Mo.
22/10/18

1 Introduction

1.1 Optimal control problems

Optimal control:

- Optimal decision making over time (over a time horizon)
- Optimisation problems with dynamic systems (differential/difference equations), e.g. constraints are differential equations

[OP] Optimisation problem (calculus)

$$\min_u f(u) \triangleq \max_u -f(u) \qquad (1.1)$$

$$\text{s.t.} \quad h_i(u) = 0 \qquad i = 1 \dots m$$
$$g_j(u) \leqslant 0 \qquad j = 1 \dots M$$

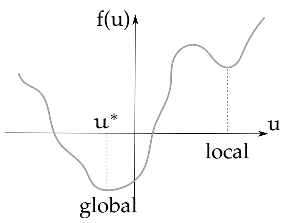
u	decision/optimisation variables (unknown)	$\mathbb{R}^n$
f	objective/cost function	$\mathbb{R}^n \to \mathbb{R}$
h_i	equality constraint $h = \begin{bmatrix} h_1 & \cdots h_m \end{bmatrix}^T$	$\mathbb{R}^n \to \mathbb{R}$
g_j	inequality constraint $g = \begin{bmatrix} g_1 & \cdots g_M \end{bmatrix}^T$	$\mathbb{R}^n \to \mathbb{R}$

Feasible/admissible set

$$FS = \left\{ u \in \mathbb{R}^n : \begin{array}{ll} h_i(u) & = 0, \quad i = 1 \dots m \\ g_j(u) & \leqslant 0, \quad j = 1 \dots M \end{array} \right\}$$

Solution: u^* is a (global) solution of

$$f^* \leqslant f(u), \; \forall \, u \in FS$$

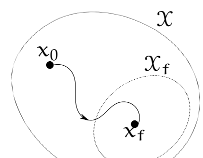


$$u = u\,(x, t) \quad \text{feedback}$$
$$u = u(.) \quad \text{all time instances}$$
$$u = u(t) \quad \text{at fixed time } t$$

[OP] Opt. control problem – continuous time

$$\min_{u(.),\, x(.)} \underbrace{\int_{t_0}^{t_f} f_0\,(t, x(t), u(t)) \;\, dt + \phi\,(t_f, x_f)}_{J} \qquad (1.2)$$

$$\begin{aligned} \text{s.t.} \quad \dot{x}(t) &= f\,(t, x(t), u(t)) \\ x(t) &\in \mathcal{X} \subseteq \mathbb{R}^n \\ u(t) &\in \mathcal{U} \subseteq \mathbb{R}^q \\ &\in C^1 \\ x(t_0) &= x_0 \in \mathcal{X}_0 \\ x(t_f) &= x_f \in \mathcal{X}_f \\ t &\in [t_0, t_f] \end{aligned}$$



J	cost/objective functional	
f_0	running/infinitesimal/incremental cost	
ϕ	terminal cost	
$\mathcal{X}$	state constraints	$\mathcal{X} \subseteq \mathbb{R}^n$
$\mathcal{X}_f$	terminal/target set	
$\mathcal{U}$	input constraints	$\mathcal{U} \subseteq \mathbb{R}^q$

For all admissible $x(.)$, $u(.)$:

$$J^* = J\,(x^*(.),\; u^*(.)) \leqslant J\,(x(.),\; u(.))$$
$$\Rightarrow u^*(.),\; x^*(.) \text{ is a solution of } J^*$$

[OP] Opt. control problem – discrete time

$$\min \sum_{k=0}^{N-1} \underbrace{f_0\,(k, x(k), u(k))}_{\text{stage cost}} + \phi\,(x(N)) \qquad (1.3)$$

$$\begin{aligned} \text{s.t.} \quad x(k+1) &= f\,(k, x(k), u(k)) \qquad k = 0, 1, \dots, N \\ x(0) &= x_0 \in \mathcal{X}_0 \\ x(N) &= x_N \in \mathcal{X}_f \end{aligned}$$

$$x^*(.) = \{x^*(1), \dots, x^*(N)\}$$
$$u^*(.) = \{u^*(0), \dots, u^*\,(N-1)\}$$

1.2 Topics/Tools

- Nonlinear programming approach (NLP)
- Dynamic programming (DP) – constraints are dynamic
- Receding horizon optimal control (MPC, MHE)
- Pontryagin maximum principle (PMP)

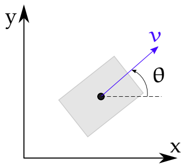
- Infinite vs. finite horizon
  $t_f = \infty$ vs. $t_f < \infty$
  $N = \infty$ vs. $N < \infty$
- Open loop vs. feedback
- Continuous time vs. discrete time
- Online (realtime) vs. offline computation

1.3 Examples

1.3.1 Shortest path problem

Robot unicycle model with constant velocity v

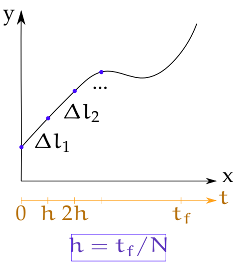
$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = u$$



Goal: Move robot along the shortest path ($A \to B$) subject to angle/velocity constraints etc. ($\therefore$ not a straight line).

Length of a curve:

$$\Delta l_1 = \sqrt{\Delta x^2 + \Delta y^2}$$
$$x(h) = x(0) + \dot{x}(0)h + \dots$$
$$\approx x(0) + \Delta x$$



$$\Delta x \approx \dot{x}(0)h$$
$$\Delta y \approx \dot{y}(0)h$$

$$\Rightarrow \Delta l_1 = h\sqrt{\dot{x}^2(0) + \dot{y}^2(0)}$$

$$\vdots$$

$$\Delta l_i = h\sqrt{\dot{x}^2((i-1)h) + \dot{y}^2((i-1)h)}$$

$$l = \sum_{i=1}^{N} \Delta l_i = \sum_{i=0}^{N-1} h\sqrt{\dot{x}^2(ih) + \dot{y}^2(ih)}$$

$$\text{For } h \to 0: \qquad l = \int_0^{t_f} \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}\,dt \quad := J(.)$$

Ex. 1.1 Shortest path problem

$$\min \int_0^{t_f} \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}\,dt$$

$$\begin{aligned}
\text{s.t.} \quad & u \in [-1,1] \\
& x(0) = x_0 \qquad x(t_f) = x_f \\
& y(0) = y_0 \qquad y(t_f) = y_f \\
& \theta(0) = \theta_0 \qquad \theta(t_f) = \theta_f \\
\text{d.v.} \quad & x(t),\ y(t),\ \theta(t),\ u(t),\ t_f
\end{aligned}$$

$$\begin{aligned}
J(.) &= \int_0^{t_f} v\sqrt{\cos^2\theta + \sin^2\theta}\,dt \\
&= v t_f \mathrel{\hat{=}} \text{time optimal control problem}
\end{aligned}$$

FR.
26/10/18

1.3.2 Time optimal control problem

t_f is the decision variable.

[OP] Time opt control problem formulation

$$\min_{t_f,\, u(.)} \int_0^{t_f} 1\,dt = t_f$$

$$\begin{aligned}
\text{s.t.} \quad & \dot{x},\ y,\ \theta = \dots \\
& u \in [-1,1] \\
\text{d.v.} \quad & t_f
\end{aligned}$$

Finite horizon, time-optimal, open-loop OC problem

1.3.3 Linear quadratic regulator problem (LQR)

Given: $\dot{x} = Ax + Bu$

Find: feedback $u^* = k^*(x,t)$ such that

- The closed loop solution $(x^*(t), u^*(t))$ minimises $J = x^T(t)Qx(t) + u^T(t)Ru(t)$ (infinite horizon, closed loop)
- The closed loop is asymptotically stable with respect to $x = 0$

[OP] LQR problem

$$\min \int_0^{\infty} x^T(t)Qx(t) + u^T(t)Ru(t)\,dt$$

$$\begin{aligned}
\text{s.t.} \quad & \dot{x} = Ax + Bu, \quad x \in \mathbb{R}^n, u \in \mathbb{R}^q \\
& x(0) = x_0
\end{aligned}$$

Q and R are positive definite matrices, i.e. their eigenvalues are positive

$$Q = Q^T > 0$$
$$R = R^T > 0$$

In 1-D: $x, u \in \mathbb{R}$
$Q \to q > 0 \quad R \to r > 0$

$$J = \int_0^{\infty} qx^2 + ru^2\,dt$$

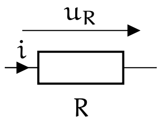
(a) Control energy

$$\int_0^{\infty} ru^2\,dt$$

e.g.
Input $u = i$

$$\Rightarrow \quad P = Ri^2 = Ru^2$$

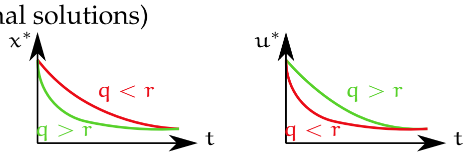
$$\Rightarrow \quad E = \int Ru^2 dt$$



(b) Penalisation of deviation from x_0

$$\int_0^\infty qx^2 \, dt$$

e.g. (both the green and red curves are optimal solutions)



$\uparrow q$	$\uparrow q, \downarrow r$
x gets more expensive.	x more expensive than u.

$\therefore$ opt solution minimises x more quickly.

$q = 0$ minimal control energy problem

$r = 0$ cheap control (energy costs nothing)

Ex. 1.2 Nonlinear QR

$$\min \int_0^\infty q(x) + u^T R(t) u \, dt$$

$$\text{s.t.} \quad \dot{x} = f(x) + Bu$$
$$u^* = k^*(x,t) = ?$$

Optimality, stability, robustness

1.3.4 State estimation problem

Given:

- System/Model (A, C)

$$x(k+1) = Ax(k)$$
$$y(k) = Cx(k)$$

- Measurements $y_0, y_1, \ldots, y_N$

Estimate: $x(0), x(N) = ?$

Idea Fit data to model in least squares sense.

