

## Programa em Python para geração de sinal de cintilação ionosférica

### *Python program for ionospheric scintillation signal generation*

### *Programa em Python para generación de señal de parpadeo ionosférico*

#### Resumo

A cintilação ionosférica equatorial pode provocar fortes flutuações na magnitude e na fase de sinais trans ionosféricos, principalmente quando os enlaces de comunicação estão colocados em regiões próximas do equador magnético. Esse fenômeno provoca grande impacto no funcionamento dos receptores do Sistema Global de Navegação por Satélite (GNSS). Presentemente, há grande interesse na investigação de novas arquiteturas de receptores GNSS que sejam mais robustas à cintilação. O modelo CSM (*Cornell Scintillation Model*) foi desenvolvido pela Universidade de Cornell, usando *scripts* em MATLAB, com o objetivo de simular os efeitos da cintilação equatorial nos sinais recebidos pelo sistema GNSS. Essa ferramenta é largamente utilizada em estudos sobre novas arquiteturas de receptores submetidos à cintilação. Neste artigo, apresenta-se um programa em Python, chamado de *Scintillation Data Generator* (SDG), que, de forma similar ao CSM, gera e armazena um sinal que simula os efeitos da cintilação. Além disso, o programa é capaz de traçar gráficos que auxiliam na avaliação das estatísticas relacionadas ao sinal gerado e um gráfico interativo que facilita que o usuário perceba de que modo o fator de cintilação  $S_4$  e o tempo de decorrelação, associado à função de autocorrelação do sinal, alteram a forma do sinal de cintilação. Além do detalhamento sobre o funcionamento do programa em Python, este trabalho revisita o modelo estatístico de um canal de comunicação submetido à cintilação e avalia o desempenho do programa proposto em termos da qualidade dos sinais gerados.

Palavras-chave: Python, Cintilação, Ionosfera.

#### Abstract

*Equatorial ionospheric scintillation can cause intense fluctuations in the magnitude and phase of trans-ionospheric signals, especially when communication links are placed in regions close to the magnetic equator. This phenomenon has a crucial impact on the functioning of the Global Navigation Satellite System (GNSS) receivers. There is great interest in investigating new GNSS receiver architectures that are more robust to scintillation. The CSM model (Cornell Scintillation Model) was developed by Cornell University, using MATLAB scripts, to simulate the effects of equatorial scintillation on signals received by the GNSS system. This tool is widely used in studies on new receptor architectures subjected to scintillation. In this article, we present a Python program called Scintillation Data Generator (SDG), which, similar to CSM, generates and stores a signal that simulates the effects of scintillation. Furthermore, the program is capable of plotting graphs that help in the evaluation of statistics related to the generated signal and an interactive graph that makes it easier for the user to understand how the  $S_4$  flicker factor and the decorrelation time associated with the signal's autocorrelation function, change the shape of the flicker signal. In addition to detailing the operation of the Python program, this work revisits the statistical model of a communication channel subjected to flickering. It evaluates the performance of the proposed program in terms of the quality of the signals generated.*

**Keywords:** Python, Scintillation, Ionosphere.

Rodrigo de Lima

Florindo  

Universidade de Fortaleza

- Unifor

rdlresearch@gmail.com

Antônio Macílio Pereira

de Lucena  

Universidade de Fortaleza

- Unifor

ampl@unifor.br



## Resumen

*El parpadeo ionosférico ecuatorial puede provocar fuertes fluctuaciones en la magnitud y en la fase de señales trans ionosféricas, principalmente cuando los enlaces de comunicación están puestos en regiones cercanas del ecuador magnético. Este fenómeno provoca gran impacto en el funcionamiento de los receptores del Sistema Global de Navegación por Satélite (GNSS). Presentemente, hay gran interés en la investigación de nuevas arquitecturas de receptores GNSS que sean más robustas al parpadeo. El modelo CSM (Cornell Scintillation Model) fue desarrollado por la Universidad de Cornell, usando scripts en MATLAB, con el objetivo de simular los efectos del parpadeo ecuatorial en las señales recibidas por el sistema GNSS. Esta herramienta es muy utilizada en estudios sobre nuevas arquitecturas de receptores sometidos al parpadeo. En este artículo, se presenta un programa de Python, llamado de Scintillation Data Generator (SDG), que, de forma similar al CSM, genera y almacena una señal que simula los efectos del parpadeo. Además de eso, el programa es capaz de trazar gráficos que auxilian en la evaluación de las estadísticas relacionadas a la señal generada y un gráfico interactivo que facilita el usuario a percibir de qué modo el factor de parpadeo  $S_4$  y el tiempo de no correlación asociado a la función de autocorrelación de la señal, alteran la forma de la señal de parpadeo. Además del detalle sobre el funcionamiento del programa en Python, este trabajo revisita el modelo estadístico de un canal de comunicación sometido al parpadeo y evalúa el rendimiento del programa propuesto en términos de la calidad de las señales generadas.*

**Palabras clave:** Python, Parpadeo, Ionosfera.

## 1 Introdução

Um efeito sob os sinais de enlaces trans ionosféricos, que prejudica significativamente as comunicações nas bandas de VHF (*Very High Frequency*) e UHF (*Ultra High Frequency*), é denominado de cintilação ionosférica e ocorre nas regiões em torno do equador magnético (cintilação equatorial) e nas regiões polares e aurorais (Crane, 1977; Dubey, 2006). A cintilação decorre da passagem do sinal transmitido por camadas localizadas na ionosfera, com irregularidades na densidade de elétrons livres, o que ocasiona flutuações rápidas tanto em amplitude como na fase do sinal recebido (Kintner, 2007; Lucena *et al.*, 2021).

