State Space Control
A First-Order Means of Control!

METR 4202: Advanced Control & Robotics
Dr Surya Singh
Lecture # 10
September 27, 2013

metr4202@itee.uq.edu.au
http://itee.uq.edu.au/~metr4202/

Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecture (F: 9-10:30, 42-212)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26-Jul</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>2-Aug</td>
<td>Representing Position &amp; Orientation &amp; State (Frames, Transformation Matrices &amp; Affine Transformations)</td>
</tr>
<tr>
<td>3</td>
<td>9-Aug</td>
<td>Robot Kinematics</td>
</tr>
<tr>
<td>4</td>
<td>16-Aug</td>
<td>Robot Dynamics &amp; Control</td>
</tr>
<tr>
<td>5</td>
<td>23-Aug</td>
<td>Robot Trajectories &amp; Motion</td>
</tr>
<tr>
<td>6</td>
<td>30-Aug</td>
<td>Sensors &amp; Measurement</td>
</tr>
<tr>
<td>7</td>
<td>6-Sep</td>
<td>Perception / Computer Vision</td>
</tr>
<tr>
<td>8</td>
<td>13-Sep</td>
<td>Localization and Navigation</td>
</tr>
<tr>
<td>9</td>
<td>20-Sep</td>
<td>State-Space Modelling</td>
</tr>
<tr>
<td>10</td>
<td>27-Sep</td>
<td>State-Space Control</td>
</tr>
<tr>
<td>11</td>
<td>11-Oct</td>
<td>Motion Planning</td>
</tr>
<tr>
<td>12</td>
<td>18-Oct</td>
<td>Vision-based control (+ Prof. P. Corke or Prof. M. Srinivasan)</td>
</tr>
<tr>
<td>13</td>
<td>25-Oct</td>
<td>Applications in Industry (+ Prof. S. LaValle) &amp; Course Review</td>
</tr>
</tbody>
</table>

Study break
### Announcements: We’re Working On It!

- **Grades:**
  - I am still working on it
  - I’ve read the course profile 1100× 😊

- **Lab 3:**
  - Working on it!

- **Cool Robotics Share Site**
  - Jared is making a “blog”. URL Soon!
  - He is still working on it !! 😊

---

**Welcome to State-Space!**

(Why the big type?)
State-Space!

государственный контроль пространства

- *It's Russian for Control*

- **Dynamic systems** are described as differential equations (as compared to transfer functions)

- **Stability** is approached via the theory of Liapunov instead of frequency-domain methods (Bode and Nyquist)

- **Optimisation of System Performance** via calculus of variations (Pontryagin) (as compared to Wiener-Hopf methods)
State-Space Control

\[ \dot{x} = Fx \]

(That can not be all of it? There has to be more to it than this...)

Benefits:
- Characterises the process by systems of coupled, first-order differential equations
- More general mathematical model
  - MIMO, time-varying, nonlinear
- Mathematically esoteric (who needs practical solutions)
- Yet, well suited for digital computer implementation
  - That is: based on vectors/matrices (think LAPACK → MATLAB)
Difference Equations & Feedback

\[ u \rightarrow H \rightarrow y \rightarrow \hat{y} \rightarrow \Sigma \rightarrow k \rightarrow H \rightarrow y \]

- Start with the Open-Loop:
  \[ y = Hu \]
- Close the loop:
  \[ u = ke = k(\hat{y} - y) \Rightarrow y = H[k(\hat{y} - y)] \]
  \[ y = \frac{Hk}{1 + Hk} \hat{y} \]
- All easy! (yesa!)

\[ 27 \text{ September 2013} \]

METR 4202: Robotics

\[ \text{Difference Equations & Feedback} \]

\[ u \rightarrow H \rightarrow y \rightarrow \hat{y} \rightarrow \Sigma \rightarrow k \rightarrow H \rightarrow y \]

