15-830 – Control 3: Control of Dynamical Systems

J. Zico Kolter

November 27, 2012

PID Control

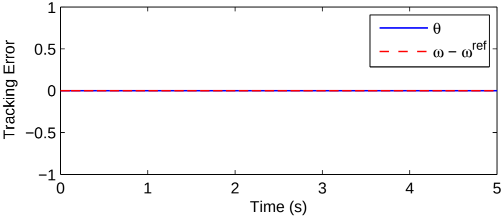
- *P*roportional *I*ntegral *D*erivative
- Remember linear system of generator

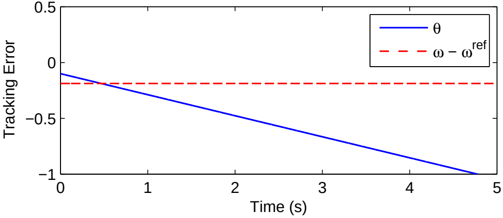
$$\dot{\theta} = \omega - \omega^{\mathrm{ref}}$$
$$\dot{\omega} = \frac{1}{2H}(u - p^{\mathrm{elec}})$$

- Goal is to achieve/maintain $\theta = 0$

- Attempt #1 (feed-forward control):

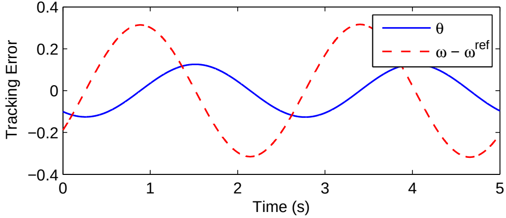
$$u_t = p^{\text{elec}}$$



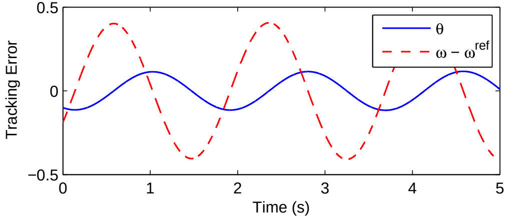


- Attempt #2 (P control):

$$u_t = p^{\text{elec}} + K_p(\theta_t^d - \theta_t)$$
$$= p^{\text{elec}} - K_p\theta_t$$



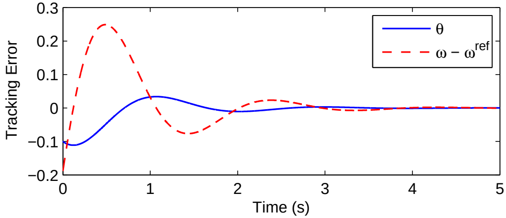
$K_p = 50$



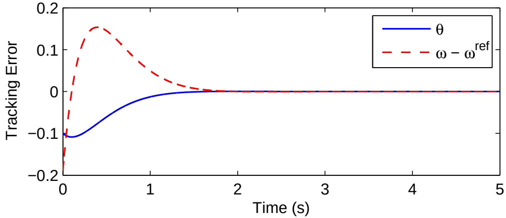
$K_p = 100$

- Attempt #2 (PD Control):

$$u_t = p^{\text{elec}} + K_p(\theta_t^d - \theta_t) + K_d(\dot{\theta}_t^{\,d} - \dot{\theta}_t)$$
$$= p^{\text{elec}} - K_p\theta_t - K_d(\omega_t - \omega^{\text{ref}})$$



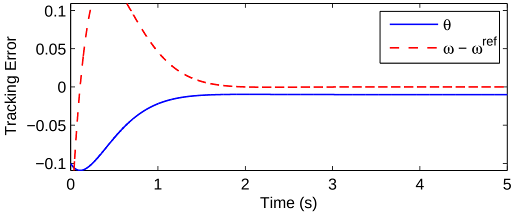
$K_p = 100,\ K_d = 20$



$K_p = 100,\ K_d = 50$

- Looks good, but what if we don't know p^{elec} beforehand?

$$u_t = K_p(\theta_t^d - \theta_t) + K_d(\dot{\theta}_t^{\,d} - \dot{\theta}_t)$$