A cintilação afeta o funcionamento do receptor de alguns sistemas de comunicação espaciais, incluindo aqueles dos sistemas GNSS (*Global Navigation Satellite System*). A variação de amplitude e de fase do sinal recebido provocada pela cintilação ionosférica pode ter grande impacto na detecção de dados e na sincronização de fase e frequência do receptor, o que acarreta deterioração na qualidade dos serviços desses sistemas. No caso dos receptores GNSS, o sincronizador de portadora, é o subsistema do receptor mais susceptível a essas alterações de amplitude e fase (Kaplan, 2006; Borre *et al.*, 2007).

Nas últimas duas décadas têm sido intensas as pesquisas sobre a cintilação ionosférica objetivando a sua caracterização e modelagem, além de sua detecção e da mitigação de seus efeitos em receptores GNSS (Vilà-Valls, 2020). Modelos matemáticos para simular canais de comunicação sob influência da cintilação ionosférica são de extrema importância em estudos envolvendo o sistema GNSS e especificamente no desenvolvimento de novas arquiteturas de receptores robustas à cintilação.

O modelo CSM (*Cornell Scintillation Model*) é um modelo estatístico amplamente utilizado para simular a cintilação equatorial em pesquisas e projetos relacionados à recuperação de portadora em receptores dos sistemas GNSS (Humphreys, 2009). Esse modelo está disponível em <https://gps.ece.cornell.edu/tools.php> e é implementado em *scripts* Matlab. O sinal correspondente ao efeito da cintilação é gerado a partir do índice de cintilação  $S_4$ , que indica a intensidade da cintilação, e do parâmetro, associado à função de autocorrelação desse sinal.

O foco desse artigo é a apresentação de um programa em Python, que, de forma similar ao CSM, gera um sinal de cintilação a partir de  $s_4$  e  $\tau_0$ . Além da série numérica de dados, correspondente ao sinal gerado, o programa também permite o traçado de gráficos relacionados a algumas propriedades do sinal. O

programa proposto é validado pela comparação das características estatísticas do sinal gerado com aquelas correspondentes ao modelo teórico.

Este trabalho está organizado como se segue. Na Secção 2, descreve-se o modelo de canal considerando a cintilação ionosférica equatorial e deriva-se o modelo discreto equivalente. O programa proposto está apresentado na Secção 3. Os resultados de simulação para validação do programa estão sintetizados na Secção 4. Por fim, na Secção 5, conclui-se o artigo.

## 2 Modelo de canal com cintilação ionosférica de baixa latitude

### 2.1 Modelo de canal

Transmissores GNSS enviam informações através de onda eletromagnética de alta frequência cujo sinal equivalente de banda-básica recebido é dado por

$$r(t) = a(t)s[t - \tau(t)]z(t)e^{j\theta(t)} + n(t), \quad (1)$$

em que  $a(t)$  corresponde à atenuação no espaço livre,  $s(t)$  representa o sinal equivalente de banda-básica transmitido,  $\tau(t)$  indica o atraso do sinal decorrente da propagação e  $\theta(t)$  modela o *offset* de fase e frequência, incluindo o desvio Doppler. O efeito da cintilação é modelado pelo sinal  $z(t)$ , enquanto  $n(t)$  denota o efeito do ruído térmico Gaussiano (Lucena *et al.*, 2021).

O sinal de cintilação  $z(t)$  pode ser expresso por (Lucena *et al.*, 2021; Humphreys *et al.*, 2009):

$$z(t) = A + y(t), \quad (2)$$

em que  $A$ , chamada de componente direta do sinal recebido, é modelada como uma constante complexa, e  $y(t)$  é um processo estocástico complexo gaussiano de média nula, com variância  $2\sigma^2$ , associado com a parte do sinal que acaba sendo dispersado, chamado de componente de múltiplos caminhos.

A função de autocorrelação do processo  $y(t)$  é definida por  $R_{yy}(\tau) = E\{y(t)y^*(t+\tau)\}$ , sendo  $E\{x\}$  a média estatística de  $x$  e  $y^*(t+\tau)$  o complexo conjugado de  $y(t+\tau)$ . Portanto,  $R_{yy}(\tau)$  pode ser expressa por (Humphreys *et al.*, 2009):