- Now add delay (image the plant is a replica with a delay \( \tau \))
  \[ y(t) = u(t - \tau) \]
- Close the loop:
  \[ u(t - \tau) = ke(t - \tau) = k [\hat{y}(t - \tau) - y(t - \tau)] \]
  \[ y(t) = k [\hat{y}(t - \tau) - y(t - \tau)] \]
- Notice we have a difference equation!
Difference Equations & Feedback

• What happens with a single delay and a unit step?
  \[ u(t) = k \text{ for } 0 < t < \tau \]
  \[ y(t) = u(t - \tau) \text{ for } \tau < t < 2\tau \]

• Then with feedback we get:
  \[ u(t) = k(1 - k) = k - k^2 \]
  \[ y(t) = k - k^2 + k^3 + \cdots + (-1)^{n-1}k^{n-1} \]

• If \( k < 1 \): then:
  \[ \lim_{t \to \infty} y(t) = \frac{k}{1+k} \]
Controllability matrix

- If you can write it in CCF, then the system equations must be linearly independent.

- Transformation by any nonsingular matrix preserves the controllability of the system.

- Thus, a nonsingular controllability matrix means $x$ can be driven to any value.

State evolution

- Consider the system matrix relation:
  \[
  \dot{x} = Fx + Gu \\
  y = Hx + Ju
  \]

  The time solution of this system is:
  \[
  x(t) = e^{F(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{F(t-\tau)} Gu(\tau) d\tau
  \]

  If you didn’t know, the matrix exponential is:
  \[
  e^{Kt} = I + Kt + \frac{1}{2!} K^2 t^2 + \frac{1}{3!} K^3 t^3 + \ldots
  \]
Stability

- We can solve for the natural response to initial conditions \( x_0 \):

\[
x(t) = e^{pt}x_0
\]

\[
\therefore \dot{x}(t) = p_i e^{pt}x_0 = Fe^{pt}x_0
\]

Clearly, a system will be stable provided \( \text{eig}(F) < 0 \)

Characteristic polynomial

- From this, we can see \( Fx_0 = p_i x_0 \)

\[
or, (p_i I - F)x_0 = 0
\]

which is true only when \( \det(p_i I - F)x_0 = 0 \)

Aka. the characteristic equation!

- We can reconstruct the CP in \( s \) by writing:

\[
\det(sI - F)x_0 = 0
\]
Great, so how about control?

• Given \( \dot{x} = Fx + Gu \), if we know \( F \) and \( G \), we can design a controller \( u = -Kx \) such that
  \[
  \text{eig}(F - GK) < 0
  \]

• In fact, if we have full measurement and control of the states of \( x \), we can position the poles of the system in arbitrary locations!

  (Of course, that never happens in reality.)

Example: PID control

• Consider a system parameterised by three states:
  - \( x_1, x_2, x_3 \)
    - where \( x_2 = \dot{x}_1 \) and \( x_3 = \dot{x}_2 \)
  \[
  \dot{x} = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix} x - Ku
  \]
  \[
  y = [0 & 1 & 0] x + 0u
  \]

  \( x_2 \) is the output state of the system;
  \( x_1 \) is the value of the integral;
  \( x_3 \) is the velocity.
• We can choose $K$ to move the eigenvalues of the system as desired:

$$\det\begin{bmatrix} 1 - K_1 & 1 - K_2 \\ -2 - K_3 & \end{bmatrix} = 0$$

All of these eigenvalues must be positive.

It’s straightforward to see how adding derivative gain $K_3$ can stabilise the system.

Just scratching the surface

• There is a lot of stuff to state-space control

• One lecture (or even two) can’t possibly cover it all in depth

Go play with Matlab and check it out!
Discretisation FTW!

- We can use the time-domain representation to produce difference equations!

\[ x(kT + T) = e^{FT} x(kT) + \int_{kT}^{kT+T} e^{F(kT+\tau)} G u(\tau) d\tau \]

Notice \( u(\tau) \) is not based on a discrete ZOH input, but rather an integrated time-series.

