CDS 110b: Lecture 1-2
Observability and State Estimation

Richard M. Murray

5 January 2007

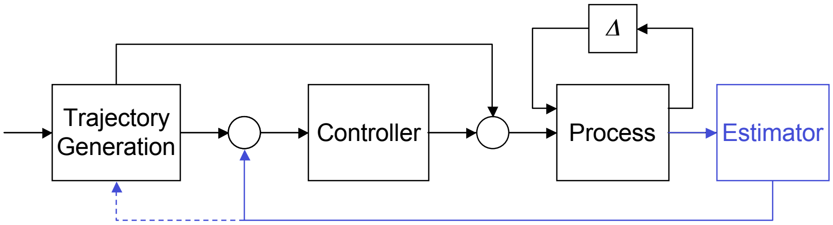
Goals:

- Define observability and give conditions for checking observability for linear control systems
- Introduce the state estimation problem and Luenberger observers
- Provide examples of state estimation in the context of closed loop design

Reading:

- Åström and Murray, *Feedback Systems,* Sections 7.1-7.3 (available online)
- Friedland, Chapters 7 and 8

Modern Control System Design

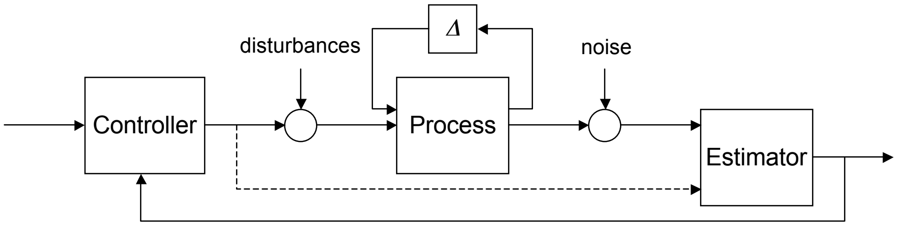


CDS 110a - Control using state feedback: $u = -K\,x + k_r\,r$

Weeks 1-4: State Estimation

- Given process measurements, how do we determine the state for use in state feedback and/or receding horizon control?
- Theory is also useful for pure estimation problems (eg, sensor fusion)
- Requires that we start talking about *noise* in a more fundamental way

The State Estimation Problem


disturbances

noise

Δ

Controller

Process

Estimator

Problem Setup

- Given a dynamical system with noise and uncertainty, estimate the state

$$\begin{aligned}\dot x &= Ax + Bu + Fv\\ y &= Cx + Du + Gw\end{aligned} \qquad \longrightarrow \qquad \begin{aligned}\dot{\hat x} &= \alpha(\hat x, y, u) \longleftarrow \text{estimator}\\ \lim_{t\to\infty} E(x-\hat x) &= 0\end{aligned}$$

expected value

- $\hat x$ is called the *estimate* of x

Remarks

- Several sources of uncertainty: noise, disturbances, process, initial condition
- Uncertainties are unknown, except through their effect on measured output
- First question: when is this even *possible?*

Observability

Defn A dynamical system of the form

$$\begin{aligned}\dot x &= f(x,u)\\ y &= h(x,u)\end{aligned}$$

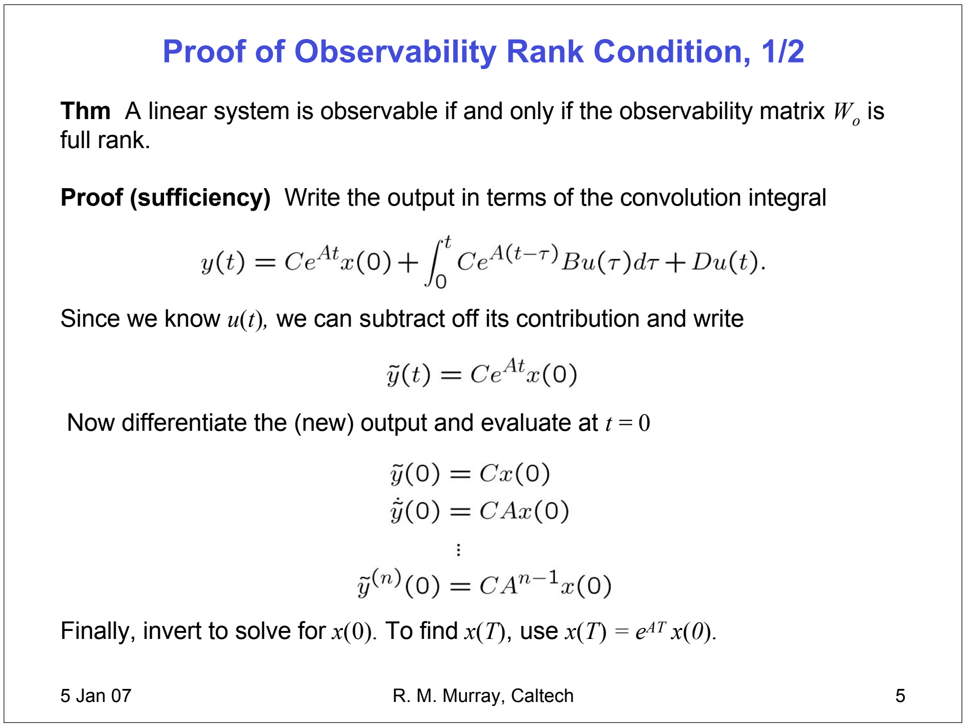
is *observable* if for any $T > 0$ it is possible to determine the state of the system $x(T)$ through measurements of $y(t)$ and $u(t)$ on the interval $[0,T]$