Ex. 1.3 Least Squares

$$\min_{\hat{x}_k, v_k} \sum_{k=0}^{N-1} \|v_k\|^2 + \|w_k\|^2 + \underbrace{\|x_0 - \xi_0\|^2}_{\text{prior knowledge}}$$

$$\begin{aligned} \text{s.t.} \quad \hat{x}_{k+1} - A\hat{x}_k &= w_k \quad \text{process noise} \\ y_k - C\hat{x}_k &= v_k \quad \text{measurement noise} \\ k = 0, \ldots, N-1 & \end{aligned}$$

ξ_0: initial guess for x_0

How to deal with unbounded data? (e.g. stream of measurements; measurements arrive at each time step $k, \uparrow N$)

$\Rightarrow$ **recursive solution (DP)**

old solution is updated, e.g. in Kalman filter

OC automatically gives recursive solution, $\because$ the dynamics $\hat{x}_{k+1} - A\hat{x}_k$ are taken into account

1.4 Discrete-time systems

Continuous

$$\dot{x}(t) = f(x(t), u(t))$$
$$x(t) = x_0 + \int_0^\infty f(x(\tau), u(\tau)) \, d\tau$$

1st order Euler approximation

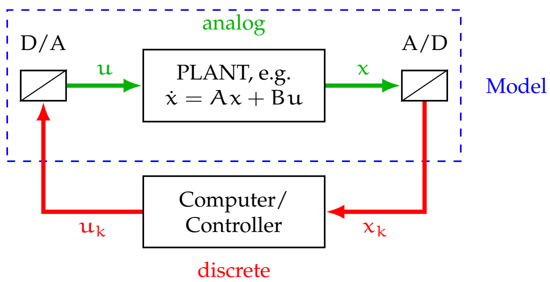
$$x(\Delta t) \approx x_0 + f(x_0, u_0) \underbrace{\Delta t}_{\ll}$$

$$x((k+1)\Delta t) \approx x(k\Delta t) + f(x(k\Delta t), u(k\Delta t)) \Delta t$$
$$x_{k+1} \approx x_k + f(x_k, u_k) \Delta t$$
$$:= \tilde{f}(x_k, u_k)$$

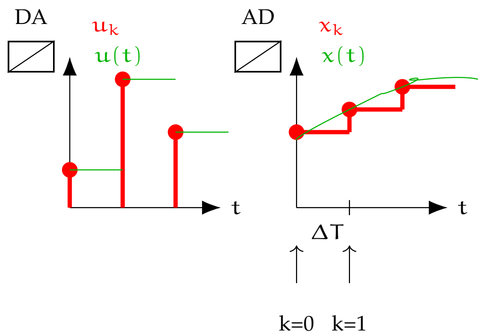
Algorithms e.g. gradient descent

$$x_{k+1} = x_k + s_k \nabla f(x_k)$$
$$\min_{x \in \mathbb{R}^n} f(x)$$

Sampled-data / digital control systems



Model with discrete input and output



ΔT : sampling time

$$u(t) = u_k \text{ for } t \in [k\Delta T, (k+1)\,\Delta T]$$

$$\Rightarrow x\left((k+1)\,\Delta T\right) = e^{A\Delta T} x\,(k\Delta T) + \int_{k\Delta T}^{(k+1)\Delta T} e^{A[(k+1)\Delta T - \tau]} b u_k \, d\tau$$

$$x(t) = e^{At} x_0 + \int_0^t e^{A(k-\tau)} b u\,(\tau)\, d\tau$$

$$x_{k+1} = \underbrace{e^{A\Delta T}}_{\tilde{A}} x_k + u_k \underbrace{\int_0^{\Delta T} e^{A(\Delta T - \tau)} b \, d\tau}_{\tilde{b}}$$

$$\boxed{\begin{gathered} x_{k+1} = \tilde{A} x_k + \tilde{b} u_k \\ \text{(Model)} \end{gathered}}$$

Exact discretisation for LTI-system with piecewise constant inputs.
'Exact': $\underbrace{x_k}_{\text{discrete}} = \underbrace{x(t)}_{\text{analog}}$

Summary of some basic facts

- **Solution**
  - Continuous

$$\dot{x} = f\,(x, u)$$
$$x(t) = x_0 + \int_0^t f\,(x\,(\tau)\,, u\,(\tau))\, d\tau$$

$$\dot{x} = Ax$$
$$x(t) = e^{At} x_0$$

  - Discrete

$$x_{k+1} = f\,(x_k, u_k)$$
$$x(k) = f \circ f\,(f\,(f\,(x_0, u_0)\,, u_1)\,, u_2)$$

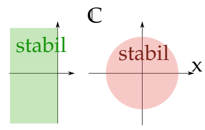
$$x_{k+1} = f\,(x_k)$$
$$x(k) = \underbrace{(f \circ \cdots \circ f)}_{\text{k-times}}(x_0)$$

$$x_{k+1} = A x_k$$
$$x(k) = A^k x_0$$

- Equilibrium points (fix points)

$$\text{Ⓒ} \quad \dot{x} = f(x), \quad f\,(x_E) = 0$$
$$\text{Ⓓ} \quad x_{k+1} = f(x_k), \quad f\,(x_E) = x_E$$

- Stability



Ⓒ $\dot{x} = Ax$ asym. stable $(x_E = 0)$
$\Leftrightarrow \forall i : \mathrm{Re}\{\lambda_i\,(A)\} < 0$

Ⓓ $x_{k+1} = Ax_k$ asym. stable
$\Leftrightarrow \forall i : \mathrm{Re}|\lambda_i\,(A)| < 1$

$A = T\Lambda T^{-1}$, then
$$x_k = A^k x_0$$
$$= T\Lambda T^{-1} x_k$$

Lyapunov theory

  - Continuous Ⓒ

$$\dot{x} = f(x), \quad f(0) = 0, \quad x \in \mathbb{R}^n$$
$$\text{let } V : \mathbb{R}^n \to \mathbb{R}$$
$$V \in C^1$$

If

* $V(0) = 0$ in neighbourhood $V(x) > 0$ of $x = 0$ $(\forall x \neq 0)$
* $\dot{V}(x) = \nabla V^T(x) f(x) < 0$ in a neighbourhood of $x = 0$ $(\forall x \neq 0)$
* $(V(x) \to \infty$ wherever $\|x\| \to \infty)$,

then $x_E = 0$ is (globally) asym. stable.

For $\dot{x} = Ax$

$$V(x) = x^T P x, \quad P = P^T > 0$$
$$\dot{V}(x) = x^T PAx + x^T A P x < 0$$
$$A \text{ Hurwitz} \Leftrightarrow \exists\, P > 0$$
$$\text{s.t. } PA + A^T P < 0$$

  - Discrete Ⓓ

$$x_{k+1} = f(x_k), \quad f(0) = 0$$
$$V : \mathbb{R}^n \to \mathbb{R}, \quad V \in C^0$$

If

* $V(0) = 0$ in neighbourhood $V(x) > 0$ of $x = 0$

* $\Delta V(x) := \underbrace{V\left(x^{k+1}\right)}_{f(x)} - V(x^k) < 0$ in a neighbourhood of $x = 0$

then $x_E = 0$ is asym. stable.

For $x_{k+1} = Ax_k$

$$\begin{aligned} V(x) &= x^T P x, \quad P = P^T > 0 \\ \Delta V(x) &= V\left(x_{k+1}\right) - V(x_k) \\ &= x_k^T A^T P A x_k - x_k^T P x_k < 0 \end{aligned}$$

A Schur $\Leftrightarrow \exists\, P > 0$

s.t. $A^T P A - P < 0$

Mo. 29/10/18

2 Nonlinear programming approach

2.1 Nonlinear programming and optimal control

Example finite horizon, discrete time, open loop OC problems

$$\underbrace{\min \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k,}_{\text{Cost } f(x)} \quad x_k \in \mathbb{R}^n$$

$$\begin{aligned} \text{s.t.}\ x_{k+1} &= Ax_k + Bu_k, \quad k = 0, \dots, (N-1) \\ x_0 &= \bar{x} \\ x_N &= 0 \end{aligned}$$

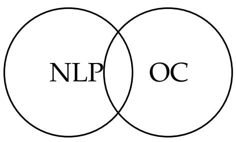
$$\text{d.v.} \qquad x = \begin{bmatrix} x_0 & u_0 & x_1 & \dots & x_{N-1} & u_{N-1} & x_N \end{bmatrix}^T$$

$$f(x) = x^T H x \qquad \Rightarrow h_i(x) = c_i^T x - d_i$$

$n(N+2)$ equality constraints

$$\begin{aligned} x_1 - Ax_0 - Bu_0 &= 0 \qquad \}n \\ x_2 - Ax_1 - Bu_1 &= 0 \qquad \}n \\ &\vdots \\ x_0 - \bar{x} &= 0 \qquad \}n \\ x_N &= 0 \qquad \}n \end{aligned}$$

$$\Rightarrow \boxed{\begin{aligned} &\min f(x) \\ &\text{s.t. } h(x) = 0 \\ &\quad g(x) \leqslant 0 \end{aligned}}$$



Remark NLP theory (algo) is a basis for OC theory (algo)

2.2 Problem setup

[OP] NLP

$$\begin{aligned} &\min\ f(x) \qquad (2.1) \\ &h(x) = 0 \\ &g(x) \leqslant 0 \end{aligned}$$

x	decision/opt variables	$\mathbb{R}^n$
f	objective/cost function	$\mathbb{R}^n \to \mathbb{R}$
h	equality constraints	$\mathbb{R}^n \to \mathbb{R}^m$
g	inequality constraints	$\mathbb{R}^n \to \mathbb{R}^M$
	$f, g, h \in C^{1/2} \leftarrow$ NLP	

Feasible set $FS = \{x : h(x) = 0, g(x) \leqslant 0\}$
all decision variables fulfilling the constraints

Optimality

- x^* is a local minimum of f, if there exists a neighbourhood $\mathcal{U}(x^*)$ such that $\forall x \in \mathcal{U}(x^*) : f(x^*) \leqslant f(x)$
  x^* is a global minimum of f, if $\forall x \in \mathbb{R}^n : f(x^*) \leqslant f(x)$
- x^* is a local minimum/solution of Problem (2.1), if $\exists \mathcal{U}(x^*)$ s.t. $\forall x \in \mathcal{U}(x^*) \cap FS : f(x^*) \leqslant f(x)$
  x^* is a minimum/solution of (2.1), if $\forall x \in FS : f(x^*) \leqslant f(x)$

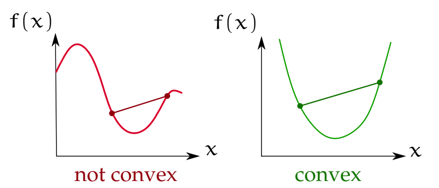
Convexity Problem (2.1) is convex if both

- $f(x)$ is a convex function

DEF: Convex functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, when

$$\forall x_1, x_2 \in \mathbb{R}^n, \forall \lambda \in (0,1) :$$
$$f(\lambda x_1 + (1-\lambda) x_2) \leqslant f(x_1) \lambda + (1-\lambda) f(x_2)$$



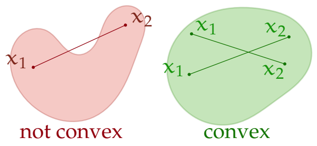
line segment always above the graph

- FS is a convex set
  ↳ h is an affine function
  ↳ g is an affine function

DEF: Convex sets

A set $C \subseteq \mathbb{R}^n$ is convex, if

$$\forall x_1, x_2 \in \mathbb{R}^n, \forall \lambda \in (0,1) :$$
$$\lambda x_1 + (1-\lambda) x_2 \in C$$



Def: Affinity

A function is affine, if

$$f(X) = c^T x + d, \quad f : \mathbb{R}^n \to \mathbb{R}$$

f is convex, $f \in C^2 \Leftrightarrow \underbrace{\nabla^2 f(x)}_{\text{Hessian}} \geqslant 0 \;\forall\, x \in \mathbb{R}$

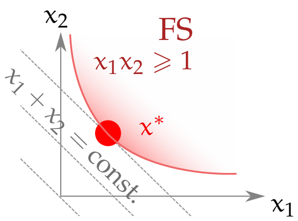
$$\nabla f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix} = \frac{\partial}{\partial x} f^T(x)$$

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(x) & \cdots & \frac{\partial^2}{\partial x_1 x_n} f(x) \\ & \ddots & \\ & & \frac{\partial^2}{\partial x_n^2} f(x) \end{bmatrix}$$

Ex. 2.4

$$\min x_1 + x_2$$

$$\text{s.t.} \quad x_2 \geqslant 0$$
$$x_1 x_2 \geqslant 1$$

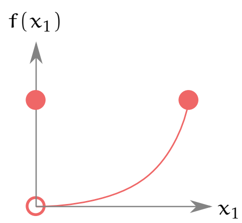


Ex. 2.5

$$\min f(x_1) = \begin{cases} x_1^2 & x_1 > 0 \\ 1 & x_1 = 0 \end{cases}$$

s.t. $x_1 \leqslant 1$
$x_1 \geqslant 0$

Minimum does not exist; $\inf f = 0$

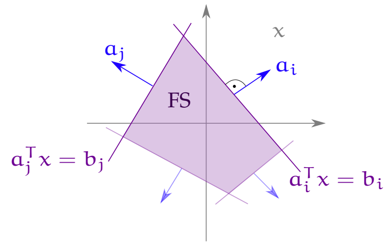


Remark Suppose f is continuous and FS is compact, then $\min_{x \in FS} f(x)$ exists.

