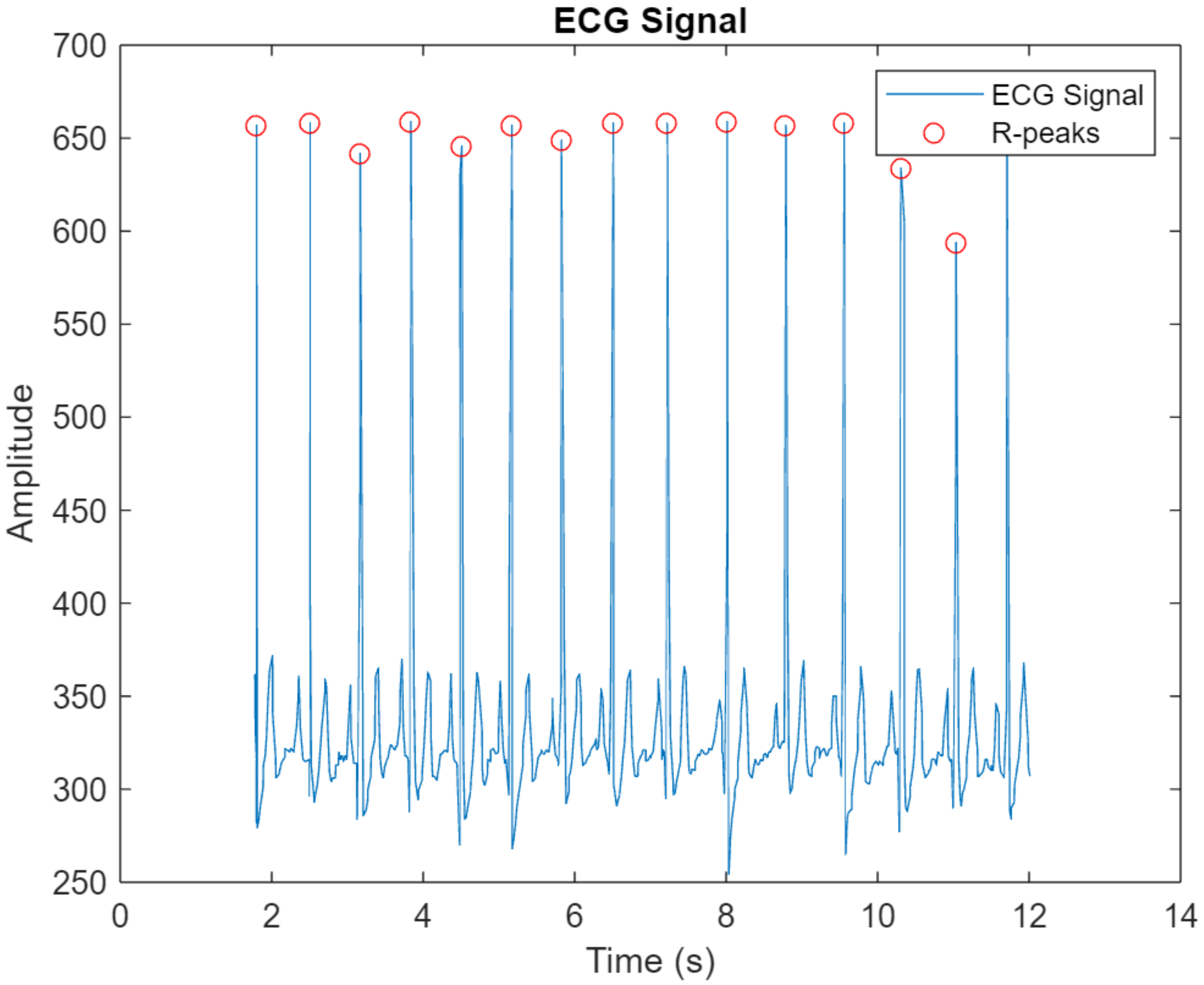
```
clear;
clc;
```

2_2:5-

```
%Load ECG data
data = csvread('ECG_Data.csv', 5, 0); % Skip 5 first rows
timestamps = data(:, 1); % First column: timestamp
ecgSignal = data(:, 2); % Second column: ECG data

%Plot the ECG signal
figure;
plot(timestamps, ecgSignal);
title('ECG Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Detect R-peaks
threshold = max(ecgSignal) * 0.7; % Set a threshold
[peaks, locations] = findpeaks(ecgSignal, 'MinPeakHeight', threshold, 'MinPeakDistance', 0.6*me

% Calculate R-R intervals
rrIntervals = diff(locations) .* mean(diff(timestamps)); % Convert to seconds

% Plot the R-peaks on the ECG signal
hold on;
plot(timestamps(locations), ecgSignal(locations), 'ro');
legend('ECG Signal', 'R-peaks');
```



```
% Calculate HRV metrics
meanRR = mean(rrIntervals); % Average R-R interval
stdRR = std(rrIntervals);   % Standard deviation of R-R intervals (HRV measure)
hr = 60 ./ rrIntervals;     % Heart rate (in bpm) for each interval

