O Modelo de McSharry para a Geração de Sinais de Eletrocardiograma Sintéticos por Meio de Equações Diferenciais Ordinais

Diogo Tuler Chaves

Universidade Federal de Minas Gerais

1 Introdução

O eletrocardiograma (ECG) é um exame que retorna graficamente uma série temporal que representa a atividade elétrica do coração durante seu funcionamento, um exemplo pode ser visto na Figura 1. Ele é um exame crucial na prevenção e identificação de problemas cardiovasculares, sendo um exame rápido, não invasivo e indolor. Tendo em vista que o exame é uma série temporal de valores, Patrick E. McSharry e sua equipe propuseram em 2003 um modelo dinâmico para a geração desses sinais através de um conjunto de Equações Diferenciais Ordinais (EDOs) [1], conseguindo simular a morfologia desses sinais. O objeto desse relatório é descrever o modelo, que ficou conhecido como Modelo de McSharry, bem como a origem e resolução das EDOs descritas nele.

2 Formato do Eletrocardiograma

s Um eletrocardiograma no computador é uma série temporal de valores,no qual o eixo x é o tempo e o y é a amplitude do sinal em Milivolts (mV), que tendem a oscilar entre 0,5 mV e 5 mV. Um exemplo de uma dessas capturas pode ser visto na Figura 1.

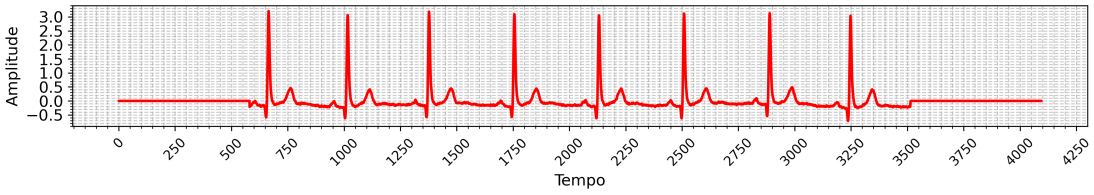

Figura 1: Exemplo de eletrocardiograma real da base do CODE-15%[2]

Cada batimento cardíaco é representado no ECG por um conjunto de cinco deflexões principais na forma de onda, identificadas como ondas P, Q, R, S e T como per ser visto na Figura 2. É comum estudar cada um desses picos e vales, bem como os intervalos entre eles, pois cada componente do ECG está relacionado a processos específicos que ocorrem no coração durante cada batimento.

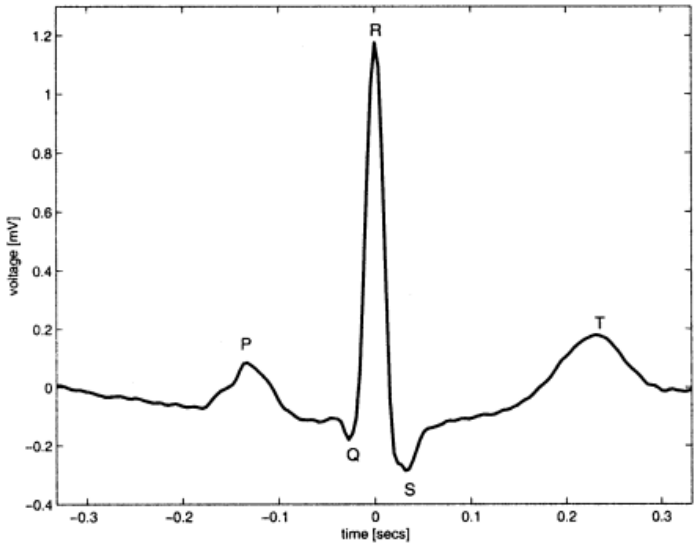

Figura 2: Exemplo retirado de [1] da divisão de um batimento cardíaco

3 Descrição do modelo

O modelo gera uma trajetória em um espaço tridimensional (3D) com coordenadas (x, y, z), como ilustrado na Figura 3. A quase periodicidade do ECG é representada pelo movimento da trajetória em torno de um ciclo limite atrativo de raio unitário no plano (x, y).Enquanto isso, as oscilações do exame ao longo de um ciclo podem ser simuladas pelo movimento da trajetória na dimensão z, identificando os pontos P, Q, R, S e T através das deflexões positivas e negativas nessa dimensão.