Consider the following problem:

[OP] Quadratic Program (QP)

$$\min \frac{1}{2} x^T Q x + c^T x$$
$$\text{s.t.} \quad a_i^T x \leqslant b_i \quad i = 1 \dots M$$



Remark $h(x) = 0 \triangleq h(x) \geqslant 0$ and $h(x) \leqslant 0$

Convexity

$$\nabla f(x) = Qx + c$$
$$\nabla^2 f(x) = Q$$

$\therefore$ the problem is convex if $Q \geqslant 0$.

QP (Quadratic Program)

$Q \ngeqslant 0$ general (worst) case, problem difficult to solve
$Q = 0$ $\Rightarrow$ linear program (LP)

2.3 Unconstrained optimisation

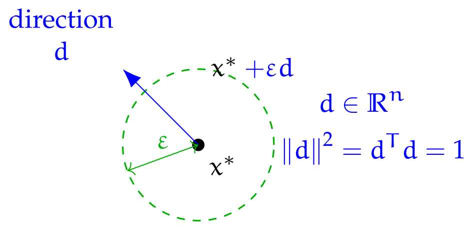
[OP] Unconstrained optimisation

$$\min_{x \in \mathbb{R}^n} f(x), \qquad f(x) \in C^{1/2}$$

a) Optimality condition
b) Algorithms
  i) Gradient descent
  ii) Newton method

2.3.1 Optimality condition

Idea 'Explore neighbourhood'.
If $\exists\, \mathcal{U}(x^*) :, \forall\, x \in \mathcal{U}(x^*)$:
$f(x^*) \leqslant f(x) \Rightarrow x^*$ is a local minimum.



Def: Optimality Cond. – Unconstrained

If $\exists\, \varepsilon_0 > 0 : \forall\, \varepsilon \leqslant \varepsilon_0, d \in \mathbb{R}^n, \varepsilon > 0, \|d\| = 1,$
$\Rightarrow f(x^*) \leqslant f(x^* + \varepsilon d)$

Taylor approximation around $\varepsilon = 0$

$$\cancel{f(x^*)} \leqslant \cancel{f(x^*)} + \varepsilon \nabla f^T(x^*) d + \frac{\varepsilon^2}{2} \underbrace{d^T \nabla^2 f(\xi) d}_{\text{upper bound } \alpha}$$

$$\xi \in [x^*, x^* + \varepsilon d]$$

$$0 \leqslant \varepsilon \nabla f^T(x^*) d + \frac{\varepsilon^2}{2} \alpha$$

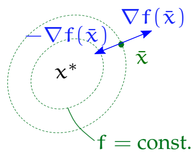
$$\Rightarrow \nabla f(x^*) = 0$$

Why? Suppose $\nabla f(x^*) \neq 0$,

choose $d = \dfrac{-\nabla f(x^*)}{\|\nabla f(x^*)\|}$

$$\forall\, \varepsilon \gg 0 \qquad 0 \leqslant -\varepsilon \underbrace{\|\nabla f(x^*)\|}_{\beta} + \alpha\varepsilon^2/2$$

$$0 \leqslant -\underbrace{\|\nabla f(x^*)\|}_{\beta} + \alpha\varepsilon/2$$

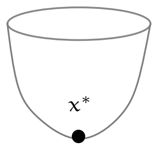


THEOREM: 1st & 2nd order necessary opt. conds

Let $f : \mathbb{R}^n \to \mathbb{R}, f \in C^2$, x^* be a local minimum. Then,

$$\nabla f(x^*) = 0 \qquad (2.2)$$

$$\nabla^2 f(x^*) \geqslant 0 \qquad (2.3)$$



Assume f is convex and $f \in C^1$,

$$\nabla f(x^*) = 0 \Leftrightarrow x^* \text{ is min.}$$

$\therefore$ 'Every local minimum of a convex function is also a global minimum.'

Remark

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon_0} f(x^* + \varepsilon d) = \nabla f^T(x^*) d \geqslant 0 \quad \forall\, d$$

$$\Updownarrow$$

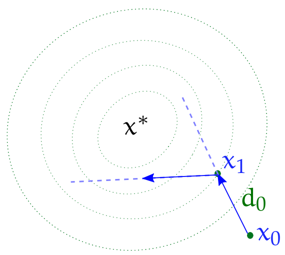
$$\nabla f(x^*) d = 0 \quad \forall\, d, \|d\| = 1$$

$$\Rightarrow \quad f(x^*) = 0$$

Fr. 02/11/18 no class

Mo. 05/11/18

2.3.2 Algorithms



Basic idea of classical optimisation algorithms

1. Initialise algorithm, choose x_0
2. Determine a search direction d_k
3. Line search, i.e. 1D optimisation problem

$$\alpha_k^* \approx \arg\min_{\alpha} f(x_k + \alpha d_k)$$

   approximate (fast) solution. $\alpha \in \mathbb{R}, \alpha \geqslant 0$

4. Update

$$x_{k+1} = x_k + \alpha_k^* d_k$$

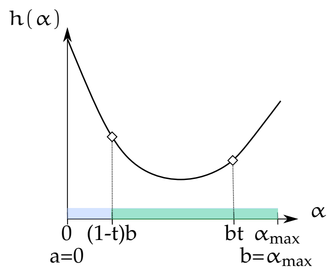
   Repeat from 2.
   Iterate until $\nabla f(x_k)$ small.

Line search (step-size rule)

Step-size rule: $\alpha_k = \dfrac{\alpha_{max}}{k+1}, \ldots$

Line search example: Golden sectioning

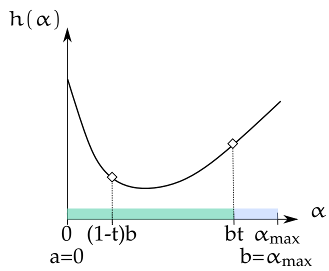
$$h(\alpha) = f(x_k + \alpha d_k), \quad t = 0.618$$



$$\text{If } h((1-t)b) \geqslant h(bt)$$

$$\text{then } a_{new} = (1-t)b$$

$$b_{new} = b$$



$$\text{If } h((1-t)b) < h(bt)$$

$$\text{then } a_{new} = a = 0$$

$$b_{new} = bt$$

Repeat the whole procedure until the interval $[a, b]$ small.

Search direction

- gradient descent direction
- Newton direction

(i) **Gradient descent algorithm**

$$d_k = -\nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Equilibrium points (fix points):

$$x_{k+1} \stackrel{!}{=} x_k \Leftrightarrow \nabla f(x) = 0 \qquad (\alpha_k > 0)$$

Stability

$$\begin{aligned}\Delta V(x_k) &= V(x_{k+1}) - V(x_k)\\ &= f(x_{k+1}) - f(x_k)\\ &= f\left(x_k - \alpha \nabla f(x_k)\right) - f(x_k), \quad \alpha \text{ const.}\\ &\approx f(x_k) + \nabla f^T(x_k)\left(-\alpha \nabla f(x_k)\right)\\ &\quad + \frac{\alpha^2}{2} \nabla f^T(x_k) \nabla^2 f(\xi) \nabla f(x_k) - f(x_k)\\ &\quad\ \xi \in [x_k, x_{k+1}]\\ &< 0 \text{ for } \alpha \ll\end{aligned}$$

(ii) **Newton method** Assumptions

A1. $f(x) \approx$ quadr. function

$$\begin{aligned}\Rightarrow h(\alpha d) &= f(x + \alpha d)\\ &= f(x) + \nabla f^T(x) d\alpha\\ &\quad + \frac{\alpha^2}{2} d^T \underbrace{\nabla^2 f(x)}_{\text{assume } >0} d + \dots\end{aligned}$$

Higher order terms ignored, ≈ 0

A2. $f(x)$ convex and quadratic
$\nabla^2 f(x)$ pos. def $\Rightarrow$ invertible

$$\min_d h(d)$$

h convex, $\therefore$ d^* is min. $\Rightarrow \nabla h(d^*) = 0$

$$\nabla h(d^*) = \nabla f(x) + \nabla^2 f(x) d^* \stackrel{!}{=} 0$$

$$d^* = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$$

$x^* = x + d^*$ minimises f

$$d_k = -\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_k \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

Hessian: curvature

Remarks Alternative derivation
$\nabla f(x_k) \neq 0$, choose d_k such that $\nabla f(x_k + d_k) \stackrel{!}{=} 0$
$\rightarrow$ Taylor linearisation up to 1st order

$$\nabla f(x_k) + \nabla^2 f(x_k) d_k = 0$$

$$d_k = -\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

Comparison

Gradient descent	Newton method
simple (1st order algo)	more complex (2nd order, matrix inversion necessary – expensive)
slow e.g. if gradient small	fast

2.4 Constrained optimisation

[OP] Constrained optimisation

$$\begin{aligned}\min\ & f(x) \\ & h(x) = 0 \\ & g(x) \leqslant 0\end{aligned} \tag{2.4}$$

x	decision/opt variables	$\mathbb{R}^n$
f	objective/cost function	$\mathbb{R}^n \to \mathbb{R}$
h	equality constraints	$\mathbb{R}^n \to \mathbb{R}^m$
g	inequality constraints	$\mathbb{R}^n \to \mathbb{R}^M$
	$f, g, h \in C^2$	

a) Optimality conditions
   i) Equality constraints
   ii) Inequality constraints
b) Lagrange multipliers, Lagrange function
c) Algorithms

2.4.1 Optimality conditions

EQUALITY CONSTRAINTS $h(x) = 0$

Idea 'Explore neighbourhood' (differentials)

$FS = M = \{x : h(x) = 0\}$ surface (manifold)

Fr. 09/11/18

- Equality constrains
- Inequality constraints

Mo. 12/11/18

2.5 NLP approach to optimal control

3 Dynamic programming

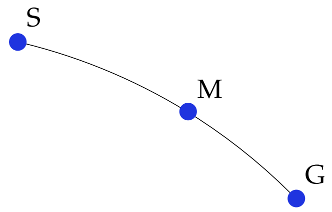
NLP	DP
finite horizon	(in)finite horizon
discrete time	discrete/continuous
open loop	feedback

3.1 Principle of optimality

In NLP: principle of 'exploring the neighbourhood'

Example
Task: Find optimal route from Stuttgart to Graz.
Optimal: fastest, shortest, cheapest, ...

