L3, Wednesday

Ch. 7 Optimal Control

- If system is controllable, can put poles anywhere.
  n-dim sys. $\Rightarrow$ n poles $\to$ where to put them?

- Rules of thumb: stabilize as needed; speed up slowest modes; leave rest untouched

- Here: define a "performance index" or "cost funct."
  – choose controller to minimize the cost

1d example:

$$J\left(u_0^\infty, x(0)\right) \equiv \int_0^\infty dt\, L(x,u) \equiv \int_0^\infty dt \cdot \tfrac{1}{2}\left(Qx^2(t) + Ru^2(t)\right)$$

$$\dot{x} = -ax + u(t), \qquad x(0) = x_0, \qquad a > 0$$

- J is a functional of $u_0^\infty \equiv u(t)$, $t \in [0, \infty)$
  and a function of $x(0) = x_0$ [but not of x_0^∞]

- need $J \geq 0$ (finite lower bound) $\Rightarrow$ $Q \geq 0$, $R \geq 0$
  - $R = 0$ $\Rightarrow$ want best performance (min $x(t)$)
    regardless of control "effort"
  - $Q = 0$ $\Rightarrow$ only care about cost (do nothing)

scale $Q \to 1$

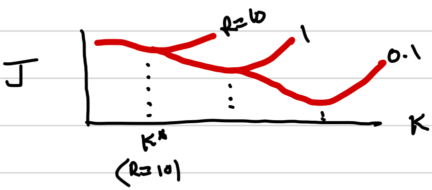
Let $u = -kx$ (will justify this form later)
$\Rightarrow$ $\dot{x} = -(a+k)x$ $\Rightarrow$ $x(t) = x_0 e^{-(a+k)t}$ $\Rightarrow$ $J = x_0^2 \left(\dfrac{1+Rk^2}{4(a+k)}\right)$

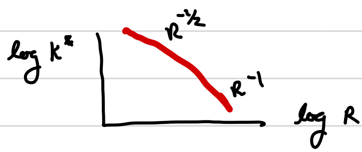
$$\frac{\partial J}{\partial K} = 0 \quad \Rightarrow \quad K^2 + 2aK - \frac{1}{R} = 0 \quad \Rightarrow \quad K^* = -a \pm \sqrt{a^2 + 1/R}$$

choose + so $K^* > 0$

$Ra^2 \ll 1$: cheap control, $J \approx \int_0^\infty dt \, \frac{1}{2} x^2$, $K^* \sim R^{-1/2} \to \infty$

$Ra^2 \gg 1$: expensive control, $J \approx \int_0^\infty dt \, \frac{1}{2} u^2$, $K^* \sim \frac{1}{2Ra} \to 0$





- Rather than choose K, we choose R (or R/Q)
- We have more intuition about R than K (here, a bit artificial)
- Optimal ≠ good! (poor R ⇒ poor control)

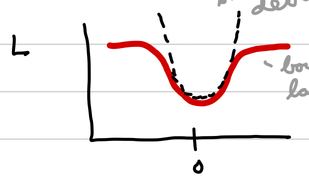
Q: unstable sys?

General setting

$$J = \varphi[x(\tau)] + \int_0^\tau dt \; L(x,u)$$

terminal cost running cost

- Dynamics $\dot{x} = f(x,u)$ are a constraint
- $J = J[u_0^\tau, x_0]$
- $\tau \to \infty \Rightarrow$ drop $\varphi(\cdot)$
- can replace soft constraint $\varphi(\cdot)$ with hard constraint
  - BC: $x(\tau) = x_\tau$



- Running cost $L(x, u)$
  - scalar
  - bounded from below
  - smooth (eg., twice differentiable)

- Large φ, small L: "ends justify the means".