We can structure this by using the form:

\[ u(\tau) = u(kT), \quad kT \leq \tau \leq kT + T \]

---

Discretisation FTW!

- Put this in the form of a new variable:

\[ \eta = kT + T - \tau \]

Then:

\[ x(kT + T) = e^{FT} x(kT) + \left( \int_{kT}^{kT+T} e^{F\eta} d\eta \right) G u(kT) \]

Let’s rename \( \Phi = e^{FT} \) and \( \Gamma = \left( \int_{kT}^{kT+T} e^{F\eta} d\eta \right) G \)
Discrete state matrices

So,
\[
x(k + 1) = \Phi x(k) + \Gamma u(k)
\]
\[
y(k) = Hx(k) + Ju(k)
\]
Again, \(x(k + 1)\) is shorthand for \(x(kT + T')\)

Note that we can also write \(\Phi\) as:
\[
\Phi = I + FT\Psi
\]
where
\[
\Psi = I + \frac{FT}{2!} + \frac{F^2T^2}{3!} + \cdots
\]

Simplifying calculation

• We can also use \(\Psi\) to calculate \(\Gamma\)
  – Note that:
  \[
  \Gamma = \sum_{k=0}^{\infty} \frac{F^k T^k}{(k + 1)!} T G
  \]
  \[
  = \Psi T G
  \]
\(\Psi\) itself can be evaluated with the series:
\[
\Psi \approx I + \frac{FT}{2} \left\{ I + \frac{FT}{3} \left[ I + \cdots + \frac{FT}{n-1} \left( I + \frac{FT}{n} \right) \right] \right\}
\]
State-space z-transform

We can apply the z-transform to our system:

\[(zI - \Phi)X(z) = \Gamma U(k)\]
\[Y(z) = HX(z)\]

which yields the transfer function:

\[\frac{Y(z)}{X(z)} = G(z) = H(zI - \Phi)^{-1}\Gamma\]

State-space control design

- Design for discrete state-space systems is just like the continuous case.
  - Apply linear state-variable feedback:
    \[u = -Kx\]

such that \(\det(zI - \Phi + \Gamma K) = \alpha_c(z)\)

where \(\alpha_c(z)\) is the desired control characteristic equation.

Predictably, this requires the system controllability matrix

\[C = [\Gamma \Phi \Phi^2 \Gamma \ldots \Phi^{n-1}\Gamma]\]

to be full-rank.
Example:
Command Shaping
Command Shaping

- Zero Vibration (ZV)
  \[
  \begin{bmatrix}
  A_i \\
  t_i
  \end{bmatrix} = \begin{bmatrix}
  \frac{1}{1+K} & K \\
  0 & \frac{1+K}{T_d/2}
  \end{bmatrix}
  \]

  \[K = e^\left(\frac{-\pi}{\sqrt{1-\zeta^2}}\right)\]

- Zero Vibration and Derivative (ZVD)
  \[
  \begin{bmatrix}
  A_i \\
  t_i
  \end{bmatrix} = \begin{bmatrix}
  \frac{1}{(1+K)^2} & \frac{2K}{(1+K)^2} & \frac{K^2}{(1+K)^2} \\
  0 & \frac{T_d}{2} & T_d
  \end{bmatrix}
  \]

Example II:
Estimation
Along multiple dimensions

State Space

• We collect our set of uncertain variables into a vector …
  \[ \mathbf{x} = [x_1, x_2, \ldots, x_N]^T \]

• The set of values that \( \mathbf{x} \) might take on is termed the state space

• There is a single true value for \( \mathbf{x} \), but it is unknown
State Space Dynamics

\[ \dot{x} = Ax + Bu \]
\[ y = Cx + Du \]

\[ H(s) = C(sI - A)^{-1}B \]

Measured versus True

* Measurement errors are inevitable

* So, add Noise to State...
  - State Dynamics becomes:
    \[ \dot{x} = Ax + Bu + w \]
    \[ y = Cx + Du + v \]

