```
% segway_eom.m
%
% "Segway-Style" Inverted Pendulum: Equations of Motion.
% (See notes from Lecture 13 for a description of the system.)
% Katie Byl, 2012.

clear all; format compact  % compact produces single-spaced output

% Define symbolic variables in matlab:
syms phiw thetab L mb Jb mw Jw Rw g b tau

% 1a. GC's (generalized coordinates), and their derivatives:
GC = [{phiw},{thetab}]; % Using ABSOLUTE angles here
dphiw = fulldiff(phiw,GC); % time derivative. GC are variables (over time)
dthetab = fulldiff(thetab,GC);

% 1b. Geometry of the masses/inertias, given GC's are freely changing...
xw = Rw*phiw;
xb = xw+L*sin(thetab);
yw = 0;
yb = L*cos(thetab);

% 1c. Define any required velocity terms (for masses):
dxw = fulldiff(xw,GC);
dxb = fulldiff(xb,GC);
dyb = fulldiff(yb,GC);

% 2. Kinetic Energy:
T = (1/2)*(mw*dxw^2 + Jw*dphiw^2 + mb*(dxb^2 + dyb^2) + Jb*dthetab^2)

% 3. Potential Energy:
V = mb*g*yb

% 4. Lagrangian:
L = T-V

% 5. EOMs:
eq1 = fulldiff(diff(L,dphiw),GC) - diff(L,phiw)
eq2 = fulldiff(diff(L,dthetab),GC) - diff(L,thetab);
eq2 = simplify(eq2)

% 6. Xi: non-conservative terms
Xi1 = tau - b*(dphiw-dthetab)  % Motor torque tau, and back emf damping b
Xi2 = -tau + b*(dphiw-dthetab)  % (equal and opposite to above)

% BELOW IS THE OUTPUT FROM THIS MATLAB SCRIPT:
```

```
Output from segway_eom.m :
(Katie Byl, 2012)
---------------------------------------------------

T =
(Jb*dthetab^2)/2 + (Jw*dphiw^2)/2 + (mb*((Rw*dphiw + L*dthetab*cos(thetab))^2 + L^2*dthetab^2*sin(thetab)^2))/2 +
(Rw^2*dphiw^2*mw)/2

V =
L*g*mb*cos(thetab)

L =
(Jb*dthetab^2)/2 + (Jw*dphiw^2)/2 + (mb*((Rw*dphiw + L*dthetab*cos(thetab))^2 + L^2*dthetab^2*sin(thetab)^2))/2 +
(Rw^2*dphiw^2*mw)/2 - L*g*mb*cos(thetab)

eq1 =
- L*Rw*mb*sin(thetab)*dthetab^2 + d2phiw*(Jw + Rw^2*mb + Rw^2*mw) + L*Rw*d2thetab*mb*cos(thetab)

eq2 =
Jb*d2thetab + L^2*d2thetab*mb - L*g*mb*sin(thetab) + L*Rw*d2phiw*mb*cos(thetab)

Xi1 =
tau - b*(dphiw - dthetab)

Xi2 =
b*(dphiw - dthetab) - tau
```

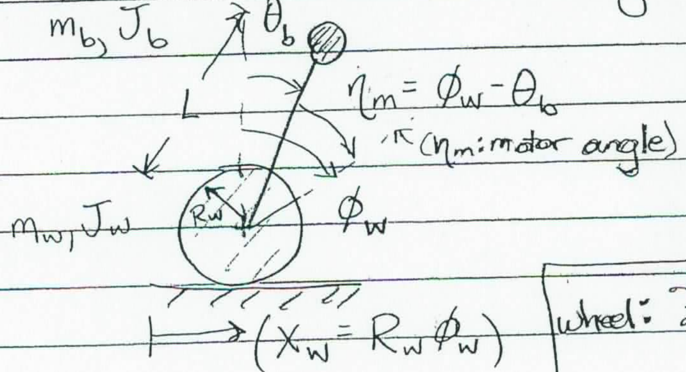
Lecture 13

$\boxed{\text{6.6 in Spong}}$

– Controllability, Observability, State Space design –

Segway example:

(neglects $n^2 J_m \ldots$)



$\xi_1 = \phi_w = $ "ξ_{ϕ_w}"

$\xi_2 = \theta_b = $ "ξ_{θ_b}"

b due to back EMF, mech. damping

wheel: $\tilde{\Xi}_1 = \pm\tilde{\tau}_m - b(\dot\phi_w - \dot\theta_b)$

body: $\tilde{\Xi}_2 = \mp\tau_m + b(\dot\phi_w - \dot\theta_b)$

sign depends on (arbitrary) Lego wiring!