Impose constraints by Lagrange multiplier

transpose of n-dim vector

$$J' \equiv \varphi[x(\tau)] + \int_0^\tau dt\, L(x,u) + \lambda^T(t)\left[f(x,u) - \dot{x}\right]$$

- Lagrange mult. $\lambda(t)$ is adjoint vector (costate)

- $\lambda(t)$ is a function of time because $\dot{x} = f$ must be imposed at every moment in time

- Solve unconstrained problem by allowing functional variations $\delta x(t)$, $\delta u(t)$, $\delta\lambda(t)$

$$\delta J' = (\partial_x \varphi)\, \delta x(\tau) + \int_0^\tau dt \Big[(\partial_x L)\,\delta x + (\partial_u L)\,\delta u + \lambda^T\big((\partial_x f)\,\delta x + (\partial_u f)\,\delta u - \delta\dot{x} \big) + \delta\lambda^T (f - \dot{x}) \Big]$$

$$= (\partial_x \varphi)\,\delta x(\tau) - \lambda^T(\tau)\,\delta x(\tau) + \lambda^T(0)\,\delta x(0) + \int_0^\tau dt \left[(\partial_x L) + \lambda^T(\partial_x f) + \dot{\lambda}^T \right] \delta x(t) + \int_0^\tau dt \left[(\partial_u L) + \lambda^T(\partial_u f) \right] \delta u(t) + \int_0^\tau dt\, (f - \dot{x})^T\, \delta\lambda(t)$$

use $a^T b = b^T a$

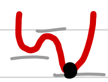
- The extra boundary terms come from integrating $\lambda^T(t)\,\delta\dot{x}(t)$ by parts
- $x(0)=x_0$ is fixed $\Rightarrow$ $\delta x(0)=0$
- In classical mechanics (CM), $u(t)$ is fixed
  $\Rightarrow$ $x(\tau)$ fixed $\Rightarrow$ $\delta x(\tau)=0$ not here!

$\delta J' = 0 \Rightarrow$ Euler-Lagrange eqs.

- Use Thm: $\int_0^\tau dt\; f(t)\,\delta\varphi(t)=0 \;\;\forall\,\delta\varphi(t) \Rightarrow f(t)=0 \;\; (0<t<\tau)$

$0\to\tau$:	$\dot{x} = f(x,u)$	$x(0)=x_0$
$\tau\to 0$:	$\dot{\lambda} = -(\partial_x f)^T\lambda - (\partial_x L)^T,$	$\lambda(\tau) = \left(\partial_x \varphi\vert_\tau\right)^T$
t :	$0 = (\partial_u f)^T\lambda + (\partial_u L)^T$	

- These eqs. give only necessary conditions

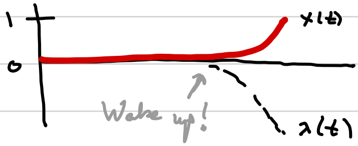


- Interpretation of adjoint $\lambda(t)$ (1d)
  - ignore λ-dependence in $(\partial_x f)$ $\Rightarrow$ $\dot{\lambda} = -(\partial_x f)\lambda - (\partial_x L)$
  - planning horizon (timescale) $\sim -(\partial_x f)^{-1}$
  - driven by $-\partial_x L$ term
  - backwards in time
- Together: Boundary-value problem (not init val.)

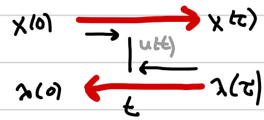
1d ex. (again) $\quad L = \frac{1}{2}(x^2 + u^2)$, $\quad \dot{x} = -x + u$

– but now let **$x(0) = x_0$** and impose **$x(\tau) = x_\tau$**

Euler-Lagrange: $\quad \dot{x} = -x + u$, $\quad \dot{\lambda} = +\lambda - x$, $\quad u = -\lambda$



- Nature of control signal $u(t)$
  - depends on past states $\quad x_0^t$
  - depends on future plans $\quad \lambda_t^\tau$



Ex: **Swing up a pendulum**

	$(\theta, \dot{\theta})$
$t = 0$:	$(0, 0)$
$t = \tau$:	$(\pi, 0)$

boundary conditions

$$\ddot{\theta} + \sin\theta = u(t)$$

torque control

$$J = \int_0^\tau dt \cdot \tfrac{1}{2} u^2(t) \qquad \text{“control effort”}$$

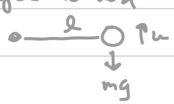
For DC motor, $u \sim$ torque $\sim I$, $\int dt\, I^2(t) \sim$ Work

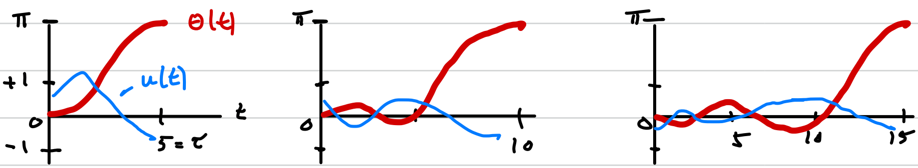
hard constraint on $x = \{\theta, \dot{\theta}\}$ at $t = \tau$ ⇒ no λ cond.

