EE 5322- Intelligent Control Systems

Updated: Saturday, August 22, 2015

Simulation of Stochastic Discrete Time Systems

The discrete time stochastic system

$$x_{k+1} = Ax_k + Bu_k + Gw_k , \quad x_0$$

with w_k the process noise, is simulated in MATLAB using the M file

```
function [ki,x]=dtsystem(a,b,g,x0,u);
N=size(u); N=N(1); ki(1)= 1;
n=size(x0); n=n(1);
x=zeros(N,n); x(1,:)=x0';
for k=1:N-1
   ki(k+1)=k+1;
   x(k+1,:)= (a*x(k,:)' + b*u(k,:)' + g*rand)' ;
        % x(k+1)=a*x(k) + b*u(k) + g*w(k);
end
ki=ki';
```

Note that the state vector is stored, e.g., as

```
x =
        0         0
   0.1790    1.1790
   1.3675    3.1514
   3.2184    5.5225
   5.6099    8.1635
   8.2577   10.7957
  10.8255   13.1127
  13.1399   14.9953
  15.1018   16.4036
```

which is a 2-D state vector for k=1,2,3,4,5,6,7,8,9. That is, one has

$$x_1^T$$
$$x_2^T$$
$$\vdots$$

over the time interval *k=[1,2,…,N]* stored as an $N \times n$ matrix, with $x_k \in R^n$.

Prior to using dtsystem, one must generate the initial condition x0 and the control input u_k, stored like the state above. i.e. the *m*-vector control input over the *k* interval of interest *[1,N]* is stored as an $N \times m$ matrix. The script code needed is, e.g.,

```
>> a=[0 1
-.89 1.8] ;
```

```
>> b=[0 1]' ;

>> g=[1 1]' ;

>> x0=[0 0]' ;

>> k=[1:1:200]' ;

>> u=ones(size(k)) ;

>> [ki,x]=dtsystem(a,b,0.2*g,x0,u);
>> plot(ki,x)
```

Generating Random Noise in MATLAB

The two main MATLAB random noise generators are:

1. RAND Uniformly distributed random numbers.

RAND(N) is an N-by-N matrix with random entries, chosen from
a uniform distribution on the interval (0.0,1.0).
RAND(M,N) and RAND([M,N]) are M-by-N matrices with random entries.
RAND(M,N,P,...) or RAND([M,N,P,...]) generate random arrays.
RAND with no arguments is a scalar whose value changes each time it
is referenced. RAND(SIZE(A)) is the same size as A.

RAND produces pseudo-random numbers. The sequence of numbers
generated is determined by the state of the generator. Since MATLAB
resets the state at start-up, the sequence of numbers generated will
be the same unless the state is changed.

2. RANDN Normally distributed random numbers.

RANDN(N) is an N-by-N matrix with random entries, chosen from
a normal distribution with mean zero, variance one and standard
deviation one.
RANDN(M,N) and RANDN([M,N]) are M-by-N matrices with random entries.
RANDN(M,N,P,...) or RANDN([M,N,P...]) generate random arrays.
RANDN with no arguments is a scalar whose value changes each time it
is referenced. RANDN(SIZE(A)) is the same size as A.

RANDN produces pseudo-random numbers. The sequence of numbers
generated is determined by the state of the generator. Since MATLAB
resets the state at start-up, the sequence of numbers generated will
be the same unless the state is changed.

Obtaining the System Output

The system output $z_k \in R^p$

$$z_k = Hx_k + v_k$$

can be obtained by adding lines to the m-file dtsystem, or by using dtsystem to generate x_k and then using

$$Y = X * H^T$$

where X is the $N \times n$ matrix above. This batch method has problems adding the measurement noise.