Table of Contents

.... 1
2.0 Setup 1
2.1 1
2.2 1
2.2.1 Find Luenberger Oberver gain L 1
2.2.2 Find State feedback gain K 2
2.3 2
2.3 Plotting 3
2.4 4
2.4 Plotting 4
2.5 5
2.5 Plotting 5
3.2 6
3.2 Plotting... 7
4.1 8
4.2 9
4.2 Plotting 9
5 10

```
clear all; close all; clc;
```

2.0 Setup

```
A = [[-2 1];[0 -3]];
B = [1;1];
C = [1 0];
D = 0;

t = linspace(0,pi,100);
u = ones(size(t));
```

2.1

```
% Check Observability

if rank([C;C*A]) == size(A)
    fprintf('the system is fully observable')
end
```

```
the system is fully observable
```

2.2

2.2.1 Find Luenberger Oberver gain L

```
W = [[5 1];[1 0]];
```

```
Omega_o = [[1 0];[-2 1]];
Tinv = W*Omega_o;
T = inv(Tinv);

K_o = [12 1];
K = K_o*T;
L = K';

% closed loop observer error dynamics response
eig(A - L*C)


ans =

  -3.0000 + 3.0000i
  -3.0000 - 3.0000i
```

2.2.2 Find State feedback gain K

```
Omega_c = [[1 -1];[1 -3]];
Tinv = Omega_c*W;
T = inv(Tinv);

K_c = [12 1];
K = K_c*T;

% closed loop state feedback response
eig(A - B*K)


ans =

  -3.0000 + 3.0000i
  -3.0000 - 3.0000i
```

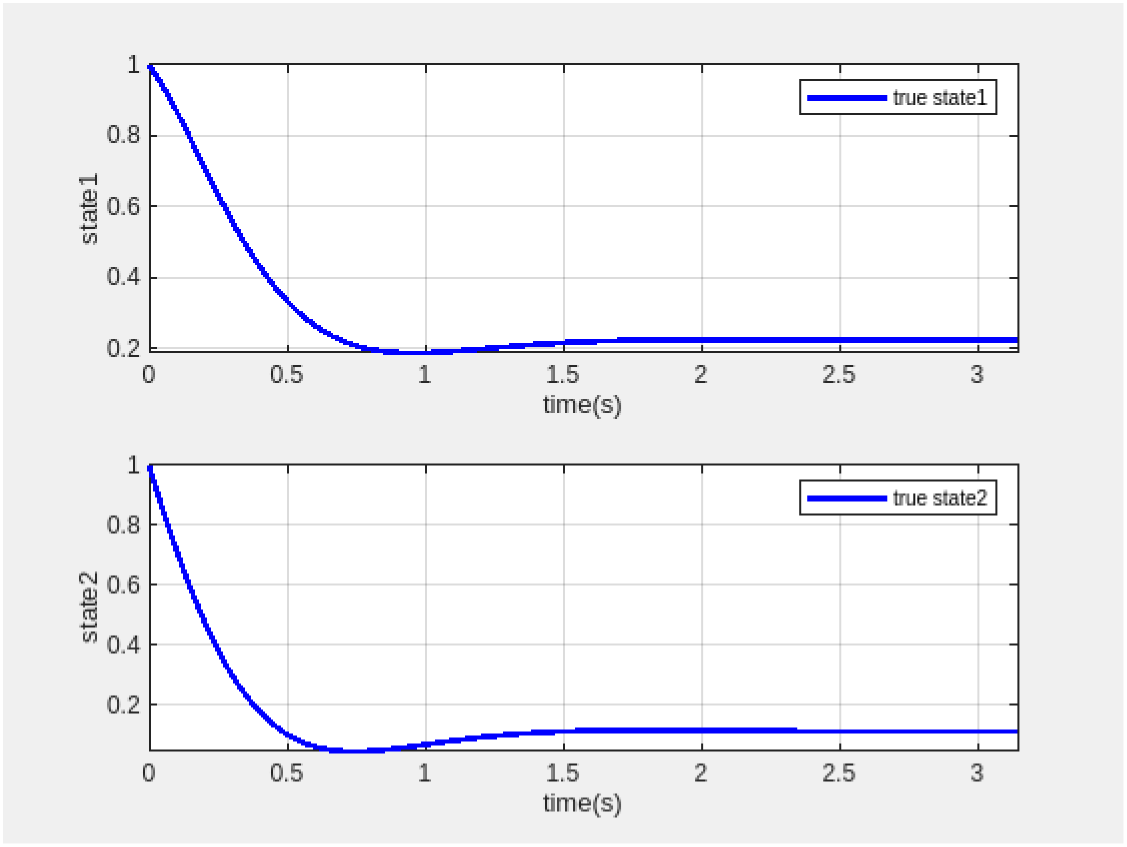
2.3

```
% Simulate unit step response of controller alone

x_true_0 = [1;1];

C_full = eye(2); % full state observability

A_cl = A-B*K;
B_cl = B;
C_cl = [1 0];
D_cl = 0;

sys_true = ss(A_cl, B_cl, C_full, D_cl); % assume the states are measured for
 feedback
```

```
[Y_true,~,X_true] = lsim(sys_true,u,t,x_true_0);

sys_true_output = ss(A_cl, B_cl, C_cl, D_cl); % in reality, only some states
 are measured
[Y_true_output,~,~] = lsim(sys_true_output,u,t,x_true_0);
```

