 2008):

- Se a linearização do sistema naquele ponto de operação for assintoticamente estável, então o sistema não linear é estável naquele ponto.
- Se a linearização do sistema naquele ponto de operação for assintoticamente instável, então o sistema não linear é instável naquele ponto de operação.
- Se a linearização do sistema naquele ponto de operação for estável, porém não assintoticamente estável, então a estabilidade do sistema não pode ser avaliada através de sua linearização.

A linearização do sistema não linear de equações diferenciais é feita através do primeiro termo da série de Taylor (Ogata, 2010). A Equação (17) mostra a linearização de uma função  $f$  de  $n$  variáveis, representadas por  $X = (x_1, x_2, \dots, x_n)$  com a série de Taylor em torno de um ponto de operação  $X_0 = (x_{10}, x_{20}, \dots, x_{n0})$ .

$$f(X) = f(X_0) + \frac{\partial f(X_0)}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f(X_0)}{\partial x_n} \Delta x_n \quad (17)$$

Dado um conjunto de equações diferenciais não lineares em (18), é possível tornar essas equações lineares para as redondezas de um ponto de operação  $X_0$ . Supondo que  $X_0$  é um ponto de equilíbrio do sistema, a linearização do sistema em torno desse ponto pode ser vista em (19).

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, \dots, x_n) \\ \vdots \\ \frac{dx_n}{dt} = f_n(x_1, x_2, \dots, x_n) \end{cases} \quad (18)$$

$$\begin{cases} \frac{d\Delta x_1}{dt} = \frac{\partial f_1(x_0)}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f_1(x_0)}{\partial x_n} \Delta x_n \\ \vdots \\ \frac{d\Delta x_n}{dt} = \frac{\partial f_n(x_0)}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f_n(x_0)}{\partial x_n} \Delta x_n \end{cases} \quad (19)$$

O sistema de equações (19) pode ser escrito na forma matricial, como mostra (20). A matriz com suas linhas compostas pelas derivadas parciais das funções  $f_1, f_2, \dots, f_n$  em relação as variáveis  $x_1, x_2, \dots, x_n$  é chamada de matriz jacobiana do sistema.

$$\begin{bmatrix} \dot{\Delta x}_1 \\ \vdots \\ \dot{\Delta x}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(x_0)}{\partial x_1} & \dots & \frac{\partial f_1(x_0)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x_0)}{\partial x_1} & \dots & \frac{\partial f_n(x_0)}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} \quad (20)$$

Com a obtenção de um sistema linear de equações diferenciais, é possível analisar a sua estabilidade através dos autovalores da matriz jacobiana. Se a matriz jacobiana possui como autovalores o conjunto  $\{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n\}$ , possuindo esse conjunto autovalores reais ( $\lambda_k = \alpha_k$ ) e complexos ( $\lambda_k = \alpha_k + j\omega_k$ ), então a resposta do desvio  $\Delta x_i$  em função do tempo será dada pela Equação (21).

$$\Delta x_i(t) = \sum |c_{ik}| e^{\alpha_k t} + \sum 2|c_{ik}| e^{\alpha_k t} \cos(\omega_k t + \phi_{ki}) \quad (21)$$

Em que  $c_{ik}$  e  $\phi_{ki}$  depende das condições iniciais do sistema e dos autovetores (a direita e a esquerda) associados a  $\lambda_k$ . Observa-se que se a parte real de um autovalor do sistema for positiva, a resposta será instável. Essa resposta instável pode ser oscilatória, caso o autovalor instável seja complexo, ou não oscilatória, caso o autovalor instável seja real.

Além da estabilidade do ponto de operação do sistema, a análise linear revela informações importantes, como as frequências de oscilação (Equação 22) e o valor de amortecimento (Equação 23) para cada modo do sistema.

$$f_k = \frac{\omega_k}{2\pi} \quad (22)$$

$$\zeta_k = \frac{-\alpha_k}{\sqrt{\alpha_k^2 + \omega_k^2}} \quad (23)$$

Esse desenvolvimento matemático é aplicável as equações dinâmicas de um sistema elétrico de potência. Através dessa análise, é possível conhecer os modos característicos de um sistema.

#### 4. FERRAMENTAS DE SIMULAÇÃO DESENVOLVIDAS

A linguagem Python possui bibliotecas de acesso livre para modelagem matemática, soluções numéricas e visualização de dados. Três bibliotecas da linguagem python para foram utilizadas para desenvolver as ferramentas de simulação não linear no tempo e análise linear dos modelos propostos, são elas as bibliotecas *Numpy* (n.d.), *Scipy* (n.d.-a) e *Matplotlib* (n.d.). O uso dessas bibliotecas é bastante comum em diversas áreas da ciência e engenharia, sendo empregadas em áreas como otimização, processamento de sinais, modelagem matemática e análise e visualização de dados.

