
```

% TSRT78, Lab 1: Fundamental Signal Processing
% Matheus Bernat (matvi959) & Caspian Süsskind (cassu286)

% ===== 5 Assignment: Vowel =====

% ----- INTRO -----
clear;
% ----- Read wav file: extract data and sampling frequency
x = audioread('aaaaa.wav');
y = audioread('ooooo.wav');
fs = 8000;
Nx = size(x,1);
Ny = size(y,1);
x = detrend(x);
y = detrend(y);
% ----- Plot signal in time axis
Ts = 1/fs;
tx = Ts*(0:Nx-1); % time vector in seconds
ty = Ts*(0:Ny-1); % time vector in seconds

figure(1);clf();
subplot(2,1,1); plot(tx, x); xlabel('time [s]'); ylabel('x[t]');
subplot(2,1,2); plot(ty, y); xlabel('time [s]'); ylabel('y[t]');

% ----- Get the 2 most consistent seconds of both signals
idx_x = (tx >= 1) & (tx <= 3); x = x(idx_x);
idx_y = (ty >= 2) & (ty <= 4); y = y(idx_y);

Nx = size(x, 1); tx = Ts*(0:Nx-1);
Ny = size(y, 1); ty = Ts*(0:Ny-1);

figure(2);clf();
subplot(2,1,1); plot(tx, x); xlabel('time [s]'); ylabel('x[t]');