* Can represent this as a “Normal” Distribution

\[ \mathcal{N}(x; \mu, \sigma) = \frac{1}{(\sqrt{2\pi})\sigma} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \]
Recovering The Truth

- Numerous methods
- Termed “Estimation” because we are trying to estimate the truth from the signal
- A strategy discovered by Gauss
- Least Squares in Matrix Representation

\[
\begin{bmatrix}
    p_0 \\
    p_1
\end{bmatrix} = 
\left[
\begin{bmatrix}
    n \\
    \sum_{i=1}^{n} t_i \\
    \sum_{i=1}^{n} t_i^2
\end{bmatrix}^{-1}
\begin{bmatrix}
    \sum_{i=1}^{n} z_i \\
    \sum_{i=1}^{n} t_i z_i
\end{bmatrix}
\right]
\]

Recovering the Truth: Terminology

\[
\dot{x} = Fx + Gu + \mathbf{w}
\]

\[
z = Hx + \mathbf{v}
\]

- $x$: the state vector
- $x_{k|j}$: the state of $x$ at time $j$ based on data taken up to time $j$.
- $\hat{x}$: estimate of the true state vector
- $F$: system dynamics matrix in continuous time equivalent to $A$ in Eq. 1
- $G$: system control matrix relating deterministic input $u$ to the state equivalent to $B$ in Eq. 1
- $H$: measurement matrix in continuous time equivalent to $C$ in Eq. 2
- $F_k$: system model in discrete time at $t = t_k$
- $H_k$: measurement model in discrete time at $t = t_k$
- $P_k$: estimate covariance in discrete time at $t = t_k$
- $\mathbf{w}$: process uncertainty (noise) vector (of type $N(0, P_k)$)
- $Q$: process noise matrix, $Q = F_k [uw^T]$
- $Q_k$: $Q$ in discrete time at $t = t_k$
- $\mathbf{v}$: measurement noise vector (of type $N(0, R_k)$)
- $R_k$: the measurement variance matrix, $R_k = E[ww^T]$, in discrete time at $t = t_k$
General Problem…

![Diagram showing the flow of true state and observed state over time](image)

Duals and Dual Terminology

<table>
<thead>
<tr>
<th>Model:</th>
<th>Estimation</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x = Fx$ (discrete: $x = F_kx$)</td>
<td>$x = Ax$, $A = F'$</td>
</tr>
<tr>
<td>Regimes:</td>
<td>$P$ (covariance)</td>
<td>$M$ (performance matrix)</td>
</tr>
<tr>
<td>Minimized function:</td>
<td>$Q$ (or $Q^2$)</td>
<td>$V$</td>
</tr>
<tr>
<td>Optimal Gain:</td>
<td>$K$</td>
<td>$G'$</td>
</tr>
<tr>
<td>Completeness law:</td>
<td>Observability</td>
<td>Controllability</td>
</tr>
</tbody>
</table>

---
Estimation Process in Pictures

**System:** (unknown)

\[ X_{k-1} \rightarrow X_k \text{ noise} \]

\[ w \sim N(0, r) \]

**Measured:**

\[ z_k + U \]

\[ H_k \]

**Estimate:**

\[ \hat{X}_{k-1} \rightarrow F_{k} + w \]

\[ \hat{X}_k \]

\[ P_{k-1} \rightarrow P_k \]

Kalman Filter Process

Initial state (x) & covariance (P)

Project state & covariance forward

Measurement (z)

Compute optimal observer gain ("Kalman gain") then update state and covariance estimates
KF Process in Equations

Prediction: \( \hat{x}_{k|k-1} = F_{k-1} \hat{x}_{k-1|k-1} \)  
\( P_{e[k-1]} = Q_{k-1} + F_{k-1} P_{k-1|k-1} F_{k-1}^T \)  
Kalman Gain: \( K_k = P_{h[k-1]} H^T (H P_{h[k-1]} H^T + R_k)^{-1} \)  
Update: \( P_{e[k]} = [I - K_k H] P_{h[k-1]} \)  
\( \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H \hat{x}_{k|k-1}) \)

