**Code:**

% Chuyển đổi thành tỷ lệ

x = x ./ n; % Tính tỷ lệ cho mỗi phần tử trong ma trận x

n=sum(x(:)); %Sum of Matrix elements

x=x./n; %proportion

r=sum(x,2); %rows sum

s=sum(x); %columns sum

Ex=r\*s; %expected proportion for random agree

pom=sum(min([r’;s])); %maximum proportion observable

po=sum(sum(x.\*f)); %proportion observed

pe=sum(sum(Ex.\*f)); %proportion expected

k=(po–pe)/(1–pe); %Cohen’s kappa

**Mô tả:**

tính toán Cohen’s Kappa - một chỉ số đo lường sự đồng thuận giữa hai người đánh giá hoặc giữa hai bộ phân loại.

**Kết quả:**

**A black text on a white background

Description automatically generated**

**Code:**

% Khởi tạo tín hiệu với một dạng sóng mẫu (ví dụ, sóng sin)

signal = sin(2 \* pi \* (1:100)/100); % Sóng sin với 100 mẫu

zeroCrossingNum = 0;

for i = 1:1:(length(signal)-1)

if signal(i) \* signal(i+1) <= 0

zeroCrossingNum = zeroCrossingNum + 1;

end

end

Z\_SQI = zeroCrossingNum / length(signal);

disp(['Zero Crossing Rate: ', num2str(Z\_SQI)]);

**Mô tả:**

signal là một sóng sin có 100 mẫu, dao động giữa giá trị âm và dương.

Vòng lặp kiểm tra **sự cắt qua điểm 0** của tín hiệu, tức là khi dấu của tín hiệu thay đổi giữa hai mẫu liên tiếp.

Nếu tích của hai giá trị liên tiếp signal(i) và signal(i+1) nhỏ hơn hoặc bằng 0, điều này cho thấy tín hiệu đã cắt qua trục hoành.

Biến zeroCrossingNum đếm số lần tín hiệu cắt qua trục hoành.

Biến Z\_SQI là **Tỷ lệ Cắt qua Điểm 0 (Zero Crossing Rate)**, được tính bằng tỷ lệ giữa số lần cắt qua trục hoành và tổng số mẫu trong tín hiệu.

**Kết quả:**

**A black text on a white background

Description automatically generated**

**Code:**

% Khởi tạo tín hiệu giả (ví dụ, tín hiệu sin với nhiễu)

Fs = 125; % Tần số lấy mẫu

t = 0:1/Fs:10-1/Fs; % Thời gian từ 0 đến 10 giây

signal = sin(2\*pi\*1\*t) + 0.5\*randn(size(t)); % Tín hiệu sin với nhiễu

% Loại bỏ giá trị trung bình của tín hiệu

signal = signal - mean(signal);

% Số lượng điểm FFT

NFFT = max(256, 2^nextpow2(length(signal)));

% Phương pháp Welch

[pxx, f] = pwelch(signal, length(signal), length(signal)/2, (NFFT\*2)-1, Fs);

% Định nghĩa các dải tần số

F1 = [1 2.25]; % Dải tần số F1

F2 = [0 8]; % Dải tần số F2

% Tính tổng công suất trong các dải tần số F1 và F2

powerF1 = trapz(f(f >= F1(1) & f <= F1(2)), pxx(f >= F1(1) & f <= F1(2)));

powerF2 = trapz(f(f >= F2(1) & f <= F2(2)), pxx(f >= F2(1) & f <= F2(2)));

% Tính chỉ số chất lượng tín hiệu (SQI)

R\_SQI = powerF1 / powerF2;

disp(['Signal Quality Index (SQI): ', num2str(R\_SQI)]);

**Mô tả:**

**Khởi tạo tín hiệu (signal)**: Tín hiệu giả được tạo ra bằng cách cộng một sóng sin (tần số 1 Hz) với một nhiễu ngẫu nhiên (tạo bởi randn).

**Loại bỏ giá trị trung bình**: Lệnh signal = signal - mean(signal); loại bỏ giá trị trung bình của tín hiệu, giúp tín hiệu có giá trị trung bình bằng 0.

**Phương pháp Welch**: Sử dụng pwelch để tính toán phổ công suất của tín hiệu, với các tham số như độ dài cửa sổ và tần số lấy mẫu. pwelch trả về phổ công suất pxx và các tần số f.

