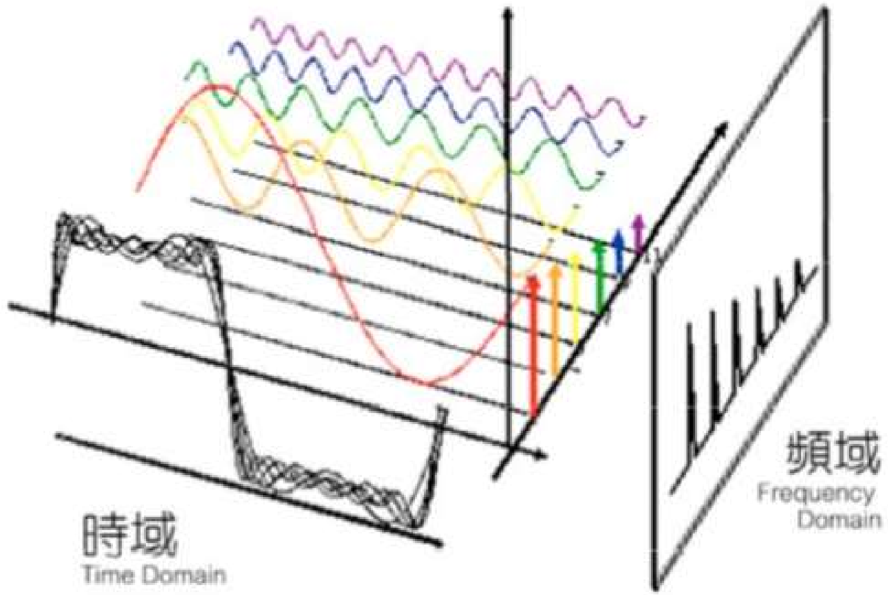
详解快速傅里叶变换 FFT 在 MATLAB 中的实现

- FFT 的由来
- FFT 变换结果详解

为什么要进行傅里叶变换？

将时域的信号，变换到频域的正弦信号

- ✓ 正弦比原信号更简单，且正弦函数很早就被充分地研究，处理正弦信号，比处理原信号更简单
- ✓ 正弦信号的频率保持性：输入为正弦信号，输出仍是正弦信号，幅度和相位可能发生变化，但频率与原信号保持一致；只有正弦信号才拥有这样的性质
- ✓ 一言难尽