2.3 Plotting

```
figure(23)
subplot(2,1,1)
plot(t,X_true(:,1),'b', 'linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state1')
legend('true state1')

subplot(2,1,2)
plot(t,X_true(:,2),'b', 'linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state2')
legend('true state2')
```

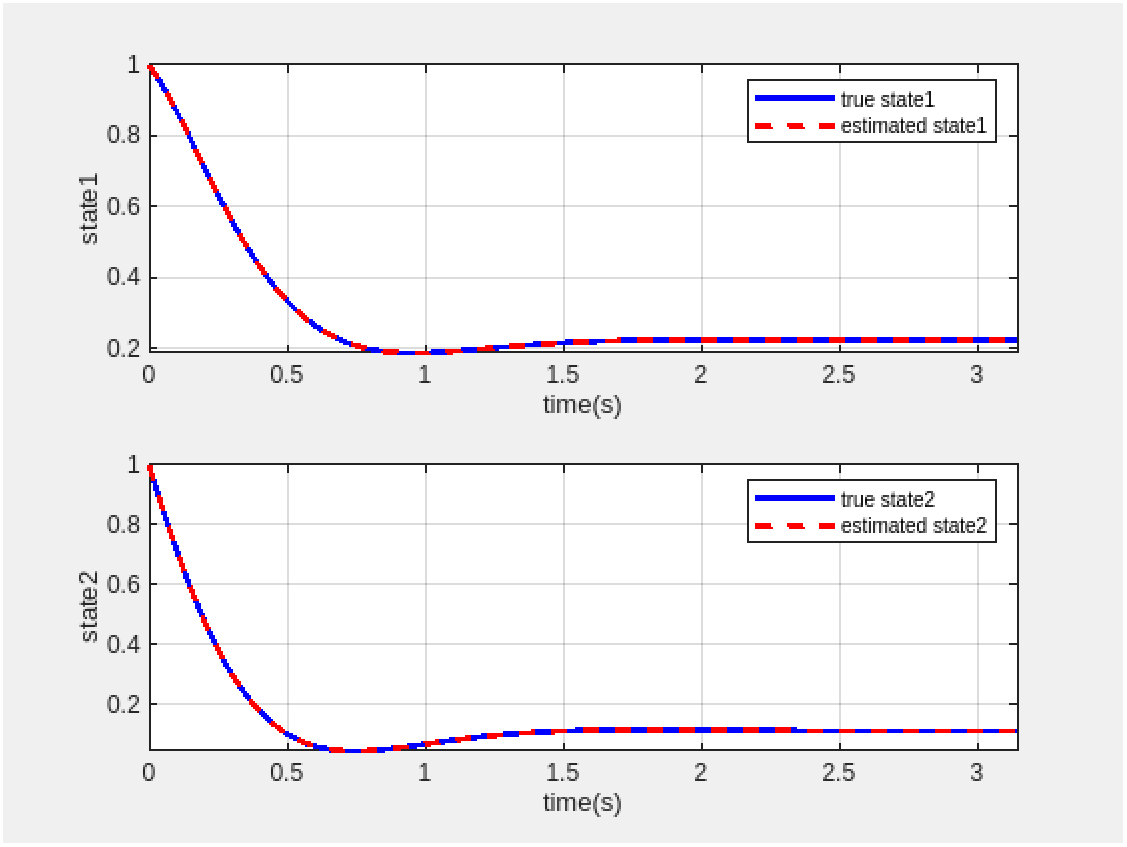


2.4

```
% Simulate unit step response

% estimated state initial condition
x_hat_0 = [1;1];

A_est = A_cl - L*C;
B_est = [B L];
C_est = C_full;
D_est = 0;

u_est = [u;Y_true_output'];

sys_est = ss(A_est, B_est, C_est, D_est); % assume the states are measured for
 feedback

