Measurement and Data Driven Modeling

Assignment Report

Trade-offs in FRF Measurements

Objective

This assignment aims to explore and demonstrate the practical trade-offs between:

1. Frequency resolution
2. Measurement time
3. Signal-to-Noise Ratio (SNR)

in the context of measuring Frequency Response Functions (FRFs). This understanding is crucial when designing excitation signals for system identification.

Background

Measuring accurate FRFs requires balancing three competing factors:

1. **Frequency Resolution**: Higher resolution allows more frequency lines to be distinguished within a given bandwidth.
2. **Measurement Time**: Longer signals allow more averaging, reducing random noise.
3. **SNR**: The clarity with which excited frequency components emerge from the noise floor.

This trade-off is fundamental and unavoidable in any practical measurement setup. The trade-offs explored in this assignment assume:

- The RMS amplitude of the excitation signal is fixed to $1\,V$.
- The frequency band of interest lies between 5 Hz and 10 Hz.

Multisine Design Summary

The table below summarizes six example multisines designed to illustrate the trade-offs under different constraints:

Table 1: Example Multisine Designs for Each Trade-off Case

Trade-off Case	**Excited Frequencies (Hz)**	**Period (s)**	**Notes**
Fixed Resolution 1	5 to 10 Hz (30 lines)	6 s	Balanced power
Fixed Resolution 2	5 to 10 Hz (30 lines)	12 s	Higher SNR
Fixed Time 1	5 to 10 Hz (30 lines)	6 s	Baseline
Fixed Time 2	5 to 10 Hz (60 lines)	6 s	Finer resolution
Fixed SNR 1	5 to 10 Hz (30 lines)	6 s	Target SNR 40 dB
Fixed SNR 2	5 to 10 Hz (60 lines)	12 s	Same SNR, more lines

Results and Explanation

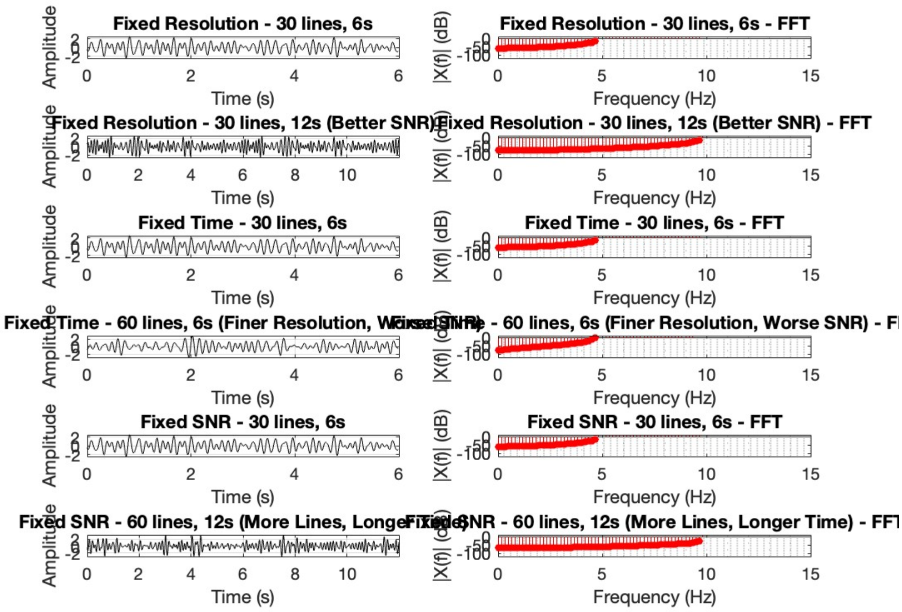


Figure 1: Time and Frequency Domain Plots for Different Multisine Designs Illustrating Trade-offs

Overview

Figure 1 displays the time domain and frequency domain representations of all six multisines. The left column shows the time-domain signals, while the right column shows the corresponding magnitudes of the Discrete Fourier Transform (DFT).