傅里叶变换的类型

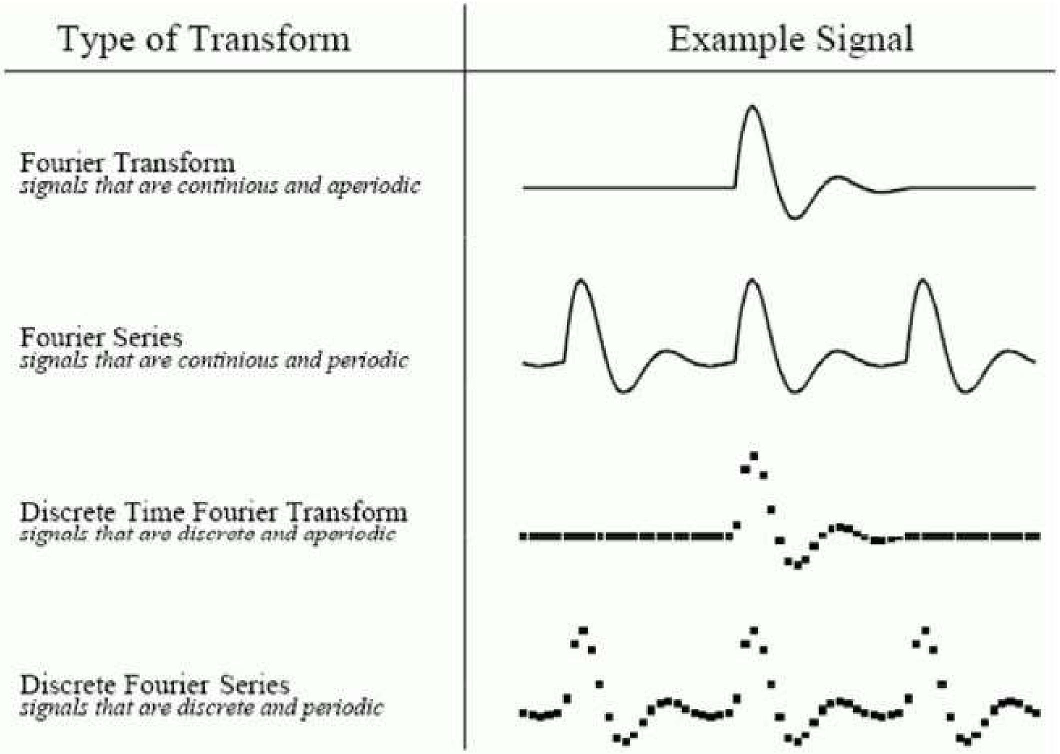


FIGURE 8-2
Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

非周期连续信号：傅里叶变换

周期连续信号：傅里叶级数

非周期离散信号：离散时间傅里叶变换

周期离散信号：离散傅里叶级数

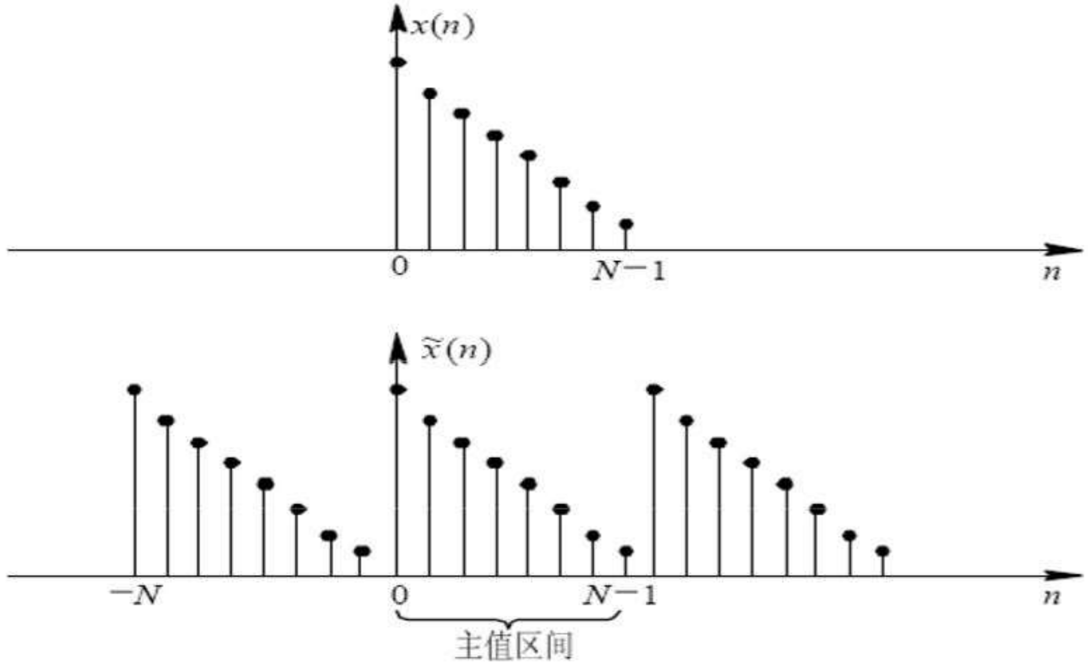
四种信号均为 **(-∞, +∞)** 上的无穷信号，计算机只能处理离散的、有限长度的信号

四种傅里叶变换总结

时域函数	频域函数	傅里叶变换/级数
连续、非周期	非周期、连续	傅里叶变换 (*FT*)
连续、周期	非周期、离散	傅里叶级数 (*FS*)
离散、非周期	周期、连续	离散时间傅里叶变换 (*DTFT*)
离散、周期	周期、离散	离散傅里叶级数 (*DFS*)

- *FT*，*FS*，*DTFT*，至少都有一个域不是离散的，计算机无法处理
- *DFS* 满足时域和频率离散的要求，但其时域为无穷长的周期序列
- 通过对 *DFS* 的推导，得到适合计算机计算的离散傅里叶变换 (*DFT*)

