Systems and Signals 414 Practical 3: Detection by means of correlation

Aim: Investigate the use of correlation in the detection of a signal corrupted by noise.

Handing in: Hand in via your `stnumber@sun.ac.za` email address. (To restate, use your student number email address! Do not use a special alias email address such as `johnsmith@sun.ac.za`.) Attach both your Jupyter `.ipynb` notebook file, and an `.html` copy of your output (`File` → `Download as` → `.html`). Email this to `sfstreicher+ss414@gmail.com` with subject as `prac3`. In summary, your email should look like this:

```
Recipient : sfstreicher+ss414@gmail.com
Subject   : prac3
Attachment: c:\...\filename.ipynb
Attachment: c:\...\filename.html
```

You may send your work multiple times; only the last submission will be marked. The process is automated (so we will not actually read the emails; i.e. trying to contact us using `sfstreicher+ss414@gmail.com` is futile). Make sure your `.ipynb` file returns your intend output when run from a clean slate (`Kernel` → `Restart`)!

Task: Do the following assignment using Jupyter. Document the task indicating your methodology, theoretical results, numerical results and discussions. Graphs should have labelled axes with the correct units indicated.

Hints: Plot "continuous-time" signals with Matplotlib's `plot` function, and discrete-time signals with `stem` instead. For example `pl.plot(np.arange(10))` and `pl.stem(np.arange(10))`. Here is some useful Jupyter preamble code:

```
In [1]: #All the necessary imports
        %matplotlib inline
        import pylab as pl
        pl.style.use('bmh') #pretty plots
        pl.rcParams['figure.figsize'] = (9, 2)
        import numpy as np

        def zpad(signal, pad_left_right):
            return np.pad(signal, pad_left_right,
                          mode='constant', constant_values=0)

        def prepare_plot(title, y_label, x_label):
            pl.figure()
            pl.title(title)
            pl.ylabel(y_label)
            pl.xlabel(x_label)
```

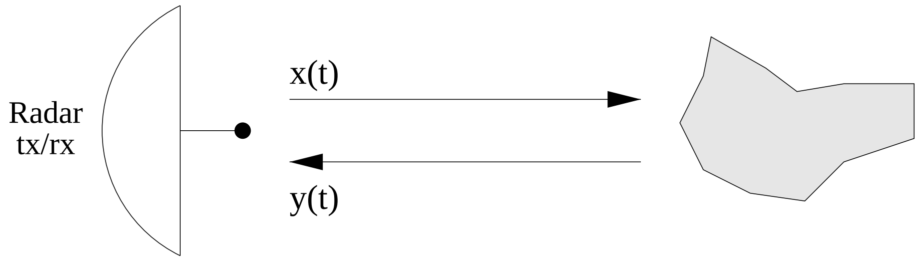
The following function might serve as a helpful `np.correlate` alternative to keep track of indices:

```
In [2]: def corr(x, y):
            idxes = np.arange(-len(y)+1, len(x))
            return idxes, np.correlate(x, y, mode = 'full')
```

Determining a signal delay by means of correlation

Consider a radar system used to determine the distance between its transmitting/receiving antenna and a particular object:

```
In [3]: import IPython.display; IPython.display.display(IPython.display.SVG('radar_image.svg'))
```



The radar transmits a signal $x(t)$, which is reflected by the object and received by the radar as $y(t)$. Now assume that this reflected signal $y(t)$ is a delayed version of the transmitted signal $x(t)$ with additive noise $w(t)$; i.e.

$$y(t) = a \cdot x(t - t_D) + w(t).$$

The distance between the radar and the object may be deduced from the time delay t_D. To proceed, let us assume that $x(t)$ and $y(t)$ have been sampled with a sampling period T, without any aliasing. This results in discrete-time signals $x[n] = x(nT)$ and

$$y[n] = y(nT) = a \cdot x(nT - DT) + w(nT) = a \cdot x[n - D] + w[n],$$

where it has been assumed that $t_D = DT$ with D an integer.

1. Derive an algebraic expression for $r_{yx}[i]$ in terms of $r_{xx}[i]$ and $r_{wx}[i]$. Do this by hand; i.e. pen and paper.

2. Using this result, explain how you could determine D from an estimate of $r_{yx}[i]$. Under what condition(s) will this scheme be effective?

3. Let $x[n]$ be the 13-point Barker sequence:

$$x[n] = \{+1, +1, +1, +1, +1, -1, -1, +1, +1, -1, +1, -1, +1\},$$

and let $w[n]$ be zero-mean Gaussian white noise with a variance $\sigma_w^2 = 0.01$ (this can be generated using `np.random.normal` with appropriate scaling).

   1. Why is this an appropriate choice of $x[n]$? Explain by displaying $r_{xx}[i]$ using `np.correlate`. Note the keyword argument `mode` for `np.correlate` with options `'valid'`, `'same'`, and `'full'`. See `help(np.correlate)` and `help(np.colvolve)` to figure out which one to use.
   2. Take $a = 0.9$ and $D = 20$, and now calculate and plot $y[n]$ for $0 \leq n \leq 199$.
   3. From your result in (3.2), use `np.correlate` to determine $r_{yx}[i]$. Plot the result and identify the delay D.
   4. Repeat (3.2) and (3.3) for $\sigma_w^2 = 0.1$ and for $\sigma_w^2 = 1.0$. What is the significance of σ_w and how does it affect the identification of D?

4. Next, let $x[n]$ instead be the following 13-point sequence:

$$x[n] = \{+1, +1, +1, +1, 0, 0, 0, 0, 0, +1, +1, +1, +1\}.$$

   1. Is this a suitable choice for $x[n]$? Why (or why not)?
   2. Repeat steps (3.2) and (3.3) of Question 3 with $w[n]$ taken as zero-mean Gaussian white noise with a variance $\sigma_w^2 = 0.1$. Can you identify the delay D?

5. Now let $x[n]$ itself consist of 200 samples of Gaussian white noise with variance $\sigma_x^2 = 1.0$.

   1. Is this an appropriate choice for $x[n]$? Why (or why not)?
   2. Repeat steps (3.2) and (3.3) of Question 3 with $w[n]$ taken as zero-mean Gaussian white noise with a variance $\sigma_w^2 = 1.0$. Can you identify the delay D from the plot of $r_{yx}[i]$?

6. Finally, let $x[n]$ consist of 13 samples of Gaussian white noise with variance $\sigma_x^2 = 1.0$. Repeat steps 3.2 and 3.3 of question 3 with $w[n]$ taken as zero-mean white noise with a variance $\sigma_w^2 = 1.0$. How do you results compare with those you obtained from the previous question? Explain your observations. Comment on the advantages and disadvantages of using the Barker sequence over the noise sequences. Under which circumstances would one or the other be a better choice?