Solving Differential Equtions with Fourier Transforms

Background

From phasor analysis, if you have a differential equation, such as

$$\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 2y = 4x$$

if you assume all functions are in the form of

$$y(t) = e^{j\omega t}$$

you can rewrite this differential equation as

$$(j\omega)^2 Y + 3(j\omega)Y + 2Y = 4X$$

or write this as a transfer function

$$Y = G(j\omega)X$$

$$Y = \left(\frac{4}{(j\omega)^2+3j\omega+2}\right)X$$

If x(t) is a sine wave

$$x(t) = 2\cos(4t) + 3\sin(4t)$$

then replace x(t) with it's phasor representation

$$X = 2 - j3$$

and find Y by evaluating the gain, G, at the frequency of x(t)

$$Y = \left(\frac{4}{(j\omega)^2+3j\omega+2}\right)_{\omega=4} \cdot (2 - j3)$$

$$Y = (-0.1647 - j0.1412) \cdot (2 - j3)$$

$$Y = -0.7529 + j0.2118$$

Convert back to time to find y(t), (recall that the frequency of the input was 4 rad/sec. The frequency of the output will also be 4 rad/sec)

$$y(t) = -0.7529\cos(4t) - 0.2118\sin(4t)$$

This also works if x(t) contains several frequencies. In this case, use superposition:

- Treat the problem as if there were N separate problems, each with an input, x(t), at a unique frequency
- Find the output at each frequency
- The total outout will be the sum of the outputs at each separate frequency.

If you have a function which is periodic in time T

$$x(t) = x(t+T)$$

but is *not* a sine wave, use your favorite Fourier transform to convert x(t) into a sum of sine waves

$$x(t) = \sum c_n \cdot e^{jn\omega_0 t}$$ *complex Fourier transform*

or

$$x(t) = \sum a_n \cos(n\omega_0 t) + b_n \cos(n\omega_0 t)$$ *sine / cosine Fourier transform*

where ω_0 is the fundamental frequency of x(t):

$$\omega_0 = \frac{2\pi}{T}$$

Now that x(t) is expressed in terms of sine waves, use superposition to solve for y(t).

Example:

Find y(t) given that x and y are related by the following differential equation

$$\frac{dy}{dt} + 3y = 6x$$

x(t) is periodic in 2π

$$x(t) = x(t+2\pi)$$

and

$$x(t) = \begin{cases} 1 & 0 < t < \pi \\ 0 & \pi < t < 2\pi \end{cases}$$

Solution: The fundamental frequency is one

$$\omega_0 = \frac{2\pi}{T} = 1$$

Step 1: Express x(t) in terms of it's Fourier transform. From before

$$x(t) = \frac{1}{2} + \sum_{\text{n odd}} \left(\frac{1}{jn\pi}\right) e^{jnt}$$

Step 2: Find the transfer function from X to Y

$$(j\omega)Y + 3Y = 6X$$

$$(j\omega + 3)Y = 6X$$

$$Y = \left(\frac{6}{j\omega+3}\right) X$$

Step 3: Use superposition and evaluate at each frequency

Harmonic	Frequency	Xn	$G(j\omega)$	Yn
n	$j\omega = jn\omega_0$	$\left(\frac{1}{jn\pi}\right)$	$\left(\frac{6}{j\omega+3}\right)$	$G(j\omega) \cdot X_n$
0	0	0.5	2.000	1.000
1	j1	-j 0.3183	1.800 - j0.600	-0.191 - j0.573
2	j2	0	1.385 - j0.923	0
3	j3	-j 0.1061	1.000 - j1.000	-0.106 - j0.106
4	j4	0	0.720 - j0.960	0
5	j5	-j 0.0637	0.529 - j0.882	-0.056 - j0.034
6	j6	0	0.400 - j0.800	0
7	j7	-j 0.0455	0.310 -j0.724	-0.033 - j0.014

Step 4: Convert back to time. Note that

- Each Yn represents y(t) at a different frequency
- You need to double the compex Fourier transform terms to get cosine and sine terms

$$\begin{aligned} y(t) = \; & 1 - 0.382\cos(t) + 1.146\sin(t) \\ & -0.212\cos(3t) + 0.212\sin(3t) \\ & -0.112\cos(5t) + 0.068\sin(5t) \\ & -0.066\cos(7t) + 0.028\sin(7t) \end{aligned}$$

(You could also use polar form if you like...)