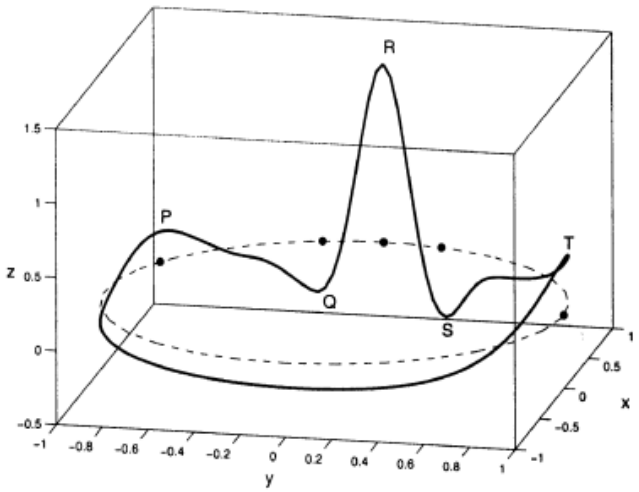

Figura 3: Exemplo retirado de [1] da trajetória gerada pelo modelo

Para gerar essas coordenadas é utilizado um conjunto de 3 equações ordinais diferenciais:

$$\begin{aligned}\dot{x} &= \alpha x - \omega y \\ \dot{y} &= \alpha y + \omega x \\ \dot{z} &= -\sum_{i \in \{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\Delta\theta_i^2 / 2b_i^2\right) - (z - z_0)\end{aligned} \tag{1}$$

onde:

$$\Delta\theta_i = (\theta - \theta_i) \mod 2\pi \tag{2}$$

$$\alpha = 1 - \sqrt{(x^2 + y^2)} \tag{3}$$

Aqui, θ_i são os ângulos correspondentes aos pontos P, Q, R, S e T no ciclo cardíaco com θ sendo $atan2(x, y)$. Os parâmetros a_i e b_i são constantes específicas do modelo para cada um dos pontos do ciclo. O artigo [1] monta a tabela da Figura 4 com base em ECG de indivíduos normais para usar no modelo.

TABLE I
PARAMETERS OF THE ECG MODEL GIVEN BY (1)

Index (i)	P	Q	R	S	T
Time (secs)	-0.2	-0.05	0	0.05	0.3
θ_i (radians)	$-\frac{1}{3}\pi$	$-\frac{1}{12}\pi$	0	$\frac{1}{12}\pi$	$\frac{1}{2}\pi$
a_i	1.2	-5.0	30.0	-7.5	0.75
b_i	0.25	0.1	0.1	0.1	0.4

Figura 4: Exemplo retirado de [1] da tabela de valores possíveis para o modelo com base em um indivíduo saudável

O artigo define o z_0, que pode ser visto como desvio inicial da linha de base do sinal, como uma função em volta da frequência respiratória $f2$:

$$z_0(t) = A \sin 2\pi f_2 t \tag{4}$$

onde:

$$A = 0,15mV \tag{5}$$

O ω é descrito no artigo como sendo a velocidade angular da trajetória no ciclo no plano (x,y) e pode ser definido como:

$$\omega = 2\pi/T(t) \tag{6}$$

Em ECGs normais existem variações no tempo de cada batimento cardíaco e nos períodos entre cada um dos pontos. Para conseguir simular essas variações tendo como base o observado em ECGs reais podemos gerar um espectro de potência a partir da soma de duas distribuições Gaussiana. Com esse espectro em mãos, podemos utilizar a transformada inversa de Fourier para convertê-lo em $T(t)$ que seria uma série temporal dos intervalos entre dois picos R.

3.1 Origem das EDOs

As EDOs do modelo de McSharry são derivadas a partir da observação das propriedades cíclicas do sinal do ECG e da necessidade de reproduzir suas características principais, como os picos e vales correspondentes às ondas P, Q, R, S e T. A abordagem baseia-se em sistemas dinâmicos e na teoria dos osciladores, ajustando parâmetros para replicar o comportamento observado nos dados reais de ECG.Um sistema dinâmico é um modelo matemático que descreve a evolução de um sistema ao longo do tempo.

Um exemplo simples de um modelo dinâmico, que até mesmo faz parte da grade curricular do ensino médio, é o movimento de um pêndulo simples. Em termos de EDOs ele pode ser escrito com a fórmula:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin(\theta) = 0 \tag{7}$$

onde:

- θ é o ângulo do pêndulo em relação à vertical,
- g é a aceleração devido à gravidade,
- L é o comprimento do fio do pêndulo.