E-L: $\quad \ddot{\lambda} + \lambda \cos\theta = 0 \qquad u(t) = -\lambda(t)$

$u = mgl \equiv 1$

torque to hold





- Increasing τ ⇒ reduce $|u|_{max}$
- # reversals increases w/ τ

7.3 Linear Quadratic Regulator (LQR)

- small deviations, quadratic costs

$$J = \int_0^\tau dt \; \tfrac{1}{2}\left(x^T Q x + u^T R u\right)$$

Q, R symmetric
$R > 0, \quad Q \geq 0$
add final cond if $Q=0$

E-L: $\dot{x} = f(x,u)$ w/ $f = Ax + Bu$; $f(0,0)=0$ $\qquad L = \tfrac{1}{2}(x^T Q x + u^T R u)$

$\dot{\lambda} = -(\partial_x f)^T \lambda - (\partial_x L)^T$

$0 = (\partial_u f)^T \lambda + (\partial_u L)^T$

$$\dot{x} = Ax + Bu, \quad \dot{\lambda} = -A^T \lambda - Qx \qquad u = -R^{-1} B^T \lambda$$

- Can try $\lambda(t) = S(t)\, x(t)$ — S: $n \times n$ matrix

this trick works only for linear dynamics!

$$u = -R^{-1} B^T S x \equiv -Kx$$

K: $m \times n$ matrix

$$\dot{\lambda} = -Qx - A^T(\underbrace{Sx}_{\lambda}) = \tfrac{d}{dt}(Sx) = \dot{S}x + S(\underbrace{Ax - BR^{-1}B^T Sx}_{\dot{x}})$$

$$\dot{S} = -Q - A^T S - SA + SBR^{-1}B^T S$$

factor $x(t)$; holds $\forall t$
continuous time Riccati Eq.

steady state

$$0 = -Q - A^T S - SA + SBR^{-1}B^T S$$

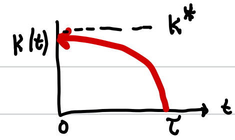
algebraic Riccati Eq.

- quadratic eq $\Rightarrow$ multiple sol'ns; only 1 is physical

7

$$k = R^{-1}(1)S = S/R$$

Ex: 1d control yet again: $A = -a$, $B = Q = 1$



$$\Rightarrow \quad \dot{S} = -1 + 2aS + S^2/R$$

$$\Rightarrow \quad S = -aR \pm \sqrt{a^2R^2 + R} \qquad \text{steady state}$$

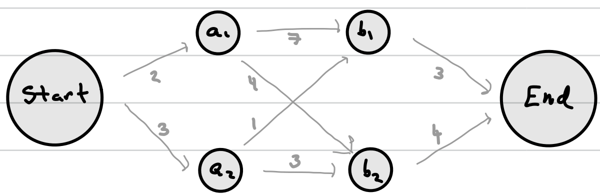
But fully optimal solution for finite τ has $K = K(t)$

$K \to 0$ as $t \to \tau$ (it costs for control, but no benefit if applied too close to end)

7.4 Dynamic Programming

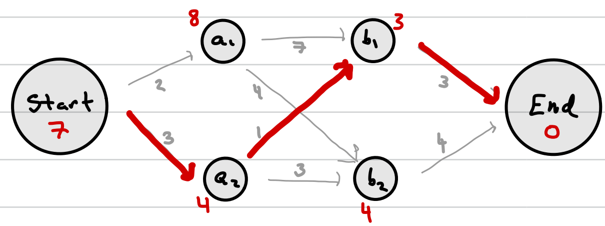
Richard Bellman, 1950s (RAND)

Shortest path between cities



$J(x) =$ cost-to-go

$J^* =$ optimal J



Day 0 1 2 3

$J^*(\text{End}) = 0$ You are already there.