A biblioteca Numpy é um agregado de diversos módulos e funções para a criação e manipulação de matrizes de dados (chamadas de *Numpy arrays*). Essa biblioteca foi utilizada nos códigos para a criação de matrizes e aplicação de operações e transformações da álgebra linear (inversão, multiplicação, obtenção de transposta, obtenção de autovalores e autovetores, etc).

A biblioteca Scipy possui diversos métodos matemáticos empregados na computação científica. Foram empregados dois métodos dessa biblioteca no trabalho, são eles o `integrate.solve_ivp`(n.d.-b) e o `optimize.approx_fprime` (n.d.-c).

O método `scipy.integrate.solve_ivp` tem como função integrar numericamente um sistema de equações diferenciais (lineares ou não) de primeira ordem, nesse método da biblioteca `scipy` é possível definir, entre outros parâmetros, o método de integração utilizado e o passo de integração. Esse método foi empregado em um código com a função de solucionar as equações diferenciais não lineares das máquinas e controladores vistos na Seção 2. O algoritmo implementado para a simulação no tempo pode ser visto na Figura 7.

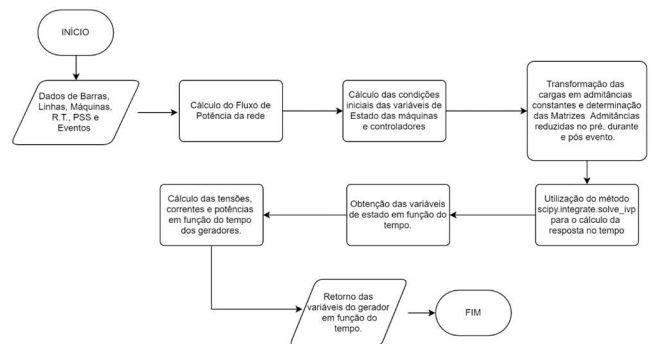


Fig. 7. Algoritmo implementado para simulação de transitórios eletromecânicos.

O método `scipy.optimize.approx_fprime` tem como função o cálculo do vetor gradiente de uma determinada função de múltiplas variáveis. Esse método foi empregado em um segundo código responsável por linearizar as equações diferenciais das máquinas e controladores vistos na Seção 2 e obter a matriz jacobiana do sistema para um determinado ponto de operação. Com a obtenção da matriz jacobiana, os autovalores da matriz são determinados com o uso do método

numpy.linalg.eig. O algoritmo implementado para análise linear é mostrado na Figura 8.

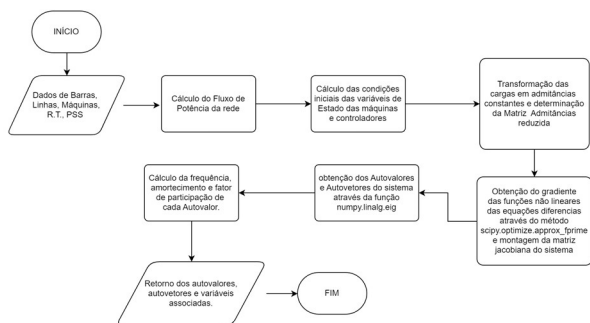


Fig. 8. Algoritmo implementado para análise linear de sistemas elétricos

Para a visualização das curvas de simulação no tempo, assim como também para a plotagem dos autovalores no plano complexo, foi empregada a biblioteca Matplotlib. Essa biblioteca possui diversas formas de plotagem e visualização de dados, como, por exemplo, plotagem de curvas, gráficos de barra e gráficos de dispersão.

### 5. SIMULAÇÕES EM UM SISTEMA COM 14 BARRAS.

Para testar os códigos propostos, foram realizadas simulações no sistema de 14 barras do IEEE (University of Washington, n.d.), o sistema é mostrado na Figura 9. O gerador da barra 1 foi definido com inércia infinita, e os dados dos quatro geradores restantes (G1, G2, G3 e G4) podem ser vistos na Tabela 1. Os dados de seus reguladores de tensão podem ser vistos na Tabela 2, e o PSS empregado somente no Gerador G1, tem seus dados descritos na Tabela 3. Desses quatro geradores, somente o G1 fornece potência ativa ao sistema, atuando os três restantes como compensadores síncronos.