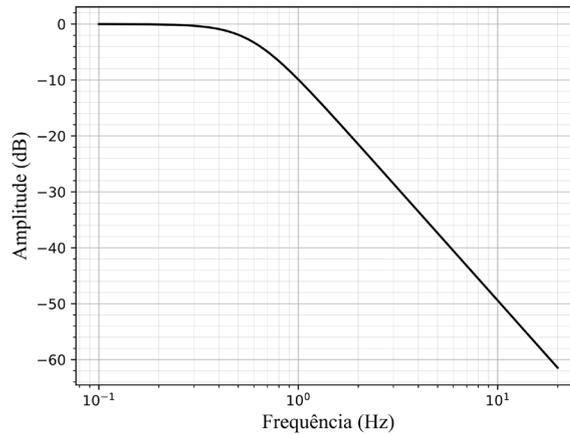
$$R_{yy}(\tau) = 2\sigma^2 e^{\left(\frac{-\beta|\tau|}{\tau_0}\right)} \left[ \cos\left(\frac{\beta\tau}{\tau_0}\right) + \text{sen}\left(\frac{\beta|\tau|}{\tau_0}\right) \right], \quad (3)$$

em que adota-se  $\beta = 1.2396464$  para assegurar que  $R_{yy}(\tau_0)/R_{yy}(0) = e^{-1}$ , sendo  $\tau_0$  o tempo de decorrelação do canal. A Equação (4) sugere que o processo  $y(t)$  pode ser sintetizado pela passagem de um processo Gaussiano complexo e branco de variância  $2\sigma^2$  e média nula por um filtro Butterworth de segunda ordem cuja magnitude da resposta em frequência é

$$|H(f)| = \frac{1}{\sqrt{1 + \left(\frac{f}{f_n}\right)^4}}, \quad (4)$$

em que  $f_n = \beta/\sqrt{2} \pi\tau_0$ . A Figura 1 ilustra  $|H(f)|$  supondo  $\tau_0 = 0.17$ .

Figura 1 – Resposta em frequência do filtro passa baixa Butterworth de segunda ordem



Fonte: Os autores usaram programa SDG.

A magnitude de  $z(t)$ , indicada por  $\alpha = |z(t)|$  é uma variável aleatória com distribuição de Rice cuja função de densidade de probabilidade é expressa por (Humphreys *et al.*, 2009)

$$f_{\alpha}(\alpha) = \frac{2\alpha(1+K)}{\Omega} I_0\left(2\alpha\sqrt{\frac{K+K^2}{\Omega}}\right), \quad (5)$$

em que  $K = A^2/2\sigma^2$  é o fator de Rice e  $\Omega = E\{\alpha^2\} = E\{|z(t)|^2\} = A^2 + 2\sigma^2$ . Assumindo a conservação da potência do sinal transmitido, resulta em  $\Omega = 1$ .  $I_0(x)$  denota a função de Bessel modificada do primeiro tipo e ordem zero.

O índice de cintilação  $S_4$  é definido por Humphreys *et al.*, (2009), como segue:

$$S_4 = \frac{\langle \alpha^4 \rangle - \langle \alpha^2 \rangle^2}{\langle \alpha^2 \rangle^2}, \quad (6)$$

em que  $\langle \cdot \rangle$  denota média temporal.

Por outro lado, o fator de Rice está relacionado ao fator de cintilação  $S_4$  pela seguinte equação (Lucena *et al.*, 2021):

$$K = \frac{\sqrt{1 - S_4^2}}{1 - \sqrt{1 - S_4^2}}. \quad (7)$$

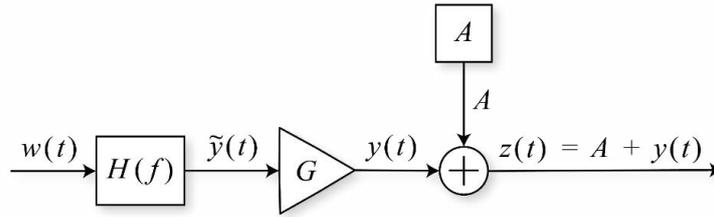
Partindo-se da definição de  $k$  e  $\Omega$ , obtém-se

$$A = \sigma\sqrt{2K}e\sigma = \sqrt{\frac{1}{2(1+K)}}, \quad (8)$$

## 2.2 Geração do sinal

Baseando-se no modelo de  $z(t)$  definido pelas Equações (3, 4, 5, 6, 7 e 8), configura-se a geração do processo de cintilação conforme o diagrama de blocos apresentado na Fig. 2.

Figura 2 – Diagrama de blocos para geração de cintilação



Fonte: Autoria própria

Na Fig. 2,  $w(t)$  representa o ruído Gaussiano complexo e branco com média nula e variância unitária,  $H(f)$  indica um filtro Butterworth de segunda ordem,  $\tilde{y}(t)$  é um processo estocástico com características espectrais semelhantes ao espectro de  $y(t)$  e  $G$  é um amplificador que ajusta a variância de  $\tilde{y}(t)$  para  $2\sigma^2$ . O valor de  $A$  é determinado a partir de  $S_4$ , usando as Equações (7 e 8), enquanto que o ganho  $G$  pode ser obtido como segue:

$$G = \sqrt{2}\sigma = \sqrt{\frac{1}{1 + K}}, \quad (9)$$

### 2.3 Discretização do gerador

Para geração da cintilação em Python, é necessário discretizar o modelo analógico detalhado acima. O ruído gaussiano branco complexo de média nula e variância unitária  $w(t)$  é amostrado a uma frequência  $F_s$ , derivando o ruído discreto correspondente  $w[n]$ . O filtro analógico do modelo é substituído por um filtro discreto  $H_c(e^{j\omega}) = H_c[k]$  com mesma resposta em frequência e ganho de potência unitário.

