16-722 Sensing and Sensors
Lab Assignment 3

Phase-sensitive detection

Design and carry out an experiment that uses phase-sensitive detection (PSD). The recommended approach is the one shown on slide #14 ("Classic lock-in detection setup") of the first Synchronous Detection lecture, with a multiplier and a lowpass filter as the primary processing components. Your carrier can be a sinusoid or a square wave (you will probably find a square wave easier to implement).

The following directions assume that you are using PSD primarily to extract a small signal from large background noise, in which case the role of the PSD is to squeeze out noise that is not only at a different frequency but also at a different phase from your signal. Alternatively, if you prefer, you may design a PSD application in which measuring a phase or a phase shift is your primary objective, e.g., an amplitude-modulated laser rangefinder, where the phase difference between outgoing and returned-echo signals is the indication of time of flight.

Design and implement the system, carry out the experiment, and present your results in a traditional laboratory report format. A good write-up will typically include at least these elements:

- your target application (what phenomena will it demonstrate?);
- your design concept (hardware and software);
- the components you actually used;
- your Arduino (or other microprocessor) code;
- any preliminary measurements you made for calibration/learning/etc.;
- the results of your experiment (data/graphs/etc.).

As part of the project you should:

- **Build a signal source.** For example, this could be an audible tone of variable frequency and amplitude, or a light beam of variable brightness. *[This is your Dataset 1.]*
- **Build a detector.** For example, this could be a microphone or a photo cell that captures and records the signal. To be realistic, the detector should be located some appreciable distance away.
- **Include a source of noise.** Depending on your situation, some large noise may already be present; if so, use it (if the noise is already present and unavoidable, then your Dataset 1 should be the "clean" drive signal that you are transmitting, before the noise gets added to it.) If not, then **generate and add an external randomly varying source of technical noise.** For example, this could be a radio (if your signal is audible) or a randomly-varying bright light source (if your signal is optical). Record the received **unmodulated** signal in the presence of this noise. *[This, your unmodulated signal + noise, is your Dataset 2.]*
- **Now modulate your signal** (but not the noise), and record the **modulated** signal in the presence of your noise. *[This is your Dataset 3.]*
- Next, perform PSD. That is, **add a connection between your source and your detector** that informs the detector of the state of the modulation at the source (that is, use the carrier signal as a reference input to the demodulator/mixer), and implement a signal-detection algorithm (i.e., PSD) that is cognizant of the modulation state (phase) and that uses this information to achieve a substantial improvement in the signal-to-noise ratio of your detection. *[This is your Dataset 4.]*
- In your report, describe your application and your system, and be sure to present graphs of Datasets 1-4 above. Dataset 4 should be approximately the same as Dataset 1 (at least, much closer than Dataset 2 is to Dataset 1).

As you know, getting above 90% on the labs involves going above and beyond the bare requirements, either through a highly creative approach, or through high quality of execution. One way that you could do that here would be by comparing the results to some other noise-reduction approach, e.g., frequency-selective filtering. (The more the frequency content of your noise overlaps with the frequency content of your signal, the greater you would expect the benefit of PSD to be.) Another way could be by developing a Simulink model of your system, and discussing differences in performance between the model and the real system.

(The limitations of your instrumentation (e.g., Arduino) will most likely not allow you to attain the truly high performance in noise reduction that makes PSD an essential technique for sensing, but the assignment will enable you to learn first-hand about implementation of the technique.)