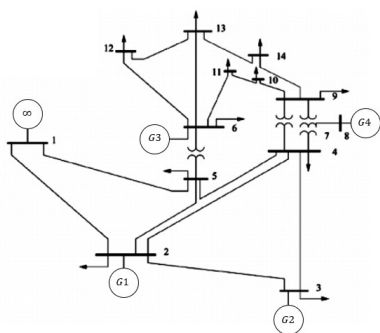


Fig. 9. Sistema elétrico de 14 barras.

Tabela 1. Dados dos Geradores.

Dados do gerador:	G1:	G2:	G3 e G4:
$S_{base}$	500 MVA	72 MVA	50 MVA
$H$	8 MW/MVA	3,588 MW/MVA	2,474 MW/MVA

$X_d$	1,138 p.u.	0,946 p.u.	1,003 p.u.
$X_q$	0,681 p.u.	0,621 p.u.	0,599 p.u.
$X'_d$	0,35 p.u.	0,31 p.u.	0,357 p.u.
$X''_d$	0,28 p.u.	0,274 p.u.	0,283 p.u.
$T'_{d0}$	5,6 s	8,13 s	7,28 s
$T''_{d0}$	0,08 s	0,048 s	0,05s
$T'''_{q0}$	0,15 s	0,14 s	0,1 s
$R_A$	0 p.u.	0 p.u.	0 p.u.
$X_l$	0,156 p.u.	0,202 p.u.	0,198 p.u.

Tabela 2. Dados dos Reguladores de Tensão.

Ger	$K_{reg}$	$T_{reg}$	$E_{f Max}$	$E_{f Min}$
1	200.	0,01 s	7 p.u.	-7 p.u.
2	100.	0,01 s	7 p.u.	-7 p.u.
3	50.	0,01 s	7 p.u.	-7 p.u.
4	50.	0,01 s	7 p.u.	-7 p.u.

Tabela 3. Dados do PSS adotado para o gerador G1.

Gerador:	G1
$K_{PSS}$	41,18922
$T_1$	0,2511851 s
$T_2$	0,4791076 s
$T_{ws}$	3
$V_{PSS Min}$	-0.1 p.u.
$V_{PSS Max}$	0.1 p.u.

Foram simulados dois cenários distintos de geração. No primeiro cenário, o gerador G1 está despachando 40 MW, e a barra infinita está fornecendo o restante de potência solicitada pelas cargas. No segundo cenário, o gerador G1 está despachando 500 MW, sendo parte dessa potência absorvida pelas cargas do sistema e a restante absorvida pela barra infinita. As simulações realizadas com os códigos desenvolvidos foram comparadas com as mesmas simulações realizadas nos softwares ANATEM (Cepel, 2020) e PacDyn (CEPEL, 2021).

#### 5.1 Simulação no tempo e comparação com software tradicional:

Realizando a simulação de um curto circuito trifásico na barra 2, com eliminação do circuito que interliga as barras 1 e 2 para ambos os cenários, é possível ver que as máquinas oscilam até

atingir um novo ponto estável de operação. A Figura 10 mostra a simulação para o cenário de menor geração por G1, enquanto a Figura 11 mostra o cenário de maior geração. É possível ver com as Figuras 10 e 11 a diferença no amortecimento das oscilações eletromecânicas para diferentes pontos de operação.

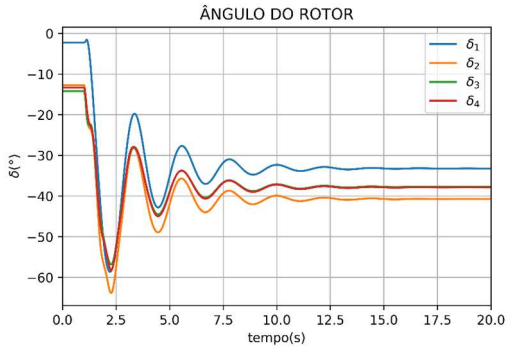


Fig. 10. Ângulo dos rotores das máquinas do sistema para o cenário base (G1 despachando 40 MW).

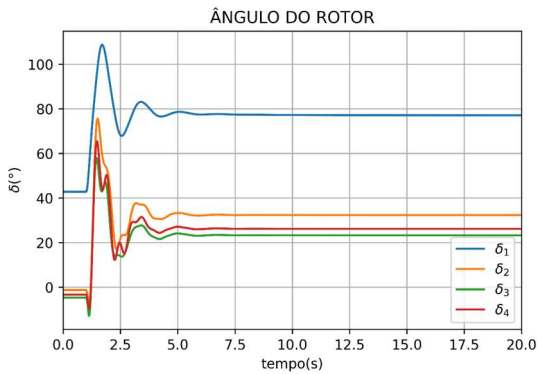


Fig. 11. Ângulo dos rotores das máquinas do sistema para o cenário de maior geração por G1 (500 MW).