A resposta em frequência do filtro discreto pode ser determinada a partir da amostragem em frequência de  $H(f)$ . Amostrando-se  $H(f)$ , no intervalo de frequência 0 até  $F_s/2$  em  $N/2$  pontos uniformemente espaçados, obtêm-se a primeira parte das amostras da resposta em frequência do filtro discreto não normalizado  $\tilde{H}_c[k]$  pela seguinte expressão, explicitada por Ye e Satorius (2003):

$$\tilde{H}_c[k] = \frac{1}{\sqrt{1 + \left(\frac{kF_s\sqrt{2}\pi\tau_0}{N\beta}\right)^4}} e^{-j2\pi\frac{Lk}{2N}}, \quad (10)$$

$$k \in N \mid 0 \leq k \leq \frac{N}{2} - 1,$$

em que  $L$  corresponde à quantidade de amostras da resposta impulsiva não normalizada desejada  $\tilde{h}_c[n]$  e  $N$  o total de amostras tomadas da resposta em frequência do filtro  $H(f)$ . Tendo em vista que  $\tilde{h}_c[n]$  é real, cada elemento da outra metade de  $\tilde{H}_c[k]$  é dado por (Ye e Satorius, 2003):

$$\tilde{H}_c[N - k] = \tilde{H}_c[k], \quad (11)$$

$$k \in N \mid 1 \leq k \leq \frac{N}{2}.$$

Dessa forma, a resposta impulsiva discreta não normalizada  $\tilde{h}_c[n]$  do filtro é determinada pela Transformada Inversa e Discreta de Fourier (IDFT) de  $\tilde{H}_c[k]$  (Ye e Satorius, 2003):

$$\tilde{h}_c[n] = \frac{1}{N}g[n] \sum_{k=0}^{N-1} \tilde{H}_c(k) e^{j2\pi n\frac{k}{N}}, \quad (12)$$

$$n \in N \mid 0 \leq n \leq L - 1,$$

em que  $g[n]$  indica a janela de Hamming.

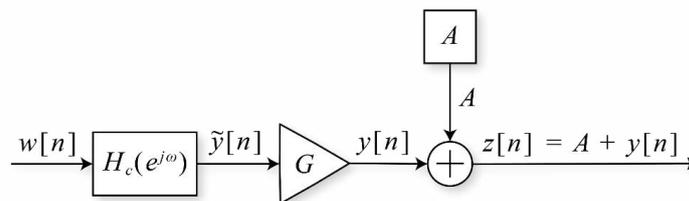
Finalmente, para fazer com que a variância do ruído na saída do filtro seja unitária, normaliza-se o filtro  $\tilde{h}_c[n]$  como se segue:

$$h_c[n] = \frac{\tilde{h}_c[n]}{\sqrt{\sum_{i=0}^{L-1} |\tilde{h}_c[i]|^2}}, \tag{13}$$

$$n \in N \mid 0 \leq n \leq L - 1.$$

O diagrama de blocos do gerador de cintilação discreto está ilustrado na Figura 3. Dado a frequência de amostragem  $F_s$  e o tempo de decorrelação  $\tau_0$ , determina-se a resposta impulsiva  $h_c[n]$  do filtro de Butterworth discreto, utilizando-se as Equações (10, 11, 12 e 13). Os valores de G e A são calculados a partir de  $S_4$ , através das Equações (7, 8 e 9).

Figura 3 – Diagrama geral de geração discreta de dados de cintilação sintética.



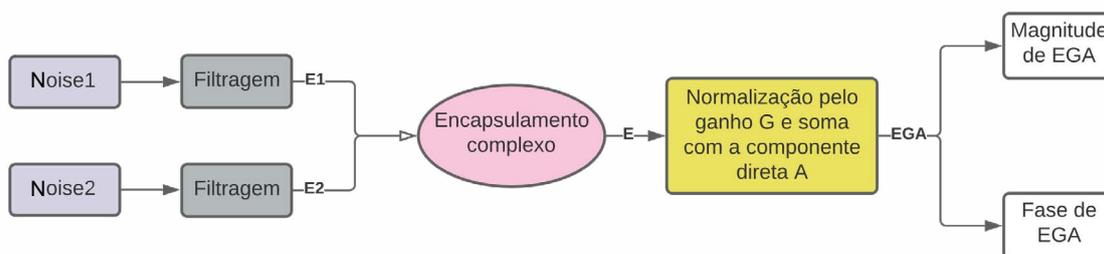
Fonte: Autoria própria

### 3 Descrição do programa

O programa *Scintillation Data Generator (SDG)* que foi desenvolvido em Python possui três funcionalidades principais: gerar e armazenar os dados do sinal de cintilação; traçar alguns gráficos relacionados à estatística do sinal e ao filtro discreto que define o espectro do sinal; e apresentar um gráfico interativo que permite ao usuário visualizar como os fatores  $S_4$  e  $\tau_0$ , afetam na forma do sinal de cintilação.

