file ccf_state

Table of Contents

revised 28 JAn 2017 Dr. Tom Chmielewski

original system

this is given in diagonal form and controllable and completely observable

```
A = [-1 0 ; 0 -2]
B = [1;1]
C = [1 1]
D = 0
```

```
A =

    -1     0
     0    -2


B =

     1
     1


C =

     1     1


D =

     0
```

find transfer function

```
tf(ss(A,B,C,D))
```

```
ans =

     2 s + 3
  -------------
  s^2 + 3 s + 2

Continuous-time transfer function.
```

write in controllable canonical form

this is done by finding the transfer function and doing a coefficient match to the model of Eq. 8.7

```
Ab = [-3 -2; 1 0]
Bb = [1; 0]
Cb = [2 3]
```

```
Ab =

    -3    -2
     1     0


Bb =

     1
     0


Cb =

     2     3
```

get characteristic equation same for both representations

```
CEA = poly(A)
```

```
CEA =

     1     3     2
```

Compute feedback gain based on characteristic equation coeffients

```
% desired closed loop ce = s^2 + 11s + 30 = (s+5)(s+6)
%  desired - actual high-1  ro low
CED = [1 11 30]

DD = CED-CEA
% strip off first element s^2 coefficient

kb = DD(2:3)

kb = [8 28]  % state feedback in CCF
```

```
CED =

     1    11    30


DD =

     0     8    28


kb =

     8    28


kb =

     8    28
```

check eigenvalues using kb

```
eig((Ab-Bb*kb)) % verify correct
```

```
ans =

   -6.0000
   -5.0000
```

find P = Cbar * inv (C) and k for original system

```
Cbar = [Bb, Ab*Bb]
CT = [B, A*B]
```

```
P = Cbar*inv(CT) % this is transformation

k= kb*P
```

```
Cbar =

     1    -3
     0     1


CT =

     1    -1
     1    -2


P =

    -1     2
     1    -1


k =

    20   -12
```

verify k with original system

```
eig((A-B*k))
```

```
ans =

   -6.0000
   -5.0000
```

Verify matrices using transform, P are same as by inspection of G(s)

```
AA = P*A*inv(P)
BB = P*B

CC = C*inv(P)
```

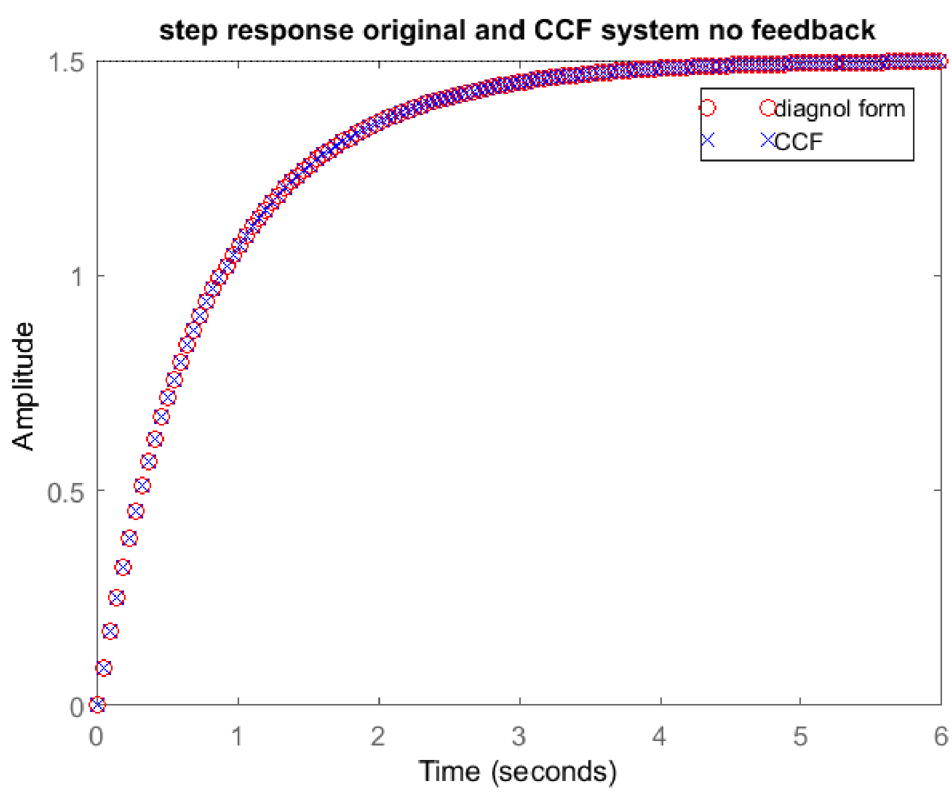
```
AA =

    -3    -2
```

```
        1         0


BB =

        1
        0


CC =

        2         3
```