As Figuras 12 e 13 mostram a comparação das respostas no tempo do ângulo do gerador G1 para os mesmos eventos simulados com o código Python e com o software ANATEM. Pode-se ver a convergência dos resultados em ambos os softwares de simulação.

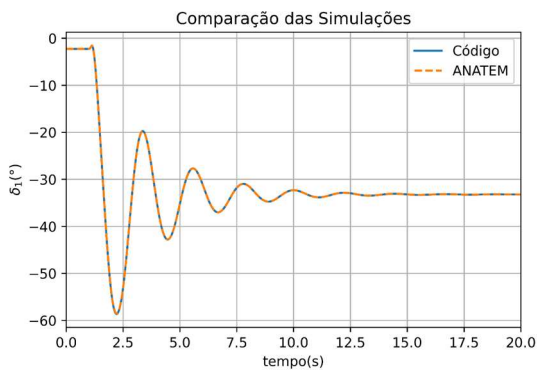


Fig. 12. Comparação das respostas no tempo do ângulo Gerador G1 para o cenário base.

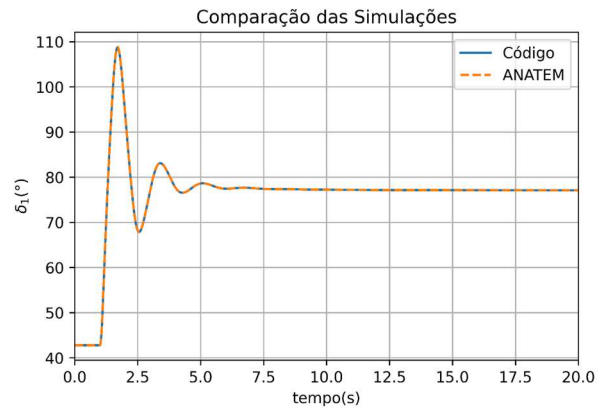


Fig. 13. Comparação das respostas no tempo do ângulo Gerador G1 para o cenário de maior geração.

### 5.2 Obtenção dos autovalores e comparação com software tradicional

Para o cenário em que o gerador G1 está despachando 500 MW e a linha que interliga as barras 1 e 2 está delgada, executou-se o código de linearização do sistema. Os autovalores mais próximos ao eixo imaginário podem ser vistos na Figura 14. É possível observar que o par de autovalores complexos mais próximo do eixo imaginário possui amortecimento superior a 20%, o que condiz com o bom amortecimento observado na simulação no tempo para o curto-circuito com perda de linha (Figura 11).

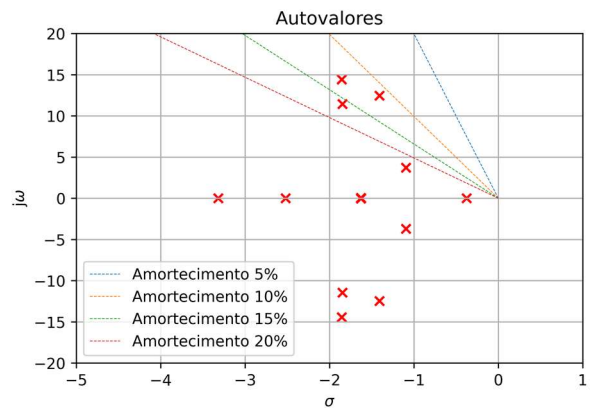


Fig. 14. Autovalores do sistema obtidos com o código de análise linear.

Fazendo a comparação desses resultados com os autovalores obtidos com o auxílio do PacDyn, foi observado que os resultados convergiram. A Figura 15 mostra a plotagem dos autovalores obtidos com o código de linearização escrito em Python e com o PacDyn no mesmo plano complexo.

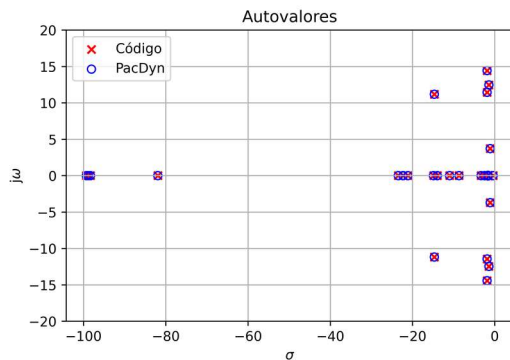


Fig. 15. Comparação dos autovalores do sistema obtidos com o código de análise linear e com o PacDyn.