$+\tilde{\tau}_m \rightarrow$ tends to $\uparrow\theta_b, \downarrow\phi_w$

1) Geometry: Represent all masses

$(y_w = 0)$

$x_w = R_w \phi_w$

$x_b = x_w + L\sin\theta_b$

$y_b = L\cos\theta_b$

2) Kinetic Energy:

for 2 wheels!

$$\boxed{T^* = \left(\tfrac{1}{2} m_w \dot{x}_w^2 + \tfrac{1}{2} J_w \dot\phi_w^2\right) + \tfrac{1}{2} m_b(\dot{x}_b^2 + \dot{y}_b^2) + \tfrac{1}{2} J_b \dot\theta_b^2}$$

To include motor inertia: $+\frac{1}{2}J_m\left(n(\dot\phi_w - \dot\theta_b)\right)^2$ (assume small)

3) Potential Energy:

$$\boxed{V = m_b g y_b = m_b g L\cos\theta_b}$$

4) Solve for any required terms: $\dot{x}_w^2, \dot{x}_b^2, \dot{y}_b^2$

ECE/ME 179D

4) (cont'd)

$\dot{x}_w = R_w \dot{\phi}_w$

$\dot{x}_b = R_w \dot{\phi}_w + L\cos(\theta_b)\dot{\theta}_b$

$\dot{y}_b = -L\sin(\theta_b)\dot{\theta}_b$

$\therefore \mathcal{L} = \left(\frac{1}{2}m_w R_w^2 + \frac{1}{2}J_w + \frac{1}{2}m_b R_w^2\right)\dot{\phi}_w^2 + \ldots$
$\left(\frac{1}{2}m_b L^2 + \frac{1}{2}J_b\right)\dot{\theta}_b^2 + \ldots$
$m_b R_w L\cos(\theta_b)\dot{\phi}_w\dot{\theta}_b + \ldots$
$-m_b g L\cos(\theta_b)$

5) $\mathcal{L} = T^* - V$

$$\boxed{\mathcal{L} = \frac{1}{2}m_w(R_w\dot{\phi}_w)^2 + \frac{1}{2}J_w\dot{\phi}_w^2 + \frac{1}{2}m_b\left((R_w\dot{\phi}_w + L\cos(\theta_b)\dot{\theta}_b)^2 + (-L\sin(\theta_b)\dot{\theta}_b)^2\right) + \frac{1}{2}J_b\dot{\theta}_b^2 - m_b g L\cos(\theta_b)}$$

$\mathcal{L} = \frac{1}{2}[m_w R_w^2 + J_w]\dot{\phi}_w^2 + \frac{1}{2}J_b\dot{\theta}_b^2 + \frac{1}{2}m_b\left(R_w^2\dot{\phi}_w^2 + 2R_w L\cos(\theta_b)\dot{\phi}_w\dot{\theta}_b + L^2\dot{\theta}_b^2\right) - m_b g L\cos\theta_b$

$(L^2\cos^2(\theta_b)\dot{\theta}_b^2 + L^2\sin^2(\theta_b)\dot{\theta}_b^2)$

6) EOM's

#1 $$\boxed{\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}_w}\right) - \frac{\partial\mathcal{L}}{\partial\phi_w} = \tilde{\tau}_{\phi_w} = +\tau_m - b(\dot{\phi}_w - \dot{\theta}_b)}$$

$$\boxed{\mathcal{L} = K_w\dot{\phi}_w^2 + K_b\dot{\theta}_b^2 + K_c\cos\theta_b\dot{\phi}_w\dot{\theta}_b - m_b g L\cos\theta_b}$$

$$\boxed{K_w \equiv \frac{1}{2}m_w R_w^2 + \frac{1}{2}J_w + \frac{1}{2}m_b R_w^2}$$

$$\boxed{K_b \equiv \frac{1}{2}m_b L^2 + \frac{1}{2}J_b}$$

$$\boxed{K_c \equiv m_b R_w L}$$

#1 $\frac{d}{dt}\left(2K_w\dot{\phi}_w + K_c\cos\theta_b\dot{\theta}_b\right) - 0 = \tilde{\tau}_{\phi_w}$