**Tính tổng công suất trong các dải tần số**:

powerF1 tính tổng công suất trong dải tần số từ 1 đến 2.25 Hz (F1).

powerF2 tính tổng công suất trong dải tần số từ 0 đến 8 Hz (F2).

**Chỉ số chất lượng tín hiệu (R\_SQI)**: Đo tỷ lệ giữa công suất trong dải F1 và F2. Chỉ số này được sử dụng để đánh giá chất lượng tín hiệu.

**Kết quả:**

**A black and blue text

Description automatically generated**

**Code:**

% Khởi tạo tín hiệu PPG giả

Fs = 125; % Tần số lấy mẫu

t = 0:1/Fs:10-1/Fs; % Thời gian từ 0 đến 10 giây

rawSignal = sin(2\*pi\*1\*t) + 0.5\*randn(size(t)); % Tín hiệu PPG giả (sóng sin với nhiễu)

% Lọc tín hiệu giả (giả sử đã áp dụng một bộ lọc)

filteredSignal = lowpass(rawSignal, 2, Fs); % Lọc tín hiệu với tần số cắt 2 Hz

% Tính độ lệch chuẩn của tín hiệu và nhiễu

std\_signal = std(abs(filteredSignal)); % Độ lệch chuẩn của tín hiệu đã lọc

std\_noise = std(filteredSignal); % Độ lệch chuẩn của tín hiệu nhiễu

N\_SQI = std\_signal / std\_noise; % Chỉ số chất lượng tín hiệu (N\_SQI)

disp(['Noise-to-Signal Quality Index (N\_SQI): ', num2str(N\_SQI)]);

**Mô tả:**

**Khởi tạo tín hiệu (rawSignal)**: Tín hiệu giả được tạo ra bằng cách cộng một sóng sin (tần số 1 Hz) với một nhiễu ngẫu nhiên (tạo bởi randn). Đây là tín hiệu PPG chưa được lọc.

**Lọc tín hiệu (filteredSignal)**: Sử dụng hàm lowpass để lọc tín hiệu với tần số cắt là 2 Hz, giúp loại bỏ các thành phần tần số cao không mong muốn.

**Tính độ lệch chuẩn của tín hiệu và nhiễu**:

std\_signal = std(abs(filteredSignal)); tính độ lệch chuẩn của giá trị tuyệt đối tín hiệu đã lọc, đại diện cho độ biến thiên của tín hiệu.

std\_noise = std(filteredSignal); tính độ lệch chuẩn của tín hiệu nhiễu.

**Chỉ số chất lượng tín hiệu (N\_SQI)**: Tính tỷ lệ giữa độ lệch chuẩn của tín hiệu và độ lệch chuẩn của nhiễu. Chỉ số này được sử dụng để đánh giá chất lượng tín hiệu, trong đó tỷ lệ càng cao cho thấy tín hiệu càng sạch, ít nhiễu.

**Kết quả:**

**A black and white text

Description automatically generated**

**Code:**

% Khởi tạo tín hiệu PPG giả

Fs = 125; % Tần số lấy mẫu

t = 0:1/Fs:10-1/Fs; % Thời gian từ 0 đến 10 giây

signal = sin(2\*pi\*1\*t) + 0.5\*randn(size(t)); % Tín hiệu PPG giả (sóng sin với nhiễu)

% Tính số lần cắt qua 0 (zero-crossing)

zeroCrossingNum = 0;

for i = 1:1:(length(signal) - 1)

if signal(i) \* signal(i + 1) <= 0

zeroCrossingNum = zeroCrossingNum + 1;

end

end

% Tính chỉ số chất lượng tín hiệu Zero Crossing Index (Z\_SQI)

Z\_SQI = zeroCrossingNum / length(signal);

disp(['Zero Crossing Quality Index (Z\_SQI): ', num2str(Z\_SQI)]);

**Mô tả:**

**Khởi tạo tín hiệu (signal)**:

Tạo một tín hiệu giả với tần số 1 Hz bằng sóng sin.

Thêm nhiễu ngẫu nhiên (randn) vào tín hiệu để mô phỏng tín hiệu PPG thực tế.

**Tính số lần cắt qua 0 (zero-crossing)**:

Biến zeroCrossingNum được khởi tạo bằng 0. Đây là biến đếm số lần tín hiệu cắt qua 0 (zero-crossings).