## 6. CONCLUSÕES

Por ser uma linguagem com diversas bibliotecas de acesso livre para modelagem numérica, análise e visualização de dados, o Python mostra-se como uma linguagem promissora para o desenvolvimento de simulações computacionais em diversas áreas do conhecimento. Na área de sistemas elétricos de potência, tal linguagem possui grande utilidade, visto que as simulações em sistemas elétricos envolvem muitas soluções numéricas.

Através do estudo realizado, foi possível obter dois códigos dedicados a simulação no tempo de transitórios eletromecânicos e análise linear de sistemas elétricos através de bibliotecas de acesso livre do Python. Com esses códigos foi possível realizar a simulação dinâmica de um sistema de 14 barras, obtendo resultados satisfatórios e convergentes com softwares tradicionais de simulação.

Um dos parâmetros não observados nesse trabalho e que pode ser abordado em futuras publicações é o emprego de técnicas para a otimização das simulações numéricas. Técnicas como processamento paralelo e compilação *just-in-time* são algumas das que podem ser empregadas para esse tipo de aplicação.

## REFERÊNCIAS

- Anderson, P. M., & Fouad, A. A. (2003). *Power System Control and Stability* (2nd ed.). John Wiley and Sons.
- Arrillaga, J., & Arnold, C. P. (1994). *Computer Analysis of Power Systems*. John Wiley and Sons.
- Brown, T., Hörsch, J., & Schlachtberger, D. (2018). PyPSA: Python for power system analysis. *Journal of Open Research Software*, 6(1). <https://doi.org/10.5334/jors.188>
- CEPEL. (2021). *PacDyn - Análise e Controle de Oscilações Eletromecânica - User's Manual Release*.
- Cepel, E. (2020). *Análise de Transitórios Eletromecânicos - Anatem*. 397.
- Demello, F. P., & Concordia, C. (1969). Concepts of Synchronous Machine Stability as Affected by Excitation Control. *IEEE Transactions on Power Apparatus and Systems*, PAS-88(4), 316–329. <https://doi.org/10.1109/TPAS.1969.292452>
- Kundur, P. (1994). *POWER SYSTEM STABILITY AND CONTROL*. McGraw Hill.
- Kundur, P., Lee, D. C., & El-Din, H. M. Z. (1981). Power system stabilizers for thermal units: Analytical techniques and on-site validation. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(1), 81–95. <https://doi.org/10.1109/TPAS.1981.316890>
- Larsen, E. V., & Swann, D. A. (1981). APPLYING POWER SYSTEM STABILIZERS PART I: GENERAL CONCEPTS. *IEEE Transactions on Power Apparatus and Systems*, PAS-100.
- Machowski, J., Bialek, J. W., & R.Bumby, J. (2008). *Power System Dynamics: Stability and Control* (2nd ed.). John Wiley and Sons.
- Matplotlib. (n.d.). *Matplotlib*. Retrieved April 1, 2021, from <https://matplotlib.org/>
- NumPy. (n.d.). *NumPy*. Retrieved April 28, 2021, from <https://numpy.org/>
- Ogata, K. (2010). *Engenharia de Controle Moderno*. Pearson.
- Python. (n.d.). *Python*. Retrieved December 30, 2021, from <https://www.python.org/>
- SciPy. (n.d.-a). *scipy.integrate.solve\_ivp*. Retrieved September 10, 2021, from [https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve\\_ivp.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html)
- SciPy. (n.d.-b). *scipy.optimize.approx\_fprime*. Retrieved September 10, 2021, from [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.approx\\_fprime.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.approx_fprime.html)
- SciPy. (n.d.-c). *SciPy*. Retrieved April 29, 2021, from <https://www.scipy.org/>
- Stevenson, W. D. (1986). *Elementos de Análise de Sistema Elétricos de Potência* (2nd ed.). McGraw Hill.
- Turner, L., Scheidler, A., Schafer, F., Menke, J. H., Dollichon, J., Meier, F., Meinecke, S., & Braun, M. (2018). Pandapower - An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems*, 33(6), 6510–6521. <https://doi.org/10.1109/TPWRS.2018.2829021>
- University of Washington. (n.d.). *University of Washington Archive*. Retrieved October 5, 2021, from <http://labs.ece.uw.edu/pstca/>