(1) $$\boxed{2K_w\ddot{\phi}_w + K_c\cos\theta_b\ddot{\theta}_b - K_c\sin\theta_b\dot{\theta}_b^2 = \tilde{\tau}_{\phi_w}} = +\tau_m - b\dot{\phi}_w + b\dot{\theta}_b$$

Eqn #2

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_b}\right) - \frac{\partial \mathcal{L}}{\partial \theta_b} = \tilde{\tau}_{\theta_b} = \tau_m + b(\dot{\phi}_w - \dot{\theta}_b)$$

$$\tilde{\tau}_{\theta_b} = \frac{d}{dt}\left(2K_b\dot{\theta}_b + K_c\cos\theta_b\dot{\phi}_w\right) - \left(-K_c\sin\theta_b\dot{\phi}_w\dot{\theta}_b + m_bgL\sin\theta_b\right)$$

$$2K_b\ddot{\theta}_b + K_c\cos\theta_b\ddot{\phi}_w \boxed{\cancel{-K_c\sin\theta_b\dot{\theta}_b\dot{\phi}_w + K_c\sin\theta_b\dot{\phi}_w\dot{\theta}_b}} - m_bgL\sin\theta_b = \tilde{\tau}_{\theta_b}$$

2

$$\therefore \boxed{2K_b\ddot{\theta}_b + K_c\cos\theta_b\ddot{\phi}_w - m_bgL\sin\theta_b = \tilde{\tau}_{\theta_b} = -\tau_m + b\dot{\phi}_w - b\dot{\theta}_b}$$

Near equilibrium, $\sin\theta_b \approx \theta_b$, $\cos\theta_b \approx 1$, $\dot{\theta}_b \approx 0$, $\dot{\phi}_w \approx 0$ (ignore H.O.T.)

#1 linearized

$$2K_w\ddot{\phi}_w + K_c\ddot{\theta}_b = +\tau_m - b\dot{\phi}_w + b\dot{\theta}_b$$

#2 linearized

$$2K_b\ddot{\theta}_b + K_c\ddot{\theta}_w - m_bgL\theta_b = -\tau_m + b\dot{\phi}_w - b\dot{\theta}_b$$

from 1: $\ddot{\theta}_b = \frac{1}{K_c}\left(-2K_w\ddot{\phi}_w + \tau_m - b\dot{\phi}_w + b\dot{\theta}_b\right)$

into 2: $2\frac{K_b}{K_c}\left(-2K_w\ddot{\phi} + \tau_m - b\dot{\phi}_w + b\dot{\theta}_b\right) + K_c\ddot{\theta}_w - m_bgL\theta = -\tau_m + b\dot{\phi}_w - b\dot{\theta}_b$

UGH!! Is there a "clean" way to solve this??

Yes!

- Can solve via **MATRIX ALGEBRA**:

Let us define all but highest order derivative of each Generalized Coordinate (GC) as a "state":

e.g. $\boxed{m\ddot{y} + b\dot{y} + ky = F}$

$\ddot{x}$ would be 2nd order deriv.

2 GC's (position variables)

4 states (pos + velocities)

first state → $x_1 = y$

second state → $x_2 = \dot{y}$

For IP robot:

$$\begin{cases} x_1 = \phi_w \\ x_2 = \theta_b \\ x_3 = \dot{\phi}_w = \dot{x}_1 \\ x_4 = \dot{\theta}_b = \dot{x}_2 \end{cases} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

We wish to use a matrix formulation to describe the set of all possible states, the "state space", over time....

i.e., to write the dynamic relationships (EOM).

$$M\dot{X} = A_m X + B_m u$$

More general goal is to write the (highest order) deriv's on LHS in terms of state variables!

$$\boxed{\dot{x} = Ax + Bu}$$

τ_m

matrix inverse

$$(M^{-1})M\dot{x} = (M^{-1}A_m)x + (M^{-1}B_m)u$$

From our equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2K_w & K_c \\ 0 & 0 & K_c & 2K_b \end{bmatrix} \begin{bmatrix} \dot{\phi}_w \\ \dot{\theta}_b \\ \ddot{\phi}_w \\ \ddot{\theta}_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -b & b \\ 0 & m_b gL & b & -b \end{bmatrix} \begin{bmatrix} \phi_w \\ \theta_b \\ \dot{\phi}_w \\ \dot{\theta}_b \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_m \\ -K_m \end{bmatrix} u$$

1) $\cancel{\dot{\phi}_w = \phi_w}$ $\rightarrow$, $\frac{d}{dt}(\phi_w) = \dot{\phi}_w$, $\boxed{\dot{X}_1 = X_3}$

2) $\frac{d}{dt}(\theta_b) = \dot{\theta}_b$ $\boxed{\dot{X}_2 = X_4}$ ← no dots on RHS!

all states have dots on LHS, only!