Vòng lặp for duyệt qua tất cả các phần tử của tín hiệu và kiểm tra điều kiện signal(i) \* signal(i + 1) <= 0, nghĩa là nếu tín hiệu chuyển từ dương sang âm hoặc ngược lại, thì đó là một lần cắt qua 0.

Nếu điều kiện này đúng, tăng biến zeroCrossingNum lên 1.

**Tính chỉ số chất lượng tín hiệu Zero Crossing Index (Z\_SQI)**:

Chỉ số Z\_SQI được tính bằng cách chia số lần cắt qua 0 (zeroCrossingNum) cho tổng số điểm trong tín hiệu (length(signal)).

Chỉ số này thể hiện chất lượng tín hiệu: tỷ lệ zero-crossing càng cao có thể cho thấy tín hiệu thay đổi nhanh và có tính ổn định thấp.

**Kết quả:**

**A black and white text

Description automatically generated**

**Code:**

% Khởi tạo tín hiệu PPG giả

Fs = 125; % Tần số lấy mẫu

t = 0:1/Fs:10-1/Fs; % Thời gian từ 0 đến 10 giây

signal = sin(2\*pi\*1\*t) + 0.5\*randn(size(t)); % Tín hiệu PPG giả (sóng sin với nhiễu)

% Lọc tín hiệu (giả sử đã có tín hiệu đã được lọc, ở đây đơn giản dùng tín hiệu gốc)

filtered\_signal = signal;

% Tính P\_SQI (Chỉ số chất lượng tín hiệu dựa trên biên độ)

P\_SQI = (max(filtered\_signal) - min(filtered\_signal)) / mean(signal) \* 100;

% Tính S\_SQI (Chỉ số chất lượng tín hiệu dựa trên độ lệch)

S\_SQI = skewness(signal);

% Tính K\_SQI (Chỉ số chất lượng tín hiệu dựa trên độ nhọn)

K\_SQI = kurtosis(signal);

% Tính E\_SQI (Chỉ số chất lượng tín hiệu dựa trên entropy)

E\_SQI = -sum((signal.^2) .\* log(signal.^2));

% Hiển thị các chỉ số

disp(['P\_SQI: ', num2str(P\_SQI)]);

disp(['S\_SQI: ', num2str(S\_SQI)]);

disp(['K\_SQI: ', num2str(K\_SQI)]);

disp(['E\_SQI: ', num2str(E\_SQI)]);

**Mô tả:**

**P\_SQI (Peak Signal Quality Index)**:

Tính toán độ biến thiên của tín hiệu bằng cách lấy hiệu giữa giá trị cực đại và cực tiểu của tín hiệu sau khi lọc (filtered\_signal), chia cho giá trị trung bình của tín hiệu gốc (signal), rồi nhân với 100 để có tỷ lệ phần trăm.

Đây là chỉ số thể hiện sự biến thiên của tín hiệu, càng lớn chứng tỏ tín hiệu có sự dao động mạnh.

**S\_SQI (Skewness Quality Index)**:

Tính độ lệch (skewness) của tín hiệu. Skewness đo độ không đối xứng của phân phối tín hiệu xung quanh giá trị trung bình.

Nếu skewness > 0, tín hiệu lệch phải (dồn về phía giá trị cao), nếu skewness < 0, tín hiệu lệch trái (dồn về phía giá trị thấp).

**K\_SQI (Kurtosis Quality Index)**:

Tính độ nhọn (kurtosis) của tín hiệu. Kurtosis đo sự tập trung của tín hiệu quanh giá trị trung bình.

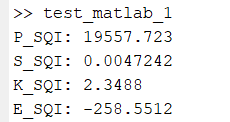
Kurtosis > 3 nghĩa là tín hiệu có phân phối nhọn hơn phân phối chuẩn (Gaussian), còn Kurtosis < 3 nghĩa là phân phối phẳng hơn.

**E\_SQI (Entropy Quality Index)**:

Tính entropy của tín hiệu, đây là một phép đo sự hỗn loạn hoặc tính không chắc chắn trong tín hiệu.

Công thức -sum((signal.^2) .\* log(signal.^2)) tính entropy theo định lý Shannon, là chỉ số đo mức độ ngẫu nhiên trong tín hiệu.

**Kết quả:**

****