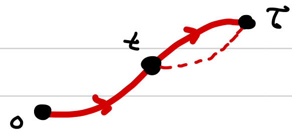
$J^*(b_1) = 3$, $J^*(b_2) = 4$ only one possibility in each case

$J^*(a_1) = \min[7 + J^*(b_1), 4 + J^*(b_2)] = 8$

$J^*(a_2) = \min[1 + J^*(b_1), 3 + J^*(b_2)] = 4$

$J^*(\text{Start}) = \min[2 + J^*(a_1), 3 + J^*(a_2)] = 7$

Principle of Optimality:



For any point on an optimal trajectory, the remaining trajectory is optimal, starting at that point.

Bellman Eq. (discrete case)

$x_k =$ state — city at time k; eg a_1, a_2

$u_k =$ action — road choice at time k

$L(x_k, u_k) \equiv L_k$ — running cost of current step, given x_k, u_k

$J(x_k, u_k) \equiv J_k$ — cost-to-go; start from x_k, choose $u_k, \dots u_{N-1}$

$\varphi(x_N)$ — terminal cost [note: $u_N = 0$]

$$J = \sum_{n=0}^{N-1} L(x_n, u_n) + \varphi(x_N) \qquad \text{total cost}$$

$$J_k = \sum_{n=k}^{N-1} L(x_n, u_n) + \varphi(x_N)$$

$$= \underbrace{L(x_k, u_k)}_{L_k} + \underbrace{\sum_{n=(k+1)}^{N-1} L(x_n, u_n) + \varphi(x_N)}_{J_{k+1}}$$

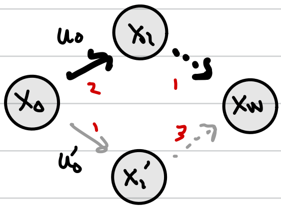
So:

$$J_k = L_k + J_{k+1} \qquad J_N = \varphi(x_N)$$

$$x_{k+1} = f(x_k, u_k) \qquad x_0 \text{ given}$$

J_k obeys a backwards recursion relation
- solving it fixes u_k^* (optimal choice at time k)
- then solve forward recursion rel'n for x_k

Bellman eq. $J^*(x_k) = \min_{\{u_k\}} \left[L(x_k, u_k) + J^*(x_{k+1}) \right]$



We choose u_0 over u_0' even though it has a higher immediate cost.

Overall future costs are lower

Solve J_k backwards; get u_k^*; iterate $x_{k+1} = f(x_k, u_k^*)$ forwards.

- Solution is possible because dynamics have a state structure $\Rightarrow$ solution decomposes into stages

Numerical algorithm: n nodes / step
$\Rightarrow$ each node has n possibilities $\Big\}$ $O(n^2)$
$\times$ N stages $\Rightarrow$ $O(N \cdot n^2)$

vs. naive search $O(n^N)$ huge difference for large N!

$\Rightarrow$ Value of planning.

Bellman algorithm (re-) discovered many times

- Cell phones (Andrew Viterbi) Qualcomm.
- Sequence alignment (bioinf.) (Needleman-Wunsch, Smith-Waterman)

"Livet skal forstås baglaens, men leves forlaens."

"Life must be understood backwards but lived forwards."

Bellman eq. (continuous case)

cost-to-go: $J(x,u) = \varphi[x(\tau)] + \int_t^\tau dt'\, L(x,u)$; $J^* = \inf_{\substack{u \\ [t,\tau]}} J(x,u)$

$$J^*(x) = \inf_u \left[\varphi[x(\tau)] + \int_t^\tau dt'\, L(x,u) \right]$$

$$= \inf_u \left[\int_t^{t+\Delta t} dt'\, L(x,u) + \varphi[x(\tau)] + \int_{t+\Delta t}^{t} dt'\, L(x,u) \right]$$

$$\simeq \inf_{\substack{u \\ [t,t+\Delta t]}} \left[L(x,u)\,\Delta t + \inf_{\substack{u \\ [t+\Delta t,\tau]}} \left[J(x + \dot{x}\Delta t,\ u(t+\Delta t)) \right] \right]$$

$$= \inf_{\substack{u \\ [t,t+\Delta t]}} \left[L(x,u)\,\Delta t + J^*(x + f\Delta t) \right]$$

$$\simeq \inf_{\substack{u \\ [t,t+\Delta t]}} \left[L(x,u)\,\Delta t + J^*(x) + (\partial_t J^*)\,\Delta t + (\partial_x J^*) f \Delta t \right]$$

$$\partial_t J^* + \inf_{u(t)} \left[L(x,u) + (\partial_x J^*)\, f(x,u) \right] = 0$$

Hamilton - Jacobi - Bellman (HJB) eq.