[Y_est,~,X_est] = lsim(sys_est, u_est, t, x_hat_0);
```

2.4 Plotting

```
figure(24)
subplot(2,1,1)
plot(t,X_true(:,1),'b','linewidth',2)
hold on
plot(t,X_est(:,1),'r--','linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state1')
legend('true state1', 'estimated state1')

subplot(2,1,2)
plot(t,X_true(:,2),'b','linewidth',2)
hold on
plot(t,X_est(:,2),'r--','linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state2')
legend('true state2', 'estimated state2')
```



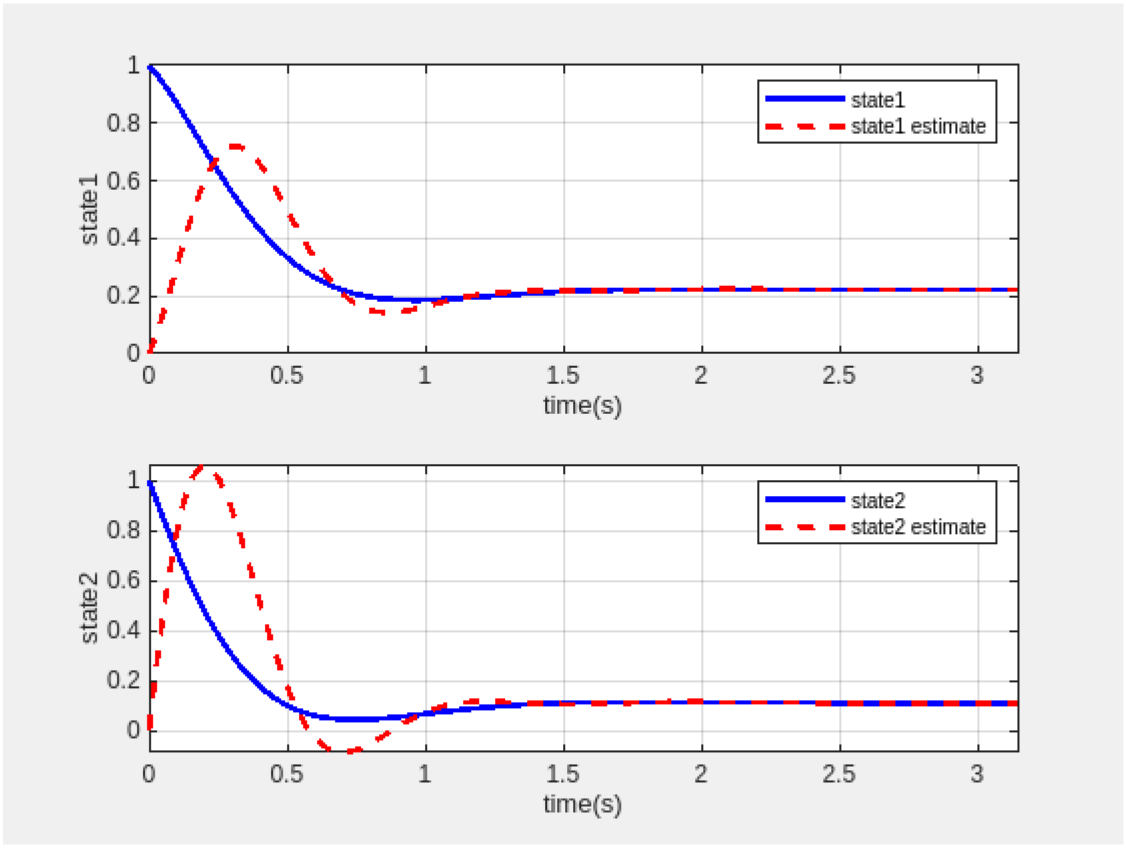
2.5

```
% estimated state initial condition
x_hat_0 = [0;0];

[~,~,X_est] = lsim(sys_est,u_est,t,x_hat_0);
```

