Introduction to State Space

What a provincial idea!

METR 4202: Advanced Control & Robotics
Dr Surya Singh
Lecture # 9
September 20, 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><strong>State-Space Modelling</strong></td>
</tr>
<tr>
<td>10</td>
<td>27-Sep</td>
<td>State-Space Control</td>
</tr>
<tr>
<td>11</td>
<td>4-Oct</td>
<td>Study break</td>
</tr>
<tr>
<td>12</td>
<td>11-Oct</td>
<td>Motion Planning</td>
</tr>
<tr>
<td>13</td>
<td>18-Oct</td>
<td>Vision-based control (+ Prof. P. Corke or Prof. M. Srinivasan)</td>
</tr>
<tr>
<td></td>
<td>25-Oct</td>
<td>Applications in Industry (+ Prof. S. LaValle) &amp; Course Review</td>
</tr>
</tbody>
</table>
Announcements:

- **Grades:**
  - I am working on assembling the scores
  - I promise you will have them by Monday night
    - (or I will, um, um … read the course profile 100x)

- **Lab 3:**
  - Will be out by Sept 27  (or I will read the course profile 1000x)

- **Integrated BE/ME Meeting** (including Mechatronics)
  - Tuesday, 24/September ➔ 10-11a @ Hawken 50-C207

- **Cool Robotics Share Site**
  - Jared is making a “blog”. URL Soon! Thanks Ashley!

Announcements:

- **Lab 2:**
  - The “Lab 2 Points ($)” may be better viewed as:
    - a "necessary, but not sufficient” condition.
  - All team members must also be able to explain their work and the principles behind it if called on
  - (As must be a broken record now): getting the right value(s) and/or points is not enough
    - Why?
      - Even a broken clock is right twice a day.
  - If you have genuinely studied the material/project,
    - Then this should be easy (so no worries)!
Affairs of state

- Introductory brain-teaser:
  - If you have a dynamic system model with history (i.e. integration) how do you represent the instantaneous state of the plant?

  Eg. how would you setup a simulation of a step response, mid-step?
Introduction to state-space

- Linear systems can be written as networks of simple dynamic elements:

\[
H = \frac{s + 2}{s^2 + 7s + 12} = \frac{2}{s + 4} + \frac{-1}{s + 3}
\]

- We can identify the nodes in the system
  - These nodes contain the integrated time-history values of the system response
  - We call them “states”
Linear system equations

• We can represent the dynamic relationship between the states with a linear system:

\[
\begin{align*}
\dot{x}_1 &= -7x_1 - 12x_2 + u \\
\dot{x}_2 &= x_1 + 0x_2 + 0u \\
y &= x_1 + 2x_2 + 0u
\end{align*}
\]

State-space representation

• We can write linear systems in matrix form:

\[
\begin{align*}
\dot{x} &= \begin{bmatrix} -7 & 12 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\
y &= \begin{bmatrix} 1 & 2 \end{bmatrix} x + 0u
\end{align*}
\]

Or, more generally:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

“State-space equations”
State-space representation

- State-space matrices are not necessarily a unique representation of a system
  - There are two common forms

  - **Control canonical form**
    - Each node – each entry in \( x \) – represents a state of the system
      (each order of \( s \) maps to a state)

  - **Modal form**
    - Diagonals of the state matrix \( A \) are the poles ("modes") of the transfer function

State variable transformation

- Important note!
  - The states of a control canonical form system are not the same as the modal states
  - They represent the same dynamics, and give the same output, but the vector values are different!

- However we can convert between them:
  - Consider state representations, \( x \) and \( q \) where
    
    \[ x = Tq \]

    \( T \) is a “transformation matrix”
State variable transformation

- Two homologous representations:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du \\
\dot{q} &= Fq + Gu \\
y &= Hq + Ju
\end{align*}
\]

We can write:

\[
\begin{align*}
\dot{x} &= T\dot{q} = ATz + Bu \\
\dot{q} &= T^{-1}ATz + T^{-1}Bu
\end{align*}
\]

Therefore, \( F = T^{-1}AT \) and \( G = T^{-1}B \)

Similarly, \( C = HT \) and \( D = J \)

Controllability matrix

- To convert an arbitrary state representation in \( F, G, H \) and \( J \) to control canonical form \( A, B, C \) and \( D \), the “controllability matrix”

\[
C = [G \ FG \ F^2G \ \cdots \ F^{n-1}G]
\]

must be nonsingular.