- Integrate backwards in time from final condition
  $J^*[x(\tau),\tau] = \varphi[x(\tau)]$
  $\Rightarrow \quad u^*(t)$

- Then integrate $\dot{x} = f(x, u^*)$ forward from $x(0) = x_0$

- $\partial_x J^* =$ n-dim row vector $\Rightarrow$ $(\partial_x J^*) f$ is a scalar

One-dim control, yet again!

$$J = \int_t^{\tau} dt' \, \tfrac{1}{2}(x^2+u^2) \,, \qquad \dot{x} = -x + u$$

HJB: $$\partial_t J^* = -\inf_u \Big[\underbrace{\tfrac{1}{2}(x^2+u^2)}_{L(x,u)} + (\partial_x J^*) \underbrace{(-x+u)}_{f(x,u)} \Big]$$

Since there are no restrictions on $u(t)$, we can do the minimization by taking ∂_u of right-hand side.

$$\partial_u \Big[\tfrac{1}{2}(x^2+u^2) + (\partial_x J^*)(-x+u) \Big] = u + (\partial_x J^*) = 0$$

$$\Rightarrow \qquad u^* = -\partial_x J^*$$

$$\Rightarrow \qquad \partial_t J^* = -\tfrac{1}{2}x^2 + \tfrac{1}{2}(\partial_x J^*)^2 + x(\partial_x J^*)$$

Looks intimidating! But try $J^*(x,t) = \tfrac{1}{2}x^2 S(t)$

$$\Rightarrow \qquad \dot{S} = -1 + S^2 + 2S \,, \qquad S(\tau) = 0$$

$$\Rightarrow \qquad u^* = -\partial_x J^* = -Sx \equiv -Kx$$

Now we can find $\dot{x} = -x - Kx = -(1+K)x$, $x(0) = x_0$, etc.

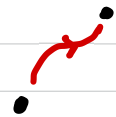
HJB vs. Euler-Lagrange

- Why do we have two apparently different ways to solve the optimal control problem?

- Same story in classical mechanics, optics, ...

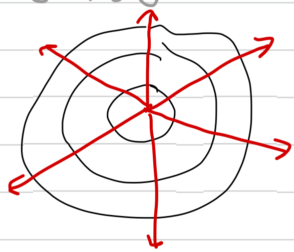
Eg., Optics

(1) Calculus of variations : $\delta \int ds \; n(s) = 0$



Fermat

(2) Eikonal (Huyghen's wavefronts)
PDE for wavefronts.



So we can solve for rays (trajectories)
or wavefronts (cost functions)

Since rays are $\perp$ wavefronts, methods are equiv.

- HJB usually much harder to solve, but you get soln for all x_0 at once.

- Euler-Lagrange is simpler but has to be redone for each x_0.

7.5 Hard constraints

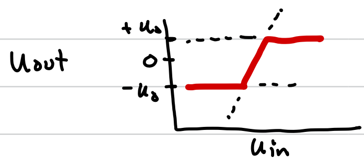
soft: $J = \varphi[x(\tau)] + \cdots$

or $= \cdots \int \frac{1}{2} dt\, R u^2(t)$

hard: $x(\tau) = x_\tau$

$|u(t)| \leq u_0$

Notice that the hard constraint $|u| \leq u_0$ is a nonlinearity.