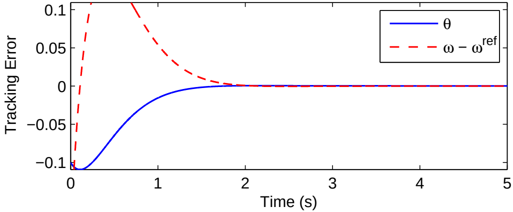


- θ never reaches desired value

- Attempt #3 (PID Control):

$$u_t = K_p(\theta_t^d - \theta_t) + K_d(\dot{\theta}_t^{\,d} - \dot{\theta}_t) + K_i \sum_{\tau=1}^{t} (\theta_\tau^d - \theta_\tau)$$

$$= -K_p\theta_t - K_d(\omega_t - \omega^{\text{ref}}) - K_i \sum_{\tau=1}^{t} \theta_\tau$$



$$K_p = 100,\ K_d = 50,\ K_i = 15\Delta t$$

Multi-variate PID control

- PID control works well for controlling “single input, single output” (SISO) systems
  - For second order linear systems, it is the “optimal” method
- For higher-order or multi-variate systems, it is no longer optimal, but often works well anyway
- Can require a fair amount of tuning

- Example: multiple generators and DC power flow approximation

$$\left.\begin{aligned}\dot{\theta}_i &= \omega_i - \omega^{\text{ref}} \\ \dot{\omega}_i &= \frac{1}{2H_i}(u_i - p_i)\end{aligned}\right\} i \in \text{GEN}$$

$$p = B\theta$$

- A set of differential algebraic equations, but since algebraic equations are linear, we can invert them directly to form ordinary differential equations

$$\begin{bmatrix} p_G \\ p_L \end{bmatrix} = \begin{bmatrix} B_{GG} & B_{GL} \\ B_{LG} & B_{LL} \end{bmatrix} \begin{bmatrix} \theta_G \\ \theta_L \end{bmatrix}$$

- Eliminate θ_L variables

$$p_L = B_{LG}\theta_G + B_{LL}\theta_L \implies \theta_L = B_{LL}^{-1}p_L - B_{LL}^{-1}B_{LG}\theta_G$$

$$p_G = B_{GG}\theta_G + B_{GL}\theta_L = (B_{GG} - B_{GL}B_{LL}^{-1}B_{LG})\theta_G + B_{GL}B_{LL}^{-1}p_L$$

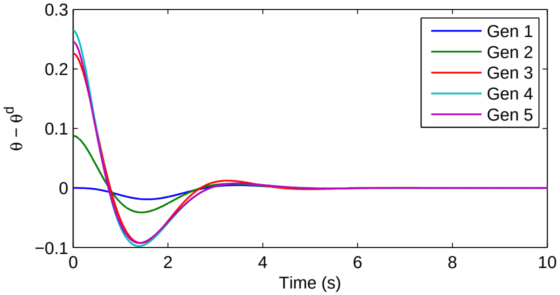
- Results in dynamical system

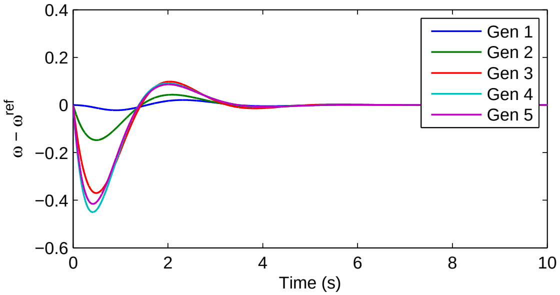
$$\dot{\theta} = \omega - \omega^{\mathrm{ref}}$$

$$\dot{\omega} = \frac{1}{2H}(u - (B_{GG} - B_{GL}B_{LL}^{-1}B_{LG})\theta - B_{GL}B_{LL}^{-1}p_L)$$

ω time derivative couples together the dynamics of the different generators

- PID control still works surprisingly well





Linear Quadratic Control

- Returning to optimal control formulation

$$\text{pick } u_{1:T} \text{ to minimize } J = \sum_{t=1}^{H} C(x_t, u_t)$$

- Remember from intro lecture that we can solve this when dynamics are linear and costs/constraints are convex

- An important special case: linear dynamics and quadratic costs, with no control or state constraints: Linear Quadratic Regulator (LQR)

$$x_{t+1} = Ax_t + Bu_t$$
$$C(x_t, u_t) = \|Qx_t\|_2^2 + \|Ru_t\|_2^2$$

- Can write as the optimization problem

$$\begin{array}{ll} \underset{x_{1:T}, u_{1:T}}{\text{minimize}} & \sum_{t=1}^{H} \left(\|Qx_t\|_2^2 + \|Ru_t\|_2^2 \right) \\ \text{subject to} & x_{t+1} = Ax_t + Bu_t \end{array}$$