This is actually a *lot* easier in Matlab:

Step 1: Input the complex Fourier transform for X (taken out to 20 terms)

```
X = zeros(20,1);

for n=1:20
   X(n) = (1 - (-1)^n) / (j*2*pi*n);
   end
```

Step 2: Compute G(jw) at each frequency corresponding to n

```
n = [1:20]';
w0 = 1;
w = n*w0;

G = 6 ./ (j*w + 3);
```

Step 3: Compute Y(n): Output is gain times input. Note that G and X are 20x1 matricies: the gain and input at each frequency for n = 1..20

```
Y = G .* X;
```

Also note that you need to use dot-times (element by element multiplication). The dot-notation tells Matlab to treat this as 20 separate problems, not a matrix multiply.

The result is

```
n = [1:20]'
[n, X, G, Y]

    n       X(n)        G(n)               Y(n)
    1.    - 0.318i    1.8 - 0.6i       - 0.191 - 0.573i
    2.      0         1.385 - 0.923i     0
    3.    - 0.106i    1. - i           - 0.106 - 0.106i
    4.      0         0.72 - 0.96i       0
    5.    - 0.064i    0.529 - 0.882i   - 0.056 - 0.034i
    6.      0         0.4 - 0.8i         0
    7.    - 0.045i    0.31 - 0.724i    - 0.033 - 0.014i
    8.      0         0.247 - 0.658i     0
    9.    - 0.035i    0.2 - 0.6i       - 0.021 - 0.007i
    10.     0         0.165 - 0.55i      0
    11.   - 0.029i    0.138 - 0.508i   - 0.015 - 0.004i
    12.     0         0.118 - 0.471i     0
    13.   - 0.024i    0.101 - 0.438i   - 0.011 - 0.002i
    14.     0         0.088 - 0.41i      0
    15.   - 0.021i    0.077 - 0.385i   - 0.008 - 0.002i
    16.     0         0.068 - 0.362i     0
    17.   - 0.019i    0.06 - 0.342i    - 0.006 - 0.001i
    18.     0         0.054 - 0.324i     0
    19.   - 0.017i    0.049 - 0.308i   - 0.005 - 0.001i
    20.     0         0.044 - 0.293i     0
```

To plot y(t), start with the DC term

```
X0 = mean(x)
     0.5

G0 = 6 / (j0 + 3)
      2.0

Y0 = G0 * X0
     1.000
```

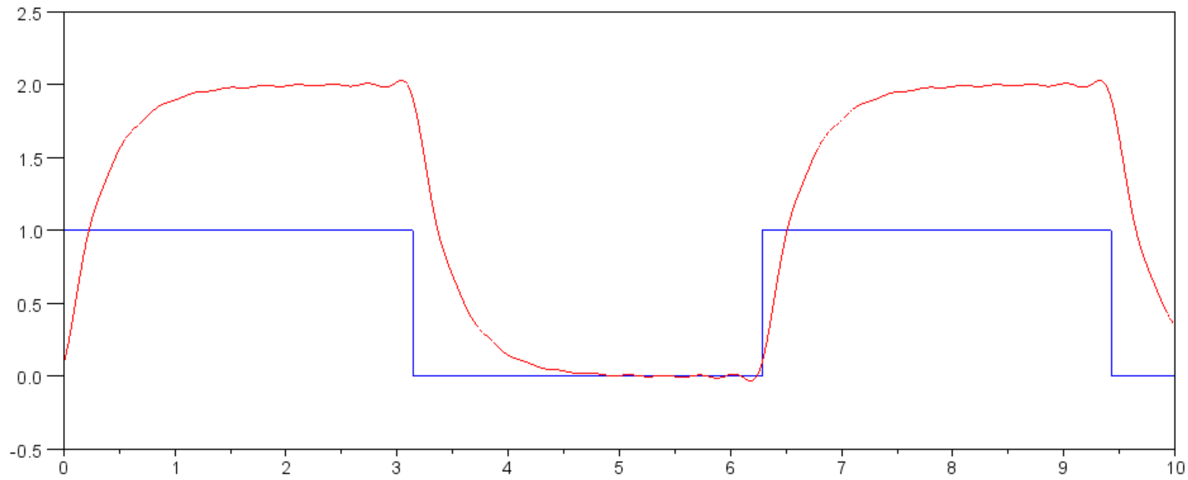
Now add in all the rest of the terms

```
t = [0:0.001:10]';;
x = 1 * (sin(t) > 0);

y = 0*t + Y0;

for n=1:20
   y = y + 2*real(Y(n))*cos(n*t)  -  2*imag(Y(n))*sin(n*t);
   end
```

```
plot(t,x,t,y)
```

