**ENGR 2013: Engineering Analysis and Applications**

**Laboratory No. 4: Can You Discover Hidden Structure in a Signal?**

**Goal:** This lab demonstrates how we can fit a model to data and then learn something about the underlying structure of the data.

**Preliminary Work:** In this lab we are going to explore the following equation:

$$y\left(t\right)=a\_{1}\sin(\left(2πf\_{1}t\right))+a\_{2}\sin(\left(2πf\_{2}t\right)+w(t))$$

where the unknowns are $a\_{1}$and $a\_{2}$. The last term, $w\left(t\right)$, is a random noise signal. Its values are unknown – think of it as measurement noise.

Write a simple Python program that generates values of $y\left(t\right)$ every $0.01$ secs. Set $f\_{1}=1.0 Hz $and $f\_{2}=2.0 Hz$. Use the Numpy function random to generate random values. Every time you need a random value, call this function:

$w\left(t\right)= $np.random.normal(0,1,1) .

For the unknowns $a\_{1}$and $a\_{2}$, use $a\_{1}=10$ and $a\_{2}=5$.

Your program should print the value of $t$ and $y(t)$ to the terminal separated by a comma for $t=[0,100 secs]$. To capture this data in a file, redirect the program output to a file using this command:

python gen\_signal.py > signal.dat

Your data should look something like this:

0.000000, 0.462001

0.010000, 2.116416

0.020000, 1.787632

…

Look at this code if you need help writing this program:

/data/courses/engr\_2011/current/labs/lab\_04/picone\_joseph/gen\_signal.py

Run your program and generate the file signal.dat, which should have 10,000 lines (use the command “wc ‑l signal.dat” to verify this).

**Tasks:**

1. Find the optimal values of $a\_{1}$ and $a\_{2}$ by writing a program that analyzes the signal and computes the values using matrix inversion.

Hint: Write a Python program that loads signal.dat into memory as a matrix. Pick two values of the signal (e.g., $t\_{1}=0.04$00 and $t\_{2}=1.800$), solve for $a\_{1}$ and $a\_{2}$. Do this for as many different values of $t\_{1}$ and $t\_{2}$ (e.g., 100 different pairs). Then average together all the values of $a\_{1}$ that you computed and average the values of $a\_{2}$. Are these average values, which we refer to as $\overbar{a}\_{1}$and $\overbar{a}\_{2}$, close to the original values? If you use these values to generate $y\left(t\right)$, are the new values close to the original values?

1. Increase the level of the noise. The second argument in *np.random.normal* is the standard deviation, or strength of the noise. Try values of $1$, $10$ and $100$. What happens to your estimates of $a\_{1}$ and $a\_{2}$? Do they get better or worse?

For these tasks, you do not need to generate a formal written report. Capture relevant output into text files and demonstrate your results to the TAs during the check-off process.

**Summary:**

This assignment is actually very similar to lab no. 2, only the format of the data is different. Later in the semester we will learn more powerful techniques for estimate these coefficients from signal data using techniques such a linear regression and least mean square error analysis. Many machine learning technologies today involve computations somewhat similar to these – fitting a model to signal data and then using that model to predict future values of the signal.