2.5 Plotting

```
figure(25)
subplot(2,1,1)
plot(t, X_true(:,1), 'b', 'linewidth', 2)
hold on
plot(t, X_est(:,1), 'r--', 'linewidth', 2)
grid on
axis tight
xlabel('time(s)')
ylabel('state1')
legend('state1', 'state1 estimate')

subplot(2,1,2)
plot(t, X_true(:,2), 'b', 'linewidth', 2)
hold on
plot(t, X_est(:,2), 'r--', 'linewidth', 2)
grid on
```

```
axis tight
xlabel('time(s)')
ylabel('state2')
legend('state2', 'state2 estimate')
```



3.2

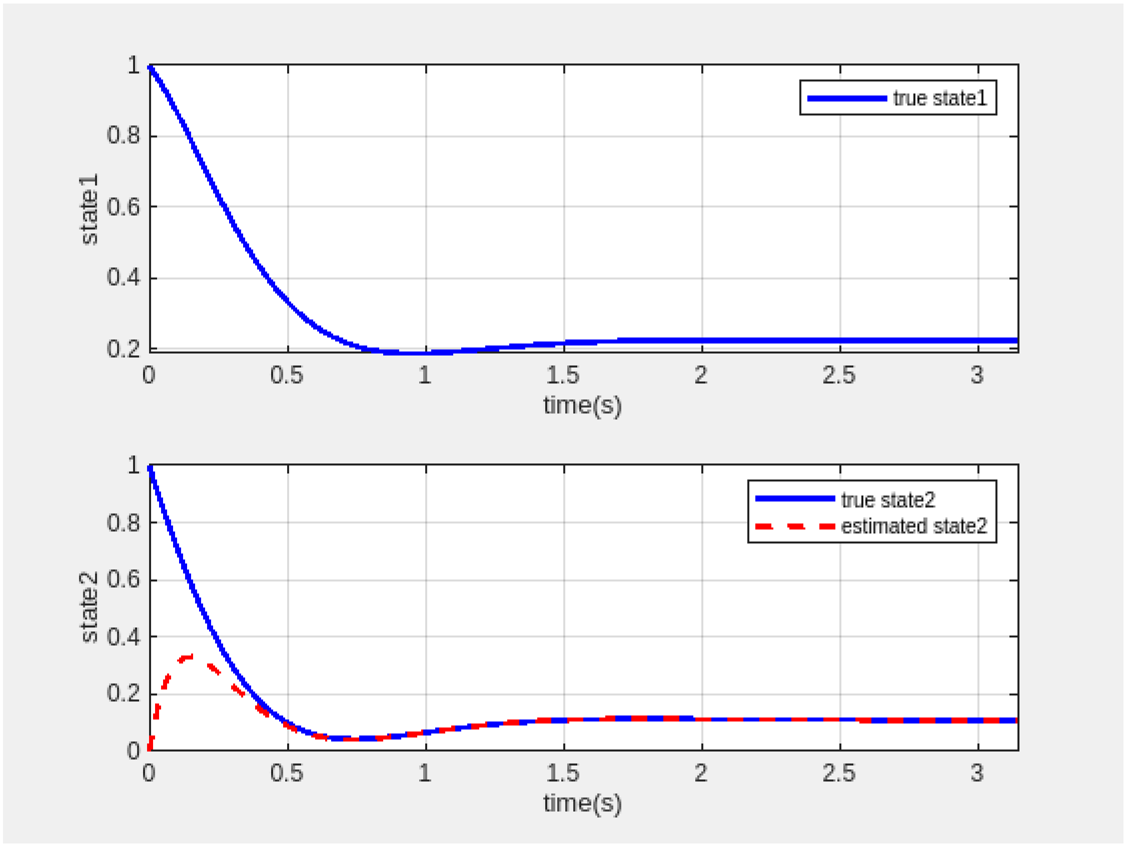
```
% check that the closed loop system has the appropriate eigenvalues
eig(A_cl);

% reduced order observer gain for eigenvalue at -5
L = 2;

% coefficients
x2_coeff = A_cl(2,2) - L * C_cl(1,1) * A_cl(1,2);
y_coeff = L*A_cl(2,2) - L^2*C_cl(1,1)*A_cl(1,2) + A_cl(2,1)*inv(C_cl(1,1)) -
 L*C_cl(1,1)*A_cl(1,1)*C_cl(1,1)*inv(C_cl(1,1));
u_coeff =  B_cl(2,1) - L * C_cl(1,1) * B_cl(1,1);

if eig(x2_coeff) == -5
    fprintf('eigenvalue of the reduced order observer succesfully placed at
 -5')
end

% encode the one dimensional reduced order equation
A_est_ro = x2_coeff;
```

```
B_est_ro = [u_coeff y_coeff];
C_est_ro = 1;
D_est_ro = 0;

% u = 1 1 1...
% Y_true_output = actual system output
u_est = [u; Y_true_output'];

% convert x_hat initial condition for z_hat
x_hat_0 = 0;
z_hat_0 = x_hat_0 - L*Y_true_output(1,1);

% estimated reduced order system
sys_est_ro = ss(A_est_ro, B_est_ro, C_est_ro, D_est_ro);
[Y_est_roo,~,Z_est_roo] = lsim(sys_est_ro, u_est, t, z_hat_0);

% transform back to X_est
X_est_roo = Z_est_roo + L*Y_true_output;

error = X_est_roo - Y_true(:,2);
ss_error = error(end,end)
```

```
ss_error =

  -1.2207e-07
```