Analysis of Trade-offs

Trade-off 1: Fixed Frequency Resolution

The first two rows represent cases where 30 equally spaced frequencies between 5 Hz and 10 Hz are excited.

- Row 1 uses a 6-second signal.
- Row 2 uses a 12-second signal.

The longer duration reduces noise variance, improving the SNR. This can be seen from the cleaner peaks in the frequency spectrum of Row 2 compared to Row 1.

Physical Reason: A longer time window improves frequency resolution by creating narrower frequency bins, making it easier to distinguish closely spaced frequencies. It also enhances SNR by allowing more averaging, which reduces noise variation and makes signal peaks clearer. This leads to a more accurate and detailed frequency spectrum.

Trade-off: Higher SNR $\rightarrow$ Longer measurement time, Shorter measurement time $\rightarrow$ Lower SNR.

Trade-off 2: Fixed Measurement Time

The third and fourth rows correspond to a fixed 6-second measurement duration.

- Row 3 excites 30 frequencies.
- Row 4 excites 60 frequencies, doubling the resolution.

Since the total power (RMS fixed to 1V) must be spread over more frequencies in Row 4, the SNR per frequency decreases. This leads to less distinct peaks in the frequency domain.

Physical Reason: The fixed measurement time limits the total energy available in the excitation signal. As more frequency lines are excited (increased resolution), the available power per frequency component decreases.

As a result, the peaks in the frequency spectrum become less pronounced relative to noise, this reduces the effective SNR per frequency line. This is a direct consequence of energy spreading across more frequencies.

Trade-off: Higher FR $\rightarrow$ Lower SNR, Lower FR $\rightarrow$ Higher SNR.

Trade-off 3: Fixed SNR

The fifth and sixth rows maintain a constant SNR.

- Row 5 excites 30 frequencies over 6 seconds.
- Row 6 excites 60 frequencies, requiring the measurement duration to increase to 12 seconds to maintain SNR.

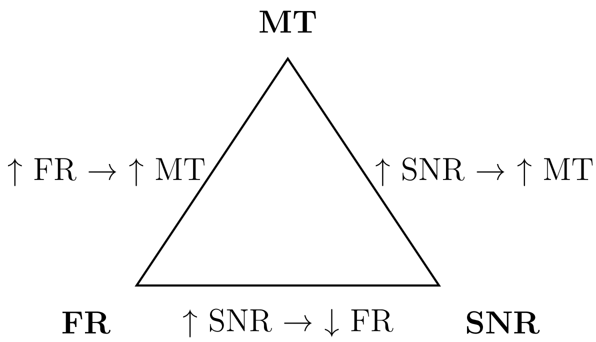
This illustrates that improving resolution (more lines) requires a proportional increase in measurement time to maintain SNR.

Physical Reason: If more frequencies are excited while keeping SNR constant, the measurement must last longer to allow enough noise averaging. This ensures each frequency gets sufficient energy and remains distinguishable from noise. Higher resolution needs longer measurement time for the same noise suppression.

Trade-off: Higher FR $\rightarrow$ Longer measurement time, Shorter measurement time $\rightarrow$ Lower FR.

Summary of Observations

The experiments confirm the core trade-off triangle:



- Increasing measurement time improves SNR at constant resolution.
- Increasing resolution reduces SNR if time is fixed.

- Maintaining SNR while increasing resolution requires longer measurement times.

This triangular trade-off directly reflects the physics of spectral analysis and how energy spreads across time and frequency when the signal RMS is fixed.

Conclusion

Understanding these physical mechanisms is essential for practical FRF measurements. We must carefully select the combination of resolution, time, and SNR to match the goals of a specific system identification experiment. This trade-off framework applies to many real-world scenarios, such as vibration testing, acoustic analysis, and electromagnetic characterization.