- However, it turns out for this special case we get an analytical solution of the form

$$u_t^\star = K_t x_t$$

for some matrices $K_t \in \mathbb{R}^{m \times n}$, $t = 1, \ldots, H$

- Derivation is a bit involved, but just linear algebra operations

- Even more interesting: we can solve the *infinite time* problem

$$\begin{array}{ll} \underset{x_{1:T}, u_{1:T}}{\text{minimize}} & \sum_{t=1}^{\infty} \left(\|Qx_t\|_2^2 + \|Ru_t\|_2^2 \right) \\ \text{subject to} & x_{t+1} = Ax_t + Bu_t \end{array}$$

  and solution is given by *steady-state* matrix

$$u_t = Kx_t$$

- Intuition: once we achieve $x_t = 0$, $u_{t'} = 0$ and $x_{t'} = 0$ for all $t' \geq t$; if system is controllable, we can achieve this in finite time, so infinite horizon cost is finite

- So common, there is a MATLAB routine for this

```
K = dlqr(A, B, Q'*Q, R'*R);
```

- Example: generator control

$$\dot{\theta} = \omega - \omega^{\text{ref}}$$
$$\dot{\omega} = \frac{1}{2H}(u - (B_{GG} - B_{GL}B_{LL}^{-1}B_{LG})\theta - B_{GL}B_{LL}^{-1}p_L)$$

- Write as linear systems

$$\dot{x} = Ax + Bu + a$$
$$x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$$
$$A = \begin{bmatrix} 0 & I \\ -\frac{1}{2H}(B_{GG} - B_{GL}B_{LL}^{-1}B_{LG}) & 0 \end{bmatrix}$$
$$B = \begin{bmatrix} 0 \\ \frac{1}{2H}I \end{bmatrix}$$
$$a = \begin{bmatrix} -\omega^{\text{ref}}1 \\ -\frac{1}{2H}B_{GL}B_{LL}^{-1}p_L \end{bmatrix}$$

- Convert to discrete-time system

$$\begin{aligned} x_{t+1} &= (I + \Delta t A)x_t + \Delta t B x + \Delta t a \\ &= \tilde{A}x_t + \tilde{B}x_t + a_t \end{aligned}$$

- Given some *equilibrium point* $x^\star$, $u^\star$

$$x^\star = \tilde{A}x^\star + Bu^\star + a$$

we can convert this affine system to a linear system in the variables $\Delta x_t = x_t - x^\star$, $\Delta u_t = u_t - u^\star$

$$\Delta x_t = \tilde{A}\Delta x_t + \tilde{B}\Delta u_t$$

- Define a cost function on deviation from optimal state

$$C(x_t, u_t) = \|Q(x_t - x^\star)\|_2^2 + \|R(u_t - u^\star)\|_2^2 = \|Q\Delta x_t\|_2^2 + \|R\Delta u_t\|_2^2$$

- Then optimal LQR solution given by

$$\Delta u_t = K\Delta x_t \iff u_t = u^\star + K(x_t - x^\star)$$

- Notice that the LQR solution

$$u_t = u^\star + K(x_t - x^\star)$$

is a generalization of the PD controller with feedforward control

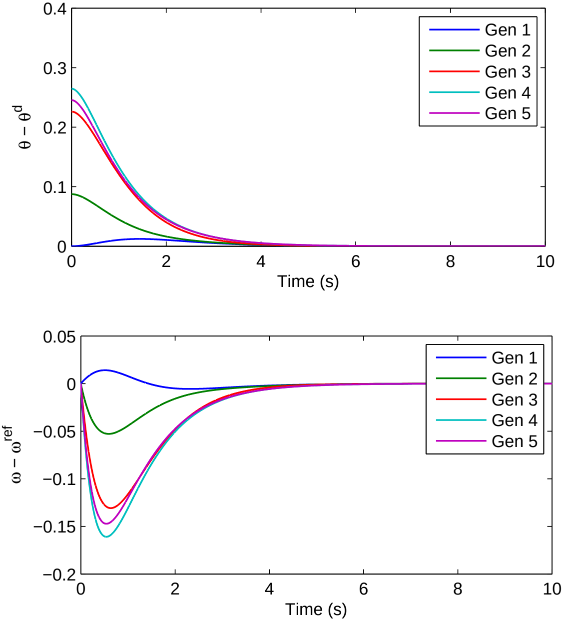
$$u_t = u^\star + \begin{bmatrix} -K_p I & -K_d I \end{bmatrix} (x_t - x^\star)$$

- However, if K is full, then LQR controller accounts for interdependence of state variables

- Also, it can be much more intuitive to specify the cost function

$$C(x_t, u_t) = \|Q(x_t - x^\star)\|_2^2 + \|R(u_t - u^\star)\|_2^2$$

than to guess control gains (cost specifies what we actually want to optimize)