compare step responses of both systems without state feedback

same y so they should yiled same results even though states are different

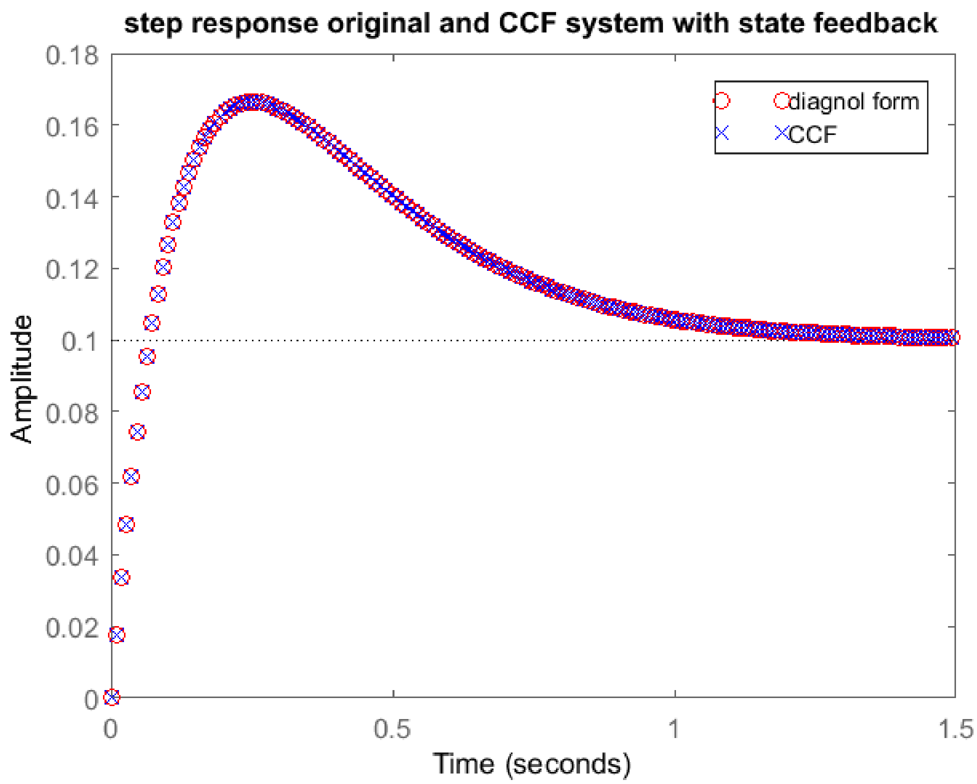
```
sys_orig = ss(A, B, C, 0);
sys_CCF = ss(Ab, Bb, Cb, 0);
figure
step(sys_orig, 'ro', sys_CCF, 'bx')
title('step response original and CCF system no feedback')
legend('diagnol form', 'CCF')
```



compare step responses systems of both systems with state feedback

same y so they should yiled same results even though states are different

```
sys_orig_f = ss((A-B*k), B, C, 0);
sys_CCF_f = ss((Ab-Bb*kb), Bb, Cb, 0);
figure
step(sys_orig_f, 'ro', sys_CCF_f, 'bx')
title('step response original and CCF system with state feedback')
legend('diagnol form', 'CCF')
```



show transfer functions

original transfer function

```
gg1= tf(sys_orig_f)
% transfer function with state feedback
gg2 = tf(sys_CCF_f)
```

```
gg1 =

      2 s + 3
```

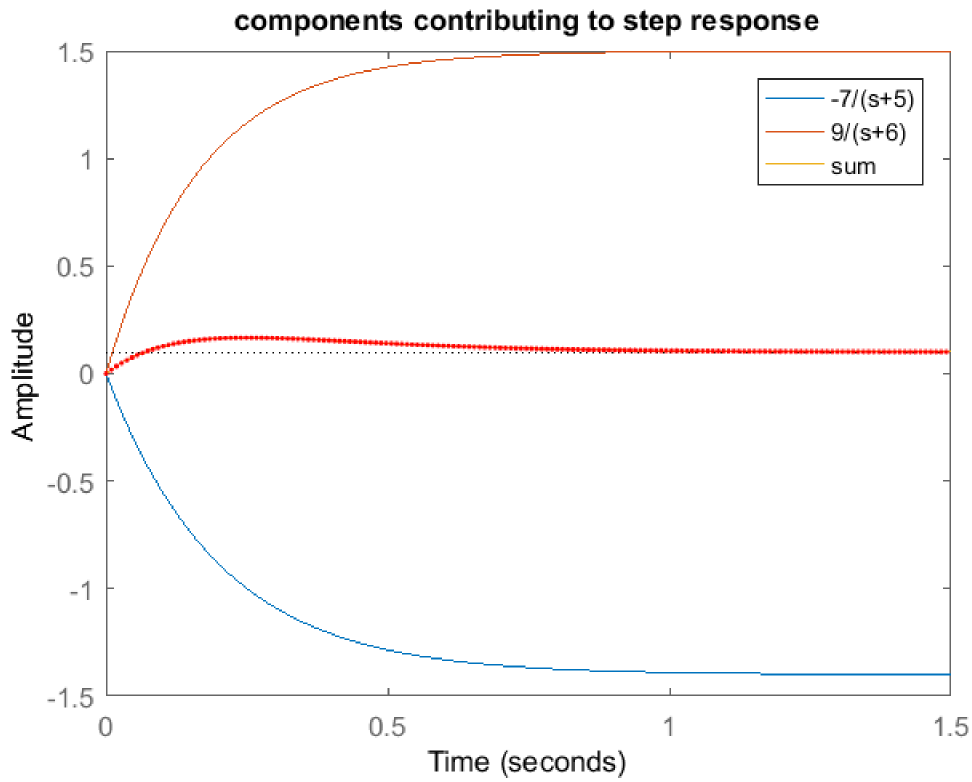
```
  ---------------
  s^2 + 11 s + 30

Continuous-time transfer function.


gg2 =

      2 s + 3
  ---------------
  s^2 + 11 s + 30

Continuous-time transfer function.
```