Why is it called the “controllability” matrix?
Remembering the Motion Models:

- Recall from Dynamics, the Required Joint Torque is:

\[ \tau_i = D_i(q) \ddot{q}_i + C_i(q, \dot{q}_i) + h(q) + b(\dot{q}_i) \]

- Dynamical Manipulator Inertial Tensor – a function of position and acceleration
- Coupled joint effects (centrifugal and coriolis) issues due to multiple moving joints
- Gravitational Effects
- Frictional Effect due to Joint/Link movement
Let's simplify the model

- This torque model is a 2\textsuperscript{nd} order one (in position) let's look at it as a velocity model rather than positional one then it becomes a system of highly coupled 1\textsuperscript{st} order differential equations

- We will then isolate Acceleration terms (acceleration is the 1\textsuperscript{st} derivative of velocity)

\[ a = \dot{v} = \ddot{q} = D_i^{-1}(q) (\tau_i - C_i(q, \dot{q}_i) - h(q) - b(\dot{q}_i)) \]

Considering Control:

- Each Link’s torque is influenced by each other links motion
  - We say that the links are highly coupled

- Solution then suggests that control should come from a simultaneous solution of these torques

- We will model the solution as a “State Space” design and try to balance the torque-in with \textit{positional control}-out – the most common way it is done!
  - But we could also use ‘force control’ to solve the control problem!
The State-Space Control Model:

Setting up a Real Control

• We will (start) by using positional error to drive our torque devices

• This simple model is called a PE (proportional error) controller
PE Controller:

• To a 1st approximation, $\tau = K_m * I$
  
  • Torque is proportional to motor current

• And the Torque required is a function of ‘Inertial’ (Acceleration) and ‘Friction’ (velocity) effects as suggested by our L-E models

$$\tau_m \approx J_{eq} \ddot{q} + F_{eq} \dot{q}$$

$\Rightarrow$ Which can be approximated as:

$$K_m I_m = J_{eq} \ddot{q} + F_{eq} \dot{q}$$

Setting up a “Control Law”

• We will use the positional error (as drawn in the state model) to develop our torque control

• We say then for PE control:

$$\tau \propto k_{pe} (\theta_d - \theta_a)$$

• Here, $k_{pe}$ is a “gain” term that guarantees sufficient current will be generated to develop appropriate torque based on observed positional error
Using this Control Type:

- It is a representation of the physical system of a mass on a spring!
- We say after setting our target as a ‘zero goal’ that:

\[-k_p e \cdot \theta_a = J\ddot{\theta} + F\dot{\theta}\]

the solution of which is:

\[
\theta_a = e^{-\frac{F}{2}t} \left[ C_1 e^{\frac{1}{2}w_t} + C_2 e^{-\frac{1}{2}w_t} \right]
\]

\(\theta_a\) is a function of the servo feedback as a function of time!

State Space Model of PD:
PID State Space Model:

State Model of Adjustable Controller
Controllability

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 \(x_0\):
  \[
  x(t) = e^{Pt} x_0 \\
  \therefore \dot{x}(t) = \dot{p}_t e^{Pt} x_0 = F e^{Pt} x_0
  \]

Clearly, a system will be stable provided
\[
eig(F) < 0
\]
Characteristic polynomial

• From this, we can see $Fx_0 = p_ix_0$
  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 $x_3 = \dot{x}_2$

\[
\begin{bmatrix}
  1 & 1 \\
  0 & -2
\end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}
\]

\[
y = [0 \ 1 \ 0]x + 0u
\]

$x_2$ 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:

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

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!

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

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

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:
    \[
    \Gamma = \sum_{k=0}^{\infty} \frac{F^k T^k}{(k+1)!} \Gamma G
    = \Psi T G
    \]
  $\Psi$ 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
    \[ C = [\Gamma \quad \Phi \Gamma \quad \Phi^2 \Gamma \quad \ldots \quad \Phi^{n-1} \Gamma] \]
    to be full-rank.
2\textsuperscript{nd} Order System Response

- Response of a 2\textsuperscript{nd} order system to increasing levels of damping.

