

---

# SubId function

```
function [As, Cs, Ks] = SubId(sid,s,n)
% SubID calculations
% IN
% sid : Slopes of Identification data
% s    : Parameter 's'. Rows of Henkel matrix.
% n    : Order of states 'n'
% OUT
% As : State Space matrix
% Cs : State Space matrix
% Ks : Kalman Gain in Observer form

% Number of sample points for Subspace Identification
N = size(sid,2);

% Dimension of sid
M = size(sid,1);

% Construct Hankel Matrix using sid
Y = [];
sid_stacked = sid(:);

for i=1:N-s+1
    Y = [Y sid_stacked(M*(i-1)+1:M*(s+i-1))];
end

Ypast = Y(:,1:N-2*s+1);
Yfuture = Y(:,s+1:N-s+1);

R = triu(qr([Ypast; Yfuture]'))';
R32 = R(M*s+1:M*2*s,1:M*s);
R22 = R(1:M*s,1:M*s);

[U,Sigma,V] = svd(R32*inv(R22)*Ypast);

% Assigning Sigma for semilogy for choosing n
assignin('base','Sigma',Sigma);

% Reduced System
U = U(:,1:n);
Sigma = Sigma(1:n,1:n);
V = V(:,1:n);

X = sqrt(Sigma)*V';

% Solve least squares problem for X
Xpast = X(:,1:end-1);
Xfuture = X(:,2:end);
Ylsq = sid(:,s+1:end-s);

F = Xpast;
```

---

```

Sys = (F*(F'))\ (F*([Xfuture; Ylsq]'));
Sys = Sys';

As = Sys(1:n,:);
Cs = Sys(n+1:end,:);

% Find K using residuals
Res = [Xfuture; Ylsq] - [As; Cs]*Xpast;
Nt = size(Res,2);
Covar = (Res*Res')/Nt;
Q = Covar(1:n,1:n);
S = Covar(1:n,n+1:end);
R = Covar(n+1:end,n+1:end);

% Choosing P from the solution of the DARE, we can define a Kalman
gain Ks
% that is asymptotically stable.
% Solution of DARE:
E = eye(n,n);
[P,~,~] = dare(As',Cs',Q,R,S,E);

% Kalman Gain:
Ks = (S + As*P*(Cs'))*inv(R + (Cs*P*(Cs')));

end

```

*Published with MATLAB® R2018a*