Remarks

- Observability must respect *causality*: only get to look at past measurements
- We have ignored noise, disturbances for now ⇒ estimate exact state
- Intuitive way to check observability:

$$\begin{aligned}\dot x &= Ax + Bu\\ y &= Cx\end{aligned} \qquad \begin{aligned} y &= \underline{Cx}\\ \dot y &= C\dot x = \underline{CAx} + CBu\\ \ddot y &= \underline{CA^2x} + CABu + CB\dot u\\ &\vdots \end{aligned} \qquad W_o = \begin{bmatrix} C\\ CA\\ CA^2\\ \vdots\\ CA^{n-1}\end{bmatrix}$$

Thm A linear system is observable if and only if the observability matrix W_o is full rank



Proof of Observability Rank Condition, 1/2

Thm A linear system is observable if and only if the observability matrix W_o is full rank.

Proof (sufficiency) Write the output in terms of the convolution integral

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t).$$

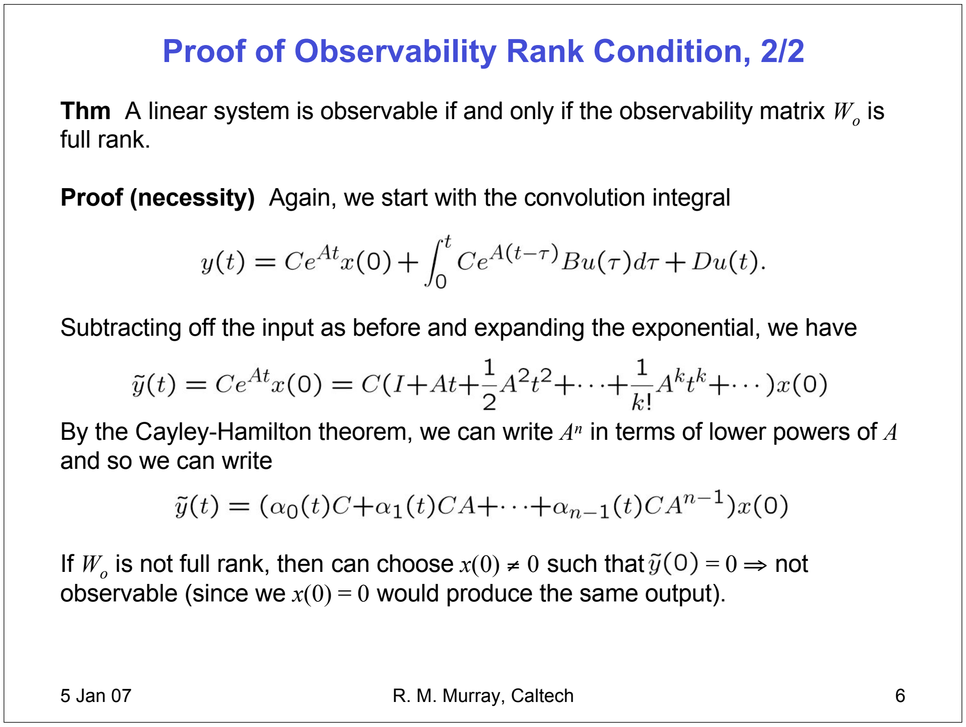
Since we know $u(t)$, we can subtract off its contribution and write

$$\tilde{y}(t) = Ce^{At}x(0)$$

Now differentiate the (new) output and evaluate at $t = 0$

$$\begin{aligned} \tilde{y}(0) &= Cx(0) \\ \dot{\tilde{y}}(0) &= CAx(0) \\ &\vdots \\ \tilde{y}^{(n)}(0) &= CA^{n-1}x(0) \end{aligned}$$

Finally, invert to solve for $x(0)$. To find $x(T)$, use $x(T) = e^{AT}\, x(0)$.