1. Suppose the optimal route passes through Munich (M)



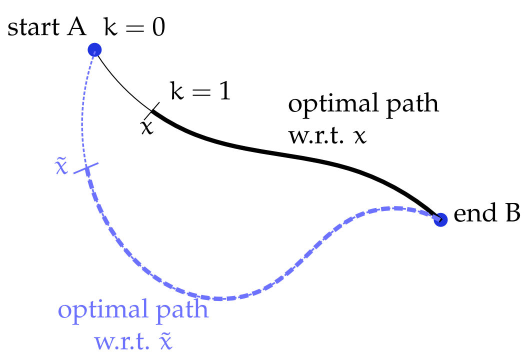
Solution:
- split task into 2: $S \to M$, $M \to G$
- recursive solution

2. Find optimal route $M \to G$ given the/an optimal route $S \overset{M}{\to} G$

Solution: "endpiece of optimal routes/trajectories are optimal" $\to$ **principle of optimality**
- endpiece of $S \overset{M}{\to} G$ is $M \to G$
- holds in general, but not always (splitting of problems)

DEF: Principle of optimality

"An optimal policy has the property that whatever the initial state and optimal first decision may be, the remaining decisions constitute an optimal policy w.r.t. the state resulting from the first decision."
– (Bellman, 1957)



FR.
16/11/18

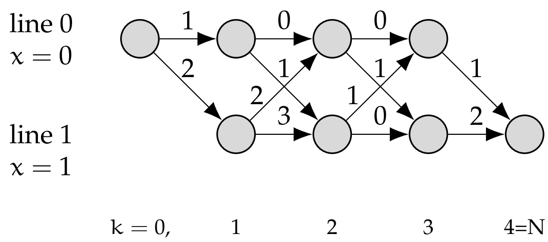
Assumption: cost sums up

$$
\begin{aligned}
J^*(A \to X) &= \alpha \\
J^*(X \to B) &= \beta \\
\Rightarrow J^*(A \to B) &= \alpha + \beta
\end{aligned}
$$

Proof by contradiction

$$
\begin{aligned}
J(A \to B) &< J^*(A \to B) \\
\alpha + \beta' &< \alpha + \beta
\end{aligned}
$$

Example two production lines
Cost: time or money



- Goal: find optimal ('shortest') path from start to end point.
- Solution:
  - Brute force
  - Solve backwards in time

Solution 1 brute force
List all possible paths and pick the best one.

- path = O XXX 1, $x \in \{0, 1\}$
  number of paths = $2^3 = 2^{N-1}$

- If M production lines:
  number of paths = M^{N-1} :(

- Complexity
  $(M^{N-1})(N-1) \approx \mathcal{O}(NM^{N-1})$

Solution 2 solving backwards
(Optimal) cost-to-go (to endpoint): $J(x, k)$

$$
\begin{aligned}
k = 3 \qquad J(0,3) &= 1 \\
J(1,3) &= 2 \\
k = 2 \qquad J(0,2) &= \min\{0 + J(0,3), 1 + J(1,3)\} \\
&= \min\{0+1, 1+2\} = 1 \\
J(1,2) &= \min\{1 + J(0,3), 0 + J(1,3)\} \\
&= \min\{1+1, 0+2\} = 2 \\
k = 1 \qquad J(0,1) &= \min\{0 + J(0,2), 1 + J(1,2)\} \\
&= \min\{0+1, 1+2\} = 1 \\
J(1,1) &= \min\{2 + J(0,2), 3 + J(1,2)\} \\
&= \min\{2+1, 3+2\} = 3 \\
k = 0 \qquad J(0,1) &= \min\{1 + J(0,1), 2 + J(1,1)\} \\
&= \min\{1+1, 2+3\} = 2
\end{aligned}
$$

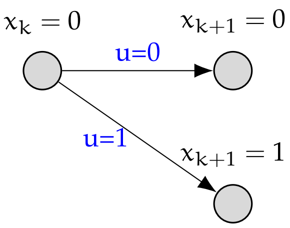
Complexity (count no. of +, -, min, max)

- $N-1$ steps (backwards in stime)
- for each production line: $2 \cdot 2\,(M-2)$
- for M production lines: M^2

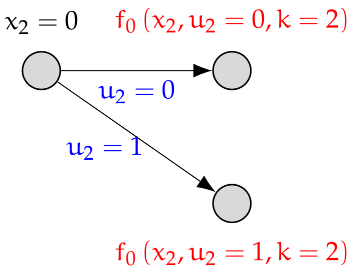
$\Sigma : \mathcal{O}\left(M^2 N\right)$

State space dynamics

$$x_k = \{0,1\}$$
$$x_{k+1} = x_k + u_k$$



$$u_k = \begin{cases} \{0,1\} & x_k = 0 \quad k = 1 \ldots N-2 \\ \{0,-1\} & x_k = 1 \end{cases}$$
$$u_0 \in \{0,1\}$$
$$u_{N-1} = \begin{cases} -1 & x_{N-1} = 0 \\ 0 & x_{N-1} = 1 \end{cases}$$



Cost (stage cost f_0)

$$\sum_{k=0}^{N-1} f_0\left(x_k, u_k, k\right)$$

s.t. $x_{k+1} = x_k + u_k$

$$u_k \in \mathcal{U}\left(x_k\right) = \begin{cases} \{0,1\} & x_k = 0 \quad k = 1 \ldots N-2 \\ \{0,-1\} & x_k = 1 \end{cases}$$

$$J\left(k, x_k\right) = \min_{u_k} \left\{ f_0\left(x_k, u_k, k\right) + J\left(k+1, x_{k+1}\right) \right\}$$

3.2 Discrete-time, finite-dimension DP

$$\min \sum_{k=k_0}^{N-1} f_0\left(k, x_k, u_k\right) + \phi\left(N, x_N\right) \tag{3.1}$$

Mo. 19/11/18 exercise Fr. 23//11/18

s.t. $x_{k+1} = f\left(k, x_k, u_k\right)$

$u_k \in \mathcal{U}\left(k, x_k\right)$

DEF: (Optimal) cost to go

$$J^*\left(k, x\right) = \min_{u_j \in \mathcal{U}\left(j, x_j\right)} \sum_{j=k}^{N-1} f_0\left(j, x_j, u_j\right) + \phi\left(N, x_N\right) \tag{3.2}$$

"optimal cost of endpiece"

DEF: Optimal cost

$$J^*\left(k_0, x\right) = V\left(x\right) \tag{3.3}$$

"optimal value starting at point x"
optimal value function

Theorem Backward DP recursion
Let:

$$\bar{J}\left(N, x\right) = \phi\left(N, x\right) \quad \forall\, x \in \mathbb{R}^n \tag{3.4}$$
$$\bar{J}\left(k, x\right) = \min_{u \in \mathcal{U}\left(k, x\right)} \left\{ f_0\left(k, x, u\right) + \bar{J}\left(k+1, f\left(x, u, k\right)\right) \right\} \tag{3.5}$$

"Going backwards in time"
$\because \bar{J}\left(k, x\right) = f\left(\ldots k+1 \ldots\right)$

$$\bar{u}\left(k, x\right) = \arg\min \left\{ f_0\left(k, x, u\right) + \bar{J}\left(k+1, f\left(x, u, k\right)\right) \right\}$$

Then

$$J^*\left(k, x\right) = \bar{J}\left(k, x\right) \tag{3.6}$$
$$u^*\left(k, x\right) = \bar{u}\left(k, x\right) \tag{3.7}$$

Proof By induction

$$J^*\left(N, x\right) = \phi\left(N, x\right) = \bar{J}\left(N, x\right) \quad \text{by definition}$$

$$J^*\left(k+1, x\right) = \bar{J}\left(k+1, x\right)$$
$$\Rightarrow J^*\left(k, x\right) = \bar{J}\left(k, x\right)$$
$$= \min_{u_j} \left\{ \sum_{j=k}^{N-1} f_0\left(j, x_j, u_j\right) + \phi\left(N, x_N\right) \right\}$$
$$= \min_{u_{k \ldots N-1}} \left\{ f_0\left(k, x_k, u_k\right) + \min_{u} \left\{ \sum_{j=k+1}^{N-1} f_0\left(j, x_j, u_j\right) + \phi\left(N, x_N\right) \right\} \right\}$$
$$= \min_{u_k \in \mathcal{U}\left(k, x_k\right)} \left\{ f_0\left(k, x_k, u_k\right) + \bar{J}\left(k+1, x_{k+1}\right) \right\}$$
$$= \bar{J}\left(k, x_k\right)$$

DP $J^*(x, k_0) = V(x)$ value function

$$\min \sum_{k=0}^{N-1} f_0(x_k, u_k, k) + \phi(x_N, N)$$

$$\begin{aligned} & x_k \in \mathbb{R}^n \\ & u_k \in \mathbb{R}^q \\ \text{s.t.} \quad & x_{k+1} = f(k, x_k, u_k), \quad k = k_0 \dots (N-1) \\ & u_k \in \mathcal{U}(x_k, k) \subseteq \mathbb{R}^q \\ & x_0 = x \end{aligned}$$

Cost-to-go

$$\bar{J}(x_k, k) = \min \left\{ \sum_{j=k}^{N-1} f_0(x_j, u_j, j) + \phi(x_N, N) \right\}$$

Backward DP recursion If $\bar{J} = \bar{J}(x, k)$ such that

$$\bar{J}(x, N) = \phi(x, N) \quad \forall x \tag{3.8}$$

$$\bar{J}(x_k, k) = \min_{u_k \in \mathcal{U}(k, x_k)} \{f_0(x_k, u_k, k) + \bar{J}(f(k, x_k, u_k), k+1)\} \tag{3.9}$$

$$\bar{u}(x_k, k) = \arg\min \{f_0(x_k, u_k, k) + \bar{J}(f(k, x_k, u_k), k+1)\} \tag{3.10}$$

then

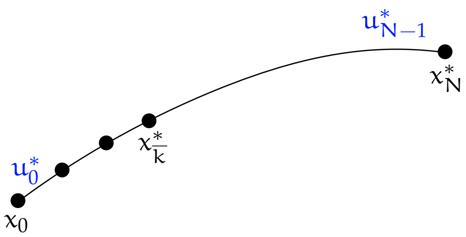
$$\bar{J}(k, x) = J^*(k, x) \,\forall\, X, k = k_0 \dots N-1 \tag{3.11}$$

$$\bar{u}(x, k) = \text{optimal feedback} \tag{3.12}$$

- (+) quite general, feedback, constructive (algorithm)
- (-) computational complexity (solve a NLP for each $x \in \mathbb{R}^n$)
  $\rightarrow$ discretisation $\rightarrow$ curse of dimensionality
  no. grid points in state space = N, dim = n
  N^n points :(
- between analytic soln (LQR) — brute force discretisation:
  approximate DP lat approx. $\bar{J}, \bar{u}, f$ by some basic function

PoO $\triangleq$ DP recursion ($k_0 = 0$ without loss of generality)

$$J^*(0, x_0) = \sum_{k=0}^{N-1} f_0(x_k^*, u_k^*, k) + \phi(x_N^*, N)$$



Discretisation algo (handout)
Idea: discretise state space and input space
$\rightarrow$ problem on a graph
(no. of states is finite, production line problem)

LQR – analytic solution

$$\min \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T S_N x_N,$$

$$R > 0, S_N \geqslant 0$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k = f(x_k, u_k)$$