EOM 1 3) $2K_w \frac{d}{dt}\dot{\phi}_w + K_c \frac{d}{dt}\dot{\theta}_b = -b\,\dot{\phi}_w + b\,\dot{\theta}_b + K_m u$

EOM 2 4) $K_c \frac{d}{dt}\dot{\phi}_w + 2K_b \frac{d}{dt}\dot{\theta}_b = m_b g L\,\theta_b + b\,\dot{\phi}_w - b\,\dot{\theta}_b - K_m u$

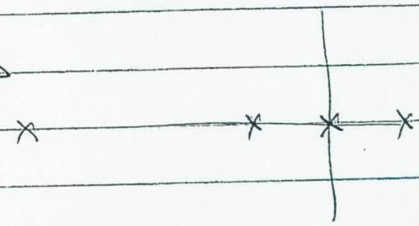
$$M \approx \begin{bmatrix} \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} & 0 \\ 0 & \begin{matrix} .0006 & .0017 \\ .0017 & .0072 \end{matrix} \end{bmatrix}$$

Form of A: $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -a & -c & +c \\ 0 & +b & \left(\frac{c}{d}\right) & \left(\frac{-c}{d}\right) \end{bmatrix}$, B: $\begin{bmatrix} 0 \\ 0 \\ f \\ \left(\frac{-f}{d}\right) \end{bmatrix}$

$$A \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -650 & -270 & 270 \\ 0 & +230 & 70 & -70 \end{bmatrix} \qquad B \rightarrow \begin{bmatrix} 0 \\ 0 \\ 280 \\ -74 \end{bmatrix}$$

State Feedback Control

eig(A) → open-loop poles



Control Law is: $\boxed{u = -Kx} + r$

Then $\dot{x} = (A - BK)x$

Closed-loop poles:

eig(A − BK) → need to be stable.

LQR − minimize a cost fn.

$u^T R u$

$$J = \int_{t=0}^{t=\infty} \left[x^T(t)\, Q\, x(t) + R u^2(t) \right] dt$$

Typically, Q is a diagonal matrix, giving penalty for each state error.

$$Q_{1,1} x_1^2 + Q_{2,2} x_2^2 + Q_{3,3} x_3^2 + Q_{4,4} x_4^2 + R u^2$$

in matlab

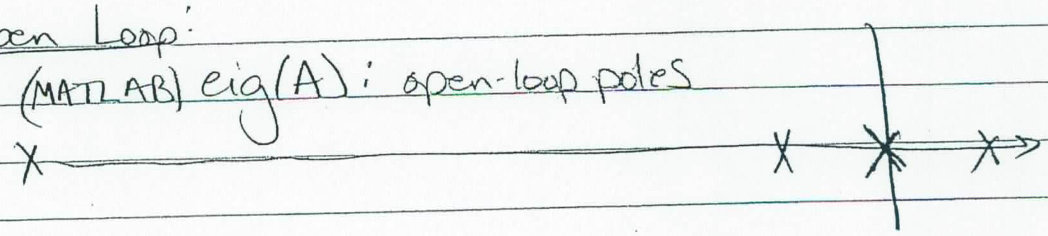
$$\boxed{K_{LQR} = \mathrm{lqr}(A, B, Q, R)}$$

Form of A & B

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -a_{32} & -a_{33} & +a_{33} \\ 0 & +a_{42} & \left(\frac{a_{33}}{d}\right) & \left(\frac{-a_{33}}{d}\right) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ b_{31} \\ \left(\frac{-b_{31}}{d}\right) \end{bmatrix}$$

Open Loop:

(MATLAB) eig(A): open-loop poles



State feedback: Recall, $\dot{x} = Ax + Bu$. What is "u"?