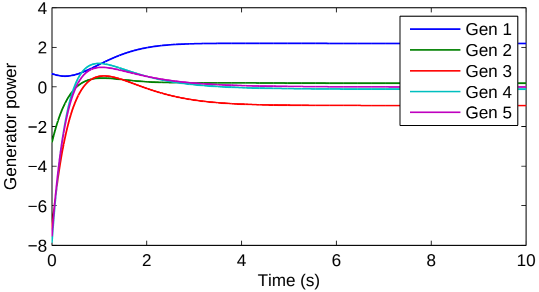


0.4
0.3
0.2
0.1
0
$\theta - \theta^{d}$
0
2
4
6
8
10
Time (s)
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
0.05
0
−0.05
−0.1
−0.15
−0.2
$\omega - \omega^{ref}$
0
2
4
6
8
10
Time (s)
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5

Issues with LQR

- Sometimes, it is difficult to express the costs/constraints of a control task with just a quadratic cost function

- Control inputs from LQR controller



Inputs are similar for the above PD/PID controller

- LQR cannot enforce bounds on control inputs, cannot enforce hard constraints on resulting states

- Some heuristics for dealing with these issues

  - Take LQR controls and clip them to allowable region

  - Tune quadratic penalties (possibly varying over time), to ensure desired behavior

- Ultimately, little can be said about how well these methods will perform

Control via Optimization

- An alternative solution: return to the paradigm of control as optimization

- Recall LQR was just solving the (convex) optimization problem

$$
\begin{aligned}
\underset{x_{1:T}, u_{1:T}}{\text{minimize}} \quad & \sum_{t=1}^{H} \left(\|Q x_t\|_2^2 + \|R u_t\|_2^2 \right) \\
\text{subject to} \quad & x_{t+1} = A x_t + B u_t \\
& x_1 = x_{\text{init}}
\end{aligned}
$$

- We can easily augment this to include explicit bounds on states and controls

$$\begin{aligned}\underset{x_{1:T},u_{1:T}}{\text{minimize}} \quad & \sum_{t=1}^{H} \left(\|Qx_t\|_2^2 + \|Ru_t\|_2^2\right) \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t \\ & x_1 = x_{\text{init}} \\ & \underline{u} \leq u_t \leq \overline{u}, \;\; \underline{x} \leq x_t \leq \overline{x}\end{aligned}$$

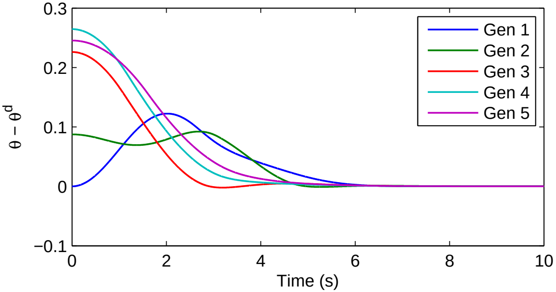
- This is a Quadratic Program, can solve using YALMIP or specialized solvers

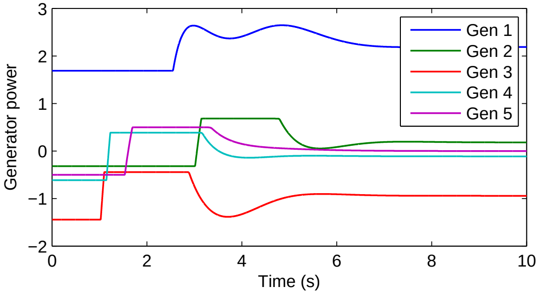
- Example: generator control with power limited to nominal power output ± 0.5 p.u.

```
x = sdpvar(2*N, T);
u = sdpvar(N, T);
C = [x(:,2:end) == A*x(:,1:end-1) + B*u(:,1:end-1) + a;
     x(:,1) == [zeros(n,1); omega_ref*ones(N,1)];
     u_star - 0.5 <= u;
     u_star + 0.5 >= u;]
solvesdp(C, norm(x-x_star,'fro')^2 + ...
         1e-3*norm(u-u_star,'fro')^2);
```

- Takes about 10 seconds to solve with YALMIP (for $T = 10000$)

- Output is a sequence of optimal control actions $u_{1:T}$, *not* a feedback controller $u_t = Kx_t$


Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
0.3
0.2
0.1
0
−0.1
0
2
4
6
8
10
Time (s)
$\theta - \theta^d$


Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
3
2
1
0
−1
−2
0
2
4
6
8
10
Time (s)
Generator power

- Many advantages and disadvantages to PID, LQR, and optimization (many others in addition to these)

	Pros	**Cons**
PID	Easy to implement (even without model)	Gain tuning can be "art"; cannot always apply to multi-variate systems
LQR	Gives feedback controller $u_t = Kx_t$; easy to compute (with MATLAB)	Can't incorporate constraints; requires model
Opt	Can incorporate constraints; directly solves optimal control problem	More time consuming; doesn't give feedback controller