Proof of Observability Rank Condition, 2/2

Thm A linear system is observable if and only if the observability matrix W_o is full rank.

Proof (necessity) Again, we start with the convolution integral

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t).$$

Subtracting off the input as before and expanding the exponential, we have

$$\tilde{y}(t) = Ce^{At}x(0) = C(I + At + \frac{1}{2}A^2t^2 + \cdots + \frac{1}{k!}A^kt^k + \cdots)x(0)$$

By the Cayley-Hamilton theorem, we can write A^n in terms of lower powers of A and so we can write

$$\tilde{y}(t) = (\alpha_0(t)C + \alpha_1(t)CA + \cdots + \alpha_{n-1}(t)CA^{n-1})x(0)$$

If W_o is not full rank, then can choose $x(0) \neq 0$ such that $\tilde{y}(0) = 0 \Rightarrow$ not observable (since we $x(0) = 0$ would produce the same output).

State Estimation: Full Order Observer

Given that a system is observable, how do we actually estimate the state?

- Key insight: if current estimate is correct, follow the dynamics of the system

$$\begin{aligned} \dot x &= Ax + Bu \\ y &= Cx \end{aligned} \qquad \dot{\hat x} = \underbrace{A\hat x + Bu}_{\text{prediction (copy of dynamics)}} + L(y - C\hat x) \longleftarrow \text{correction (based on output error)}$$

- Modify the dynamics to correct for error based on a linear feedback term
- L is the *observer gain matrix*; determines how to adjust the state due to error
- Look at the error dynamics for $\tilde x = x - \hat x$ to determine how to choose L:

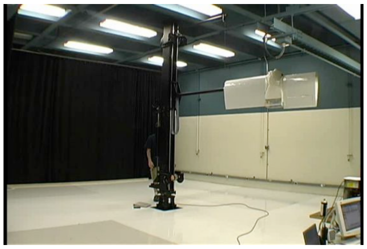
$$\dot{\tilde x} = \dot x - \dot{\hat x} = Ax + Bu - (A\hat x + Bu + LC(x - \hat x)) = (A - LC)\tilde x$$

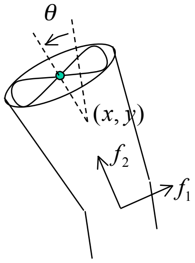
Thm If the pair (A, C) is observable (associated W_o is full rank), then we can place the eigenvalues of A-LC arbitrarily through appropriate choice of L.

Proof Note that the transpose of A-LC is A^T - C^T L^T and in this form, this is the same as the eigenvalue placement problem for state space controllers**.**

Remark: In MATLAB, use `L' = place(A', C', eigs)` to determine L

Example: Ducted Fan





Estimation:

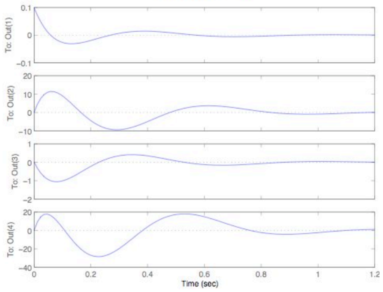
- Given the xy position of the fan and the inputs (f_1, f_2), determine the full state of the system:

$$x, y, \theta, \dot x, \dot y, \dot\theta$$

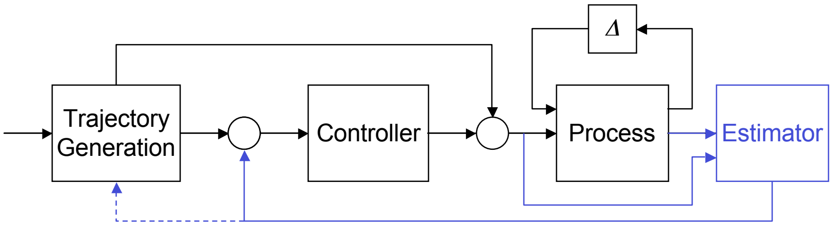
Equations of motion

$$\begin{aligned} m\ddot x &= f_1 \cos\theta - f_2 \sin\theta - c_{d,x}(\theta, \dot x) \\ m\ddot y &= f_1 \sin\theta + f_2 \cos\theta - mg - c_{d,y}(\theta, \dot y) \\ J\ddot\theta &= rf_1 - mgl \sin\theta - c_{d,\theta}(\theta, \dot\theta) \end{aligned}$$

Estimator design: see obs_dfan.m



Separation Principle


Trajectory Generation

Controller

Process

Estimator

Δ

What happens when we apply state space controller using *estimate* of x?