O sinal de cintilação é obtido a partir da criação de dois vetores de variáveis aleatórias gaussianas com nomes *Noise1* e *Noise2*, de médias nulas e variâncias  $s^2 = 0.5$  que, em sequência, são filtrados pelo filtro discreto  $h_c[n]$ , determinado conforme explicitado na Seção 2.3, gerando os vetores *E1* e *E2*, respectivamente. Por conseguinte, os dois vetores são unidos para formar o vetor complexo dado por  $E = E1 + jE2$ , o qual representa a componente de múltiplos caminhos não normalizada e discreta definida por  $\tilde{y}[n]$ . Em seguida, a componente *E* é normalizada pelo ganho *G* e somado ao componente direta *A*, definidos pelas Equações (7, 8 e 9), gerando o sinal EGA, o qual representa o sinal de cintilação complexo e discreto  $z[n]$ . Por fim, o sinal é separado em magnitude e fase.

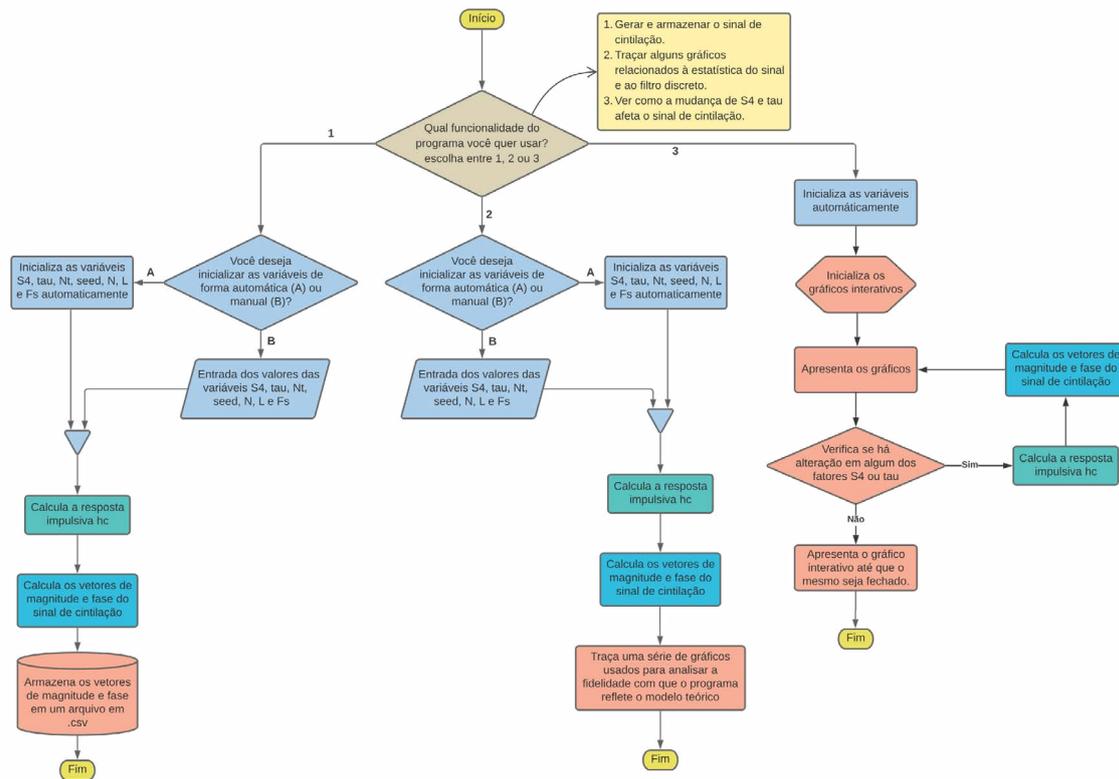
Figura 4 – Fluxograma de geração do sinal de cintilação complexo EGA



Fonte: Autoria própria.

O funcionamento do programa está descrito pelo fluxograma apresentado na Figura 5.

Figura 5 – Fluxograma de funcionamento do programa em Python



Fonte: Autoria própria.

Os dados gerados pela opção de número 1 são salvos em um arquivo .csv, cujas primeira e segunda linhas guardam os dados de magnitude do sinal de cintilação e de fase do sinal de cintilação, respectivamente. Os dados do arquivo salvo podem ser facilmente utilizados para incluir o sinal de cintilação à simulação de um sistema de comunicação.

Na opção de número 2, o programa é capaz de gerar os gráficos de: autocorrelação e correlação cruzada das variáveis gaussianas, histograma das variáveis gaussianas, histograma da magnitude do sinal de cintilação e a resposta em frequência e impulsiva do filtro Butterworth discreto associado ao espectro do sinal gerado.

O gráfico interativo da opção 3 é responsável por traçar as curvas de magnitude e de fase do sinal de cintilação, usando apenas duas sementes (*seeds*) fixadas e dos valores de  $S_4$  e  $\tau_0$  (tau) que podem ser definidos em tempo real na interface gráfica do traçado das curvas. O objetivo deste gráfico é facilitar a compreensão de como estes fatores alteram o sinal de cintilação ionosférica.