KF Considerations

\( \hat{x}_{k|k-1} = F_{k-1} \hat{x}_{k-1|k-1} + G_{k-1} u_{k-1} \)  
\( P_{h[k-1]} = Q_{k-1} + F_{k-1} P_{h[k-1]} F_{k-1}^T \)  
\( K_k = P_{h[k-1]} H^T (H P_{h[k-1]} H^T + R_k)^{-1} \)  
\( P_{e[k]} = [I - K_k H] P_{h[k-1]} \)  
\( \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H \hat{x}_{k|k-1} - H G_k u_{k-1}) \)
Ex: Kinematic KF: Tracking

• Consider a System with Constant Acceleration

\[
\begin{align*}
\ddot{y} &= -g \\
\dot{y} &= gt + p_1 \\
y &= p_0 + p_1 t + \frac{gt^2}{2}
\end{align*}
\]

\[
\begin{bmatrix}
\dot{y} \\
\ddot{y}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
y \\
\dot{y}
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
0 & -1
\end{bmatrix}
\begin{bmatrix}
g
\end{bmatrix}
\]

\[
F =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & \ddot{t} & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
F_k =
\begin{bmatrix}
0 & t_s & \frac{t_s^2}{2} \\
0 & 0 & t_s \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\hat{x}_k = F_{k-1}\hat{x}_{k-1} + K_k(z_k - HF_{k-1}\hat{x}_{k-1})
\]

In Summary

• KF:
  – The true state (x) is separate from the measured (z)
  – Lets you combine prior controls knowledge with measurements to filter signals and find the truth
  – It regulates the covariance (P)
    • As P is the scatter between z and x
    • So, if P \(\rightarrow\) 0, then z \(\rightarrow\) x (measurements \(\rightarrow\) truth)

• EKF:
  – Takes a Taylor series approximation to get a local “F” (and “G” and “H”)
Example III: 2nd Order System Response

- Response of a 2nd order system to increasing levels of damping.
Damping and natural frequency

\[ z = e^{st} \]  where \[ s = -\zeta \omega_n \pm j\omega_n \sqrt{1 - \zeta^2} \]

- Poles inside the unit circle are stable
- Poles outside the unit circle are unstable
- Poles on the unit circle are oscillatory
- Real poles at \( 0 < z < 1 \) give exponential response
- Higher frequency of oscillation for larger
- Lower apparent damping for larger \( r \)
Characterizing the step response:

- Rise time (10% $\rightarrow$ 90%): $t_r \approx \frac{1.8}{\omega_0}$
- Overshoot: $M_p \approx \frac{e^{-\pi \zeta}}{\sqrt{1 - \zeta^2}}$
- Settling time (to 1%): $t_s = \frac{4.6}{\zeta \omega_0}$
- Steady state error to unit step: $e_{ss}$
- Phase margin: $\phi_{PM} \approx 100\zeta$

Rise time, overshoot, settling time, and phase margin are key specifications for 2nd Order System Specifications. These parameters help assess the system's response to a step input, with rise time indicating how quickly the system responds, overshoot showing the extent of the initial response, settling time showing how long it takes to reach steady state, and phase margin indicating robustness to frequency changes.

Mathematically:
- $t_r \approx \frac{1.8}{\omega_0}$
- $M_p \approx \frac{e^{-\pi \zeta}}{\sqrt{1 - \zeta^2}}$
- $t_s = \frac{4.6}{\zeta \omega_0}$
- $e_{ss} = \lim_{z \to 1} \{(z - 1) F(z)\}$
1. Mass $m$ \+ force $F$

\[ F = ma = m \ddot{x} \quad \therefore \quad \ddot{x} = \frac{F}{M} \]

\[ v = \dot{x} \quad \therefore \quad \dot{x} = v \]

\[ \frac{d^2x}{dt^2} = \frac{dv}{dt} = \frac{f}{m} \]

\[ X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \]