DP: $(k = N)$
Step 0:

$$(\bar{J}(x, k) =: J(x, k))$$
$$\bar{J}(x_N, N) = x_N^T S_N x_N$$

Step 1: $(k = N-1)$

$$\begin{aligned} & J(x_{N-1}, N-1) \\ &= \min_u \{x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \\ &\quad + J(A x_{N-1} + B u_{N-1}, N)\} \\ &= \min_u \{x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \\ &\quad + (A x_{N-1} + B u_{N-1})^T S_N (A x_{N-1} + B u_{N-1})\} \\ &= \min_{u(x)} \{x_{N-1}^T \left(Q + A^T S_N A\right) x_{N-1} \\ &\quad + u_{N-1}^T \underbrace{\left(R + B^T S_N B\right)}_{>0} u_{N-1} \\ &\quad + x_{N-1}^T A^T S_N B u_{N-1} + u_{N-1}^T B^T S_N A x_{N-1}\} \end{aligned}$$

$$\frac{\partial}{\partial u} \{\} = 2\left(R + B^T S_N B\right) u + 2 B^T S_N A x \stackrel{!}{=} 0$$

$$u^*(x) = -\left(R + R^T S_N B\right)^{-1} B^T S_N A x$$

$$u^*(x_{N-1}) = -K_{N-1} x_{N-1}$$

$$\begin{aligned} & J(x, N-1) \\ &= x^T \Big(Q + K_{N-1}^T R K_{N-1} \\ &\quad + (A - B K_{N-1})^T S_N (A - B K_{N-1})\Big) x \\ &= x^T S_{N-1} x \end{aligned}$$

Because $J(x, k)$ is $\forall\, k$ always quadratic and $\bar{u}(x)$ can be computed analytically, a recursive (efficient/analytic) solution is feasible.

Ricatti equation, discrete time:

$$S_{i-1} = Q + K_{i-1}^T R K_{i-1} + (A - B K_i)^T S_i (A - B K_i) \tag{3.13}$$

$$u^*(i-1) = -K_{i-1} x_{i-1} = -\left(R + B^T S_i B\right)^{-1} B^T S_i A x_{i-1} \tag{3.14}$$

$$J(i-1, x_{i-1}) = x_{i-1}^T S_{i-1} x_{i-1} \tag{3.15}$$

Ex. 3.6 Forward DP recursion

$$\min \sum_{k=0}^{N-1} f_0\left(x_{k+1}, u_k\right) + \underbrace{\phi\left(x_0\right)}_{\text{note}^1}$$

$$\text{s.t.} \quad x_{k+1} = f\left(x_k, u_k\right)$$

$$J_a\left(k+1, x_{k+1}\right) \tag{3.16}$$

$$= \min_u \left\{ f_0\left(x_{k+1}, u_k\right) + J_a\left(k, x_k\right) \right\} \tag{3.17}$$

J_a arrival cost (summarises "past cost")

$J_a(0, x)$ optimal cost

[1] Kalman filter / estimation: $\phi\left(x_0\right) = \|x_0 - \xi\|^2$, where ξ is the initial guess/estimate of state. $f_0 = \|y_k - c x_k\|^2 + \ldots$

Why correct? time transformation

$$\bar{k} = N - k \quad k = 0 \ldots N-1$$
$$\bar{k} = 1 \ldots N$$

$$x_k = \xi_{\bar{k}} = \xi_{N-k}$$
$$\left(x_0 = \xi_N, x_1 = \xi_{N-1}, \ldots x_N = \xi_0\right)$$
$$u_k = \nu_{\bar{k}-1} = \nu_{N-k-1}$$
$$\left(u_0 = \nu_{N-1}, u_1 = \nu_{N-2}, \ldots u_{N-1} = \nu_0\right)$$

$$\sum_{k=0}^{N-1} f_0\left(x_{k+1}, u_k\right) + \phi\left(x_0\right)$$
$$= \sum_{\bar{k}=1}^{N} f_0\left(\xi_{\bar{k}-1}, \nu_{\bar{k}-1}\right) + \phi\left(\xi_N\right)$$
$$\text{where } \xi_{\bar{k}-1} = f\left(\xi_{\bar{k}}, \nu_{\bar{k}}\right)$$
$$\xi_0 = x_N = f\left(x_{N-1}, u_{N-1}\right)$$
$$= \sum_{\tilde{k}=0}^{N-1} f_0\left(\xi_{\tilde{k}}, \nu_{\tilde{k}}\right) + \phi\left(\xi_N\right)$$
$$\text{where } \xi_{\tilde{k}-1} = f\left(\xi_{\tilde{k}+1}, \nu_{\tilde{k}}\right)$$
$$g\left(\xi_k, \xi_{k+1}, \nu_k\right) = 0$$

Ex. 3.7 Backward DP recursion

$$J\left(\tilde{k}, \xi_{\tilde{k}}\right) = \min_{\nu_{\tilde{k}}} \left\{ f_0\left(\xi_{\tilde{k}}, \nu_{\tilde{k}}\right) + J\left(\tilde{k}+1, \xi_{\tilde{k}+1}\right) \right\}$$
$$\text{s.t.} \quad \xi_{\tilde{k}} = f\left(\xi_{\tilde{k}+1}, \nu_{\tilde{k}}\right)$$

Consider $\tilde{k} = N-1$

$$x_1 = \xi_{N-1}$$
$$u_0 = \nu_{N-1}$$

$$J\left(N-1, x_1\right) = \min_{u_0} \left\{ f_0\left(x_i, u_0\right) + J\left(N, x_0\right) \right\}$$

…compare with (3.17)

$$\Rightarrow J_a\left(k, x\right) = J\left(N - \tilde{k}, x\right)$$

3.3 Discrete time, infinite horizon DP

3.3.1 Problem formulation

Ex. 3.8 Problem setup

$$\min \sum_{k=0}^{N=\infty} f_0\left(x_k, u_k\right) \tag{3.18}$$

$$\text{s.t.} \quad x_{k+1} = f\left(x_k, u_k\right)$$
$$u_k \in \mathcal{U}\left(x_k\right)$$

(Optimal) cost to go

$$J\left(k, x_k\right) = \min \sum_{j=k}^{\infty} f_0\left(x_j, u_j\right) \tag{3.19}$$

Value function

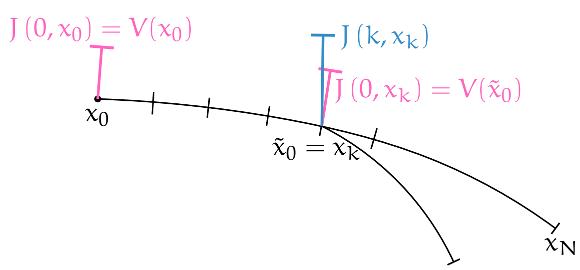
$$V(x) = J\left(0, x\right) \tag{3.20}$$

Remark: 'Cannot go backwards starting from infinity!'

Mo. 26/11/18

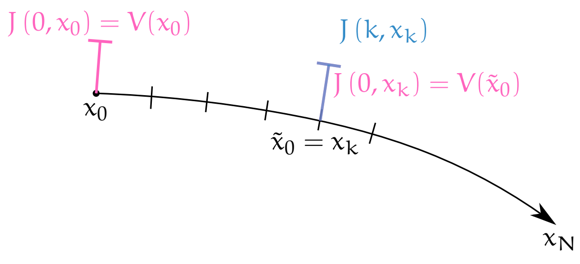
- N finite ($N < \infty$):

$$J\left(k, x_k\right) = \min_{u \in \mathcal{U}(x)} \left\{ f_0\left(x, u\right) + J\left(k+1, f\left(x, u\right)\right) \right\}$$



$J\left(k, x_k\right) \neq V\left(\tilde{x}_0\right)$

- N infinite $N = \infty$



$J(k, x_k) = V(\tilde{x}_0)$
time-invariant case!

$J(k,x)$ replaced by $V(x)$ in DP recursion:

DP equation | Bellman equation

$$V(x) = \min_{u \in \mathcal{U}(x)} \{f_0(x,u) + V(f(x,u))\} \tag{3.21}$$

THEOREM: (necessity)

Consider:

$$\overline{V}(x) = \min_{u \in \mathcal{U}(x)} \{f_0(x,u) + \overline{V}(f(x,u))\} \tag{3.22}$$

$$\overline{u}(x) = \arg\min \{f_0(x,u) + \overline{V}(f(x,u))\} \tag{3.23}$$

Suppose V in problem setup exists, then V solves $\overline{V}(x)$ in (3.22) (and the optimal feedback solves $\overline{u}(x)$ in (3.23))

Remark

- Not sufficient!
- DP equation is a fix point equation (in a function space) in the value function

$$\overline{V} = \overline{f}(\overline{V})$$

Value function iteration and policy iteration algorithm

$$\min \sum_{k=0}^{N=\infty} \alpha^k f_0(x_k, u_k), \quad \alpha \in (0,1) \tag{3.24}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$u_k \in \mathcal{U}(x_k)$$

α discount factor (forgetting)

Assumption Suppose $|f_0(x,u)| \leqslant M \quad \forall\, x, u.$

$$\sum \alpha^k f_0(x_k, u_k) \leqslant \sum_{k=0}^{N=\infty} \alpha^k M = M \frac{\alpha}{1-\alpha}$$

Notice

$$J(k, x_k) = \min \sum_{j=k}^{\infty} \alpha^j f_0(x_j, u_j)$$
$$= \alpha^k \min \sum_{j=0}^{\infty} \alpha^j f_0(\tilde{x}_j, \tilde{u}_j)$$
$$= \alpha^k V(\tilde{x}_0) = \alpha^k V(x_k)$$
$$\alpha^k V(x) = \min \left\{ f_0(x,u)\, \alpha^k + \alpha^{k+1} V(f(x,u)) \right\}$$
$$V(x) = \min_{u \in \mathcal{U}(x)} \{f_0(x,u) + \alpha V(f(x,u))\}$$

Ex. 3.9 DP operator T

$$TV(.) := \min_{u \in \mathcal{U}(.)} \{f_0(.,u) + \alpha V(f(.,u))\} \tag{3.25}$$

$$V(.)\text{function} \longrightarrow \boxed{T} \longrightarrow TV(.)\text{function}$$

$$T_{w(x)} V(x) := \{f_0(x, w(x)) + \alpha V(f(x, w(x)))\} \tag{3.26}$$

Properties Let V_1, V_2 be some functions

contribution:

$$\|TV_1 - TV_2\|_\infty \leqslant \alpha \|V_1 - V_2\|_\infty \tag{3.27}$$

monotonity:

$$V_1 \geqslant V_2 \Rightarrow TV_1 \geqslant TV_2 \tag{3.28}$$

$$\exists \overline{V} : T\overline{V} = \overline{V} \tag{3.29}$$

iteration converges to unique fix point:

$$T \dots T(TV_1) =: T^\infty V_1 = \overline{V} \tag{3.30}$$

Remark

$$\overline{V} = T\overline{V} \quad (\triangleq \text{ DP equations})$$
$$T^\infty V_1 = \overline{V}$$
$$\left(V_2 := TV_1 \dots V_{k+1} = TV_k \overset{k\to\infty}{\longrightarrow} \overline{V}\right), \text{ note}^2$$

Proof Banach space fixpoint theorem
Let $(X, \|.\|)$ be a Banach space.
Let $\Phi : X \to X$ be continuous and let $\|\Phi(x) - \Phi(y)\| \leqslant \alpha \|x - y\|, \alpha \in (0,1)$.