Caso o usuário escolha as opções de número 1 ou 2, será necessário escolher se pretende gerar os dados do sinal a partir da entrada automática ou de forma manual dos parâmetros de entrada. Os parâmetros de entrada são os seguintes: o índice de cintilação “S4”, o tempo de decorrelação “tau”, a quantidade de amostras do sinal de cintilação a serem geradas “Nt”, a *seed* de geração das variáveis aleatórias gaussianas “seed”, a quantidade de amostras da resposta em frequência do filtro de Butterworth analógico “N”, o tamanho do filtro discreto “L” e a frequência de amostragem “Fs”. Se a entrada automática for escolhida, os valores dos parâmetros são como segue:

$$S4 = 0.87; \tau = 0.17; Nt = 300000; seed = 2; N = 512; L = 256; Fs = 50.$$

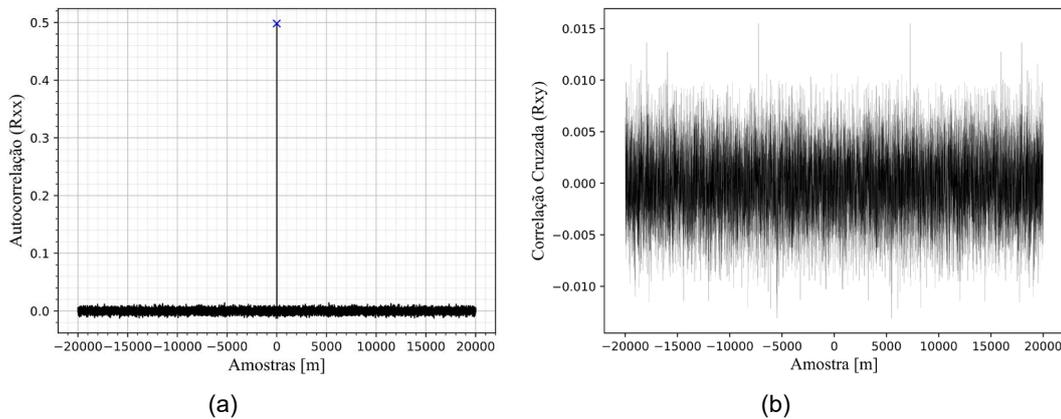
A opção de entrada automática serve para a familiarização do usuário com o programa.

## 4 Avaliação do programa

### 4.1 Estatística dos vetores gaussianos

Dois vetores gaussianos e brancos (x e y) de média nula e variância  $\sigma^2 = 0.5$  com um total de 20.000 amostras foram gerados pelo programa. As Figuras 6a e 6b ilustram a estimativa da função de autocorrelação de x ( $R_{xx}$  [m]) e a estimativa da correlação cruzada entre x e y ( $R_{xy}$  [m]). Na Fig. 6a, observa-se que  $R_{xx}$  [m] é nula para  $m \neq 0$ , evidenciando a característica de brancura do ruído, e  $R_{xx}$  [0] = 0.5 que corresponde ao valor de sua variância. A correlação cruzada sendo praticamente zero, conforme apresentado na Fig. 6b, demonstra que os ruídos gerados x e y são independentes para qualquer m.

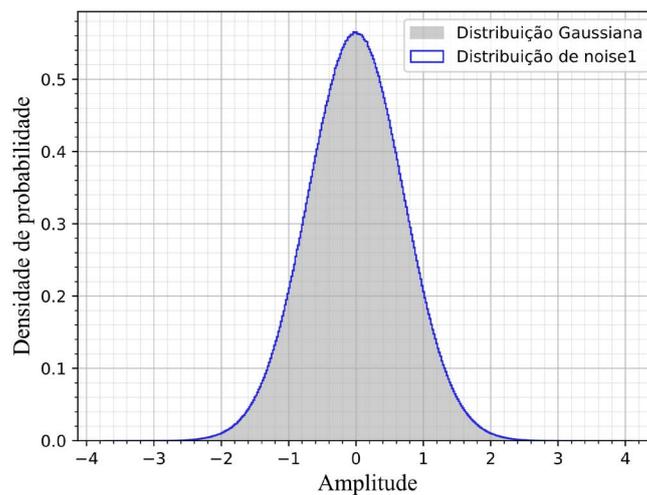
Figura 6 – (a) Gráfico da estimativa de função de autocorrelação de um vetor gaussiano gerado pelo programa. (b) Gráfico da função de correlação cruzada estimada dos dois vetores gaussianos gerados.



Fonte: Uso do programa SDG.

Em sequência, a Figura 7 mostra um histograma com 400 intervalos de amplitude de um vetor gaussiano de 32.000.000 amostras geradas pelo programa. Está notório que a distribuição de probabilidade é gaussiana com média nula e variância 0.5.

Figura 7 – Histograma de vetor gaussiano com 32.000.000 amostras.