% Display results
disp('R-R Intervals (seconds):');
```

```
R-R Intervals (seconds):
```

```
disp(rrIntervals);
```

```
    0.6995
    0.6796
    0.6596
    0.6596
    0.6396
    0.6596
    0.6596
    0.7195
    0.7795
    0.7595
    0.7595
    0.7395
    0.7195
    0.6596
```

```
disp(['Mean R-R Interval: ', num2str(meanRR), ' s']);
```

```
Mean R-R Interval: 0.69954 s
```

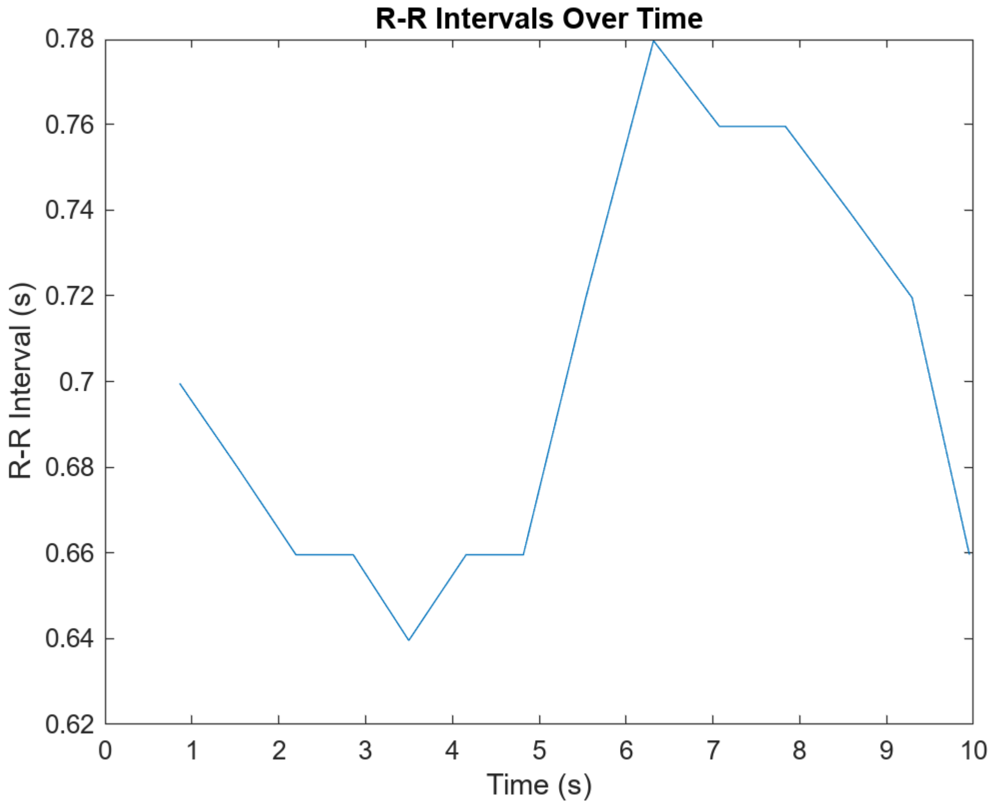
```
disp(['HRV (SD of R-R intervals): ', num2str(stdRR), ' s']);
```

```
HRV (SD of R-R intervals): 0.046379 s
```

```
disp(['Heart Rate (bpm): ', num2str(mean(hr)), ' bpm']);
```

```
Heart Rate (bpm): 86.1142 bpm
```

```

% Plot R-R intervals over time
figure;
plot(locations(2:end) .* mean(diff(timestamps)), rrIntervals);
title('R-R Intervals Over Time');
xlabel('Time (s)');
ylabel('R-R Interval (s)');
```



2-5-

```
% fs = 4; % Target sampling frequency (4 Hz recommended for HRV analysis)
%Sampling freq.
