Harmonic Balance Method

Project #1

3 December 2015

Daniel Clark

Mackenzie Kirby

Utsav Parmar

Soumith Vodnala

**Format of Harmonic Balance Subroutine**

The subroutine for finding the harmonic balance response begins by creating a ‘sampled frequency’. This involves defining the number of samples the user wishes to take and creating a vector of evenly spaced positions in time. For this analysis, it was determined that we could guess an initial response function which is equal to the forcing function. Therefore *xbar* is set equal to the forcing function which is in turn a function of the time vector.

*N = 99*

*t = np.linspace(0, 10\*np.pi, N+1) # Time sample vector*

*t = t[0:-1] # Removing the extra sample*

*f = np.cos(2\*t) # My forcing function*

*T = t[-1]*

*xbar = f #Initial guess of the response function*

The second step in the analysis is to put the response function into the form of a periodic response using fast Fourier transforms and to calculate the remainder, *R*, when the results are plugged into the original differential equation. This step is incorporated into a subroutine within the process.

*def FUNCTION(xbar):*

*N = len(xbar)*

*Xbar = np.fft.fft(xbar) #Calculate the fft of xbar*

*omega = np.fft.fftfreq(N, T/(2\*np.pi\*N) ) # list of frequencies*

*dotxbar = np.fft.ifft(np.multiply((1j\*omega),Xbar)) #Calculate the derivative of xbar*

*dotdotxbar = np.fft.ifft(np.multiply((1j\*omega)\*\*2,Xbar)) #Calculate the derivative again*

*R = dotdotxbar + dotxbar + xbar - 0\*xbar\*\*3 – f #Substitute into original eqn*

*R = R\*\*2*

*R = np.sum(R) #Calculate the remainder*

*return R*

After the function calculating the remainder has been defined, it can be inserted into a ‘minimize’ function and optimized with respect to the response function. The results of this minimize function are the output of the harmonic balance method.

*optimizedResults = sci.minimize(FUNCTION, xbar, method='SLSQP')*

*xbar = optimizedResults.x*

*print(optimizedResults)*

*print(xbar)*

*pl.scatter(t,xbar)*

*pl.show()*

**Results**

For this project, three functions were evaluated with the harmonic balance function defined above. These functions are shown below:

*The Duffing Oscillator*

*The Nonlinear, Damped Pendulum*

*The Van der Pol Oscillator*

The time response of each of these functions was found using both an integration function and the harmonic balance method. The results are compared with respect to the simulation time and both the qualitative and quantitative difference between the signals.

*The Duffing Oscillator*

When plotted together, the two responses for the duffing oscillator are qualitatively very similar. There is a transient component in the integrated solution which does not appear in the harmonic balance response. However, it disappears after approximately the first cycle. It should also be noted that there is a phase shift in the response which both methods capture. The Root Mean Square Error (RMSE) between the two signals for the specified time sample is 0.0381. As a percentage of the RMS of the integrated solution, the error is 15%. It is reasonable to assume that this value would decrease slightly as time goes to infinity and the effect of the transient response becomes relatively small.

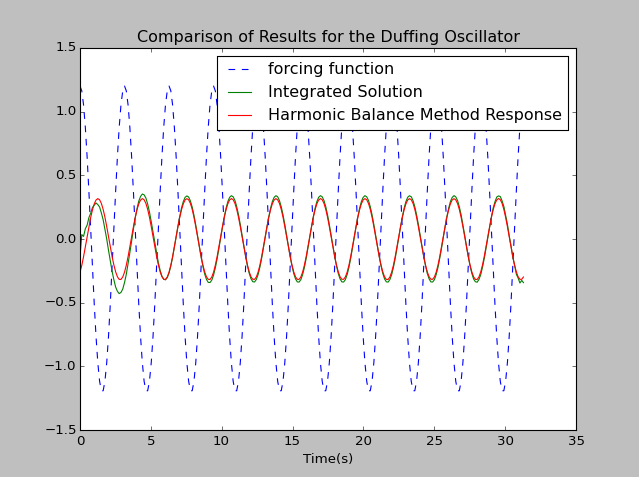
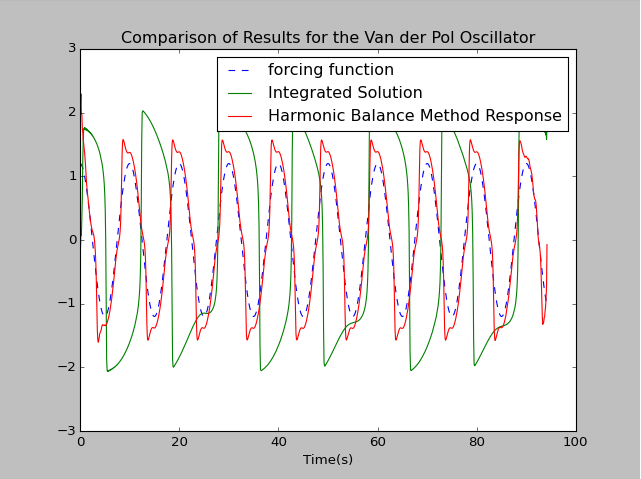


Figure : Calculated result comparison for the Duffing Oscillator (RMSE = 0.0381)

Overall, the harmonic balance method provides

*The Nonlinear, Damped Pendulum*

*The Van der Pol Oscillator*



RSME = 2.08188231465