SIMULATION OF STOCHASTIC SYSTEMS IN MATLAB

With simulation we mean here an approximative realization of the stochastic process.

Matlab's function randn produces realizations of white noise. A realization \( \{e(k)\}_{k=1}^n \) of the stochastic process \( \{\varepsilon(k)\} \), which is defined as normally distributed white noise with \( E\varepsilon(k) = 0 \) and the variance \( E\varepsilon(k)^2 = 1 \), is produced by the instruction
\[
e = \text{randn}(n,1)
\]

How do you realize a process \( \{v(k)\} \) with \( Ev(k) = 0 \) and \( Ev(k)^2 = 0.81 \)?

\[
v(k) = a\varepsilon(k), \quad Ev(k)^2 = E[a\varepsilon(k)]^2 = a^2E\varepsilon(k)^2 = a^2,
\]

The variance is \( a^2 = 0.81 \), hence \( a = 0.9 \) (= standard deviation). A realization of length \( n \) in Matlab of the process \( \{v(k)\} \) is obtained with the instruction
\[
v = 0.9*\text{randn}(n,1).
\]

Autocovariance, cross covariance and the autocovariance function of a sample

Consider two stochastic processes \( \{\xi(k)\} \) and \( \{\upsilon(k)\} \) with expectations \( E\xi(k) = 0 \) and \( E\theta(k) = 0 \), the autocovariance functions
\[r_{\xi}(h) = E\xi(k)\xi(k-h)
\]
\[r_{\upsilon}(h) = E\upsilon(k)\upsilon(k-h)
\]
and the cross covariance function
\[r_{\xi\upsilon}(h) = E\xi(k)\upsilon(k-h).
\]
The sequence \( \{x(k)\}_{k=1}^N \) is a given realization of \( \{\xi(k)\} \).

The autocovariance function \( r_{\xi}(h) \) can be estimated from the realization using the estimator
\[
\hat{r}_{\xi}(h) = r_{\xi}(h) = \frac{1}{N} \sum_{k=1}^{N-h} x(k+h)x(k)
\]
or using
\[
\hat{r}_{\xi}(h) = r_{\xi}(h) = \frac{1}{N-h} \sum_{k=1}^{N-h} x(k+h)x(k)
\]
Matlab performs the estimations in the function xcov in "Signal Processing Toolbox". In xcov the estimation is called (3) "biased" and the estimation (4) "unbiased". In fact neither (3) or (4) are unbiased. The estimate (3) gives a smaller estimation error. Therefore the estimator (3) "biased" is recommended for the exercises in MCSS.
\[
\hat{r} = \text{xcov}(x, x, \text{'biased'})
\]
gives an estimate \( r_{\xi}(h) \) for \( -(N-1) \leq h \leq (N-1) \). The estimate of the variance, \( r_{\xi}(0) \), is the \( N \)th element in the vector \( r \) given as a result from xcov.
"System Identification Toolbox" in Matlab uses a compact description of dynamic system models. Models are described by objects in "idpoly"-format. We can e.g. identify an ARMAX model using the instruction

```matlab
M = armax(z, nn)
```

The variable M is an object containing the A-, B- and C-polynomials of the ARMAX model. The different polynomials are found as vectors under the notations

- `M.a` coefficients of the A-polynomial
- `M.b` coefficients of the B-polynomial
- `M.c` coefficients of the C-polynomial.

See closer definitions with the instruction `set(idpoly)`.

**Theta format**

In older versions of Matlab the "System Identification Toolbox" uses another model format. This holds e.g. for Matlab version 5.3. The model format is here called a "theta" format (a matrix). Use the instruction `help theta` for information on transformation to other model forms.

**PRBS**

PRBS signals are most easily generated in Matlab using the functioned `idinput` in "System Identification Toolbox".