n=length(timestamps);
fs = n / (timestamps(n)-timestamps(1));
timeRR = cumsum(rrIntervals); % Time of each R-R interval
resampledRR = interp1(timeRR, rrIntervals, linspace(timeRR(1), timeRR(end), length(timeRR)*fs)

% Perform FFT
n = length(resampledRR);
frequencies = (-n/2:n/2-1)*(fs/n);
fftResult = abs(fftshift(fft(resampledRR)- mean(resampledRR))); % Subtract mean for zero-center

% Extract LF and HF bands
lfBand = (frequencies >= 0.04 & frequencies <= 0.15); % Low-frequency band
hfBand = (frequencies >= 0.15 & frequencies <= 0.4);  % High-frequency band

% Calculate power in LF and HF bands
lfPower = sum(fftResult(lfBand).^2);
hfPower = sum(fftResult(hfBand).^2);
lfHfRatio = lfPower / hfPower;

% Display frequency domain metrics
disp('Frequency Domain Analysis:');
```

```
Frequency Domain Analysis:
```

```
disp(['LF Power: ', num2str(lfPower)]);
```

```
LF Power: 28.9007
```

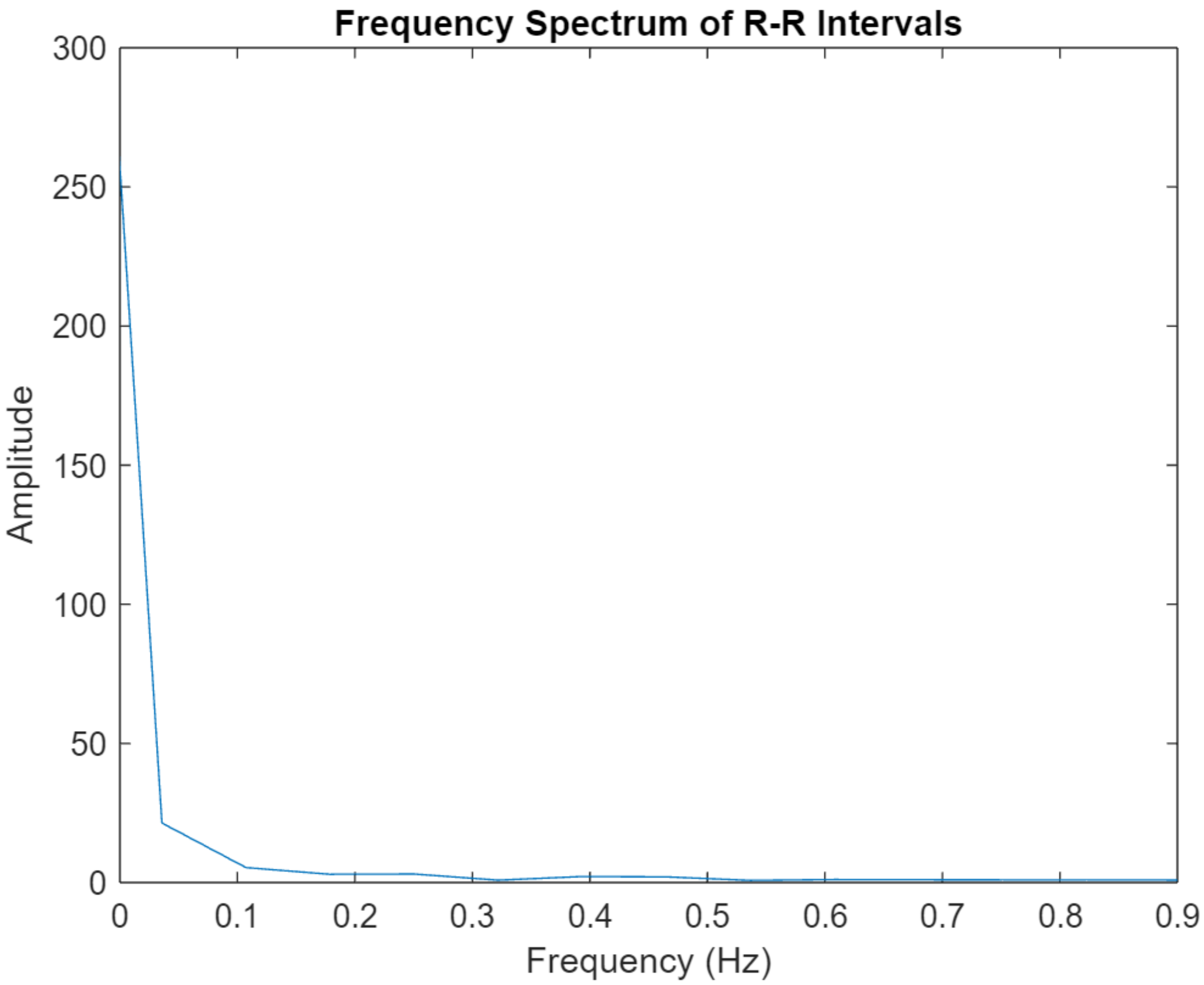
```
disp(['HF Power: ', num2str(hfPower)]);
```

```
HF Power: 23.5849
```

```
disp(['LF/HF Ratio: ', num2str(lfHfRatio)]);
```

```
LF/HF Ratio: 1.2254
```

```

% Plot frequency spectrum
figure;
plot(frequencies, fftResult);
xlim([0 0.9]);
title('Frequency Spectrum of R-R Intervals');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
```



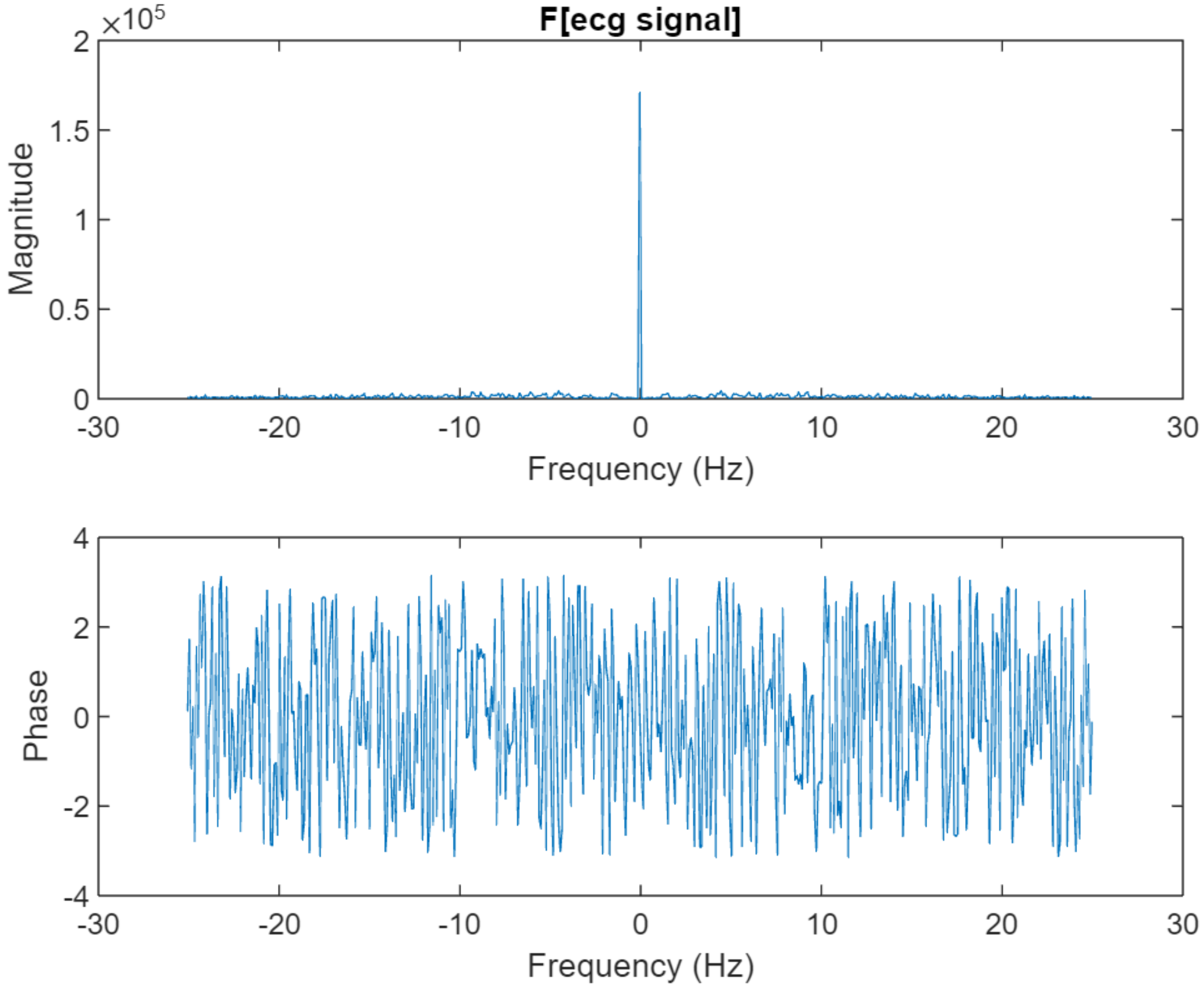
2-7-

```
%Sampling freq.
