Tarefa M3 - PDS 2024-2

OBS: Os sinais enviados estão com frequência de amostragem (FS = 8kHz), mono e quantizados em 16 bits.

1) Considere a identificação de um sistema desconhecido via filtro adaptativo (Fig 1) e os sinais de entrada x[n] (ruido_branco.pcm) e d[n] (dn_Q1.pcm), obtenha :
    a) Faça um programa em Python para executar o sistema. Plote o sinal de erro para diferentes valores do passo de adaptação (u) e tamanho do filtro adaptativo. Para quais valores o algoritmo converge (e[n] tende a zero).
    b) Dado os valores que fizeram o algoritmo adaptativo convergir faça um programa em "C" para executar o sistema. Salve o sinal de saída (erro_Q1_C.pcm) e o carregue no Ocenaudio ou em Python.
    c) Faça esse código em "C" rodar no VisualDSP. Salve o sinal de saída (erro_Q1_VDSP.pcm) e o carregue no Ocenaudio ou em Python.

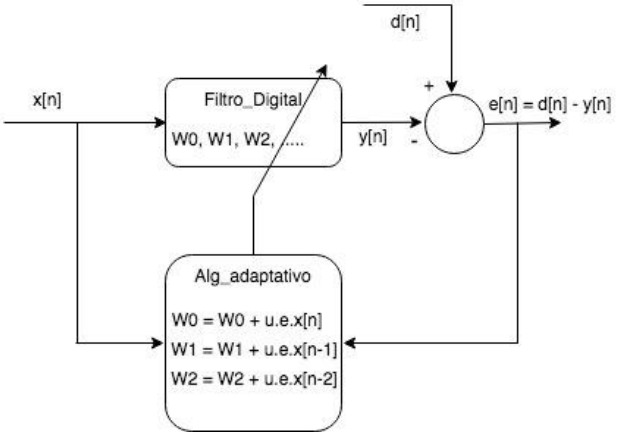

Fig 1- Identificação de um sistema desconhecido.

2) Utilize o Python para projetar um filtro IIR passa-alta de segunda ordem com frequência de corte (fc = 400Hz) e frequência de amostragem (FS = 8kHz), obtenha:
    a) Utilize o sinal x[n] (ruido_branco.pcm) como entrada do filtro IIR e gere sua saída correspondente, denominando de dn_Q2.pcm.
    b) Utilize esses sinais x[n] (ruido_branco.pcm) e d[n] (dn_Q2.pcm) como as novas entradas para a identificação de sistemas, mostrado na Fig 1. Faça um programa em Python para executar o sistema. Plote o sinal de erro para diferentes valores do passo de adaptação (u) e tamanho do filtro adaptativo. Para quais valores o algoritmo converge (e[n] tende a zero).
    c) Dado os valores que fizeram o algoritmo adaptativo convergir, faça um programa em "C" para executar o sistema. Salve o sinal de saída (erro_Q2_C.pcm) e o carregue no Ocenaudio ou em Python.
    d) Faça esse código em "C" rodar no VisualDSP. Salve o sinal de saída (erro_Q2_VDSP.pcm) e o carregue no Ocenaudio ou em Python.

3) Considere o programa em “C” do exercício 2 executado no VisualDSP, obtenha :
    a) O número de ciclos de clock (mínimo, máximo e médio) para a execução do programa.
    b) Utilize uma das duas alternativas mostradas em sala para tornar o algoritmo executado mais eficiente. Após executar essa tarefa certifique que o algoritmo ainda funciona, ou seja salve o sinal de erro (erro_Q3_VDSP.pcm) e avalie em Python/Ocenaudio que o algoritmo converge (e[n] tende a zero)
    c) Meça e apresente o novo número de ciclos de clock (mínimo, máximo e médio) para a execução do programa. Faça uma tabela com esses valores mostrando o quanto se obtém de redução de ciclos de clock.