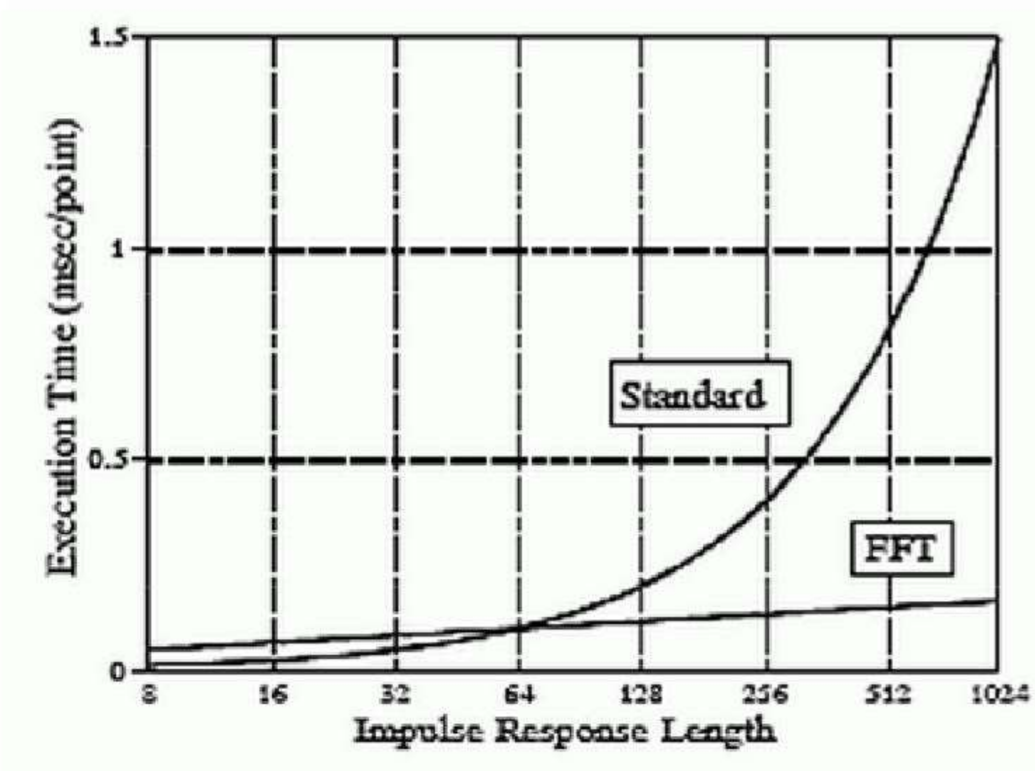
从离散傅里叶级数 (DFS) 到离散傅里叶变换 (DFT)



- 周期序列虽为无穷长序列，但是只要知道一个周期的内容，便可知其全貌
- 因此，周期序列实际上只有 ***N* 个样值**有信息，通过推导可得到 ***DFT*** **(此处略)**
- 时域和频域 (***DFT***) 上的有限长序列，可以用来“代表”周期序列
- ***DFT*** 在时域和频域上均离散，且为有限长序列，可以用计算机进行处理

从离散傅里叶变换 (DFT) 到快速傅里叶变换 (FFT)

- ***DFT*** 虽好，但是其计算的次数太多，不利于大数据量的计算
- ***FFT*** 是 ***DFT*** 的快速算法，可以节省大量的计算时间，其本质仍然是 ***DFT***



MATLAB 中实现 FFT 的计算

Y = fft(x)　　% x 为一个序列 (向量)，存放采集信号的数据

Y = fft(x,n)　% x 的定义同上，n 定义计算数据的个数

- ✓ 如果 n 大于 x 的长度，在 x 的末尾添加0，使得 x 的长度等于 n
- ✓ 如果 n 小于 x 的长度，截取 x 中的前 n 个数来进行计算
- ✓ Y 返回 fft 的结果，为一个复数序列 (向量)
- ✓ **建议：** 采用第一种格式的用法，并且保证 x 的个数为偶数

说明： fft 的其他格式的用法，请参阅 MATLAB 的 **help文档**

MATLAB 中实现 FFT 的计算

实例： 请对以下的信号序列进行 fft 计算，并解读 fft 的运算结果。其中，f_1 = 33 Hz，f_2 = 200 Hz，分别为两个余弦信号的频率。信号采集的频率 Fs = 1000 Hz，采集的数据点个数 N = 2000