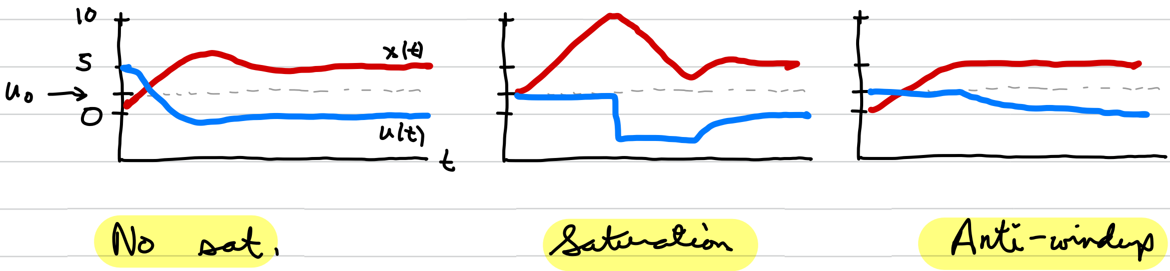
Strategies

0) Method 0: Buy a bigger actuator "increase control authority"
   – ie, increase u_0

1) Anti-windup – simple extension of PID

2) Minimum principle of Pontryagin (bang-bang control)

1) Anti-windup   Ex: $G = 1/s$, $\dot{x} = u$ (overdamped part.)
PI control, $K(s) = (1 + 1/s)$



What happened in middle case?

$$u \sim \int_0^t dt' \, e(t') \qquad e(t) = x_r - x(t)$$

When you exceed u-limit ($|u_0|$) integral keeps accum. to switch sign of $u(t)$, $x(t)$ must be below the setpoint long enough to "erase" the cumulated input.

Anti-windup: Stop integrating when u would exceed sat.

2) Minimum Principle + Bang-bang control

- Assume $u(t)$ normalized so $|u(t)| \leq 1$
  - m-components $\Rightarrow$ m-dim. hypercube

Ex: 1d dynamics, w/ constraints

$$J = \int_0^\infty dt \; \tfrac{1}{2}(x^2+u^2), \qquad \dot{x} = -x+u, \qquad x(0)=x_0$$

$$|u(t)| < 1$$

$\Rightarrow \quad u = -kx \qquad k^* = \sqrt{2}-1 \quad \Rightarrow \quad x(t) = x_0 e^{-\sqrt{2}t}$

$u(t) = -k^* x_0 e^{-\sqrt{2}t}$

When $x_0 > (k^*)^{-1} = \sqrt{2}+1$, $u(0) < -1 \Rightarrow$ sat.

Intuitive resolution: $u(t) = -1 \quad t < \tau_0$

$= -kx \quad t \geq \tau_0$

where $x(\tau_0) = x_0 e^{-\tau_0} = 1$ fixes τ_0

Hamiltonian formalism

Classical Mech: Lagrangian $L(q,\dot{q}) = T - U$ (kinetic en. / potential en.)

Generalized momenta $p = \dfrac{\partial L}{\partial \dot{q}}$

Legendre transf: $H = p\dot{q} - L$

Hamilton's eqs: $\dot{q} = (\partial_p H)^T \qquad \dot{p} = -(\partial_q H)^T$

If $H = H(p,q)$ does not have an explicit t-depend.

$$\frac{dH}{dt} = \frac{\partial H}{\partial p}\dot{p} + \frac{\partial H}{\partial q}\dot{q} = 0 \qquad \text{(via Ham. eqs.)}$$

– note change of sign!

Control-theory version: $H = L + \lambda^T \dot{x} = L + \lambda^T f$

$$\dot{x} = (\partial_\lambda H)^T = f \qquad \dot{\lambda} = -(\partial_x H)^T \qquad (\partial_u H)^T = 0$$

Much simpler than Euler-Lagrange (p. 4)

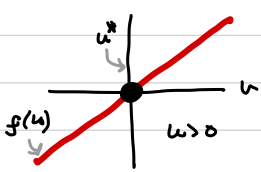
Pontryagin: $\mathcal{H}(x,\lambda) = \inf_{u \in U} H(x,\lambda,u)$

$$\dot{x} = (\partial_\lambda \mathcal{H})^T, \qquad \dot{\lambda} = -(\partial_x \mathcal{H})^T$$

$$x(0) = x_0, \qquad \lambda(\tau) = (\partial_x \varphi)^T\big|_{t=\tau}$$

If $u(t)$ is not constrained $\inf_u \Rightarrow \partial_u = 0$

With constraint, $u(t)$ might get stuck at boundary



$$\inf_{u>0} f(u) = f(0)$$

(17)

