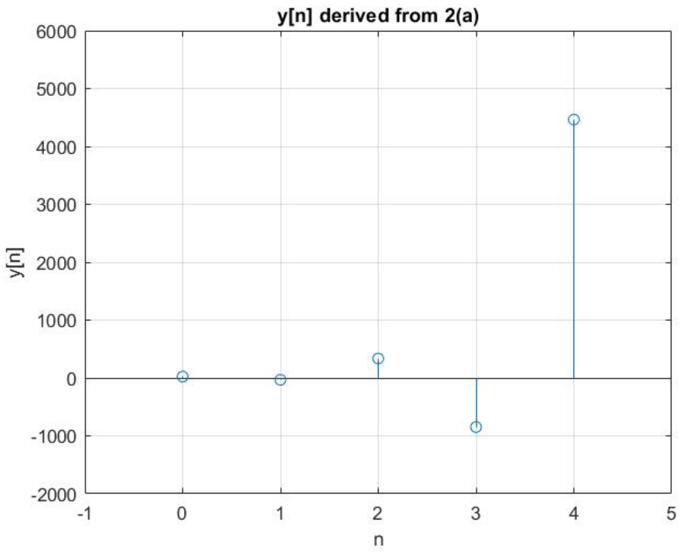
3(a) Implement the LCCDE for 2(a), and determine the output y[n] for 0≤n≤4.

經由第二題的計算，y[n]為： $y[n] = 24 \cdot 2^n + 16 \cdot (-4)^n - 16$

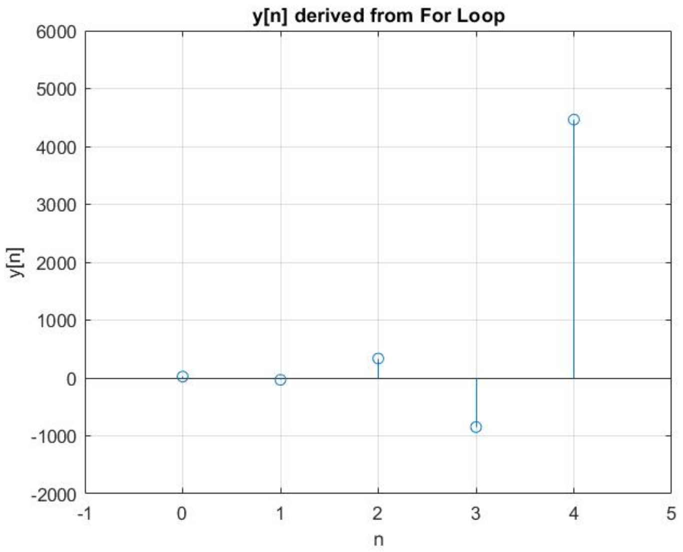
因此在MATLAB裡直接輸入 `y = 24.*2.^n + 16.*(-4).^n - 16;`

接著使用 stem，得到以下的結果：



3(b) Use for loop to implement and determine the output y[n] for 0≤n≤4

由於 y[n]與 y[n-1]、y[n-2]、x[n]有關係，接著又知道了 y[-1]與 y[-2]兩者的初始值，因此可以使用迭代的方式，將當下 n 的 y[n]透過代入過去 y[n-1]與 y[n-2]的值來計算。舉例來說，要計算 y[0]的值，就會使用到 y[-1]、y[-2]、x[0]的值，而這三者都是已知的，因此可以算出 y[0]，接著可以利用 y[0]、y[-1]、x[1]算出 y[1]，以此類推。以下為 for loop 之結果：



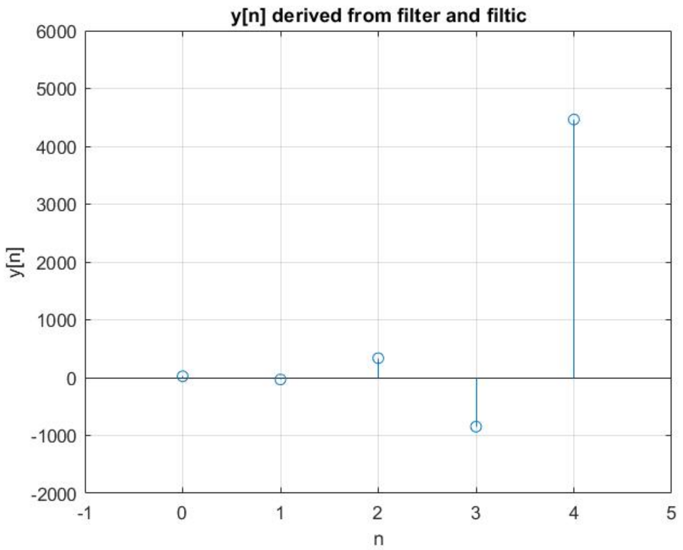
3(c) Use filtic and filter function to determine the output y[n] for 0≤n≤4

使用 filtic 與 filter 函式前，需要先將原函式$y[n]+2y[n-1]-8y[n-2]=80x[n]$的係數先寫成矩陣，如下：

```
a = [1 2 -8]; % coef y
b = [80];     % coef x
y0 = [-8 -9];
x0 = [1];
```

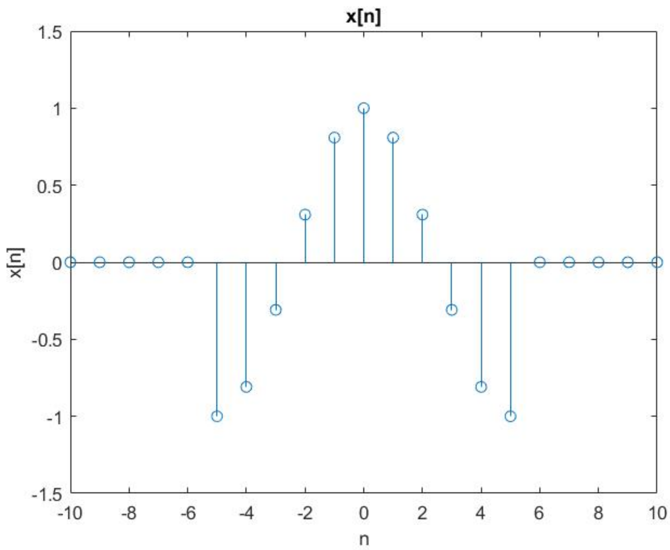
接著將這些矩陣放入 filtic 函式裡，算出零輸入時的響應 zi。

最後將 a, b, zi 代入 filter 函式，可以算出最終的輸出響應，結果如下：



4(a) Generate a sinusoidal signal.

x[n]在 n = -5 ~ 5 之間才有 $\cos(\pi n/5)$的值，其餘都會是 0。因此先創造 n = -5 ~ 5 所對應的 $\cos(\pi n/5)$的矩陣，再將此矩陣的左右邊各補上五個 0，便可以達成要求。



4(b) Use stem function to plot the autocorrelation of $x[n]$.

Autocorrelation 與 convolution 相似，兩者公式如下：

$$r_{xx} = \sum_{k=-\infty}^{\infty} x[k]\boldsymbol{x}[\boldsymbol{k}-\boldsymbol{n}] \qquad x \circledast x = \sum_{k=-\infty}^{\infty} x[k]\boldsymbol{x}[\boldsymbol{n}-\boldsymbol{k}]$$

兩者差別在於訊號是否有先左右翻轉，再相乘後相加。不過現在的 x[n]是對稱的，因此 autocorrelation 和 convolution 會是相同的，因此可以直接使用 conv 函式來計算，結果如下：