[2] value function iteration algorithm

Then there exists a unique fixpoint $x^* = \Phi(x^*)$ and $x_{k+1} = \Phi(x_k, x_k) \to x^*$.

$$\begin{aligned} V(x) &= \|x - x^*\| \\ V(x_{k+1}) - V(x_k) &= \|\Phi(x_k) - \Phi(x^*)\| - \|x_k - x^*\| \\ &\leqslant \alpha \|x_k - x^*\| - \|x_k - x^*\| < 0 \end{aligned}$$

(and the opt. feedback)

$$\left\| \min_u f(u) - \min_u g(u) \right\| \leqslant \max_u \|f(u) - g(u)\|$$

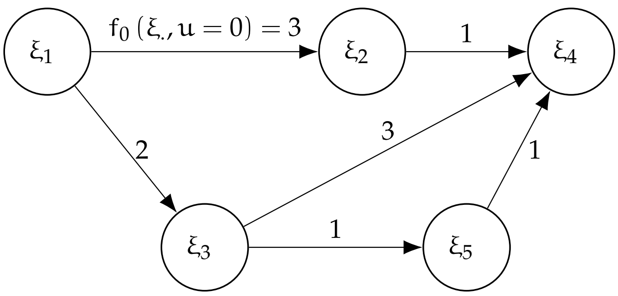
Value function iteration

- Take any V_0
- Compute $V_{k+1} = TV_k$
- Iterate until 'convergence'

Example

$$\begin{aligned} \mathcal{X} &= \{\xi_1, \dots \xi_5\} && \text{state space} \\ \mathcal{U} &= \{0, 1\} && \text{input space} \end{aligned}$$

Dynamics:



$$V_1 = V(\xi_1)$$

$$\begin{aligned} f(\xi_1, u=0) &= \xi_2 \\ f(\xi_1, u=1) &= \xi_3 \\ f(\xi_2, u=0|1) &= \xi_4 \\ &\vdots \\ f(\xi_4, u=0|1) &= \xi_4 \end{aligned}$$

$$\text{Cost: } \sum_{k=0}^{N=\infty} \underbrace{(0.9)}_{\alpha}{}^{k} f_0(x_k, u_k)$$

$f_0(\xi_2, 0(1) = 1)$

$$\begin{aligned} V &= [V(\xi_1) \dots V(\xi_5)] = [V_1 \dots V_5] \\ V^{(k+1)} &= TV^{(k)} \end{aligned}$$

$$\begin{aligned} V(\xi_1^{\text{new}}) &= V_1^{\text{new }(k+1)} \\ &= \min \left\{ f_0(\xi_1, 0) + 0.9 \underbrace{V_2^{\text{old}}}_{\text{note}^3}, 2 + 0.9 V_3^{\text{old}} \right\} \\ &\vdots \\ V_2^{\text{new}} &= \\ &\vdots \\ V_5^{\text{new}} &= \end{aligned}$$

MC:

$$\begin{aligned} V_{\text{old}} &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix} \\ V_{\text{new}} &= V_{\text{old}} \end{aligned}$$

for $i = 1 : 100$

$V_{\text{new}})1_= \min\{3 + \alpha V_{\text{old}}(2), 2 + \alpha V_{\text{old}}(3)\}$

$\vdots$

$V_{\text{now}}(5) = \dots$

$V_{\text{old}} = V_{\text{new}}$

end

3.3.2 Policy iteration algorithm

$$u = [u(\xi_1), u(\xi_2), \dots u(\xi_5)]$$

1. u^0 arbitrary (initial policy)
2. Solve $V^*(x) = Tu^k V^k(x) \quad \forall\, x \in \mathcal{X}$
3. Solve

$$\begin{aligned} u^{k+1} &= \arg\left(TV^k\right)(x) \\ &= \arg\min \left\{ f_0(x, u) + \alpha V^k(f(x, u)) \right\} \end{aligned}$$

4. Go to 2.

Remark $V^k \geqslant V^{k+1}$

Fr.
30/11/18

3.3.3 Summary

Discrete time DP

$$\min \sum_{k=0}^{N-1} \alpha^k f_0(x_k, u_k) + \phi(x_N) \tag{3.31}$$

$$\begin{aligned} \text{s.t.} \quad x_{k+1} &= f(x_k, u_k) \quad k = 0 \dots N \\ u_k &\in \mathcal{U}(x_k) \end{aligned}$$

[3] $V_2^{\text{old}} = V(f(\xi_1, 0)) = V(\xi_2)$

For $N < \infty$: DP recursion

$$J(k, x_k) = \min_{u_k \in \mathcal{U}(x_k)} \left\{ \alpha^k f_0(x_k, u_k) + \underbrace{J(k+1, x_{k+1})}_{\text{cost to go}} \right\} \tag{3.32}$$

s.t. $x_{k+1} = f(x_k, u_k) \quad k = 0 \dots N$

$$J(N, x) = \phi(x) \tag{3.33}$$

- Necessary and sufficient optimal condition

$$\text{opt FB } \overline{u}(k, x_k) = \arg\min \left\{ \alpha^k f_0(x_k, u_k) + J(k+1, x_{k+1}) \right\} \tag{3.34}$$

- Algorithms: discretisation (Approximated Dynamic Programming / ADP algorithm) $\to$ curse of dimensionality

For $N = \infty \quad (\phi = 0)$: DP (Bellman) equation

$$V(x_k) = \min_{u_k \in \mathcal{U}(x_k)} \{ f_0(x_k, u_k) + \alpha V(x_{k+1}) \}$$

- Necessary (and sufficient under additional assumptions) condition

$$\text{opt FB } \overline{u}(x_k) = \arg\min \{ f_0(x_k, u_k) + \alpha V(x_{k+1}) \} \tag{3.35}$$

- Algorithms: fix point iteration algorithm (+ discrete $\to$ ADP algorithm) $\alpha \in (0, 1)$

Remark Mann iteration
T : Hilbert space $\to \mathcal{X}$

$$x_{k+1} = (1 - \lambda_k) x_k + \lambda_k T(x_k) \overset{k \to \infty}{\longrightarrow} x^* \tag{3.36}$$

$$T(x^*) = x^*, \exists\, x^*, \text{if } \|TV_1 - TV_2\| \overset{\alpha}{\leqslant} \|V_1 - V_2\|$$

3.4 Continuous time, finite horizon DP (HJB theory)

HJB: Hamilton-Jacobi-Bellman

3.4.1 Problem formulation

Ex. 3.10 Problem setup

$$\min_{u(.)} \int_{t_0}^{t_f} f_0(t, x, u) \; dt + \phi(t_f, x_f) \tag{3.37}$$

$$\text{s.t.} \quad \dot{x} = f(t, x, u) \qquad t \in [t_0, t_f]$$
$$u(t) \in \mathcal{U}(t, x(t)) \subset \mathbb{R}^q$$
$$u \text{ piecewise continuous}$$

- Optimal cost to go

$$J^*(t_1, x_1) = \min_{u(.)} \int_{t_1}^{t_f} f_0(t, x, u) \; dt + \phi(x_f) \tag{3.38}$$

- Value function

$$J(t_0, x)) = V(x)$$

$$\begin{aligned} J(t_0, x_0) &\overset{\text{Def.}}{=} \min \int_{t_0}^{t_f} f_0(t, x, u) \; dt + \phi(x_f) \\ &= \int_{t_0}^{t_f} f_0(t, x^*(t), u^*(t)) \; dt + \phi(x_f^*) \\ &= \int_{t_0}^{t_1} f_0(t, x^*(t), u^*(t)) \; dt \\ &\quad + \underbrace{\int_{t_1}^{t_f} f_0(t, x^*(t), u^*(t)) \; dt}_{\text{PoO } J(t_1, x_1)} \\ &= \int_{t_0}^{t_1} f_0(t, x^*(t), u^*(t)) \; dt + J(t_1, x_1) \end{aligned}$$

PoO as a formula:

$$J(t_0, x_0) = \min_u \left\{ \underbrace{\int_{t_0}^{t_1} f_0(t, x, u) \; dt}_{\text{note}^4} + J(t_1, x_1) \right\}$$

[4] 'DP recursion' but not as useful as in discrete-time since $\int_{t_0}^{t_1} \dots \, dt$

3.4.2 Principle of optimality

Idea Infinitesimal version of the PoO:

$$t_1 = t_0 + \Delta t, \quad \Delta t \to 0, (J \in C^1)$$

$$\begin{aligned}
&J(t_0, x_0) \\
&= \min_{u(.)} \left\{ \int_{t_0}^{t_0+\Delta t} f_0(t, x, u)\, dt + \underbrace{J(t_0 + \Delta t, x(t_0 + \Delta t))}_{g(t_0+\Delta t),\ \text{note}^5} \right\} \\
&= \min_{u(.)} \{ f_0(t_0, x_0, u_0)\Delta t + J(t_0, x_0, u_0) \\
&\quad + \left[\frac{\partial J(t_0, x_0, u_0)}{\partial t} + \frac{\partial J(t_0, x_0, u_0)}{\partial x} f(t_0, x_0, u_0) \right] \Delta t \\
&\quad + \mathcal{O}(\Delta t^2) \}
\end{aligned}$$

Divide by Δt:

$$\begin{aligned}
\cancel{J(t_0, x_0)} = &\min_{u(.)} \{ f_0(t_0, x_0, u_0) + \cancel{J(t_0, x_0, u_0)} \\
&+ \frac{\partial J(t_0, x_0, u_0)}{\partial t} \\
&+ \frac{\partial J(t_0, x_0, u_0)}{\partial x} f(t_0, x_0, u_0) \\
&+ \mathcal{O}(\Delta t) \} \\
0 = &\min_{u(.)} \left\{ f_0(t_0, x_0, u_0) + \frac{\partial J(t_0, x_0, u_0)}{\partial t} \right. \\
&+ \frac{\partial J(t_0, x_0, u_0)}{\partial x} f(t_0, x_0, u_0) \\
&+ \mathcal{O}(\Delta t) \}
\end{aligned}$$

$\Delta t \to 0$:

$$\begin{aligned}
&-\frac{\partial J(t_0, x_0)}{\partial t} \\
&= \min_{u(.)} \left\{ f_0(t_0, x_0, u_0) + \frac{\partial J(t_0, x_0, u_0)}{\partial x} f(t_0, x_0, u_0) \right\}
\end{aligned} \tag{3.39}$$

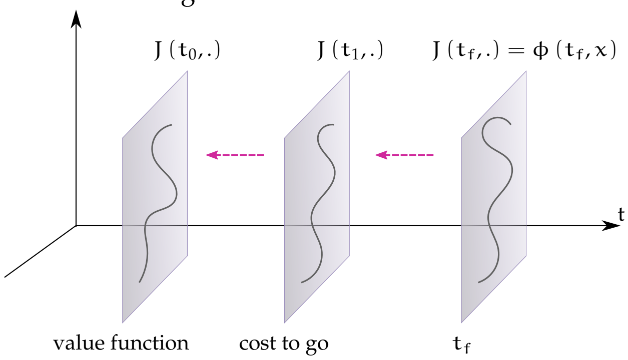
3.4.3 Hamilton-Jacobi-Bellman equation

PDE, nonlinear[6]

$$\begin{aligned}
&-\frac{\partial J(t_0, x_0)}{\partial t} \\
&= \min_{u(.)} \left\{ f_0(t_0, x_0, u_0) + \frac{\partial J(t_0, x_0, u_0)}{\partial x} f(t_0, x_0, u_0) \right\}
\end{aligned} \tag{3.39}$$

$$J(t_f, x) = \phi(t_f, x) \tag{3.40}$$

Integration backwards in time.