3.2 Plotting...

```
figure(3)
subplot(2,1,1)
plot(t, X_true(:,1), 'b','linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state1')
legend('true state1')

subplot(2,1,2)
plot(t, X_true(:,2), 'b', 'linewidth',2)
hold on
plot(t, X_est_roo(:,1), 'r--', 'linewidth',2)
grid on
axis tight
xlabel('time(s)')
ylabel('state2')
legend('true state2', 'estimated state2')
```



4.1

```
A = [[-2 1];[0 -3]];
B = [1;1];
C = [1 0];
D = 0;

Bd = B;
A_d = zeros(1,size(Bd,2));

A_aug = [
    [A Bd];
    [zeros(1,size(A,2)) A_d]
    ];
B_aug = zeros(size(Bd,1)+1,1);
D_aug = 0;
C_aug = [C D_aug];

K = acker(A_aug',C_aug',[-1,-1,-1]);
L = K';

eig(A_aug - L*C_aug)
```

```
ans =
  -1.0000 + 0.0000i
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i
```

4.2

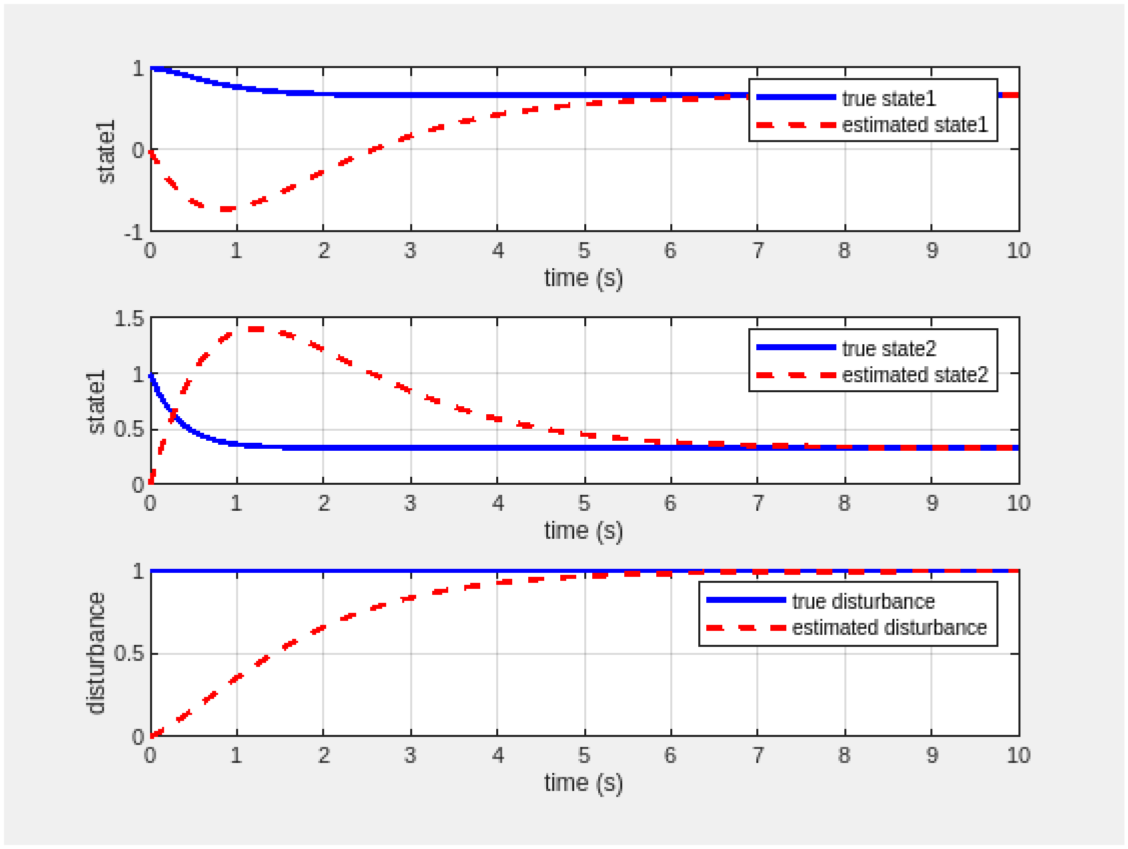
```
if rank(obsv(A_aug, C_aug)) == 3
    fprintf('the augmented system is still fully observable')
end

% true output of the augmented system
sys_aug = ss(A_aug, B_aug, C_aug, D_aug);

x_true_0 = [1;1;1];
x_hat_0 = [0;0;0];

t = linspace(0,10,100);
u_aug = zeros(size(t));

[Y_aug_true,~,X_aug_true] = lsim(sys_aug, u_aug, t, x_true_0);

% estimated output of the augmented system
A_aug_est = A_aug - L*C_aug;
B_aug_est = [B_aug L];
C_aug_est = eye(3);