MATLAB Code Implementation

The MATLAB code used to generate these results is included below:

```
% Trade-offs in FRF Measurement
clc; clear; close all;

% Global parameters
fs = 100;            % Sampling frequency (Hz)
RMS_des = 1;         % Desired RMS value
T1 = 6;              % Measurement time in seconds
T2 = 12;             % Extended measurement time

% Ensure integer number of periods
N1 = fs * T1;        % 6 seconds worth of samples
N2 = fs * T2;        % 12 seconds worth of samples

% Select frequencies that fit exactly in FFT bins and
   within [5,10] Hz
df = fs / N1;  % Frequency resolution
frequencies1 = linspace(5, 10, 30); % 30 frequencies in
   range [5,10] Hz
frequencies2 = linspace(5, 10, 60); % 60 frequencies in
   range [5,10] Hz

% Display frequencies to confirm range
disp('Frequencies for 30 lines:');
disp(frequencies1);
disp('Frequencies for 60 lines:');
```

```
disp(frequencies2);

% Generate multisines
x1 = generate_multisine(N1, frequencies1, fs, RMS_des);
x2 = generate_multisine(N2, frequencies1, fs, RMS_des);
x3 = x1;  % Fixed Time, same as x1
x4 = generate_multisine(N1, frequencies2, fs, RMS_des);
x5 = x1;  % Fixed SNR, same as x1
x6 = generate_multisine(N2, frequencies2, fs, RMS_des);

% ---- Plot all results ----
multisines = {x1, x2, x3, x4, x5, x6};
titles = {
    'Fixed␣Resolution␣-␣30␣lines,␣6s', ...
    'Fixed␣Resolution␣-␣30␣lines,␣12s␣(Better␣SNR)', ...
    'Fixed␣Time␣-␣30␣lines,␣6s', ...
    'Fixed␣Time␣-␣60␣lines,␣6s␣(Finer␣Resolution,␣Worse␣
       SNR)', ...
    'Fixed␣SNR␣-␣30␣lines,␣6s', ...
    'Fixed␣SNR␣-␣60␣lines,␣12s␣(More␣Lines,␣Longer␣Time)
       '
};

figure;
for i = 1:6
    subplot(6, 2, 2*i-1);
    plot((0:length(multisines{i})-1)/fs, multisines{i},
       'k'); % Black for better contrast
    xlabel('Time␣(s)');
    ylabel('Amplitude');
    title(titles{i});
    grid on;

    % Apply Hann window to reduce leakage (only if
       necessary)
    window = hann(length(multisines{i}))';
    x_windowed = multisines{i} .* window;

    % FFT Calculation
    X = fft(x_windowed);
    X_mag = abs(X(1:N1/2+1));  % Keep only positive half
        of spectrum
    f = (0:N1/2) * (fs/N1);    % Positive frequency axis
```

```
    % Convert to dB scale
    X_dB = 20 * log10(X_mag + 1e-16);

    % Filter out noise floor for clean plot
    threshold_dB = -120;  % Only show components above
       -120 dB
    valid_indices = X_dB > threshold_dB;

    % Reduce plotted points for clarity
    plot_step = 5; % Display every 5th frequency bin

    % Plot log-magnitude spectrum with fewer markers
    subplot(6, 2, 2*i);
    stem(f(valid_indices(1:plot_step:end)), X_dB(
       valid_indices(1:plot_step:end)), ...
         'r', 'filled', 'MarkerSize', 4); % Red markers
            for better visibility
    xlabel('Frequency␣(Hz)');
    ylabel('|X(f)|␣(dB)');
    title([titles{i}, '␣-␣FFT']);
    grid minor;
    ylim([-120, 10]);  % Keeps all FFT plots on the same
        scale
    xlim([0, 15]);     % Focus only on the relevant
       frequency range
end

disp('All␣multisines␣generated␣and␣plotted␣successfully.
   ');

% ===== Helper Function =====
function x = generate_multisine(N, frequencies, fs,
   RMS_des)
    t = (0:N-1) / fs;  % Time vector
    x = zeros(1, N);   % Initialize signal

    for f = frequencies
        phase = 2 * pi * rand();  % Random phase for
           each frequency
        x = x + cos(2*pi*f*t + phase);
    end
```

```
    % Scale to desired RMS
    RMS_x = sqrt(mean(x.^2));
    x = x * (RMS_des / RMS_x);
end
```