3.4.4 Verification theorem

THEOREM: necessity

If $J \in C^1$, then J is a solution of the HJB equation (3.39).

THEOREM: sufficiency, verification theorem

Suppose $\bar{J}(t, x) \in C^1$, and

$$\begin{aligned}
-\frac{\partial \bar{J}(t, x)}{\partial t} &= \min_{u \in \mathcal{U}(x)} \{ f_0(t, x, u) \\
&\quad + \frac{\partial \bar{J}(t, x, u)}{\partial x} f(t, x, u) \}
\end{aligned} \tag{3.39}$$

$$J(t_f, x) = \phi(t_f, x) \tag{3.40}$$

$$\begin{aligned}
\bar{u}(t, x) &= \arg\min \{ f_0(t, x, u) \\
&\quad + \frac{\partial \bar{J}(t, x, u)}{\partial x} f(t, x, u) \}
\end{aligned} \tag{3.41}$$

Then $\bar{J} = J$ and $\bar{u}$ is the (an) optimal feedback.

Proof:

$$g(t_1) - g(t_0) = \int_{t_0}^{t_1} \dot{g}(t)\, dt$$

Find theorem of calculus.

Step 1 $\bar{J} \geqslant J$

$$\begin{aligned}
-\frac{\partial \bar{J}(t, x)}{\partial t} &= \min_{u \in \mathcal{U}(x)} \{ f_0(t, x, u) \\
&\quad + \frac{\partial \bar{J}(t, x, u)}{\partial x} f(t, x, u) \} \\
&= f_0(t, x, \bar{u}(t, x)) \\
&\quad + \frac{\partial \bar{J}(t, x, \bar{u}(t, x))}{\partial x} f(t, x, \bar{u}(t, x)) \\
-\dot{\bar{J}}(t, x) &= f_0(t, x, \bar{u}(t, x))
\end{aligned}$$

[5] $g(t_0 + \Delta t) = g(t_0) + \dot{g}(t_0)\Delta t + \mathcal{O}(\Delta t^2)$
[6] nonlinear due to min...

Integrate from t_0 to t_f:

$$\begin{aligned}\bar{J}(t_0, x_0) &= \int_{t_0}^{t_f} f_0(t, x, \bar{u}(t,x))\ dt + \bar{J}(t_f, x_f) \\ &= \int_{t_0}^{t_f} f_0(t, x, \bar{u}(t,x))\ dt + \phi(t_f, x_f)\end{aligned}$$

$\therefore \bar{J}$ indeed represents the cost to go.

Step 2 $\bar{J} \leqslant J, \forall\, u(t,x) \in \mathcal{U}(t,x)$

$$\begin{aligned}-\frac{\partial \bar{J}(t,x)}{\partial t} &\leqslant f_0(t,x,u) + \frac{\partial \bar{J}(t,x,u)}{\partial x} f(t,x,u) \\ -\dot{\bar{J}}(t,x) &\leqslant f_0(t,x,u)\end{aligned}$$

Integrate from t_0 to t_f:

$$\begin{aligned}\bar{J}(t_0,x_0) &\leqslant \int_{t_0}^{t_f} f_0(t,x,u)\ dt + \bar{J}(t_f,x_f) \\ \bar{J}(t_0,x_0) &\leqslant \int_{t_0}^{t_f} f_0(t,x,u)\ dt + \phi(t_f,x_f) \\ &\qquad \forall\, u(t,x) \in \mathcal{U}(t,x), \text{note}^7 \\ \bar{J}(t_0,x_0) &\leqslant J(t_0,x_0)\end{aligned}$$

$\bar{u}$ is optimal, because $(\leqslant) \to (=)$.

Remarks

- Solving HJB (3.39) is in general hard (analytically + computationally[8])
- Sometimes helpful:

$$\text{opt FB } = \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right) \tag{3.42}$$

$$\bar{u}(t,x,\lambda) = \arg\min\left\{f_0(t,x,u) + \lambda^T f(t,x,u)\right\} \tag{3.43}$$

$$\begin{aligned}-\frac{\partial \bar{J}}{\partial t} &= f_0\left(t, x, \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right)\right) \\ &\quad + \underbrace{\frac{\partial \bar{J}(t,x)}{\partial x}}_{\lambda^T} f\left(t, x, \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right)\right)\end{aligned} \tag{3.44}$$

Example: LQR

$$\begin{aligned}&\min \int_{t_0}^{t_f} x^T Q X + u^T R u\ dt + x_f^T S_f x_f, \quad R > 0, S_f > 0 \\ &\text{s.t.} \quad \dot{x} = Ax + Bu\end{aligned}$$

Find optimal feedback $\bar{u}(t,x)$

$$\bar{u}(t,x) = \arg\min\left\{x^T Q X + u^T R u + \lambda^T (Ax + Bu)\right\} \tag{3.43}$$

$$\begin{aligned}\frac{\partial}{\partial u}\left(x^T Q X + \bar{u}^T R \bar{u} + \lambda^T (Ax + B\bar{u})\right) &= 0^T \\ 2R\bar{u} + B^T \lambda &= 0 \\ \bar{u}(\lambda) &= -\frac{1}{2} R^{-1} B^T \lambda\end{aligned}$$

$$\begin{aligned}-\frac{\partial \bar{J}}{\partial t} &= x^T Q x + \frac{1}{4}\lambda^T B R^{-1} B^T \lambda + \lambda^T A x \\ &\quad + \lambda^T B\left(-\frac{1}{2} R^{-1} B^T \lambda\right) && (3.44) \\ &= x^T Q x - \frac{1}{4}\lambda^T B R^{-1} B^T \lambda + \lambda^T A x && (3.45)\end{aligned}$$

$$\bar{J}(t_f, x) = x^T S_F x \tag{3.40}$$

$$\begin{aligned}\text{We try:} \quad \bar{J}(t,x) &= x^T S(t) x, \text{ with} \\ \frac{\partial \bar{J}}{\partial t} &= x^T \dot{S}(t) x \\ \frac{\partial \bar{J}}{\partial x} &= 2x^T S(t) = \lambda^T \\ &\to \text{ plug into } -\frac{\partial \bar{J}}{\partial t} \text{ in (3.45)}\end{aligned}$$

$$\begin{aligned}-x^T \dot{S}(t) x &= x^T Q x - \frac{4}{4} x^T S(t) B R^{-1} B^T S(t) x \\ &\quad + 2x^T S(t) A x \quad \forall\, x \\ 0 &= x^T \underbrace{\left(\dot{S}(t) + Q - S(t) B R^{-1} B^T S(t) + 2S(t) A\right)}_{\overset{!}{=}0} x\end{aligned}$$

Ricatti equation (ODE)[9]:

$$\begin{aligned}0 &= \dot{S}(t) + Q - S(t) B R^{-1} B^T S(t) + 2S(t) A && (3.46) \\ S(t_f) &= S_f && (3.47)\end{aligned}$$

$$\bar{u}(\lambda) = -\frac{1}{2} R^{-1} B^T \lambda = -R^{-1} B^T S(t) x$$

Mo. 03/12/18

3.5 Continuous time infinite horizon DP

Ex. 3.11 Problem setup

$$\min \int_0^{t_f = \infty} f_0(x,u)\ dt$$

$$\begin{aligned}\text{s.t.} \quad \dot{x} &= f(x,u) \\ u(t) &\in \mathcal{U}(x(t))\end{aligned}$$

Assumptions

- $f_0(x,u) \geqslant 0 \quad \forall\, x, u, \quad x \in \mathbb{R}^n, u \in \mathbb{R}^q$
- $f_0(x,u) > 0$ whenever $u \neq 0$
- $f_0(x,0)$ is zero state detectable
  i.e., $y = f_0(x(t), 0) \to 0 \Rightarrow x(t) = 0$
- $f_0(0,0) = 0$

[7] Holds also for the optimal feedback u^*

[8] $\dim(x) \leqslant 10$

[9] PDE is converted into an ODE

Cost-to-go $J(x_1, t_1)$
"Cost with x_1 as new initial condition."

$$J(x_1, t_1) = \min_u \int_{t_1}^{\infty} f_0(x, u)\, dt$$

$t_f < \infty$

$$-\frac{\partial J(t,x)}{\partial t} = \min_{u \in \mathcal{U}(x)} \left\{ f_0(x,u) + \frac{\partial J(t,x)}{\partial x} f(x,u) \right\}$$

$t_f = \infty$

cost-to-go $\triangleq$ cost/value function
$J(t_1, x_1) = V(x_1)$

$$0 = \min_{u \in \mathcal{U}(x)} \left\{ f_0(x,u) + \frac{\partial V(x)}{\partial x} f(x,u) \right\}$$

Main questions

a) Optimality, stability
b) Robustness

a) Optimality, stability

THEOREM:

Suppose the assumptions above hold true. Let $\overline{V} \in C^1$, which is positive definite (and radially unbounded), be

$$\text{s.t.} \quad 0 = \min_{u \in \mathcal{U}(x)} \left\{ f_0(x,u) + \frac{\partial \overline{V}(x)}{\partial x} f(x,u) \right\}$$

$$\text{and } \overline{u}(x) = \arg\min \left\{ f_0(x,u) + \frac{\partial \overline{V}(x)}{\partial x} f(x,u) \right\}$$

then $\overline{V} = V$ (value function V^*) and $\overline{u}$ is an optimal feedback which is (globally) asymptotically stabilising (w.r.t. $x = 0$).

Proof
Step 1: stability

$$0 = f_0(x, \overline{u}(x)) + \underbrace{\frac{\partial \overline{V}(x)}{\partial x} f(x, \overline{u}(x))}_{\dot{\overline{V}}(x) = L_f \overline{V}(x)} \quad \text{HJBE}$$

$\dot{\overline{V}}(x) = -f_0(x, \overline{u}(x)) \leqslant 0$
$\Rightarrow x = 0$ is (globally) stable

Provided $\overline{V}$ is positive definite (and radially unbounded).