subplot(2,1,2); plot(ty, y); xlabel('time [s]'); ylabel('y[t]');

% ----- Plot frequency content, count peaks to get model
order
X = fft(x);
Y = fft(y);
fx = (0:Nx-1)*fs/Nx;
fy = (0:Ny-1)*fs/Ny;
figure(3); clf();
subplot(2,1,1); plot(fx, abs(X)); % Order of AR model should be 14*2 =
28
xlabel('frequency [Hz]'); ylabel('X[f]');
title('Frequency spectrum for a')
subplot(2,1,2); plot(fy, abs(Y)); % Order of AR model should be 6*2 =
12
xlabel('frequency [Hz]'); ylabel('Y[f]');
title('Frequency spectrum for o')

```

```

% ----- Model order estimation with loss/AIC/BIC
figure;
subplot(2,1,1);
maxOrder = 50;
arorder(x, maxOrder);
legend('Minimal loss function',...
        'Akaikes information criterion (AIC)',...
        'Akikes information criterion B (BIC)');
title('Order selection analysis of vowel a');
xlabel('Model order');
subplot(2,1,2);
arorder(y, maxOrder);
legend('Minimal loss function',...
        'Akaikes information criterion A (AIC)',...
        'Akikes information criterion B (BIC)');
title('Order selection analysis of vowel o');
xlabel('Model order');

% ----- Create AR models

% AR models for vowel 'a' of orders 2, 9 and 18
estIdx_x = floor(2*Nx/3);
modelOrder_x = 2;
arMod_x_2 = ar(x(1:estIdx_x), modelOrder_x, 'Ts', Ts);
modelOrder_x = 9;
arMod_x_9 = ar(x(1:estIdx_x), modelOrder_x, 'Ts', Ts);
modelOrder_x = 18;
arMod_x_18 = ar(x(1:estIdx_x), modelOrder_x, 'Ts', Ts);

% AR models for vowel 'o' of orders 3, 5 and 10
estIdx_y = floor(2*Ny/3);
modelOrder_y = 3;
arMod_y_3 = ar(y(1:estIdx_y), modelOrder_y, 'Ts', Ts);
modelOrder_y = 5;
arMod_y_5 = ar(y(1:estIdx_y), modelOrder_y, 'Ts', Ts);
modelOrder_y = 10;
arMod_y_10 = ar(y(1:estIdx_y), modelOrder_y, 'Ts', Ts);

% ----- Validation of model -----

% Validation method 1: compare power spectra
f = 0:0.05:1;
Phi1 = arMod_x_18.NoiseVariance*Ts*abs(freqz(1, arMod_x_18.a,
    pi*f)).^2;
Phi2 = arMod_x_9.NoiseVariance*Ts*abs(freqz(1, arMod_x_9.a, pi*f)).^2;