É interessante notar a alta conectividade entre os fenômenos físicos; um eletrocardiograma pode ser modelado por um sistema semelhante a um sistema de pêndulo.

3.2 Análise das EDOs

Com base no estudado na disciplina é possível classificar cada uma das EDOs em função da seu grau, linearidade e homogeneidade:

- $\dot{x} = \alpha x - \omega y$

  Esta equação é de primeira ordem (maior derivada é a derivada primeira), não homogênea (o termo que depende de y,nesse caso, não é 0) e não linear (pois α é em função de x e y).

- $\dot{y} = \alpha y + \omega x$

  Esta equação é de primeira ordem (maior derivada é a derivada primeira), não homogênea (o termo que depende de x,nesse caso, não é 0) e não linear (pois α é em função de x e y).

- $\dot{z} = -\sum_{i\in\{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\Delta\theta_i^2/2b_i^2\right) - (z - z_0)$

  Esta equação é de primeira ordem (maior derivada é a derivada primeira). Sua homogeneidade e linearidade são mais difíceis de determinar que nos outros casos. Não consegui encontrar referências a esse respeito para essas equações, mas a minha avaliação é de que ela é não homogênea ($\sum_{i\in\{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\Delta\theta_i^2/2b_i^2\right)$ retorna uma constante e não 0) e é linear (pois não há termos que multipliquem z nem z').

3.3 Resolução das EDOs

O artigo menciona que as equações podem ser resolvidas utilizando o método de Runge-Kutta de quarta ordem descrito em C em [3]. Esse método utiliza quatro estimativas intermediárias para calcular a próxima posição da solução. Para aplicar esse método precisamos ter um problema de valor inicial da seguinte forma:

$$y' = f(x, y) \text{ } e \text{ } y(x_0) = y_0 \tag{8}$$

Esse método calcula a solução aproximada usando combinações de $f(x, y)$ em vários pontos do intervalo $x_k \leq x \leq x_k + 1$ ajustando parâmetros para que $y_k + 1$ coincida com a expansão de Taylor da ordem 4.

$$\begin{aligned}
k_1 &= hf(t_n, y_n) \\
k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\
k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\
k_4 &= hf(t_n + h, y_n + k_3)
\end{aligned}$$

onde h é o tamanho do passo. A próxima aproximação y_{n+1} é dada por:

$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Matematicamente cada um dos K_i valores possuem uma interpretação:

- k_1: é a inclinação no início do intervalo.
- k_2: é a inclinação no ponto médio do intervalo.
- k_3: é outra inclinação no ponto médio do intervalo, essa é mais precisa pois utiliza k_2.
- k_4: é a inclinação no final do intervalo.

Temos um exemplo de implementação em python [4] que pode ser visto abaixo:

```python
def feval(funcName, *args):
    return eval(funcName)(*args)

def RK4Ordem(func, yinit, x_range, h):
    m = len(yinit)
    n = int((x_range[-1] - x_range[0])/h)
    x = x_range[0]
    y = yinit
    xsol = np.empty(0)
    xsol = np.append(xsol, x)
    ysol = np.empty(0)
    ysol = np.append(ysol, y)

    for i in range(n):
        k1 = h*feval(func, x, y)
        k2 = h*feval(func, x+h/2, y + k1*(h/2))
        k3 = h*feval(func, x+h/2, y + k2*(1/2))
        k4 = h*feval(func, x+h, y + k3)

    for j in range(m):
        y[j] = y[j] + (1/6)*(k1[j] + 2*k2[j] + 2*k3[j] + k4[j])
        x = x + h
        xsol = np.append(xsol, x)

    for r in range(len(y)):
        ysol = np.append(ysol, y[r])

    return [xsol, ysol]

def myFunc(x, y):

    # Aqui e adicionado sua funcao especifica

    return dy
```

Listing 1: Código em Python de [4] para a resolução utilizando o método de Runge-Kutta de quarta ordem

Vale lembrar que as bibliotecas mais recentes de Python já oferecem funções prontas para resolver equações diferenciais ordinárias (EDOs). Para resolver uma EDO utilizando o método de

Runge-Kutta de quarta ordem, você pode usar a seguinte função descrita na documentaçao [5] :

```
from scipy.integrate import solve_ivp

def myFunc(x, y):

    # Aqui e adicionado sua funcao especifica

    return dy

t_span = # Aqui voce define seu intervalo de tempo
y0 = # Aqui voce define sua condicao inicial

sol = solve_ivp(myFunc, t_span, y0, method='RK45')
```

Listing 2: Código em Python de como utilizar [5] para a resolução de EDOs pelo método de Runge-Kutta de quarta ordem

É possível escolher em "method"várias outras formas de resolução, todas elas estão descritas em [5].