$\frac{d}{dt}\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}u = \begin{pmatrix}x_2\\u\end{pmatrix}$

Ex. **Moving in minimal time**

- free particle in space: $\dot{x}_1 = x_2, \quad \dot{x}_2 = u \qquad (u = F/m)$

Goal: go from $\begin{pmatrix}x_1\\x_2\end{pmatrix}_0 \to \begin{pmatrix}0\\0\end{pmatrix}$ as fast as possible

limits $|u(t)| \le 1$ $\qquad J = \int_0^\tau dt\,(1)$ $\qquad \tau$ not fixed

Running cost $L = 1 \quad \Rightarrow \quad H = 1 + \underbrace{\lambda_1 x_2 + \lambda_2 u}_{(\lambda_1\ \lambda_2)\begin{pmatrix}x_2\\u\end{pmatrix}}$

$$\Rightarrow \quad \dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = 0, \qquad \dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = -\lambda_1$$

$$\Rightarrow \quad \lambda_1 = \text{const}, \qquad \lambda_2 = -c_1 t + c_2$$

Can't min. H by $\frac{\partial H}{\partial u} = 0$, but can min. at boundary.

min H

$$\Rightarrow \quad \boxed{u = -\,\mathrm{sign}(\lambda_2)} \quad \text{at all times}$$

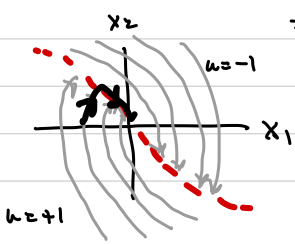
$$\Rightarrow \quad u(t) \text{ always} = \pm 1$$

Since $\lambda(t)$ varies linearly, $u(t)$ switches once.

Solve eqs. of motion separately for $u = \pm 1$

$$\Rightarrow \quad \underbrace{x_1 = \tfrac{1}{2}x_2^2 + c_p}_{u=+1}, \qquad \underbrace{x_1 = -\tfrac{1}{2}x_2^2 + c_n}_{u=-1}$$

c_p, c_n det. by init condition



Bang-bang control

In the previous problem, the control "lived" at $u = \pm u_0$

This behavior, "bang-bang control," results when $H(x, \lambda, u)$ is linear in u (or has no min. in range)

7.6 Feedback

The calc. of variations approach (or Hamiltonian) lead to a control $u^*(t)$ to take sys from $x_0 \to x_T$. via optimal $x(t)$
This is open loop, feedforward control.

If there are disturbances or modeling inaccuracies, then trajectory will deviate from optimal.
$\Rightarrow$ need some kind of feedback

1) Local linear feedback

2) Model Predictive Control

① Local linear feedback

$$J = \varphi[x(\tau)] + \int_0^\tau dt \; L(x,u), \qquad \dot{x} = f(x,u), \quad x(0) = x_0$$

$$\text{E-L} \Rightarrow \quad \dot{\lambda} = -(\partial_x f)^T \lambda - (\partial_x L)^T, \qquad 0 = (\partial_u f)^T \lambda + (\partial_u L)^T$$

- Assume we have solved the "nominal feedforward" problem:
  $\Rightarrow \quad x_{ff}(t), \quad u_{ff}(t), \quad \lambda_{ff}(t)$

- Perturb about nominal solution:

$$x(t) = x_{ff}(t) + \delta x(t), \qquad u(t) = u_{ff}(t) + \delta u(t), \qquad \lambda(t) = \lambda_{ff}(t) + \delta\lambda(t)$$

- Because the nominal "ff" solution is optimal, $\delta J = 0$

$$\Rightarrow \quad \delta^2 J = \frac{1}{2}\,\delta x^T (\partial_{xx}\varphi)\,\delta x \Big|_{t=\tau} + \int_0^\tau dt\, (\delta x^T \;\; \delta u^T) \begin{pmatrix} Q & M \\ M & R \end{pmatrix} \begin{pmatrix} \delta x \\ \delta u \end{pmatrix}$$

with $Q(t) = \partial_{xx} L, \quad M(t) = \partial_{xu} L, \quad R(t) = \partial_{uu} L$
– evaluated along $x_{ff}(t)$ and $u_{ff}(t)$