Phi3 = arMod_x_2.NoiseVariance*Ts*abs(freqz(1, arMod_x_2.a, pi*f)).^2;
[Phi4, f4] = sig2blackmantukey(x(estIdx_x+1:end), 30, Ts);
figure;
subplot(2,1,1);
semilogy(fs*f/2,Phi1, fs*f/2,Phi2,fs*f/2, Phi3, f4/2, Phi4);
legend('AR(18)', 'AR(9)', 'AR(2)', 'Blackman-Tukey Estimate');
xlabel('Frequency (Hz)'); title('Spectrum Estimates for a');

```

```

Phi5 = arMod_y_10.NoiseVariance*Ts*abs(freqz(1, arMod_y_10.a,
    pi*f)).^2;
Phi6 = arMod_y_5.NoiseVariance*Ts*abs(freqz(1, arMod_y_5.a, pi*f)).^2;
Phi7 = arMod_y_3.NoiseVariance*Ts*abs(freqz(1, arMod_y_3.a, pi*f)).^2;
[Phi8, f8] = sig2blackmantukey(y(estIdx_x+1:end), 30, Ts);

subplot(2,1,2);
semilogy(fs*f/2,Phi5, fs*f/2,Phi6,fs*f/2, Phi7, f8/2, Phi8);
legend('AR(10)', 'AR(5)', 'AR(3)', 'Blackman-Tukey Estimate');
xlabel('Frequency (Hz)'); title('Spectrum Estimates for o');

% Validation method 2: residual whiteness test
eps_x_18 = pe(arMod_x_18, x(estIdx_x+1:end));
[Rex_18, k] = sig2crosscovfun(eps_x_18, x(estIdx_x+1:end));