- We assumed we measured x directly in analyzing controller; extra dynamics in the estimator could cause closed loop to go unstable

Thm If K is a stabilizing compensator for (A, B) and L gives a stable estimator for (A, C), then the control law $u = -K(\hat{x} - x_d) + u_d$ is stable (for x_d, u_d an equil pt)

- This is an example of a *separation principle*: design the controller and estimator separately, then combine them and everything is OK
- Be careful with signs on gains (MATLAB vs LQR vs AM05)

Proof of Separation Theorem

Proof. Write down the dynamics for the complete system (assuming WLOG that x_d, $u_\mathrm{d} = 0$):

$$\dot{x} = Ax + Bu \qquad \dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$
$$y = Cx \qquad u = -K\hat{x} + u_d$$

Rewrite in terms of the error dynamics $\tilde{x} = x - \hat{x}$ and combined state $x, \tilde{x}$:

$$\dot{\tilde{x}} = (A{-}LC)\tilde{x} \qquad \frac{d}{dt}\begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} u_d \\ 0 \end{bmatrix}$$

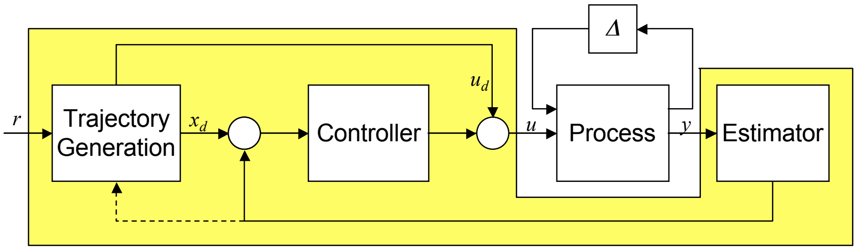
Since the dynamics matrix is block diagonal, we find that the characteristic polynomial of the closed loop system is

$$\det(sI - A + BK)\det(sI - A + LC).$$

This polynomial is a product of two terms, where the first is the characteristic polynomial of the closed loop system obtained with state feedback and the other is the characteristic polynomial of the observer error.

Since each was designed to be stable ⇒ the entire system is stable

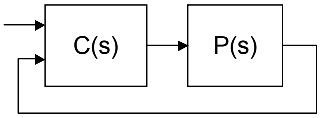
Transfer Function Analysis



Assume trajectory generation is open loop:

$x_d = Nr$ ← desired (steady) state

$u_d = K_r r$ ← nominal input



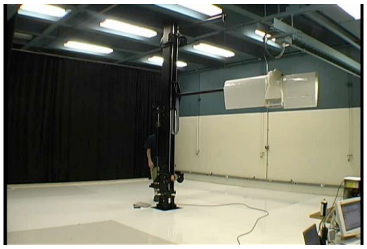
$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$
$$u = K(\hat{x} - Nr) + K_r r$$

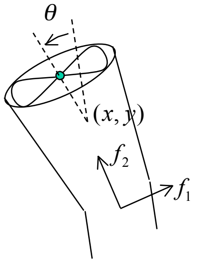
Can now write entire state space controller as a 2 input, 1 output transfer function

- $H_{uy}(s)$ gives feedback
- $H_{ur}(s)$ gives feedforward

$$H_{uy}(s) = K\big(sI - (A - BK - LC)\big)^{-1}L$$

Example: Ducted Fan



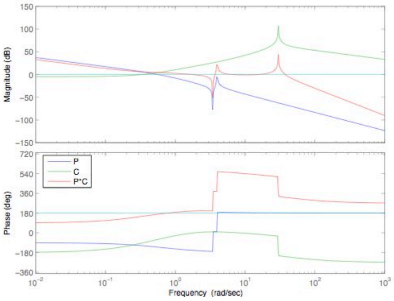


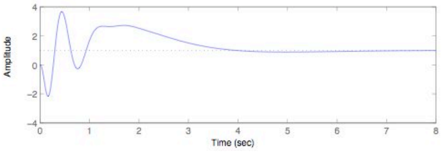
Estimation:

- Full order observer

Control

- LQR (state feedback)

