Possible classes of input sequences

Sine sweeps

Linear and logarithmic sine sweeps are input signals that are used to identify the frequency response of a system. A sine sweep is a sinusoidal signal that varies in frequency over time. Linear sine sweeps increase or decrease in frequency at a constant rate, while logarithmic sine sweeps increase or decrease in frequency at a rate that is proportional to the logarithm of the frequency.
The general equation for a sine sweep can be written as:

$$x(t) = A \sin(2 \pi f(t)\, t + \phi)$$

where x(t) is the output of the sine sweep at time t, A is the amplitude of the sine sweep, f(t) is the frequency of the sine sweep at time t, t is time, and phi is the initial phase of the sine sweep.
The equation for a linear sine sweep is given by:

$$f(t) = f1 + (f2 - f1) * (t - t1) / (t2 - t1)$$

where f1 is the start frequency, f2 is the stop frequency, t1 is the start time, t2 is the stop time, and t is the time.
For example, if you wanted to generate a linear sine sweep with an amplitude of 1, a start frequency of 100 Hz, a stop frequency of 1000 Hz, a start time of 5 seconds, a stop time of 15 seconds, and an initial phase of 0, you could use the following equation:

$$x(t) = 1 \sin(2 \pi (100 + (1000 - 100) (t - 5) / (15 - 5))\ \ t + 0)$$

This equation would generate a sine sweep that starts at a frequency of 100 Hz at t = 5 seconds, increases to a frequency of 1000 Hz at t = 15 seconds, and has an amplitude of 1.

Similarly, logarithmic sine sweeps, with a different distribution of the harmonic content, can be considered.

Sine sweeps are useful for model identification because they are persistently exciting, meaning that they excite all frequencies over the sweep range. This makes them useful for identifying systems with a wide range of dynamics.

Random Binary Sequences

A random binary sequence (RBS) is a sequence of random binary digits (0's and 1's) that is used as an input signal. RBS signals are often used as persistently exciting inputs for model identification because they have a wide spectrum of frequencies.

The general equation for an RBS signal can be written as:

$$x(n) = (-1) \hat{}\ s(n)$$

where x(n) is the output of the RBS at time n, and s(n) is the nth sample of the RBS.

To generate a random binary sequence in Matlab, you can use the rand() function to generate random numbers between 0 and 1, and then round them to the nearest integer using the round() function. For example, to generate a random binary sequence of length 10, you could use the following code: s = round(rand(1, 10))

To generate a time-varying signal from an RBS, you can use the values in the RBS sequence to switch between two different states. The mod() function in Matlab returns the remainder after division of one number by another.

Here is the algorithm for generating a time-varying signal from an RBS in Matlab:

1. Initialize a time variable t to 0.
2. Initialize a state variable s to the first value in the RBS sequence (s(1)).
3. Initialize an index variable i to 1.
4. While t < T * length(s), do the following: a. Set the output signal y to the value of s. b. Wait for T seconds. c. Increment t by T. d. Increment i by 1. e. Set s to the next value in the RBS sequence (s(mod(i, length(s)) + 1)).

This algorithm will generate a time-varying signal that switches between the high and low states according to the values in the RBS sequence, with a time between switching of T seconds.

The spectrum of the time-varying signal generated from the RBS will depend on the value of T, which is the time between switching states. If T is small, the spectrum of the signal will contain a wider range of frequencies, because the signal is switching states more frequently. On the other hand, if T is large, the spectrum of the signal will contain a narrower range of frequencies, because the signal is switching states less frequently. In general, the spectrum of the time-varying signal will be flat over a range of frequencies that extends from 0 to the inverse of T.