## · Linear time invariant convex optimal control

$\min_{x,u} \; J = \sum_{t=0}^{100} \ell(x(t), u(t))$ $\quad$ $x$: state $\quad$ $u$: action

s.t. $u(t) \in \mathcal{U} \quad x(t) \in \mathcal{X}$

$x(t+1) = A x(t) + B u(t)$ $\quad$ $x(t) \in \mathbb{R}^n \quad u(t) \in \mathbb{R}^m$

$x(0) = z$

variables: $x(0), x(1), \cdots \in \mathbb{R}^n$ $\quad$ can think $x(t)$ as variable and $x(t+1) = Ax(t) + Bu(t)$ as equality constraints

$u(0), u(1) \cdots \in \mathbb{R}^m$ $\quad$ or think $x(t+1) = Ax(t) + Bu(t)$ as recursion expression

problem data:

dynamics and input matrices $A \in \mathbb{R}^{n \times n} \quad B \in \mathbb{R}^{n \times m}$

convex stage cost function $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R} \quad \ell(0,0) = 0$

convex state set and convex input set $\mathcal{X}, \mathcal{U} \quad 0 \in \mathcal{X} \quad 0 \in \mathcal{U}$

Greedy control: $u(t) = \operatorname{argmin}_w \{ \ell(x(t), w) \mid w \in \mathcal{U}, \; Ax(t) + Bw \in \mathcal{X} \}$

ignores the cost in the future, except for $x(t+1) \in \mathcal{X}$

Greedy control typically works poorly

But it can work very well:

if $A$ has very small spectral radius / small norm

then $x(t+1)$ does not depend much on $x(t)$, the dynamic system has "strongly fading memory"

## · Solution via dynamic programming (Reinforcement learning !!!)

the optimal value of the convex optimization problem is a convex function of $z$

(Bellman) value function $V(z)$ is the optimal value of control problem as a function of initial state $z$

$V$ satisfies Bellman or dynamic programming equation

$$V(z) = \inf_w \{ \ell(z, w) + V(Az + Bw) \mid w \in \mathcal{U}, \; Az + Bw \in \mathcal{U} \}$$

$\ell(z,w)$: immediate cost $\quad$ $V(Az+Bw)$: minimum future cost

optimal $u$ given by

$$u^*(t) = \operatorname*{argmin}_{w \in \mathcal{U}, \; Ax(t)+Bw \in \mathcal{X}} \ell(x(t), w) + V(Ax(t) + Bw)$$

the optimal input is a function of state $u^*(t) = \phi(x(t))$ $\quad$ state feedback form

eg. Linear quadratic regulator

Special case of linear convex optimal control with

$U = \mathbb{R}^m \quad X \in \mathbb{R}^n$

$\ell(X(t), U(t)) = X(t)^T Q X(t) + U(t)^T R U(t) \qquad Q \geq 0 \quad R > 0$

Solved using dynamic programming

value function $V(z) = \min_{U,X} \sum_t \ell(X(t), U(t)) = z^T P z$

minimizing a quadratic function over a subset of variables gives a quadratic function (described by Schur complement)

P can be found by solving an algebric Riccati equation (ARE)

$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$

## · Finite horizon approximation

use finite horizon T, impose terminal constraint $X(T) = 0$

$\min_{X,U} \sum_{t=0}^{T-1} \ell(X(t), U(t))$

$U(t) \in \mathcal{U} \quad X(t) \in \mathcal{X}$

$X(t+1) = A X(t) + B U(t)$

$X(0) = z \quad X(T) = 0$

gives a suboptimal input of the original optimal input problem

## · Model predictive control (MPC)

at each time t, solve the (planning) problem

$\min_{X,U} \sum_{\tau=t}^{t+T-1} \ell(X(\tau), U(\tau))$

s.t. $U(\tau) \in \mathcal{U} \quad X(\tau) \in \mathcal{X}$

$X(\tau+1) = A X(\tau) + B U(\tau)$

$X(t+T) = 0$

do a complete planning exercise, use the first input, and replan

this gives a complicated state feedback control $U(t) = \phi_{mpc}(X(t))$

## · Variations on MPC

— add final state cost $\hat{V}(X(t+T))$ instead of insisting on $X(t+T) = 0$

if $\hat{V} = V$, MPC gives optimal input

— convert hard constraints to violation penalties

avoids problem of planning problem infeasibility

— Solve MPC every k steps ($k \geq 1$)

## · Explicit MPC

MPC with $l$ quadratic, $X$ and $U$ polyhedral

can show $\phi_{mpc}$ is piece-wise affine:

$$\phi_{mpc}(z) = K_j z + g_j \qquad z \in R_j$$

$R_1 \cdots R_N$ is polyhedral partition of $X$

solution of any QP is piece-wise affine in rhs of constraints

eg. $l_1$ regularized problem

$f(x) + \lambda \|x\|_1$ $\quad x^*$ is piece-wise affine in $\lambda$

can build a look-up table for $\lambda, x^*$

$\phi_{mpc}$ (ie. $K_j$ $g_j$ $R_j$) can be computed explicitly, off-line

## · MPC problem structure

mpc problem is high-structured

Hessian is block diagonal

equality constraint matrix is block banded

eg

$l(x(t), u(t))$

$x(t)$ is directly connected to $\begin{cases} u(t) \\ u(t-1) \\ x(t+1) \\ x(t-1) \end{cases}$

$x(t-1)$ $x(t)$ $x(t+1)$ $u(t)$ $u(t-1)$

block of variables $\Rightarrow$ banded

Use block-elimination to compute newton step

## · Supply chain management

n nodes (warehouses / buffers)

m unidirectional links between nodes, external world

$x_i(t)$ is amount of commodity at node $i$, in period $t$

$u_j(t)$ is amount of commodity transported along link $j$

incoming / outgoing node incidence matrices

$$A^{in/out}_{ij} = \begin{cases} 1 & \text{link } j \text{ enters/exits node } i \\ 0 & \text{otherwise} \end{cases}$$

dynamics $\quad x(t+1) = x(t) + A^{in} u(t) - A^{out} u(t)$

buffer limits $\quad 0 \le x_i(t) \le x_{max}$ $\quad$ (could allow $x_i(t) < 0$, to represent back order)

link capacity: $\quad 0 \le u_i(t) \le u_{max}$

$A^{out} u(t) \leq x(t)$ (can not ship out wheels not on hand)

objective: $\sum_{t=0}^{10} S(u(t)) + w(x(t))$

shipping cost + storage cost