eps_x_9 = pe(arMod_x_9, x(estIdx_x+1:end));
Rex_9 = sig2crosscovfun(eps_x_9, x(estIdx_x+1:end));
eps_x_2 = pe(arMod_x_2, x(estIdx_x+1:end));
Rex_2 = sig2crosscovfun(eps_x_2, x(estIdx_x+1:end));

eps_y_10 = pe(arMod_y_10, y(estIdx_y+1:end));
Rey_10 = sig2crosscovfun(eps_y_10, y(estIdx_y+1:end));
eps_y_5 = pe(arMod_y_5, y(estIdx_y+1:end));
Rey_5 = sig2crosscovfun(eps_y_5, y(estIdx_y+1:end));
eps_y_3 = pe(arMod_y_3, y(estIdx_y+1:end));
Rey_3 = sig2crosscovfun(eps_y_3, y(estIdx_y+1:end));

figure;
subplot(2,1,1);
plot(k, Rex_18, 'xm', k, Rex_9, '--b', k, Rex_2, '.-r');
legend('Rea(k) for AR(18)', 'Rea(k) for AR(9)', 'Rea(k) for AR(2)');
title('Cross correlation of residuals for a');
xlabel('k'); ylabel('Rea(k)');
subplot(2,1,2);
plot(k, Rey_10, 'xm', k, Rey_5, '--b', k, Rey_3, '.-r');
legend('Reo(k) for AR(10)', 'Reo(k) for AR(5)', 'Reo(k) for AR(3)');
title('Cross correlation of residuals for o');
xlabel('k'); ylabel('Reo(k)');

% ----- Simulation of model -----
b = 1;
