

Retake «An introduction to real-time programming (with python)»

28th June 2024

Retake the Banyan part of the lab work, that is: plotting streamed data coming from the Arduino board using two processes communicating through the Banyan backplane. Instead of hooking up your Arduino board, you will use the following recursive data generation function to mimick a data stream in python:

```
from scipy.stats import invgauss, norm from math import sqrt
def generate_next_sample(current_sample=None, tau=.01, sigma=1):
    ''' generator of an irregularly sampled time series
    Arguments:
    - current_sample: of the form (t, xt) where t is a timestamp and xt a value
    - tau: the time constant for the generation of the successive values of t
    - sigma: the variance of the values xt
    Returns:
    - (t, xt): the next sample where t is a timestamp and xt a value
    Example use:
    (t,xt) = generate_next_sample() # will return (0, 0)
    while True: # generate and print samples until the time limit of 10s is reached
        (t, xt) = generate_next_sample((t, xt))
        if t < 10:
            print(t, xt)
        else:
            break
    '''
    if current_sample is None:
        return 0, 0
    else:
        T = invgauss(mu=tau)
        dt = T.rvs()
        X = norm(loc=0, scale=sigma)
        t, xt = current_sample
        t += dt
        xt += X.rvs() * sqrt(dt)
        return t, xt
```

Since you cannot just stream the data to your plotting client (since plotting is a blocking event), you must first collect half a second of data on the server side (you may generate these data all at once on the server side, you do not need to implement a third, independent sample-generating process). For this to work, you must of course configure your buffers on both sides so as to give an updated figure every half a second, showing only the last two seconds of data (use the stem function to plot your data).

Your work is to be submitted as a lab report serving as a user manual for your code and specific design choices in the code need to be discussed (choice of the buffer, choice of a specific time constant, ...) together with a fully functional code (as one .py file or multiple .py files zipped together in a single zip file). Date of submission: 25 August 2024, submission on the Chamilo platform, informing me by email when done so.