D_aug_est = 0;

u_aug_est = [zeros(size(t)); Y_aug_true'];
sys_aug_est = ss(A_aug_est, B_aug_est, C_aug_est, D_aug_est);

[Y_aug_est,~,X_aug_est] = lsim(sys_aug_est, u_aug_est, t, x_hat_0);
```

```
the augmented system is still fully observable
```

4.2 Plotting

```
figure(4)
subplot(3,1,1)
plot(t,X_aug_true(:,1), 'b', 'linewidth',2)
grid on
hold on
plot(t,X_aug_est(:,1), 'r--', 'linewidth', 2)
xlabel('time (s)')
ylabel('state1')
legend('true state1', 'estimated state1')

subplot(3,1,2)
plot(t,X_aug_true(:,2), 'b', 'linewidth',2)
grid on
hold on
plot(t,X_aug_est(:,2), 'r--', 'linewidth', 2)
xlabel('time (s)')
```

```
ylabel('state1')
legend('true state2', 'estimated state2')

subplot(3,1,3)
plot(t,X_aug_true(:,3), 'b', 'linewidth',2)
grid on
hold on
plot(t,X_aug_est(:,3), 'r--', 'linewidth', 2)
xlabel('time (s)')
ylabel('disturbance')
legend('true disturbance', 'estimated disturbance')
```



5

```
A5 = [[4 -1];[-1 4]];
B5 = [1;1];
C5 = [1 1];

% derived K for the observable part...
K5 = [5 0];
L5 = K5';

% confirm correct eigenvalue placement
if eig(A5 - L5*C5) == [ -2 ; 5 ]
    fprintf('observer eigenvalues placed appropriately')
end
```

```
observer eigenvalues placed appropriately
```

Published with MATLAB® R2023a