4 Implementação do modelo

O modelo pode ser implementado da seguinte forma em python:

```
def mcsharry(t,T,state):
    x, y, z = state
    omega = 2 * np.pi / T
    alpha = 1 - np.sqrt(x**2 + y**2)
    theta = np.arctan2(y, x)
    z0 = np.sin(2 * np.pi * f_2 * t)

    dxdt = alpha * x - omega * y
    dydt = alpha * y + omega * x
    dzdt = -sum(A[j] * (theta - theta_i[j]) * np.exp(-((theta - theta_i[j
        ]) % (2 * np.pi))**2 / (2 * b_i[j]**2)) for j in range(len(A))) - (
        z - z0)

    return [dxdt, dydt, dzdt]
```

Listing 3: Código em Python do modelo de MC Sharry

Usando como variáveis:

```
A = [1.2, -5.0, 30.0, -7.5, 0.75]
theta_i = [-np.pi/3, -np.pi/12, np.pi, 3 * np.pi / 2, 2 * np.pi]
b_i = [0.25, 0.1, 0.1, 0.1, 0.4]
f_2 = 0.25
T = 1.0  # Periodo constante
```

Listing 4: Código em Python das variáveis usadas

Com essa implementação super simples utilizando a função [5] é possível receber como saída o ECG simulado da Figura 5.

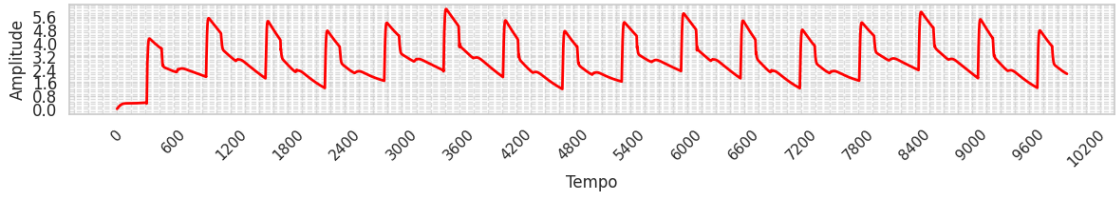

Figura 5: Exemplo de saída da implementação simplificada do modelo

Vale lembrar que é uma implementação super simples do modelo, sem entrar nos detalhes de cada um dos parâmetros e até mesmo usando um T(f), que é uma série temporal dos intervalos entre dois picos R, constante. Mesmo com essa implementação super simples já possível enxergar uma similaridade do exame gerado com um exame real. Felizmente para conseguir gerar um exame de ECG com o modelo de McSharry seguindo todos os detalhes de implementação é super simples. A biblioteca [7] já possui isso implementado, resultando em um código extremamente simplificado:

```
import neurokit2 as nk

ecg1 = nk.ecg_simulate(duration=10, method="ecgsyn") # "ecgsyn usa o
    modelo de McSharry para a simulacao
```

Listing 5: Código em Python da implementação com [7]

Utilizando essa biblioteca temos como resultado a Figura 6. O resultado é impecável, ao compararmos a Figura 6 e Figura 1 é muito difícil definir qual foi gerada e qual é o exame real.

