

Universidade de Brasília - Faculdade de Tecnologia - Dep. de Eng. Elétrica

Neuroengenharia

Lista de exercícios 1

Prof. Antonio Padilha L. Bo[†]

[†] antonio.plb@lara.unb.br

Os resultados dessa lista de exercícios devem ser fornecidos de acordo com as instruções de cada questão. Caso o código não seja fornecido no formato correto ou gerar algum erro de execução, será conferida nota 0 (zero) à questão. Dessa forma, use obrigatoriamente o esqueleto de função Matlab fornecido. Além disso, não serão aceitos códigos que utilizem funções de alto nível que implementem a funcionalidade solicitada no exercício (e.g. não deve ser utilizada diretamente uma *toolbox* de classificação).

Acerca da entrega, o envio deve ser realizado através do *moodle* até a data e horário indicados pelos instrutores da disciplina. Aos arquivos *.m* enviados, deve ser acrescido um breve relatório, contendo uma descrição do método empregado na solução de cada questão (utilize no máximo uma página para descrever cada questão). Esse conjunto de arquivos deve ser comprimido em um arquivo *.zip* antes do envio pelo *moodle*. O arquivo *.zip*, bem como os arquivos individuais, devem ser nomeados com o seu nome (e.g. antonio).

1 Detecção e classificação de spikes (spike sorting)

A partir de medições realizadas próximas a neurônios, é possível determinar o instante em que houve o disparo de potenciais de ação. Assim, em geral a primeira etapa de processamento em sistemas realizam medições desse tipo é identificar em sinais de tensão com diferentes níveis de ruído os instantes em que ocorreram tais disparos. Para solucionar tal problema, repare que assumir um limiar fixo de tensão para detectar os disparos e desconsiderar um período refratário não é suficiente para fornecer desempenho satisfatório.

Além disso, em muitos casos um eletrodo é capaz de detectar sinais provenientes de diferentes neurônios. Nesses casos, são necessários algoritmos que permitam também classificar spikes que compõem o sinal de tensão proveniente de um eletrodo.

Diante do exposto, desenvolva as quatro funções Matlab descritas abaixo. Note que são fornecidos diferentes conjuntos de dados para avaliação dos algoritmos: uns mais simples, outros mais complexos.

- (a) Função que determine os instantes em que ocorreram os disparos a partir de registros eletrofisiológicos. A função deve ser da forma `spikeTimes = nome_q1a(time, electrodeVoltage, signal)`, em que `time` e `electrodeVoltage` são vetores que representam, respectivamente, o tempo e as medidas realizadas. `spikeTimes` é um vetor contendo os instantes de tempo nos quais foi detectada a ocorrência de um spike. Não pode ser utilizada a função `findPeaks` do Matlab.
- (b) Função auxiliar para plotar o sinal eletrofisiológico e indique os instantes em que ocorreram disparos, i.e., como um *raster plot*. Use o formato `nome_q1b(time, electrodeVoltage, spikeTimes, signal)`.
- (c) Função auxiliar para plotar o formato de cada disparo que foi identificado. Use o formato `nome_q1c(time, electrodeVoltage, spikeTimes, signal)`.

(d) Função `[spikeTimesA, spikeTimesB] = nome_q1(time, electrodeVoltage, signal)`, em que `time` e `electrodeVoltage` são vetores que representam, respectivamente, o tempo e a medida realizada. As variáveis `spikeTimesA` e `spikeTimesB` devem representar vetores com `NspikesA` e `NspikesB` colunas, em que `Nspikes` é o número de spikes detectados. Cada elemento dos vetores representam o instante no qual ocorreu o spike. A seguinte bibliografia é sugerida para auxiliar na resolução da questão:

- Pascal Wallisch. Principal Component Analysis (cap. 14), in Matlab for Neuroscientists. Elsevier, 2009
- Rodrigo Quian Quiroga, Spike Sorting, Scholarpedia, 2(12):3583, 2007
- Di Ge, Dario Farina, Spike Sorting (cap. 8), in Introduction to neural engineering for motor rehabilitation, edited by Dario Farina, Winnie Jensen, Metin Akay, IEEE Press, 2013