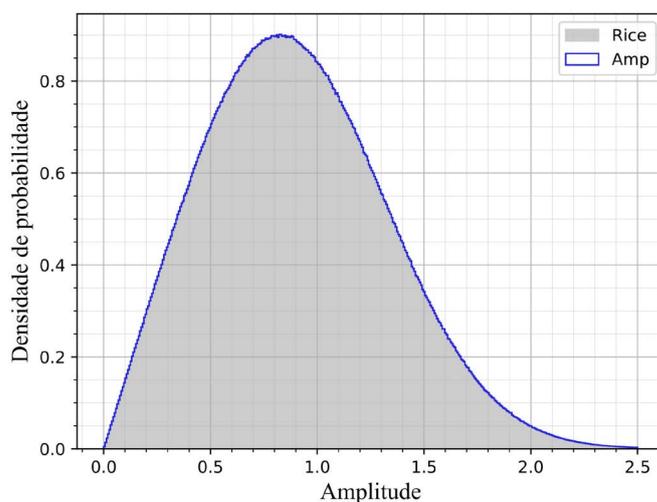
Fonte: Uso do programa SDG.

Portanto, é possível afirmar que os resultados dos gráficos refletem adequadamente o que se esperava das variáveis gaussianas, validando a função de geração de variáveis gaussianas do programa em Python

#### 4.2 Distribuição de probabilidade do sinal de cintilação

A Figura 8 exibe o histograma da magnitude do sinal de cintilação criado pelo programa em Python. Há uma concordância quase perfeita entre o histograma e a distribuição de Rice, que é a distribuição de probabilidade prevista pela teoria para o sinal de cintilação. Neste gráfico foi adotado  $S_4 = 0.87$  e  $\tau_0 = 0.17$ . Vários histogramas, com diferentes valores de  $S_4$  e  $\tau_0$ , foram traçados e todos eles seguem corretamente a distribuição teórica.

Figura 8 – Comparação da curva de distribuição de probabilidade Riciana com o histograma de magnitude do sinal de cintilação com 32.000.000 amostras e 400 intervalos ( $S_4 = 0.87$  e  $\tau_0 = 0.17$ )

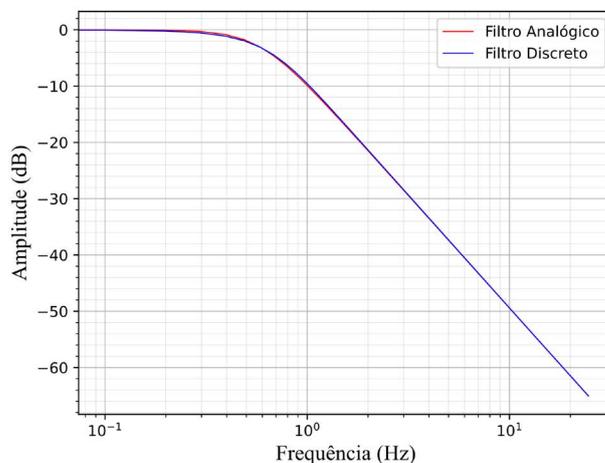


Fonte: Uso do programa SDG.

#### 4.3 Resposta em frequência do filtro de Butterworth discreto

Na Figura 9, comparam-se as respostas em frequência do filtro Butterworth de segunda ordem analógico, expressa por  $|H(f)|$ , e do filtro discreto, dada por  $|H_c[k]|$ , implementada pelo programa em Python. É importante destacar que a resposta em frequência do filtro discreto cada vez mais se aproxima da resposta em frequência do filtro analógico à medida que os valores de  $N$  e  $L$  crescem.

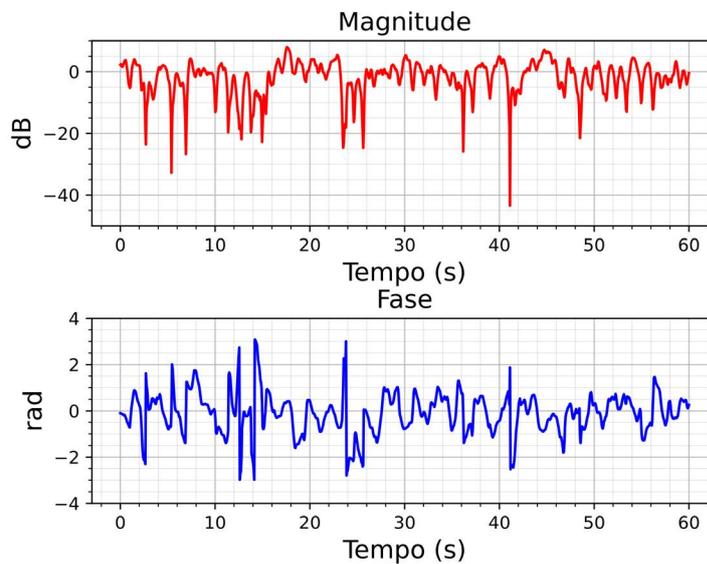
Figura 9 – Comparação entre o filtro Butterworth de segunda ordem analógico e o filtro discreto correspondente com  $N = 512$  e  $L = 256$ . O valor do tempo de decorrelação usado foi de  $\tau_0 = 0.17$ .



Fonte: Uso do programa SDG.

A magnitude em dB e a fase em radianos do sinal de cintilação gerado pelo programa por um período de 60 s estão ilustrados pela Figura 10. Nesse caso foi adotado  $S_4 = 0.87$ ,  $\tau_0 = 0.17$ ,  $F_s = 50$  Hz,  $N = 512$  e  $L = 256$ .