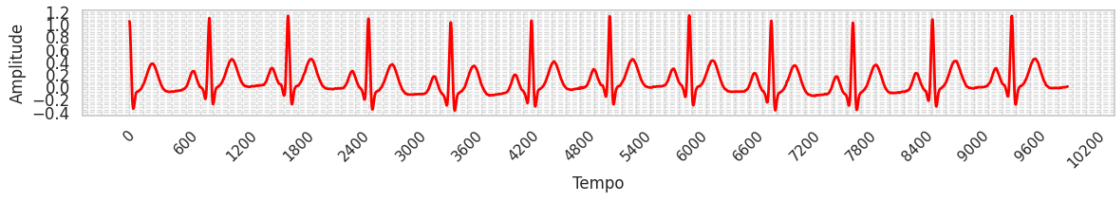

Figura 6: Exemplo de saída da implementação do modelo utilizando [7]

5 Utilidades do modelo

A capacidade de criar exames com o formato desejado é uma ferramenta valiosa e poderosa em pesquisas científicas sobre ECGs. O artigo [1] apresenta uma lista de cinco possíveis aplicações dessa abordagem. Considerando minha experiência e pesquisas na área, discutirei como esse modelo pode ser aplicado, ou provavelmente será aplicado, nas minhas pesquisas. É importante ressaltar que minhas pesquisas são voltadas na implementação de classificadores para ECGs,

- Um dos maiores problemas que estamos enfrentando é em relação à exames com interferência. O classificador acaba tentando classificar exames ilegíveis, prejudicando o seu treino e a sua acurácia final. Com a aplicação desse modelo, é possível gerar uma base de dados artificial que contém exames com e sem interferência. Podemos então desenvolver um classificador capaz de identificar a presença de interferência nos exames e pré-treiná-lo com esses dados simulados. Em seguida, ao aplicar esse classificador ao nosso conjunto de dados real, seremos capazes de identificar e remover exames com interferência, o que contribuirá significativamente para a melhoria da qualidade e precisão da análise.

- Os modelos que estamos usando são redes de aprendizado profundo. Uma das características negativas dessas desses é falta de interpretabilidade dos resultados. Todavia, podemos criar diversos ECGs sintéticos com esse modelo e coloca-los nas nossas redes. Podemos ir criando exames com certas características e analisar os resultados de classificação. Assim sendo, poderíamos de certa forma fazer um processo reverso e tentar entender quais características de cada exame levam a cada resultado na classificação.

6 Conclusão

As EDOs (Equações Diferenciais Ordinárias) são ferramentas matemáticas extremamente poderosas para a simulação de fenômenos físicos e químicos. Elas vão muito além do que é abordado nas disciplinas básicas, oferecendo um framework robusto para modelar e compreender a dinâmica de sistemas complexos tendo a capacidade de descrever mudanças contínuas ao longo do tempo.

A alta similaridade entre os resultados obtidos por meio desses modelos e os exames reais demonstra a eficácia das EDOs na replicação de comportamentos complexos. Além disso, é notável observar como problemas de natureza aparentemente distinta podem ser mapeados para sistemas dinâmicos, destacando a versatilidade das EDOs.

A aplicação de EDOs em ECGs dessa pesquisa revelou uma conexão profunda entre teoria e prática. A função de [7] já havia sido usada por mim diversas vezes para inúmeras tarefas, foi interessante saber em detalhes sobre seu funcionamento e descobrir como ela se integra completamente com os conceitos de EDOs.

Referências

[1] McSharry, P. E., Clifford, G. D. (2003). *A Dynamical Model for Generating Synthetic Electrocardiogram Signals*. Disponível em: `https://lcp.mit.edu/pdf/McSharryTBE03.pdf`

[2] CODE-15 Disponível em: `https://paperswithcode.com/dataset/code-15`

[3] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T. (1992). *Numerical Recipes in C* (2nd ed.). Cambridge, U.K.: Cambridge University Press. Disponível em: https://www.grad.hr/nastava/gs/prg/NumericalRecipesinC.pdf

[4] Doherty Andrade, P. (n.d.). *Método de Runge-Kutta de ordem 4: um exemplo usando Python*. Disponível em: www.metodosnumericos.com.br

[5] SciPy Documentation. (n.d.). *scipy.integrate.solve_ivp*. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

[6] Soares, J. J. O. (n.d.). *Estimação Paramétrica para um Modelo de Sinais de Eletrocardiograma Sintético Utilizando o Algoritmo da Região de Confiança*. Disponível em: https://www.repositorio.ufal.br/jspui/bitstream/123456789/12847/1/Estimao%20paramtrica%20para%20um%20modelo%20de%20sinais%20de%20eletrocardiograma%20sinttico%20utilizando%20o%20algoritmo%20da%20regio%20de%20confiana.pdf

[7] Neuropsychology. (n.d.). *ECG analysis*. Disponível em: https://neuropsychology.github.io/NeuroKit/functions/ecg.html. Acesso em: 21 de julho de 2024.