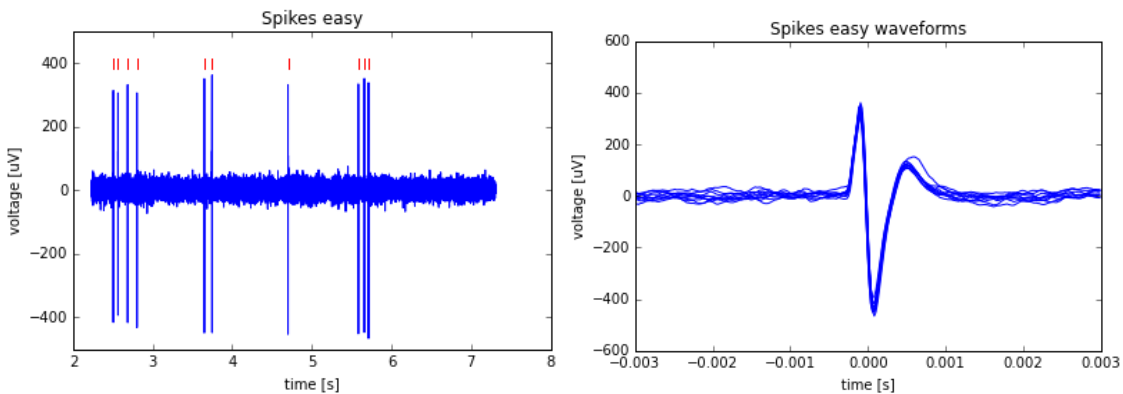


Figura 1: Exemplos de gráficos que devem ser gerados com as funções auxiliares das letras (b) e (c).

2 Simulação de neurônio LIF

Um dos modelos mais simples do neurônio capaz de representar a não-linearidade fundamental do sistema nervoso (potencial de ação), bem como algumas características dinâmicas, é o *Leaky Integrate-and-Fire neuron* (neurônio LIF). Implemente um neurônio LIF com as seguintes características: $V_{rest} = -70mV$, $V_{threshold} = -50mV$, $V_{spike} = 40mV$, $R_m = 1K\Omega$, $C_m = 20\mu F$ e $\tau_{refr} = 4ms$. Baseie sua simulação em um período de amostragem $T_s = 1ms$ e duração total $T_f = 1s$. O seu neurônio deve ser representado pela função `vMembrane = nome_q2(time, inputCurrent, signal)`. A parte linear que descreve a dinâmica do neurônio LIF pode ser representada por:

$$C_m \dot{V}_m + \frac{V_m - V_{rest}}{R_m} = i_m \quad (1)$$

- (a) Analise a frequência de disparo resultante considerando uma corrente na faixa de $10\mu A$ a $250\mu A$. Obtenha os histograma dos intervalos entre os disparos para cada entrada. Dicas: (i) a função criada na Questão 1 pode ser adaptada para identificar os *spikes* e (ii) pode ser utilizada a função `hist` do Matlab.
- (b) Adicione agora o efeito do ruído na corrente de entrada. Para tanto, considere um ruído branco Gaussiano ($\sigma = 30\mu A$). Para a mesma faixa de corrente da letra (a), obtenha os histogramas correspondentes. Discuta o resultado no relatório.

3 Eletromiografia (EMG) e eletroencefalografia (EEG)

Ambos sinais de EMG e EEG são úteis em muitos sistemas de neuroengenharia. Entretanto, para sua efetiva utilização, é necessária a realização de procedimento de pré-processamento. Os itens (a) e (b) abaixo correspondem, respectivamente, ao pré-processamento *offline* de EMG e EEG. O item (c) refere-se à classificação de padrões de EEG.

- (a) Forneça `emgProcessedVoltage = nome_q3a(time, emgVoltage, signal)`, em que `time` e `emgVoltage` são vetores que representam, respectivamente, o tempo e a tensão proveniente de par de eletrodos EMG pré-amplificado e `emgProcessedVoltage` é o sinal pré-processado, que deve ser um vetor de mesma dimensão de `emgVoltage`. As seguintes etapas de pré-processamento devem ser realizadas, nessa ordem: filtro passa-faixa (10 - 400Hz), retificação do sinal, filtro para detectar envoltório do sinal.
- (b) Forneça `[delta, theta, alpha, beta, gamma] = nome_q3b(time, eegVoltage, signal)`, em que `time` e `eegVoltage` são vetores que representam, respectivamente, o tempo e a tensão proveniente de par de eletrodos EEG pré-amplificado e `delta`, `theta`, `alpha`, `beta` e `gamma` são vetores de mesma dimensão de `eegVoltage` e que representam as faixas de frequência características do sinal de EEG, descritas abaixo:
- Delta: 0.5 - 4 Hz
 - Theta: 4 - 8 Hz
 - Alpha: 8 - 13 Hz
 - Beta: 13 - 20 Hz
 - Gamma: 20 - 40 Hz
- (c) Nesse item deve ser realizada tarefa de detecção de sinais provenientes de eventos de epilepsia (*seizures*). Forneça `classificationResult = nome_q3c(time, eegVoltage, signal)`, em que `time` e `eegVoltage` são vetores que representam, respectivamente, o tempo e a tensão proveniente de par de eletrodos EEG e `classificationResult` deve assumir valor 1 para sinal proveniente de evento de epilepsia e 0 caso contrário. Dica: use características baseadas nas faixas de frequências calculadas no item (b) para classificar os sinais.