SYSTEMS AND CONTROL
DEPARTMENT OF INFORMATION TECHNOLOGY
UPPSALA UNIVERSITY
http://www.it.uu.se/research/systems_and_control

Useful Functions

Introduction

Below follows a list of the most useful MATLAB functions provided with the Spectral Analysis of Signals textbook. These custom functions follow the notation and definitions in the book. Some useful MATLAB built-in functions are also listed for convenience. For more detailed information regarding each functions, use the `help functionname` command in MATLAB. At least for the custom functions, it is also possible to inspect the code directly.

Custom functions

- `phi=argamse(gamma,a,L)`
  Generates L samples of an ARMA spectral density function ϕ from the ARMA coefficients γ and a.
- `order=armaorder(mvec,sig2,N,nu)`
  Order estimation for a generic ARMA model using: AIC, AICc, GIC, and BIC.
- `phi=armase(b,a,sig2,L)`
  Generates L samples of an AR(MA) spectral density function ϕ from the AR(MA) polynomial coefficients B and A, and the input noise variance σ^2.
- `phi=bartlettse(y,M,L)`
  The Bartlett spectral estimator for estimating the power spectral density (PSD).
- `phi=btse(y,w,L)`
  The Blackman-Tukey spectral estimator for estimating the power spectral density (PSD).
- `phi=capon(y,m,L)`
  The Capon spectral estimator for estimating the power spectral density (PSD).
- `phi=correlogramse(y,L)`
  The correlogram spectral estimator for estimating the power spectral density (PSD).
- `phi=daniellse(y,J,L)`
  The Daniell spectral estimator for estimating the power spectral density (PSD).
- `w=esprit(y,n,m)`
  The ESPRIT method for frequency estimation.
- `[a,varphi]=freqaphi(y,w)`
  Finds the least-squares amplitude and phases for sinusoidal data once the frequencies have been estimated. Uses equation (4.3.8).
- `w=hoyw(y,n,L,M)`
  The Higher-Order Yule-Walker method for frequency estimation.
- `[beta,relMSE,yHat]=lsa(y,w)`
  Computes the complex amplitudes of sinusoidal components given the frequencies by a least-squares fit to the data.

- `[a,sig2]=lsar(y,n)`
  The covariance Least-Squares AR method, given by equation (3.4.12) with $N_1 = n+1$ and $N_2 = N$.

- `[a,b,sig2]=lsarma(y,n,m,K)`
  The two-stage Least-Squares ARMA method, given in section (3.7.2)

- `w=minnorm(y,n,m)`
  The Root Min-Norm method for frequency estimation.

- `w=music(y,n,m)`
  The Root MUSIC method for frequency estimation.

- `[a,gamma]=mywarma(y,n,m,M)`
  The modified Yule-Walker ARMA method given by equation (3.7.8) together with the AR coefficients estimated using the overdetermined set of equation (3.7.4), where W is the identity matrix.

- `phi=periodogramse(y,v,L)`
  The windowed periodogram spectral estimator for estimating the power spectral density (PSD).

- `phi=rfb(y,K,L)`
  The statically stable Refined Filter Bank spectral estimator estimating the power spectral density (PSD).

- `order=sinorder(mvec,sig2,N,nu)`
  Order estimation for sinusodial models using: AIC, AICc, GIC, and BIC.

- `phi=welchse(y,v,K,L)`
  The Welch spectral estimator for estimating the power spectral density (PSD).

- `[a,sig2]=yulewalker(y,n)`
  The Yule-Walker AR method given by equation (3.4.2).

MATLAB built-in functions

- `Y = fft(X)`
  Computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

- `Y = fftshift(X)`
  Rearranges the outputs of fft, fft2, and fftn by moving the zero-frequency component to the center of the array.

- `Y = filter(X)`
  Filters the data in vector X with the filter described by vectors A and B to create the filtered data Y.

- `X = rand(N,1)`
  Returns an N-by-1 matrix (vector) of uniformly distributed independent random numbers.

- `X = randn(N,1)`
  Returns an N-by-1 matrix (vector) of normally distributed independent random numbers (white Gaussian noise).

- `r = roots(p)`
  Returns the roots of the polynomial represented by the vector p, as a column vector.

- `subplot(n,m,p)`
  Divides the current figure into an m-by-n grid and creates an axes for a subplot in the position specified by p.

- `[U,S,V]=svd(X)`
  Singular value decomposition. Produces a diagonal matrix S, of the same dimension as X and with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that $X = USV^*$.

- `zplane(b,a)`
  Plots the poles and zeros of a discrete-time system (AR/ARMA etc) in the complex plane. Note that the inputed coefficient vectors must be row vectors.