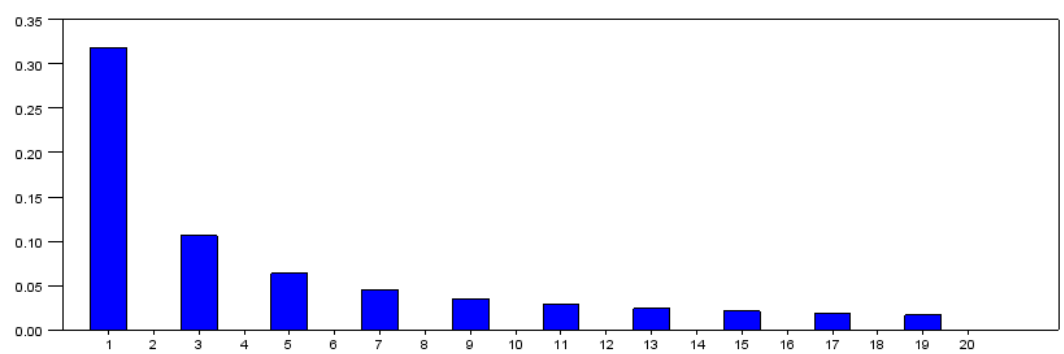


x(t) (blue) and y(t) (red)

In theory, the Fourier transform goes out to infinity. In practice, you only need a few terms to approximate the ouput. This is for two reasons:

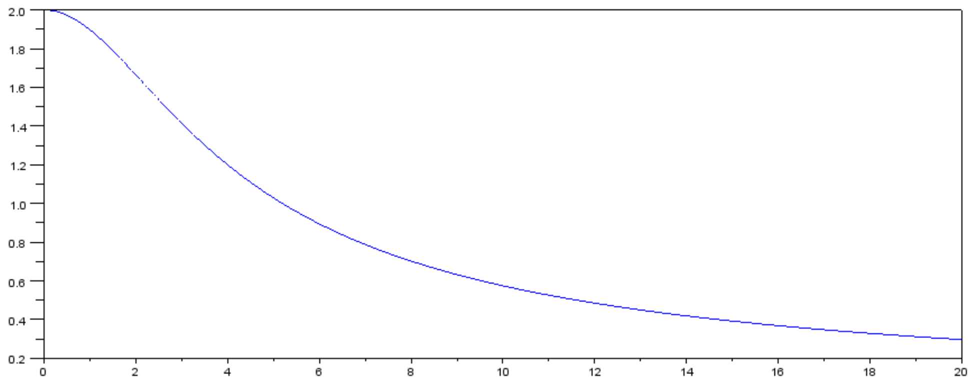
First, the Fourier transform tends to have most of its energy in the lower harmonics. If you plot a bar graph of the magnitude of X(n)

```
bar(abs(X))
```



Amplifude of the Fourier coefficients for x(t)

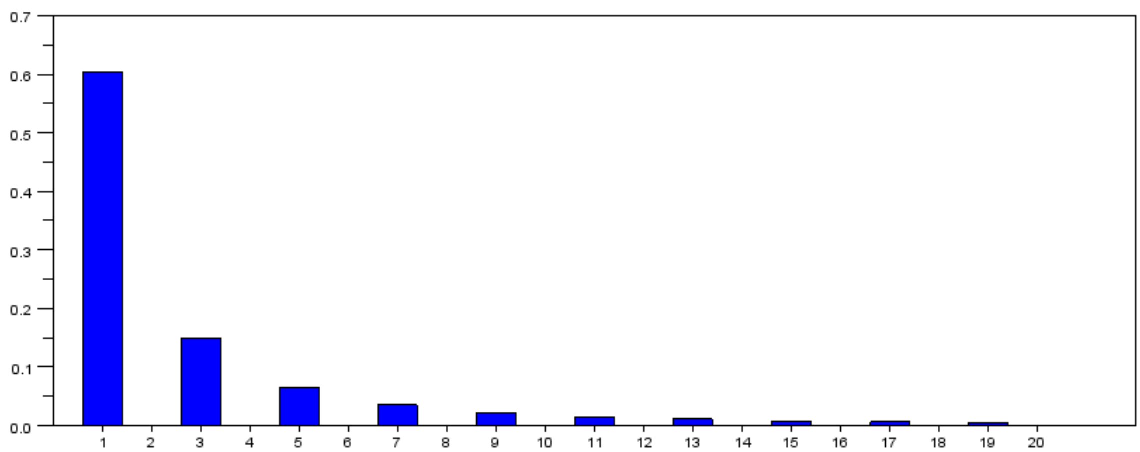
Second, most differential equations act as low pass filters. If you plot the amplitude of G(jw) vs frequency



Magnitude of G(jw) vs. frequnecy

Put the two together (output is gain times input) and you get an output which has most of its energy in the lower harmonics

```
bar(abs(Y))
```



Magnitude of the complex Fourier coefficients of y(t)

In theory, you need to go out to infinity.

In practice, if you only include a few terms (20 in this case), you've captured most of y(t)