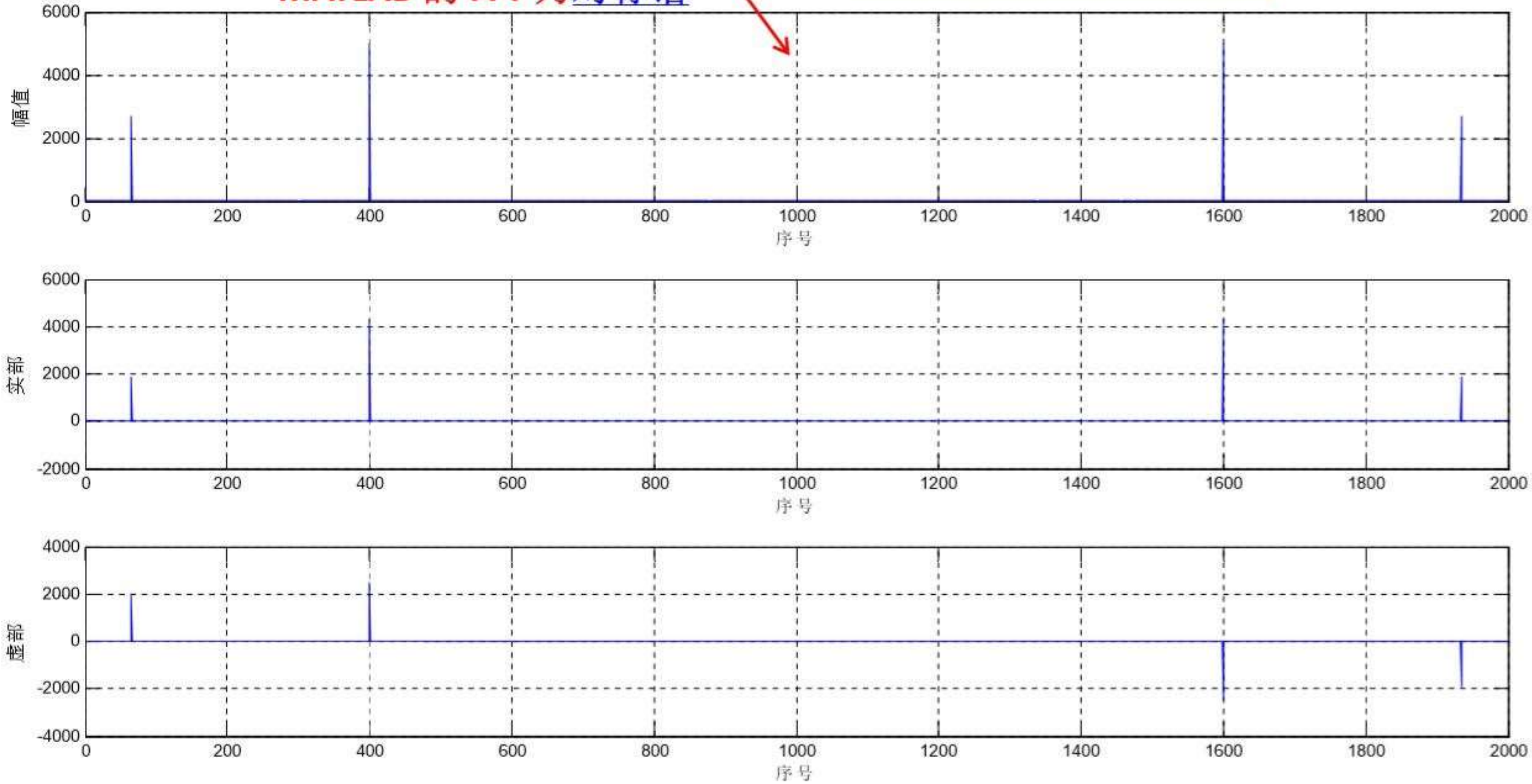
$$y = 1.2 + 2.7\cos\left(2\pi f_1 t + \frac{\pi}{4}\right) + 5\cos\left(2\pi f_2 t + \frac{\pi}{6}\right)$$