Augment the cost pert. w/ perturbed dynamics. ie $Q = \frac{\partial^2 L(x,u)}{\partial x^2}\Big|_{x_{ff},u_{ff}}$

$$\delta^2 J' = \delta^2 J + \delta\lambda^T \left[A\,\delta x + B\,\delta u - \delta\dot{x} \right] \qquad A(t) = \partial_x f, \quad B(t) = \partial_u f$$

$$\text{E-L:} \quad \delta\dot{\lambda} = -Q\,\delta x - M\,\delta u - A^T \delta\lambda, \qquad \delta u = -R^{-1}\left(M\,\delta x + B^T \delta\lambda\right)$$

This is just a slightly more complicated LQR problem!

- all matrices time dependent $A \to A(t)$, etc.
- quadratic costs have cross-term $M(t)$

Fortunately, all steps of LQR carry through analogously.

Let $\delta\lambda = S \cdot \delta x \Rightarrow \dot{S} = -Q + MR^{-1}M^T - A'^T S - SA' + SBR^{-1}$

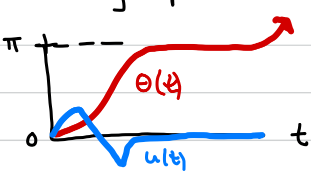
where $A' = A - BR^{-1}M^T$

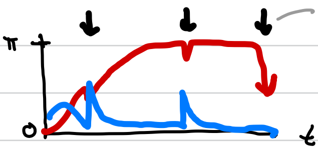
Optimal feedback gains: $K(t) = R^{-1}\left[B^T S + M^T\right]$, $\delta u_{fb} = -K\,\delta x$

The full control signal is then

$$u(t) = u_{ff}(t) + u_{fb}(t) = u_{ff}(t) + K(t)\left[x_{ff}(t) - x(t)\right]$$

Ex: Swing up + balance pendulum





turn off fb. for last parts.

Can choose different feedback laws

1. Time-dep. LQR what we should do...
2. Quasistationary LQR
3. Basic LQR fixed gains for "typical" x_0
4. Heuristic control: pick gains by heuristic method

Nonlinear Observers (set $u=0$ for simplicity)

$$\dot{x} = f(x), \quad y = h(x) \quad (p \text{ outputs})$$

Obs: $$\dot{\hat{x}} = f(\hat{x}) + L(y - \hat{y}), \quad \hat{y} = h(\hat{x})$$

notice that $e = x - \hat{x} \Rightarrow$
$$\begin{aligned}\dot{e} &= f(x) - f(\hat{x}) - L[h(x) - h(\hat{x})] \\ &\approx (\partial_x f)\, e - L(\partial_x h)\, e \\ &= (A - LC)\, e \quad \text{as before}\end{aligned}$$

Because we approx $f(x) - f(\hat{x}) \approx Ae$, this scheme works if systems begin (and stay) "close enough".

Model Predictive Control (MPC)

- Compute ff from $x(t)$ to $x(t+\tau)$ $\Rightarrow u_{ff}(t)$
- Apply $u_{ff}(t)$ for short time ("one step")

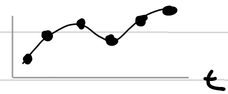
"Feedback by repeated ff"

- good for problems w/ constraints
- costly to compute $\Rightarrow$ good for "slow" problems
- widely used in "slow" industries (eg chemical plants)

Numerical Methods

Direct: $\min_{u(t)\in\mathcal{U}} \; J[u(t); x_0]$

Project $u(t)$ on finite set



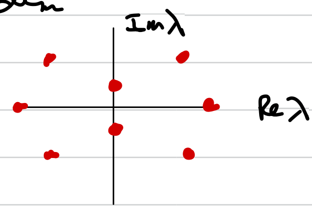
or: $u(t) = \sum_n a_n \varphi_n(t)$ basis funct.

Then solve directly (often non-convex)

Indirect: solve nec. eqs. (Euler-Lagrange, PMP)

- $n \to 2n + N$ constraints dim

Hamiltonian H real $\Rightarrow$



$\Rightarrow$ stiff eqs.

(bad for shooting methods)

$\to$ Solve by Newton's method

$\to$ can write Jacobian as band diag $\Rightarrow O(N)$

can even solve each stage in $\parallel$!