Systems and Signals 414 Practical 1: Sampling of continuous-time signals

Aim: Exposure to sampled signals and their quirks, specifically (a) a periodic frequency representation, and (b) the fact that the period of a periodic signal is not necessarily the inverse of its frequency.

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 `prac1`. In summary, your email should look like this:

```
Recipient : sfstreicher+ss414@gmail.com
Subject   : prac1
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 intended output when run from a clean slate!

Task: Do the following assignment using Jupyter. If you struggle with Python, remember the `help()` function. NumPy for MATLAB users and A Python Primer for Matlab Users might also be a useful read. Document the task indicating your methodology, theoretical results, numerical results and discussions. Graphs should have labelled axes with the correct units indicated. Add a code cell at the beginning of the document to include the necessary imports and settings:

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

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))`.

Have a look at `np.linspace` to, for example, generate 50 points linearly spaced from 0 to-but-excluding 10: `np.linspace(0,10,50,False)`. Note the goal of the 4th parameter being set to `False` is to exclude 10 and obtain $0, \ldots, 9.6, 9.8$. If it is set to `True`, we will undesirably obtain $0, \ldots, 9.7959, 10$ (why is this in almoast all our cases undesirable?).

Have you ever wondered why π feels so awkward? Join the "one revolution".

1. Simulation of continuous-time (analogue) sinusoidal signals:

We would like to have an analogue signal generator (like the ones in the fourth-floor lab) to generate signals that can then be digitised using the soundcard of the computer. Unfortunately this is currently not achievable and therefore we will simulate a signal generator in software in this section of the practical.

Implement a function `def sinewave(A,F,t)` that computes the amplitude values of the sine wave

$$x(t) = A\sin(F \cdot 2\pi t)$$

with maximum amplitude A, frequency F in Hz (cycles/second) and t an arbitrary time in seconds. Specifically ensure that t can be a vector of time values, for which x will be the corresponding vector of amplitudes. This function can now be regarded as a symbolic entity that simulates an analogue sine-wave generator. Test it by setting the parameters to `A = 2` and `F = 50` and checking the results by hand.

2. Sampling of the continuous-time signal:

Take the sample indices as $n = 0, 1, \ldots, 39$, see `np.arange(40)`, and the sampling frequency as $f_s = 2000\,\text{Hz}$. Let $A = 10$ and $t = nT$ in all the following cases, where $T = 1/f_s$ is the sampling period. Evaluate the above function for

$$F = 0, 100, 900, 1000, 1100, 1900, 2000, \text{ and } 2100\,\text{Hz}$$

and plot the result using the Matplotlib function stem. Provide your plots with correctly scaled axes so that you can directly read off the correct time (in seconds) and amplitude. Explain your observations.

How would you determine the digital frequency f_ω (cycles/sample) in the above cases? Verify your simulator above by determining the function

$$x[n] = A\sin(f_\omega \cdot 2\pi n)$$

directly for one of the above cases.

3. Look closely:

You probably would have wondered about the strange shape of the sampled 900 Hz signal in Question 2. Sample this signal again, but this time let

$$n = 0, ..., 799 \quad \text{and} \quad f_s = 40000\,\text{Hz}.$$

Plot the result using `plot` together with your previous version (plotted with `stem`) and Comment on the result. (Repeat this process for the other signals above to obtain extra insight.)

4. Sampling a sum of sinusoids:

Now use your function `sinewave(A,F,t)` to superimpose a 100 Hz and a 2100 Hz signal, each with an amplitude of 10, i.e. generate the signal

$$x(t) = 10\sin(100 \cdot 2\pi t) + 10sin(2100 \cdot 2\pi t).$$

As before, take the sample indices as $n = 0, 1, \ldots, 39$ and the sampling frequency as $f_s = 2000\,\text{Hz}$, with $t = nT$.

Plot the sampled signal, and explain your observations.

5. The period of a discrete-time signal:

What are the periods T_p (in seconds) of the continuous-time 100 Hz and 900 Hz signals $x(t)$ above?

What are the periods N_p (in samples) of their sampled versions $x[n]$? (No, the sampled 900 Hz signal does not have a period of 2.2. Make sure that $x[n + N_p] = x[n], \forall n$.)

What is the equivalent period in terms of seconds? Explain!

What would N_p be for $\omega = 900\,\text{rad/s}$? (No, it is not 13.96.)