TuanKhai Nguyen
EECE5356_DSP

AR Model Inverse Filter

In the matlab file ARModelInverseFilter.mat are two signals named:

- ytrain
- y

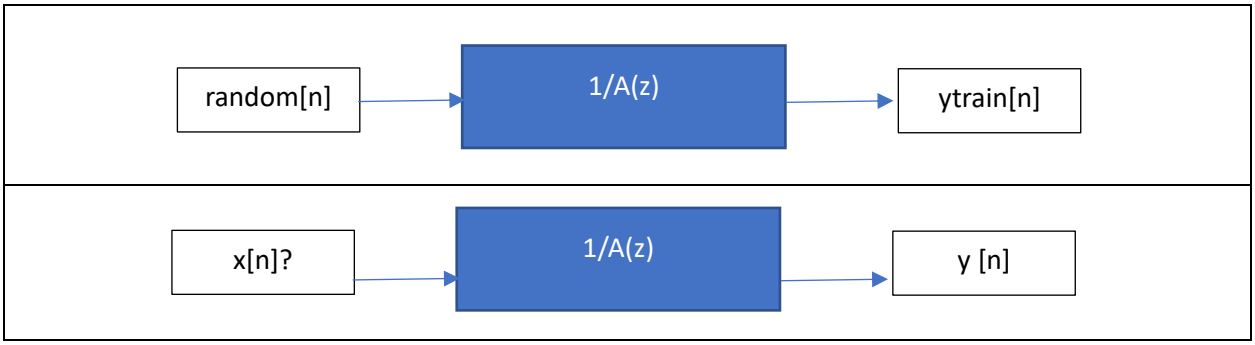
You are to model an unknown communications channel. All you know about it is that it is a **third-order** AR model. That is to say, it's system function is H(z) = 1/A(z) where A(z) is a third order polynomial. You are to estimate the coefficients of this polynomial.

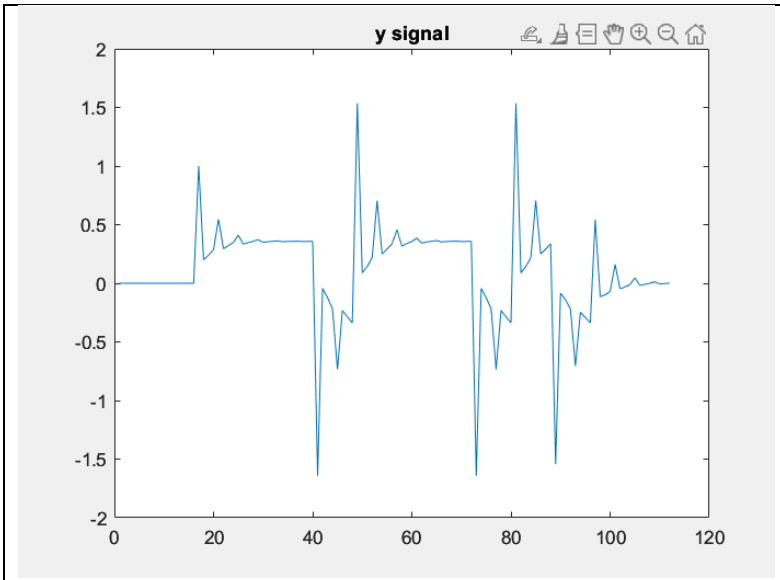
A random noise signal was used as an input to the channel and the resulting signal**, ytrain**, has been observed. Using ytrain, estimate the coefficients of **A(z).**

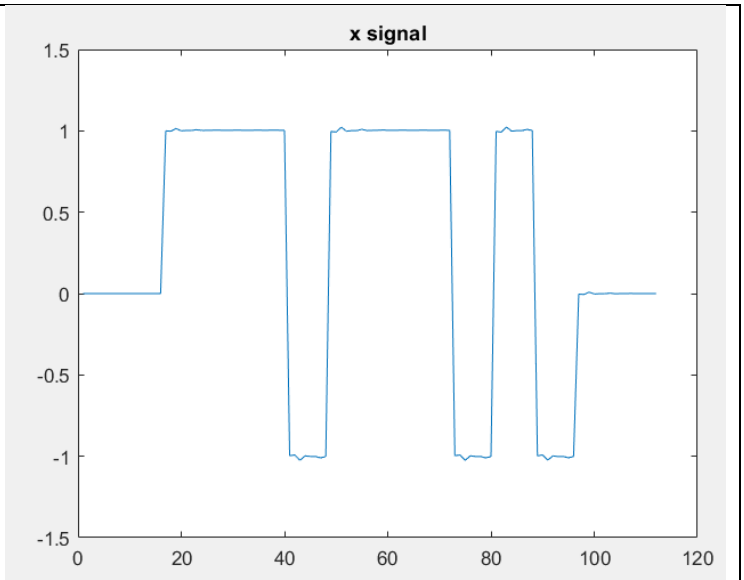
A data signal was then put at the **input of the channel** and the received signal was observed to be **y.** Using your estimated A(z), **inverse filter** the received y to estimate the **original input signal x.**

Plot both y and x.

Turn in the **plots** and **the matlab code** (i.e., the program you write) you used to solve this.








```
p = 3; % 3rd order
N=length(ytrain);
```

TuanKhai Nguyen
EECE5356_DSP

```
ytrain_v = ytrain((p+1):N);
ytrainFirstCol = ytrain(p:(N-1));
ytrainFirstRow=ytrain(p:(-1):1);
Ytrain=toeplitz(ytrainFirstCol,ytrainFirstRow);
ahls=pinv(Ytrain)*ytrain_v
A=[1
    -ahls];
x = filter(A,1,y) % normally if H = B/A we did y = filter (B,A,x)
```

Coefficients of A

```
>> ARsignal

ahls =

   -0.7976
   -0.6146
   -0.3980
```