n=length(timestamps);
Fs = n / (timestamps(n)-timestamps(1))
```

```
Fs = 50.1305
```

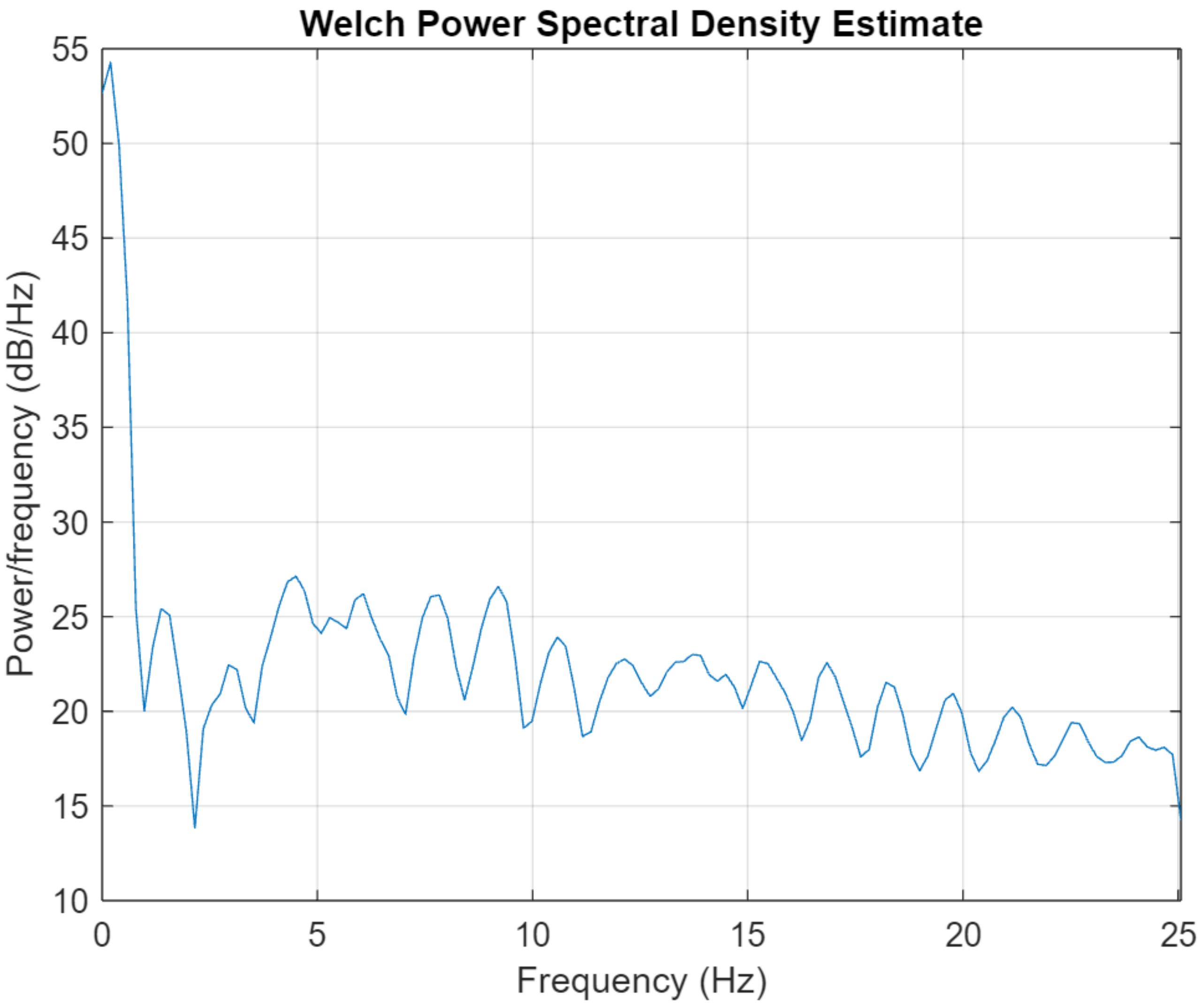
```


F_ecg = fftshift(fft(ecgSignal));
freq_vector = (-n/2:n/2-1)*(Fs/n);


figure;
subplot(2,1,1);
plot(freq_vector,abs(F_ecg));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('F[ecg signal]');
```

```
subplot(2,1,2);
plot(freq_vector,angle(F_ecg));
xlabel('Frequency (Hz)');
ylabel('Phase');
```



2-8-

```
figure;
pwelch(ecgSignal, [], [], [], Fs);
```


Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
55
50
45
40
35
30
25
20
15
10
0
5
10
15
20
25
Frequency (Hz)