2. Electric Motor with Inertial Load

(torque) $\tau = K_1 i$  \[ \text{[Motor Torque]} \]

(voltage) $V = K_2 \omega$  \[ \text{[Back emf]} \]

\[ f_0 - \frac{1}{M} \int \frac{V}{\tau} \quad \text{acceleration} \]

\[ \int \frac{V}{\tau} + \frac{B}{\tau} \quad \text{velocity} \]

\[ \frac{B}{\tau} \quad \text{linear viscous friction} B \]

\[ \frac{k\ell}{\tau} \quad \text{Hookean, linear spring} k \]

Electrical power to the motor:

\[ P = vi = \frac{k_2 \omega^2}{k_1} \]

Mechanical power:

\[ P_m = \omega^2 \]

Note:

If $\eta = 100\% \rightarrow P_e = \frac{k_2}{k_1} P_m \quad \frac{k_2}{k_1} > 1 \quad \therefore \eta < 1$
$e_0^+ \rightarrow \text{DC motor} \rightarrow J \rightarrow \theta$

$e$: Driving voltage (from battery)
$V$: Back emf

$e - V = R \mathbf{i}$ (Ohm's Law)

$\mathbf{j} = J \frac{d\omega}{dt}$

$J \frac{d\omega}{dt} = k_1 \mathbf{i} = k_1 \frac{e - V}{R}$

$= \frac{k_1}{R} e - \frac{k_1 k_2}{R} \omega$

$\therefore \frac{d\omega}{dt} = \frac{k_1 k_2}{JR} \omega + \frac{k_1}{JR} e$

$\omega = \frac{d\theta}{dt}$

$x = [\theta] = \begin{bmatrix} \theta \\
\omega \end{bmatrix}$

$[\ddot{\theta}] = [\dot{\omega}] = \begin{bmatrix} 0 & 1 \\
0 & \frac{k_1 k_2}{JR} \end{bmatrix} [\omega] + \begin{bmatrix} 0 \\
\frac{k_1}{JR} \end{bmatrix} [e]$
Generalized Step

\[ \delta(t) = \begin{cases} \delta_1 & \text{for } t > T \\ 0 & \text{for } t \leq T \end{cases} \]

Lagrange's Equations

\[ L = T - V = KE - PE \] (in generalized coordinates)

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \mathbf{q}_i \quad i = 1, 2, 3, \ldots \]

Inverted Pendulum (moving on a cart)

\[ T_1 = \frac{1}{2} M \ddot{y}_1^2 \]

\[ T_2 = \frac{1}{2} m \left( \dot{y}_2^2 + \dot{z}_2^2 \right) \]

\[ y_2 = y + l \sin \Theta = \frac{y}{L} \delta_1 \]

\[ z_2 = l \cos \Theta = \delta_2 \]

\[ \dot{x}_2 = \dot{y} + l \dot{\Theta} c_1 \]

\[ z_2 = -l \dot{\Theta} s_1 \]
\[ T = T_1 + T_2 = \frac{1}{2} k y^2 + \frac{1}{2} m \left( (y + l \theta)^2 + l^2 \dot{\theta}^2 \right) \]

Potential Energy

\[ V = mgz_2 = mglc_1 \]

Lagrangian

\[ L = T - V = \frac{1}{2} (M+m) \dot{y}^2 + mlc_1 \dot{y} \dot{\theta} + \frac{1}{2} ml^2 \dot{\theta}^2 - mglc_1 \]

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial L}{\partial y} = f \]

II

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0 \]

\[ \frac{\partial L}{\partial y} = (M+m) \dot{y} + mlc_1 \dot{\theta} \]

\[ \frac{\partial L}{\partial \dot{y}} = 0 \]

\[ \frac{\partial L}{\partial \dot{\theta}} = mlc_1 \dot{y} + ml^2 \dot{\theta} \]

\[ \frac{\partial L}{\partial \theta} = mgl S_1 - mls_1 \dot{y} \dot{\theta} \]