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

Damping and natural frequency

\[ \zeta = 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1.0 \]
Pole positions in the z-plane

- 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 $r$

2\textsuperscript{nd} Order System Specifications

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: $\phi_{PM} \approx 100\zeta$
2nd Order System Specifications

Characterizing the step response:

- Rise time (10% → 90%) & Overshoot: $t_r, M_p \rightarrow \zeta, \omega_0$: Locations of dominant poles
- Settling time (to 1%): $t_s \rightarrow$ radius of poles: $|z| < 0.01 T_s$
- Steady state error to unit step: $e_{ss} \rightarrow$ final value theorem $e_{ss} = \lim_{z \rightarrow 1} \{z - 1\} F(z)$

Ex: System Specifications $\rightarrow$ Control Design [1/4]

Design a controller for a system with:
- A continuous transfer function: $G(s) = \frac{0.1}{s(s + 0.1)}$
- A discrete ZOH sampler
- Sampling time ($T_s$): $T_s = 1s$
- $C u_k \rightarrow -0.5 u_{k-1} + 13 (e_k - 0.88 e_{k-1})$

The closed loop system is required to have:
- $M_p < 16\%$
- $t_s < 10 \text{ s}$
- $e_{ss} < 1$
Ex: System Specifications → Control Design [2/4]

1. (a) Find the pulse transfer function of \( G(s) \) plus the ZOH

\[
G(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} = \left( \frac{z-1}{z} \right) \mathcal{Z} \left\{ \frac{0.1}{s^2(s + 0.1)} \right\}
\]

e.g. look up \( \mathcal{Z} \{ \alpha/s^2(s + \alpha) \} \) in tables:

\[
G(z) = \frac{(z-1)}{z} \left\{ \frac{(0.1 - 1 + e^{-0.1})z + (1 - e^{-0.1} - 0.1e^{-0.1})}{0.1(z-1)^2(z-e^{-0.1})} \right\} = \frac{0.0484(z + 0.9672)}{(z-1)(z-0.9048)}
\]

(b) Find the controller transfer function (using \( z = \) shift operator):

\[
U(z) = D(z) = 13 \left( \frac{1 - 0.88z^{-1}}{1 + 0.5z^{-1}} \right) = 13 \frac{(z - 0.88)}{(z + 0.5)}
\]


2. Check the steady state error \( e_{ss} \) when \( r_k = \) unit ramp

\[
e_{ss} = \lim_{k \to \infty} e_k = \lim_{z \to 1} (z-1)E(z)
\]

\[
E(z) = \frac{1}{1 + D(z)G(z)}
\]

\[
R(z) = \frac{Tz}{(z-1)^2}
\]

\[
\therefore e_{ss} = \lim_{z \to 1} \left\{ \frac{(z-1)}{(z-1)^2} \frac{T}{1 + D(z)G(z)} \right\} = \lim_{z \to 1} \frac{T}{(z-1)D(z)G(z)} = \frac{1 - 0.9048}{0.0484(1 + 0.9672)D(1)} = 0.96
\]

\( e_{ss} < 1 \) (as required)
Ex: System Specifications $\rightarrow$ Control Design [4/4]

3. Step response: overshoot $M_p < 16\%$ $\Rightarrow$ $\zeta > 0.5$
   settling time $t_s < 10$ $\Rightarrow$ $|\zeta| < 0.01^{1/10} = 0.63$

The closed loop poles are the roots of $1 + D(z)G(z) = 0$, i.e.

$$1 + 13\frac{(z - 0.88)}{(z + 0.5)} \frac{0.0484(z + 0.9672)}{(z - 1)(z - 0.9048)} = 0$$

$\Rightarrow$ $z = 0.88, -0.050 \pm j0.304$

But the pole at $z = 0.88$ is cancelled by controller zero at $z = 0.88$, and

$$z = -0.050 \pm j0.304 = re^{\pm j\theta}$$

$$\begin{cases} r = 0.31, \theta = 1.73 \\ \zeta = 0.56 \end{cases}$$

all specs satisfied!