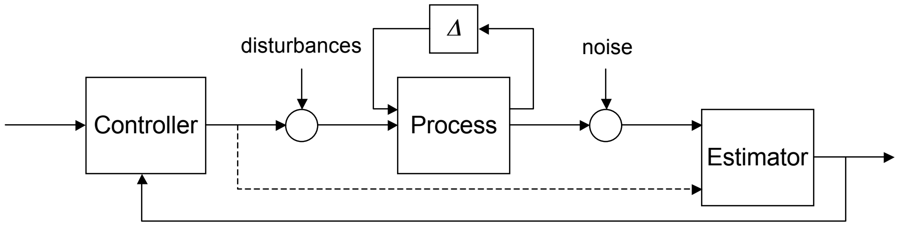




Remarks

- RHP give limits to performance
- RHC with feedforward gives better perf (but still need a good state estimate!)

Summary: Observers and State Estimation



Observability

- Derived conditions for when we could determine state from inputs & outputs: check rank of observability matrix

State Estimators

- Construct state estimate based on prediction and correction (no noise yet)

Closed Loop Performance

- Computed transfer function for overall controller (near equilibrium point)

Next: add noise to the problem formulation → Kalman filter

```
% obs_dfan.m − observability and estimation for the ducted fan
% RMM, 7 Jan 03

%%
%% Ducted fan dynamics
%%
%% These are the dynamics for the ducted fan, written in state space
%% form.
%%

% System parameters
J = 0.0475;                     % inertia around pitch axis
m = 1.5;                        % mass of fan
r = 0.25;                       % distance to flaps
g = 10;                         % gravitational constant
gamma = 0.51;                   % counterweighted mass
d = 0.2;                        % damping factor (estimated)
l = 0.05;                       % offset of center of mass

% Tweak the damping for better plots
d = 1;

% System matrices (entire plant: 2 input, 2 output)
A = [  0     0     0     1     0     0;
       0     0     0     0     1     0;
       0     0     0     0     0     1;
       0     0  −gamma −d/m    0     0;
       0     0     0     0   −d/m    0;
       0     0 −m*g*l/J  0     0     0 ];

B = [  0     0;
       0     0;
       0     0;
      1/m    0;
       0    1/m;
      r/J    0    ];

C = [  1     0    0      0     0     0;
       0     1    0      0     0     0 ];

D = [  0     0;   0      0];

%% Extract out the lateral dynamics
Ax = A([1 3 4 6], [1 3 4 6]);
Bx = B([1 3 4 6], 1);
Cx = C(1, [1 3 4 6]);
Dx = D(1, 1);

dfsys = ss(Ax, Bx, Cx, Dx);

%%
%% State space control design
%%
%% We use an LQR design to choose the state feedback gains
%%

K = lqr(Ax, Bx, eye(size(Ax)), 0.1*eye(size(Bx’*Bx)));

%%
%% Estimator
%%
%% Pole placement to set estimator gains.
%%

% Determine desired pole locations using butterworth configuration
[Af,Bf,Cf,Df]=butter(4,10,’s’);
[numf, denf] = ss2tf(Af,Bf,Cf,Df);
eigs = pole(tf(numf,denf));
```

```
L1 = transpose( place(Ax’, Cx’, eigs) );

Lk = lqe(Ax, 0.1*eye(size(Ax)), Cx, 0.001*eye(size(Ax)), ...
        0.001*eye(size(Cx*Cx’)), 0*eye(size(Ax*Cx’)));

% Look at the estimator response to initial condition error
% Do this by constructing a system without output = estimated states
est = ss(Ax − L1*Cx, Bx, eye(size(Ax)), 0);
figure(1); initial(est, [0.1; 0.01; 1/180*pi; 0]);

%%
%% Closed loop plant
%%
%% Here we use the reg command to construct the regulator from the
%% state space estimator and controller.  We invert out the minus sign
%% that reg assumes so that we can use the negative feedback in the
%% frequency domain (usual sign convention)
%%

dfctr1 = −tf(reg(dfsys, K, L1));
H1 = feedback(dfctr1*dfsys,1);

dfctrk = −tf(reg(dfsys, K, Lk));
Hk = feedback(dfctrk*dfsys,1);

%%
%% Analysis of the controller structure
%%
margintf=tf(−1);                % for seeing gain/phase margins
figure(2);
  W = logspace(−2, 3, 500);
  bode(dfsys, dfctr1, dfctr1*dfsys, margintf, W);
  legend(’P’, ’C’, ’P*C’, ’Location’, ’NorthWest’);

% Plot out the step response to a step change in desired output
% Note: we can do much better than this (and will)
figure(3);
  subplot(211);
  step(feedback(dfctr1*dfsys, 1), 8);
```