ax = arMod_x_9.A; % coefficients of the AR-model
ay = arMod_y_5.A;

pulseTrain = ones(1, Nx);

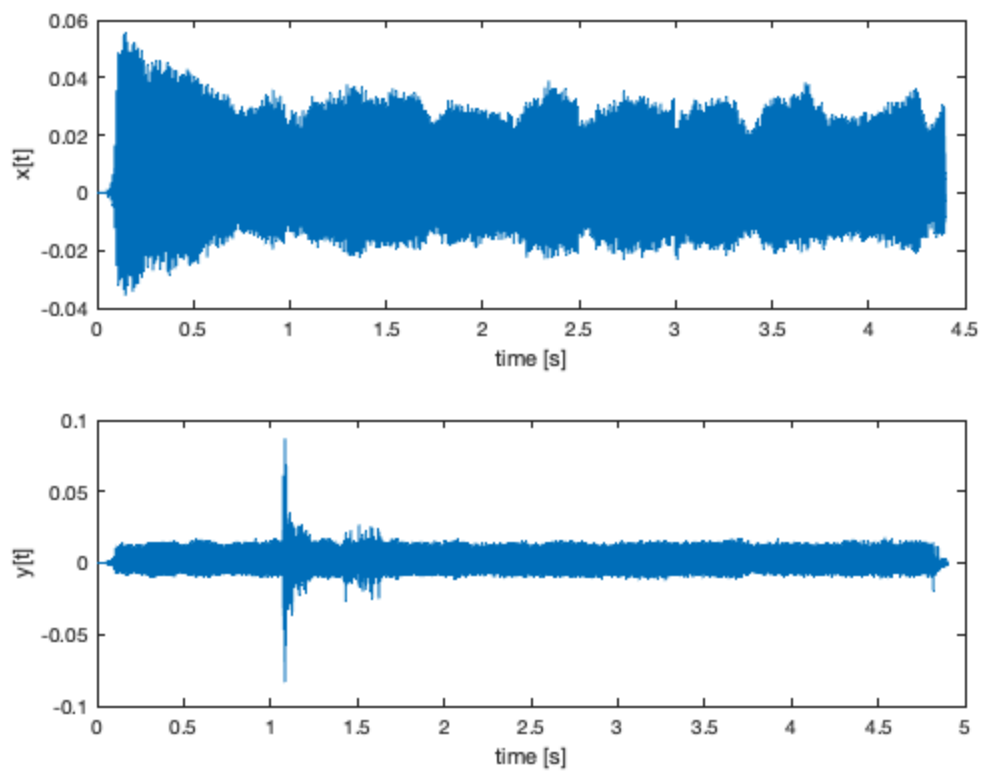
e = filter(ax, 1, x);
r = covf(e, 100);
[A, D] = max(r(20:end));
D = D + 19; % Look at max from t>19, so add 19 to time lag
ehat = (mod(1:Nx, D) == 0);
sim_x = filter(1,ax,ehat);

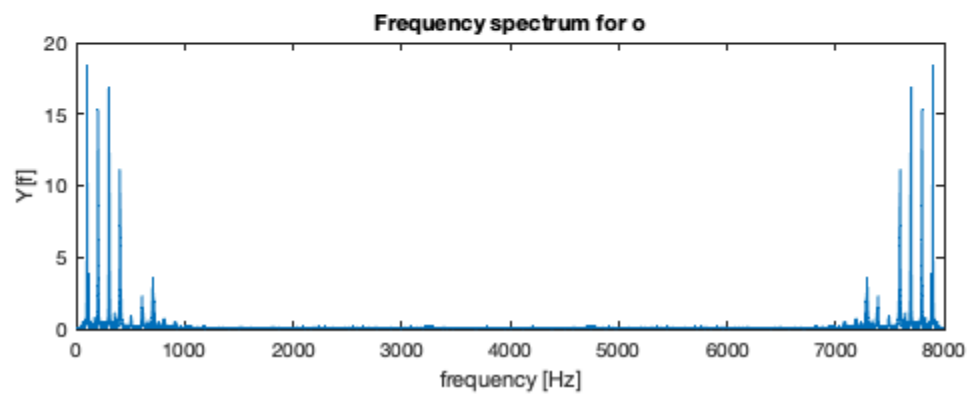
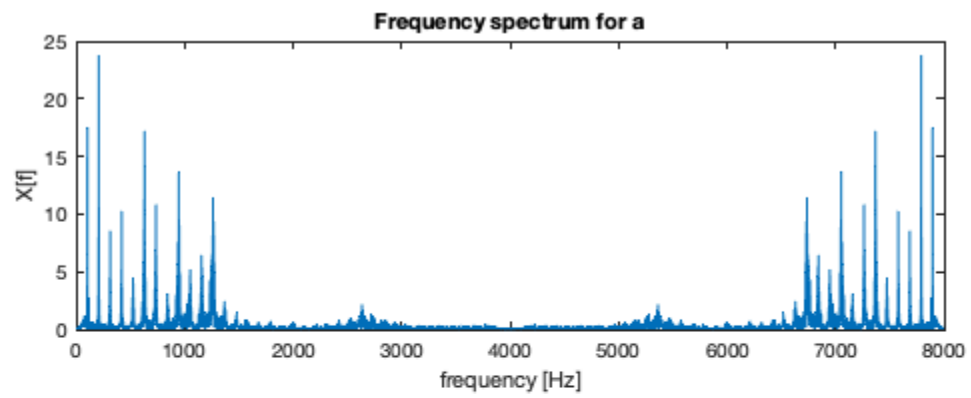
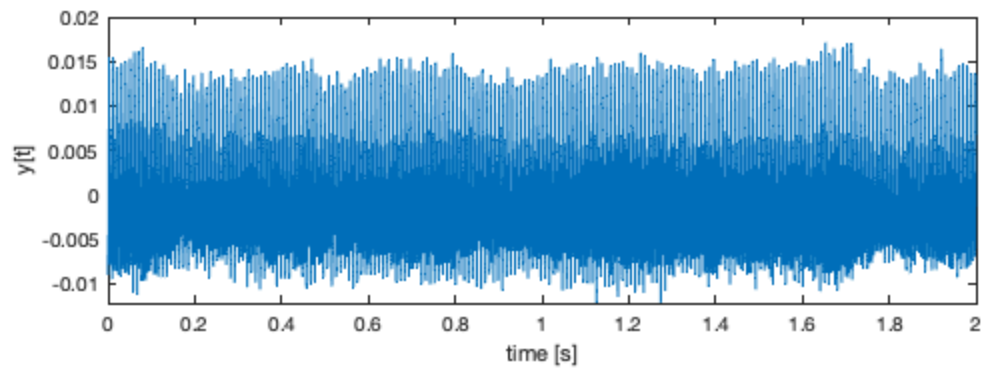
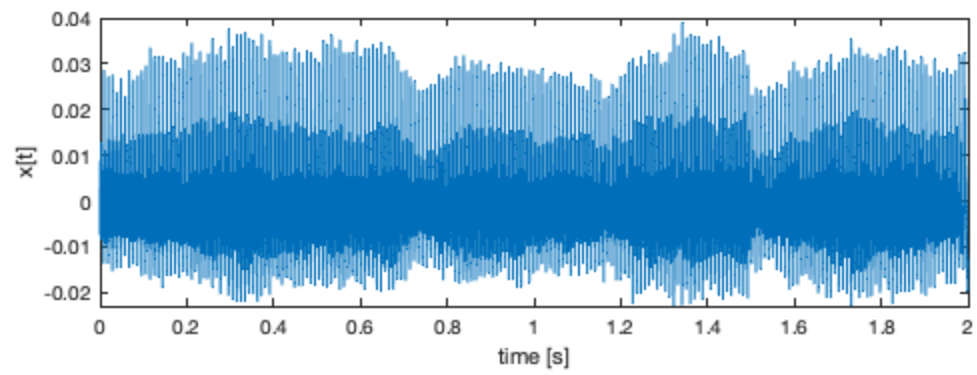
```

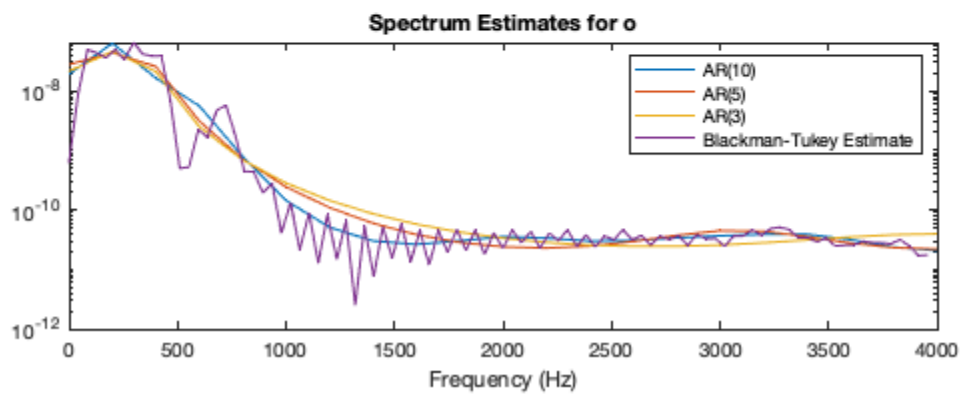
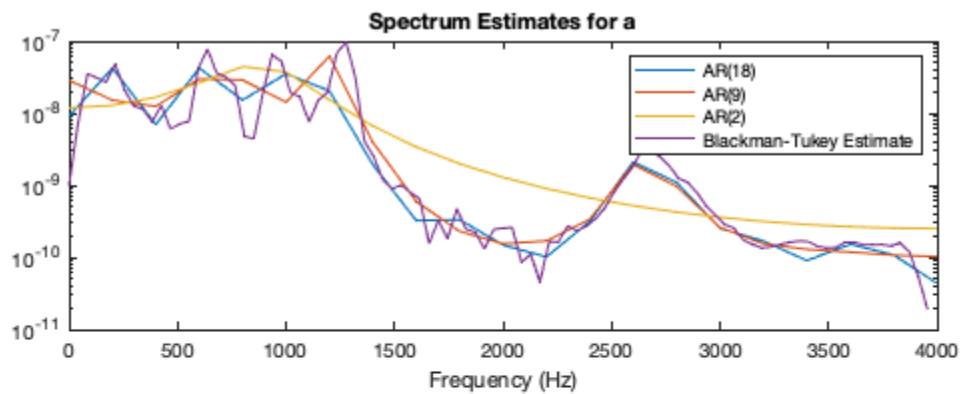
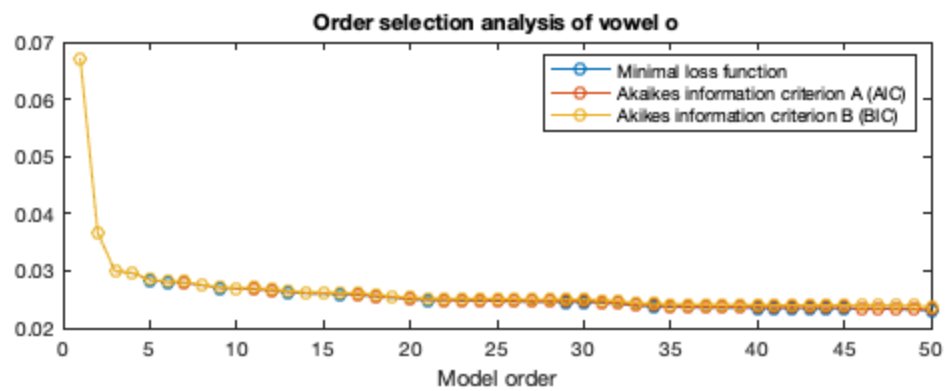
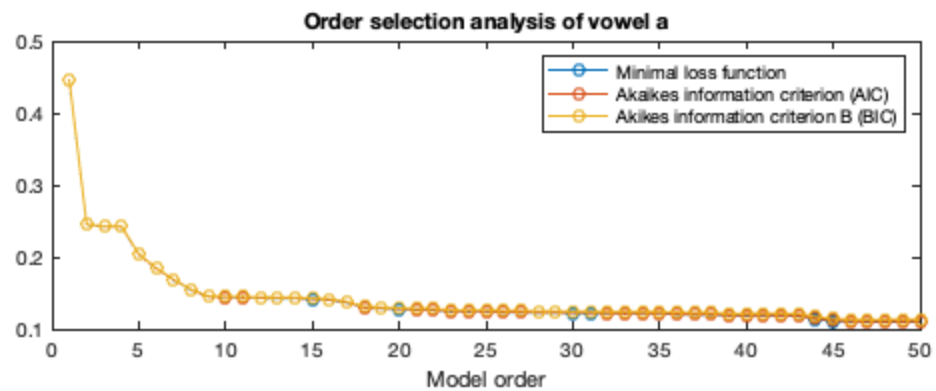
```
e = filter(ay, 1, y);
r = covf(e, 100);
[A, D] = max(r(20:end));
D = D + 19; % Look at max from t>19, so add 19 to time lag
ehat = (mod(1:Ny, D) == 0);
sim_y = filter(1,ay,ehat);

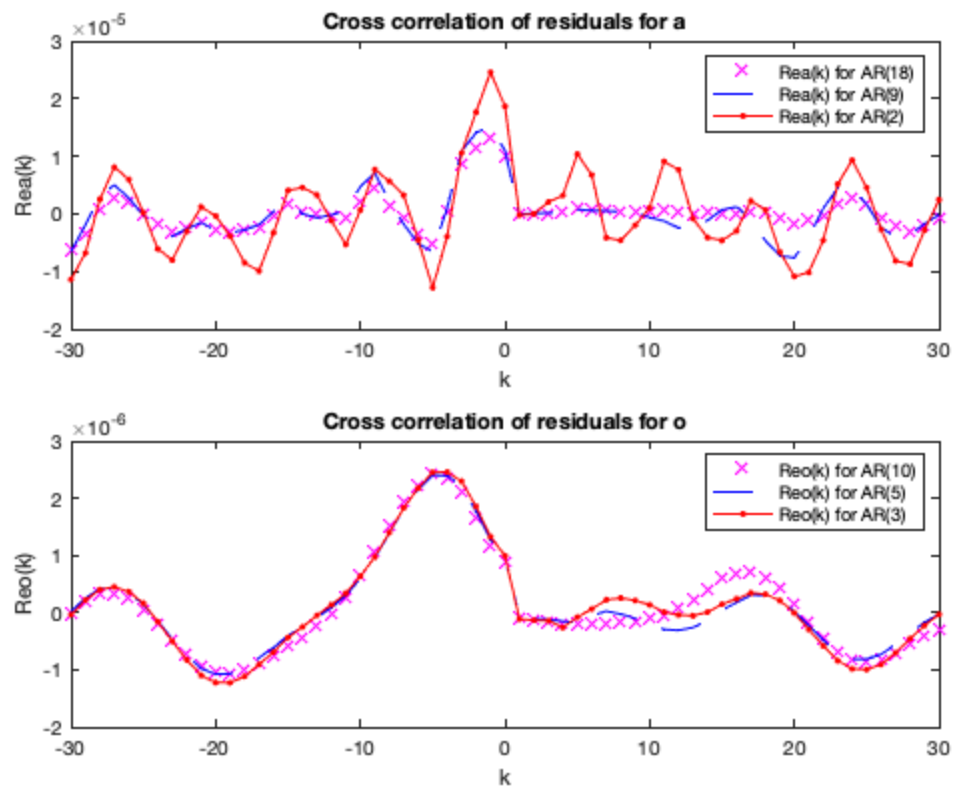
sim_X = fft(sim_x);
sim_Y = fft(sim_y);

sound([sim_x, sim_y]);
```









Published with MATLAB® R2020a