先对一个已知的信号，进行 FFT 计算，弄清楚 FFT 输出结果的含义，以及如何去处理和观测 FFT 的输出结果。

MATLAB 中实现 FFT 的计算

频谱关于中间位置对称
(序号位置 0 和 N/2 除外)
MATLAB 的 FFT 为对称谱

看 MATLAB 中 FFT 的频谱，只需要看一半



MATLAB 中实现 FFT 的计算

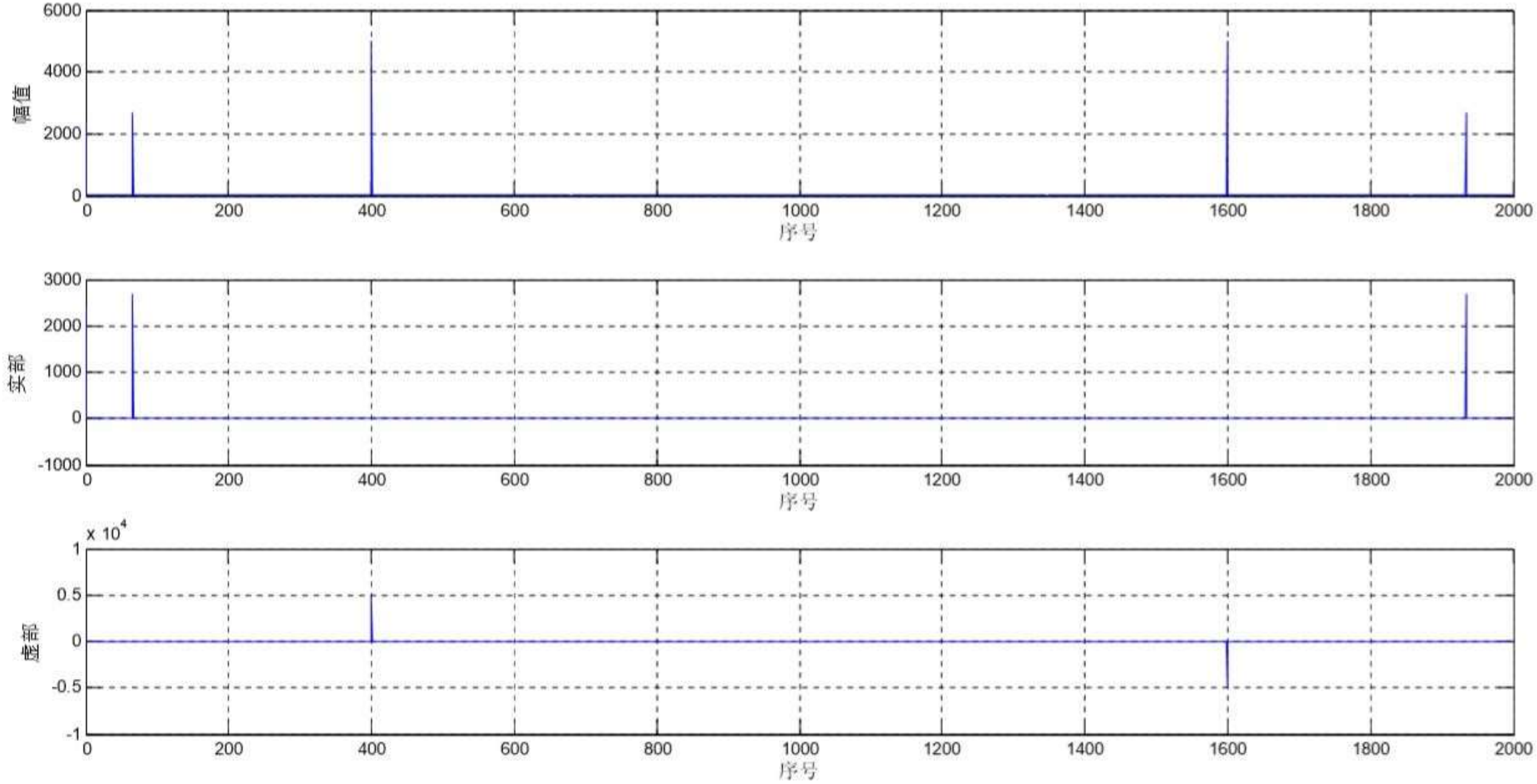
更改设置：将第一个余弦信号的相位改为**0**，第二个余弦信号的相位改为 **π/2**，观察**幅值、实部和虚部**的变化

$$y = 1.2 + 2.7\cos\left(2\pi f_1 t\right) + 5\cos\left(2\pi f_2 t + \frac{\pi}{2}\right)$$