look at components of transfer function with state feedback

```
s = tf('s')

g1 = -7/(s+5)
g2 = 9/(s+6)

[y1, t1] = step(g1);
[y2, t2] = step(g2, t1);
figure
plot(t1, y1, t2, y2, t1,(y1+y2))

hold on
step((g1+g2), 'r.')
legend('-7/(s+5)', '9/(s+6)', 'sum')
title('components contributing to step response')
```

```
s =

  s

Continuous-time transfer function.


g1 =

   -7
  -----
  s + 5

Continuous-time transfer function.


g2 =
```

```
    9
  -----
  s + 6

Continuous-time transfer function.
```



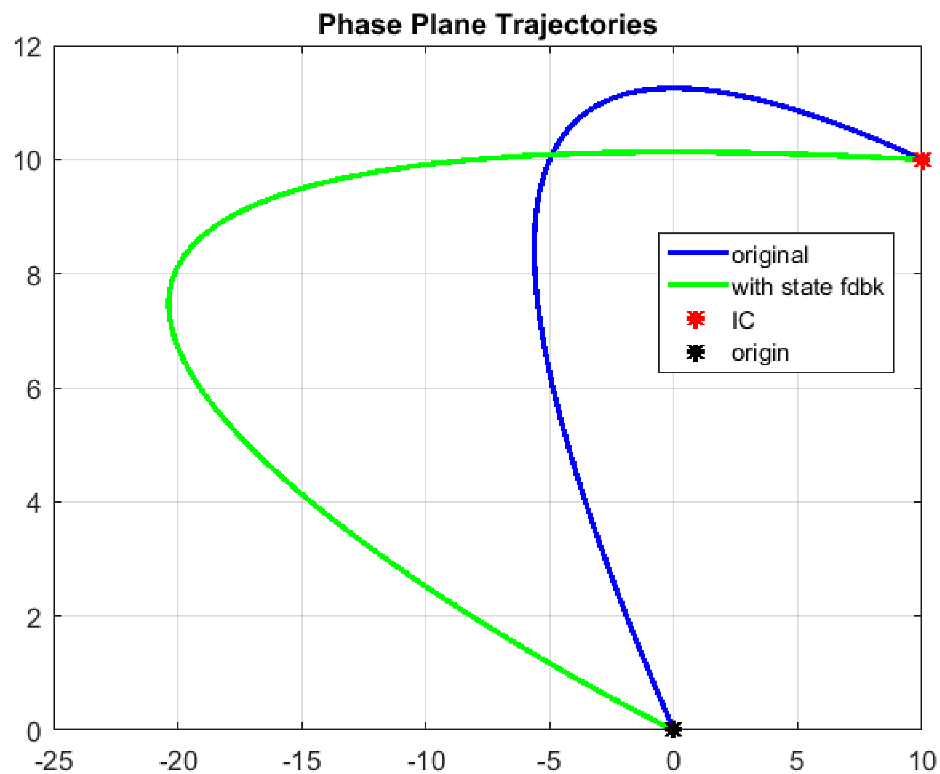
Phase plane analysis for system without and with state feedback

this is solution to homogeneous system x_dot = Ax with x(0) or zdot= (A-Bk)z note with state feedback we saw overshoot in y(t)

```
t = 0:.01:10;  % create time vector
u = zeros(size(t)); % input u = 0;
x0 = [10;10] % initial condition;
C = eye(2) ; % to see both states
sys_CCF = ss(Ab, Bb, C, 0);  % modified to see both states
sys_CCF_f = ss((Ab-Bb*kb), Bb, C, 0);% modified to see both states
X = lsim(sys_CCF,u,t, x0);   % CCF
Xf = lsim(sys_CCF_f,u,t, x0);   % CCF with state feedback
figure
plot(X(:,1), X(:,2),  'b', 'LineWidth', 2)
hold on
```

```
plot(Xf(:,1), Xf(:,2), 'g', 'LineWidth', 2)

plot(10, 10, 'r*', 'LineWidth', 2)
plot(0, 0, 'k*', 'LineWidth', 2')
legend('original', 'with state
 fdbk', 'IC', 'origin','Location', 'Best')
title('Phase Plane Trajectories')
grid on
```

```
x0 =

    10
    10
```



Published with MATLAB® R2016a