- Linear control laws, where actuators get inputs that are a linear combination of the states:

$$\boxed{u(t) = -Kx + r}$$

Gain matrix (K) ; reference input (r)

Example: Segway. One actuator. (2 wheels get same inputs...)

$$\underset{(4\times1)}{x} \equiv \begin{bmatrix} \phi_w \\ \theta_b \\ \dot{\phi}_w \\ \dot{\theta}_b \end{bmatrix}, \quad K = [K_1 \quad K_2 \quad K_3 \quad K_4]$$

$$r = 0$$

$$\therefore \boxed{u = -K_1\phi_w - K_2\theta_b - K_3\dot{\phi}_w - K_4\dot{\theta}_b = -Kx}$$

$$\dot{x} = Ax + Bu \rightarrow u = -Kx$$
$$= Ax - BKx$$
$$= (A - BK)x$$

→ Now, instead of A, the dynamics are characterized by $(A - BK)$

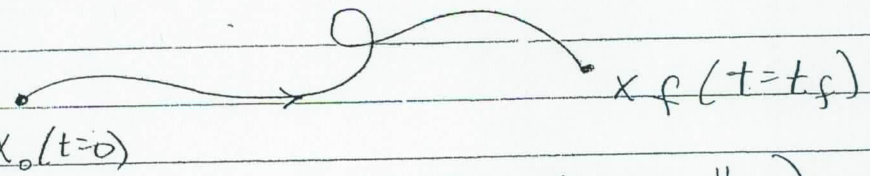
MATLAB:
eig(A-BK) : closed-loop poles. 4th order, so 4 poles.
char. eqn can be written: $s^4 + \alpha_4 s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1 = 0$

e.g., for 2nd order system: $(s - p_1)(s - p_2) = 0$
$$s^2 - (p_1 + p_2)s + p_1 p_2 = 0$$
$$s^2 + \alpha_2 s + \alpha_1 = 0$$
$$\alpha_2 = -p_1 - p_2 \qquad \alpha_1 = p_1 p_2$$

Two (related) questions of interest:
1) Given matrices A & B, can we find a control law, $u = -Kx$, to produce any, arbitrary polynomial in s as a characteristic eqn?
{ i.e. can we set $\alpha_1, \alpha_2, \alpha_3$, etc. arbitrarily? i.e., can we place poles anywhere?

eig(A-BK) ⟷ Pick value anywhere on s-plane?

(2) Can we take the system from any initial state, x_0, to any final state, x_f, in a finite interval of time?



(For non linear system only Def'n #2 "makes sense"...)

For a linear system, these are two ways of asking the SAME question!!!

$$\boxed{\text{"Is the system Controllable?"}}$$

Answer: Yes, iff $\det\underbrace{[B, AB, A^2B, \ldots A^{n-1}B]}\neq 0$

Controllability matrix: $C_{\text{segway}} \equiv [B, AB, A^2B, A^3B]$

if $\operatorname{rank}(C)$: $\begin{cases} =n, & \text{then controllable} \\ <n, & \text{not controllable} \end{cases}$

$$\boxed{\operatorname{rank}(C_{\text{segway}}) = 4}$$

→ C, more or less, shows how states can be affected at later times, due to an input now...

If system is controllable, to set (or "place") pole locations for the closed-loop (CL) system using MATLAB, can use either:

$\boxed{\texttt{place(A, B, pdes)}}$

(pdes cannot have "repeated" poles w/ multiplicity greater than # inputs in u)
(for place command)

or:

$\boxed{\texttt{acker(A, B, pdes)}}$

not so accurate for higher-order systems (n>10)
(for acker command)

This is known as "pole placement"

Problems → Commanding arbitrary pole locations may require very large gains (& therefore saturate real-world actuator(s))!

Alternative "LQR" - linear quadratic regulator
L Q R

Solves a linear quadratic optimal control problem.

[increasing R tends to reduce required u(t)]

$$J = \int_{t=0}^{t=\infty} \left[x^T(t)\, Q\, x(t) + u^T(t)\, R\, u(t) \right] dt$$

[for scalar x & u, $x^T Q x = Q x^2$, $u^T R u = R u^2$ } "quadratic", as in SQUARED]

Above, J is a cost function to minimize.
Optimal choice of gains, $\boxed{K = K_{opt}}$, minimizes this cost.
in MATLAB: $\boxed{\texttt{K = lqr(A,B,Q,R)}}$

Q & R are typically DIAGONAL:

for Segway → $J = \int (Q_{11}x_1^2 + Q_{22}x_2^2 + Q_{33}x_3^2 + Q_{44}x_4^2 + Ru^2)\, dt$