MATLAB 中实现 FFT 的计算

幅值不受影响，但是实部或虚部的值，会出现 **0** 的情况

看 **MATLAB** 中 **FFT** 的频谱，应该看幅值



MATLAB 中实现 FFT 的计算

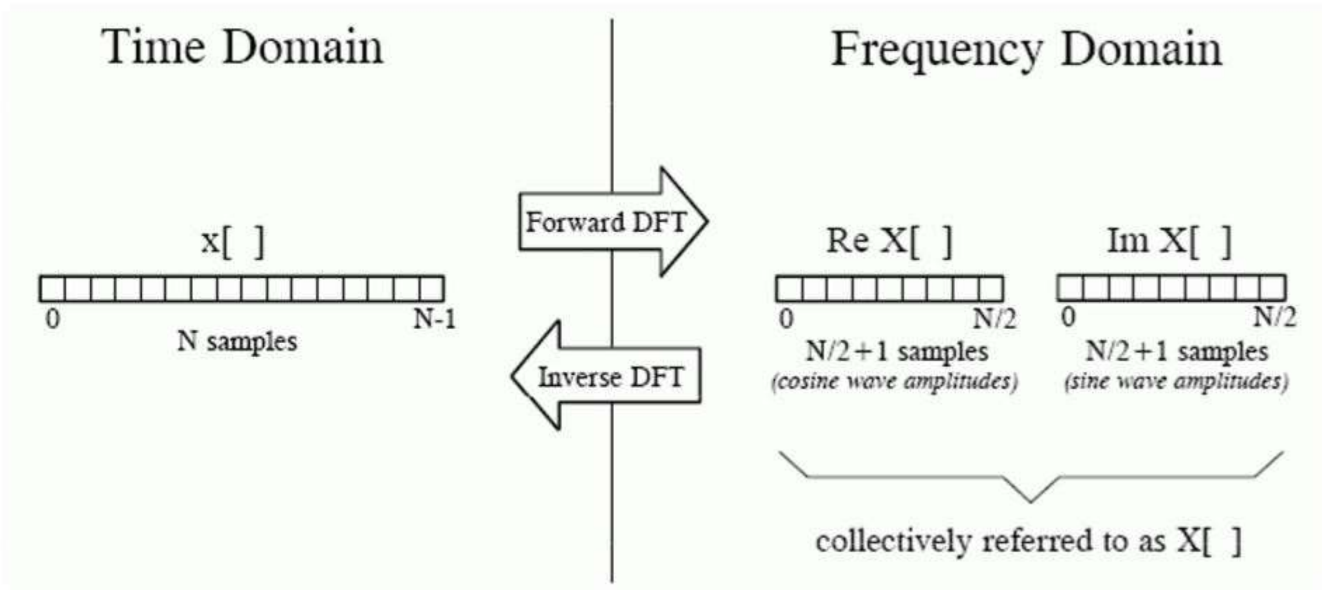
更改设置并绘制频谱：将信号的定义改回 Page 8 中定义，绘制 fft 的频谱图 (幅值半谱图)

$$y = 1.2 + 2.7\cos\left(2\pi f_1 t + \frac{\pi}{4}\right) + 5\cos\left(2\pi f_2 t + \frac{\pi}{6}\right)$$

f_1 = 33 Hz，f_2 = 200 Hz

MATLAB 中实现 FFT 的计算

- ✓ **FFT 结果的数据长度**：时域 ***N*** 个点 **-->** 频域为 ***N/2 +1*** 个点
- ✓ **x 轴频率点的设置**：采样频率为 **Fs** 时，频谱图的最高频率为 **Fs/2** (具体请参照**采样定理**)
- ✓ **综合上述两点**：x 轴的频率点为： ***(0:1:N/2)*Fs/N***



MATLAB 中实现 FFT 的计算

- ✓ 复数的幅值修正：复数序列 Y 的幅值，需要进行转换，才能得到与时域中对应信号的幅值。具体计算方法如下：

$$\text{修正后的幅值} = \begin{cases} \dfrac{2 \times Y\text{的幅值}}{N} & \left(\text{当频率介于}0 \sim Fs/2\text{时}\right) \\ \dfrac{Y\text{的幅值}}{N} & \left(\text{当频率等于}0\text{或者}Fs/2\text{时}\right) \end{cases}$$

- ✓ 复数的相位：计算 Y 的相位，得到与时域中对应信号的相位值
- ✓ 分别查看 **0 Hz, 33 Hz, 200 Hz** 处的幅值和相位角

MATLAB 中实现 FFT 的计算

两个基本问题

- 采样频率为多少合适？
  - 根据采样定理：$F_s \geq 2F_c$，实际应用中需要更大的Fs
- 需要采集多少个点？
  - 频谱图中，频率的坐标间隔（频率分辨率）：**Fs/N**
  - Fs = 2000Hz，N = 100，Fs/N = 20
  - 原信号含有60Hz，72Hz 频率成分，(72 – 60) < 20 **x**
  - N增大至1000，Fs/N = 2，(72 – 60) > 2 **√**

实例： 请分析一个未知的采集信号 (example.mat)，并确定该采集信号的频率成分。其中，信号的采集频率 $Fs = 2500\ Hz$

参考文献

- ✓ **Steven W. Smith, Ph.D. , The Scientist and Engineer's Guide to Digital Signal Processing: http://www.dspguide.com/pdfbook.htm**
- ✓ 郑君里等，信号与系统（第二版），上下册