Figura 10 – Gráfico da magnitude e fase do sinal de cintilação gerados pelo programa em Python.



Fonte: Uso do programa Python.

## 5 Conclusão

Neste artigo revisitamos o canal de comunicação com cintilação ionosférica e apresentamos o modelo CSM (*Cornell Scintillation Model*), já consagrado na literatura para geração de sinais que refletem os efeitos da cintilação ionosférica equatorial em sistemas GNSS, no tocante a recuperação de fase de portadora.

Um programa em Python para implementação desse modelo foi apresentado em detalhes. Várias simulações foram desenvolvidas para avaliar o desempenho do programa em termos da qualidade dos sinais gerados.

As estimativas das funções de autocorrelação e correlação cruzada de vetores gaussianos gerados pelo programa se mostraram em conformidade com as previsões teóricas. A magnitude do sinal de cintilação sintetizado pelo programa também foi testada quanto a sua distribuição estatística, aproximando-se quase perfeitamente da distribuição de Rice. O espectro do sinal também foi avaliado comparando as magnitudes das respostas em frequência do filtro discreto implementado pelo programa e do filtro Butterworth analógico.

Embora o programa desenvolvido seja muito simples, ele é extremamente útil porque sintetiza com perfeição o sinal de cintilação ionosférica equatorial, sendo uma ferramenta essencial para as pesquisas de arquiteturas de receptores GNSS mais robustas à cintilação ionosférica.

## Referências

BORRE, K. *et al.* **A software-defined GPS and Galileo receiver: a single-frequency approach**. Berlim: Springer Science & Business Media, 2007.

CRANE, R. K. Ionospheric scintillation. **Proceedings of the IEEE**, [s. l.], v. 65, n. 2, p. 180-199, 1977. DOI: <https://doi.org/10.1109/PROC.1977.10456>

DUBEY, S. *et al.* Ionospheric effects on GPS positioning. **Advances in Space Research**, Bhopal, v. 38, n. 11, p. 2478-2484, 2006. DOI: <https://doi.org/10.1016/j.asr.2005.07.030>

HUMPHREYS, T. E.; PSIAKI, M. L.; HINKS, J. C.; O'HANLON, B.; KINTNER, P. M. Simulating Ionosphere-Induced Scintillation for Testing GPS Receiver Phase Tracking Loops. **Ieee Journal of Selected Topics in Signal Processing**, [s. l.], v. 3, n. 4, p. 707–715, 2009. DOI: <https://doi.org/10.1109/JSTSP.2009.2024130>

KAPLAN, E. **Understanding GPS: principles and applications**. Massachusetts: Artech House, 2006.

KINTNER, P. M.; LEDVINA, B. M.; DE PAULA, E. R. GPS and ionospheric scintillations. **Space weather**, [s. l.], v. 5, n. 9, p. 1-23, 2007. DOI: <https://doi.org/10.1029/2006SW000260>

LUCENA, A. M. P. de *et al.* Scintillation Effects in S-Band Telemetry Link of INPE's Earth Station in Cuiabá-Brazil. Springer Science & Business Media, Berlim, v. 30, n. 4, p. 739-748, 2021. DOI: <https://doi.org/10.13164/re.2021.0739>

VILÀ-VALLS, J.; LINTY, N.; CLOSAS, P.; DOVIS, F.; CURRAN, J. T. Survey on signal processing for GNSS under ionospheric scintillation: detection, monitoring, and mitigation. **Navigation**, [s. l.], v. 67, n. 3, p. 511–536, 2020 DOI: <https://doi.org/10.1002/navi.379>

YE, Z.; SATORIUS, H. Channel modeling and simulation for mobile user objective system (MUOS)—part 1: flat scintillation and fading. *In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS*, 5., 2003, Anchorage. **Proceedings** [...]. Anchorage: IEEE, 2003. p. 3503–3510. DOI: <https://doi.org/10.1109/ICC.2003.1204106>

## Sobre os autores

---

### Rodrigo de Lima Florindo

Graduando de Engenharia eletrônica na Universidade de Fortaleza (UNIFOR). Possui interesse nas áreas de processamento de sinais, sincronismo, filtragem adaptativa e Kalman Filters.

### Antônio Macílio Pereira de Lucena

Graduação em Engenharia Eletrônica pelo Instituto Tecnológico de Aeronáutica (1980), mestrado em Eletrônica e Telecomunicações pelo Instituto Nacional de Pesquisas Espaciais (1986) e doutorado em Engenharia de Teleinformática pela Universidade Federal do Ceará (2006). Trabalhou no Instituto Nacional de Pesquisas Espaciais (INPE) de 1983 até 2022 onde desenvolveu vários projetos nas áreas de telecomunicações espaciais, eletrônica e rádio-astronomia. Desde 2007 é professor da Universidade de Fortaleza (UNIFOR). Seus atuais interesses em pesquisas incluem técnicas de sincronização, telecomunicações espaciais, processamento de sinal, processamento de imagem e teoria de comunicação.

---

**Avaliado em:** 05.12.2023

**Aceito em:** 11.12.2023