Step 2: asymptotic stability

$$\dot{\overline{V}}(x) = -f_0(x, \overline{u}(x))$$

$$\text{(La Salle)} \quad x(t) \to \left\{ x : \dot{\overline{V}}(x) = 0 = f_0(x, \overline{u}(x)) = 0 \right\}$$

Fox $f_0(x,u) > 0, u \neq 0$:

$$\Rightarrow x(t) \to \{x : f_0(x, 0) = 0\}$$

Step 3: HJBE

$$0 \leqslant f_0(x,u) + \underbrace{\frac{\partial \overline{V}(x)}{\partial x} f(x,u)}_{\dot{\overline{V}}} \quad \forall\, u(x) \in \mathcal{U}(x)$$

$$\left(0 = f_0(x,u) + \frac{\partial \overline{V}(x)}{\partial x} f(x,u) \right)$$

$$\dot{\overline{V}}(x) \leqslant f_0(x,u)$$

$$\overline{V}(x_0) - \overline{V}(x(t)) \leqslant \int_0^t f_0(x,u)\, d\tau, \quad t \to \infty$$

$$\overline{V}(x_0) \leqslant \int_0^{\infty} f_0(x,u)\, d\tau \quad \forall\, u(x) \in \mathcal{U}(x), n^{10}$$

- $\overline{V}(x_0)$ is a lower bound of the value function ($\overline{V} \leqslant V$)
- $\overline{V}$ is achieved by $\overline{u}$, hence $\overline{V} = V, \overline{u} = u^*$

b) Robustness
Setup:

$$\min \int_0^{\infty} q(x) + u^R(x) u\, dt \quad u \in \mathbb{R}^q, x \in \mathbb{R}^n$$

$$\text{s.t. } \dot{x} = f(x) + G(x)u, \quad q(x) > 0, R(x) > 0$$

HJBE:

$$0 = \min \Big\{ q(x) + u^T R(x) u + \frac{\partial \overline{V}(x)}{\partial x} f(x) + \frac{\partial \overline{V}(x)}{\partial x} G(x) u \Big\} \tag{3.48}$$

$$\frac{\partial}{\partial u}\{.\} = 2R(x)\overline{u} + G^T(x) \frac{\partial \overline{V}^T(x)}{\partial x} = 0$$

$$\overline{u}(x) = -\frac{1}{2} R^{-1}(x) G^T(x) \frac{\partial \overline{V}^T(x)}{\partial x} \tag{3.49}$$

$$\overline{u}(x) = -\frac{1}{2} R^{-1}(x) L_g V(x), \quad u \in \mathbb{R}, (G = \varphi)$$

'$L_g V$' control law $\to$ nonlinear control

Substitute (3.49) in (3.48)

$$0 = q(x) + \frac{1}{4} \frac{\partial V(x)}{\partial x} G(x) R(x) G^T(x) \frac{\partial V^T(x)}{\partial x} + \frac{\partial V(x)}{\partial x} f(x) - \frac{1}{2} \frac{\partial V(x)}{\partial x} G(x) R^{-1}(x) G^T(x) \frac{\partial V^T(x)}{\partial x}$$

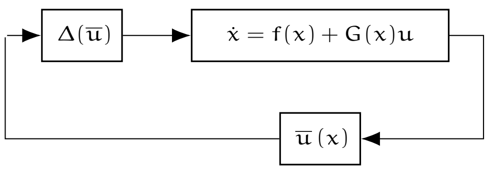
$$0 = q(x) - \frac{1}{4} \frac{\partial V(x)}{\partial x} G(x) R^{-1}(x) G^T(x) \frac{\partial V^T(x)}{\partial x} + \frac{\partial V(x)}{\partial x} f(x)$$

$$0 = q(x) + \frac{1}{2} \frac{\partial V(x)}{\partial x} G(x) \overline{u}(x) + \frac{\partial V(x)}{\partial x} f(x)$$

[10] in particular $u^* \in \mathcal{U}(x)$

Uncertainties e.g. actuator nonlinearities (unmodelled)

input uncertainty



For which Δ's is the closed-loop asymptotically stable?

$$(\dot{\bar{V}} =)\dot{V}(x) = \frac{\partial V(x)}{\partial x}\left(f(x) + G(x\Delta(\bar{u}))\right) \overset{!}{\leqslant} 0$$

$$\begin{aligned}\dot{V}(x) &= \frac{\partial V(x)}{\partial x} f(x) + \frac{1}{2}\frac{\partial V(x)}{\partial x} G(x)\bar{u}(x) \\ &\quad - \frac{1}{2}\frac{\partial V(x)}{\partial x} G(x)\bar{u}(x) + \frac{\partial V(x)}{\partial x} G(x)\Delta(\bar{u}) \\ &= \underbrace{-q(x)}_{<0} + \underbrace{\frac{\partial V(x)}{\partial x} G(x)\left(\Delta(\bar{u}) - \frac{1}{2}\bar{u}(x)\right)}_{*}\end{aligned}$$

Asymptotically stable if $* \leqslant 0$:

$$\underbrace{\frac{\partial V(x)}{\partial x} G(x)}_{-2\bar{u}^T R}\left(\Delta(\bar{u}) - \frac{1}{2}\bar{u}(x)\right) \leqslant 0$$

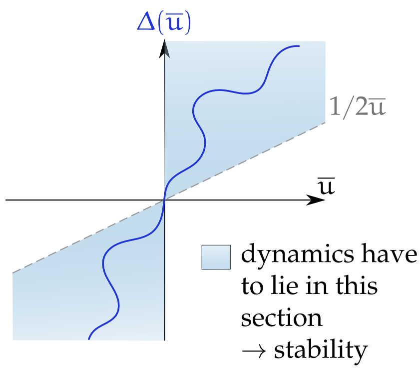
$$-2\bar{u}^T(x)R(x)\left(\Delta(\bar{u}) - \frac{1}{2}\bar{u}(x)\right) \overset{?}{\leqslant} 0$$

$$\bar{u}^T(x)R(x)\left(\Delta(\bar{u}) - \frac{1}{2}\bar{u}(x)\right) \overset{?}{\geqslant} 0$$

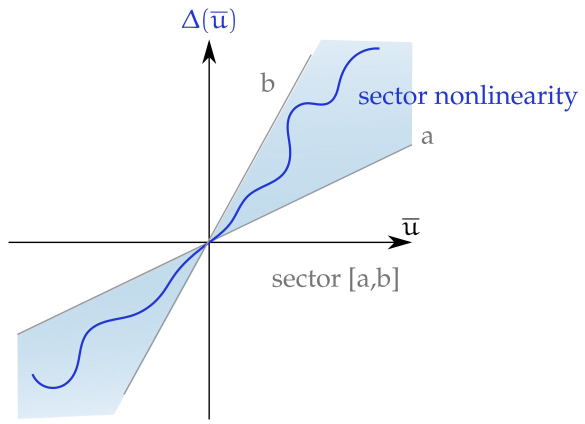
q=1 $(u \in \mathbb{R})$ $R(x) = r(x) > 0$

$$\bar{u} \cdot \left(\Delta(\bar{u}) - \frac{1}{2}\bar{u}(x)\right) \geqslant 0$$

$$\bar{u}\Delta(\bar{u}) - \frac{1}{2}\bar{u}^2 \geqslant 0$$



Remark:



THEOREM: (similar to $q = 1$)

Suppose

$$R(x) = \begin{bmatrix} r_x(1) & & 0 \\ & \ddots & \\ 0 & & r_q(x) \end{bmatrix} \tag{3.50}$$

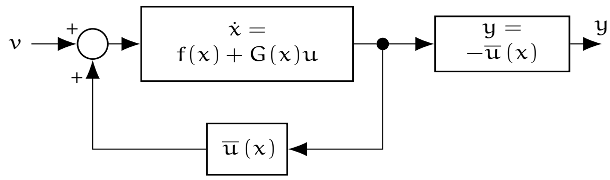
$$\Delta(\bar{u}) = \begin{bmatrix} \Delta_1(\bar{u}_1 \\ \vdots \\ \Delta_q(\bar{u}_q \end{bmatrix} \tag{3.51}$$

then $\bar{u}$ achieves a sector margin of $[1/2, \infty]$

$q > 1$

Remark:

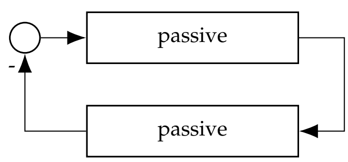
- LQR gain margin $[1/2, \infty]$
- disc margin (as a generalisation of phase margin) $\to$ nonlinear control



From $v \to y$, system is passive, i.e.

$$\dot{V} = -\frac{1}{2}y^T y + y^T v$$

Passive systems



$\Rightarrow$ " overall system is passive"

Example

$$\dot{x} = x^2 + u$$

$$u = -x^2 - x \Rightarrow \text{ closed loop}$$

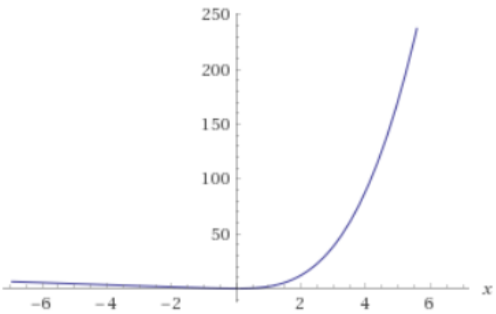
$$\Rightarrow \dot{x} = -x \quad \text{FB linearisation}$$

$$\Delta(u) = (1+\varepsilon)\,u, \quad \varepsilon > 0$$
$$\dot{x} = x^2 + (1+\varepsilon)\left(-x^2 - x\right)$$
$$-(1+\varepsilon)\,x - \varepsilon x^2$$

$\rightarrow$ closed loop is not globally stable – even worse, shows finite escape behaviour[11], even for small uncertainties.

Integrating the HJBE:

$$V(x) = \frac{2}{3}x^3 + \frac{2}{3}\left(1+x^2\right)^{\frac{3}{2}} - \frac{2}{3} > 0 \quad \text{pos. def.}$$



$$\min \int_0^\infty \underbrace{x^2 + u^2}_{f_0}\; dt$$
$$\overline{u}\,(x) = -x^2 - \underbrace{\left(\sqrt{1+x^2}\right)}_{\text{state-dependent gain}}\; x$$

Remark

- **Inherent robustness**
  uncertainties/unmodelled dynamics are not taken into account in the control design
- **Robust design**
  uncertainties are explicitly taken into account in the control design
  worst case represented by maximum disturbance w (unmodelled dynamics) $\rightarrow \max_w$

$$\min_u \max_w \int_0^\infty f_0\,(x,u,w)\; dt$$
$$\text{s.t.} \quad \dot{x} = f\,(x,u,w)$$

  HJBIE[12]:

$$0 = \min_u \max_w \left\{ f_0 + \dot{V} \right\}$$

  (2 player zero sum differential game)
  - u: control engineer
  - w: nature/disturbance

  $\rightarrow$ LQ-Setup: H_2/H_∞ control (robust control)

- **Inverse optimality**
  "normal case": use value function as Lyapunov function
  "inv. opt": use control/Lyapunov function as value function

  Applications: robust stabilisation, nonlinear control

[11] finite escape behaviour $\hat{=}\ |x(t)| \rightarrow \infty, t \rightarrow t^*$, "system explodes"
[12] I: Isaacs

Fr.
07/12/18

4 Receding Horizon Optimal Control

So far:

NLP	DP
finite horizon	(in)finite horizon
discrete time	discrete/continuous
open loop	feedback
"efficient algo"	fix point eqn, PDE

Now:

Receding horizon optimal control (RHOC) merges advantages of NLP (computability) and DP (feedback, infinite horizon)

- MPC (model predictive control)
- MHE (moving horizon estimation)

Motivation optimal feedback design

$$\begin{aligned} \min \int_0^\infty & f_0(x,u)\ dt \\ \text{s.t.}\quad & \dot{x} = f(x,u) \\ & u \in \mathcal{U} \\ & x \in \mathcal{X} \end{aligned}$$

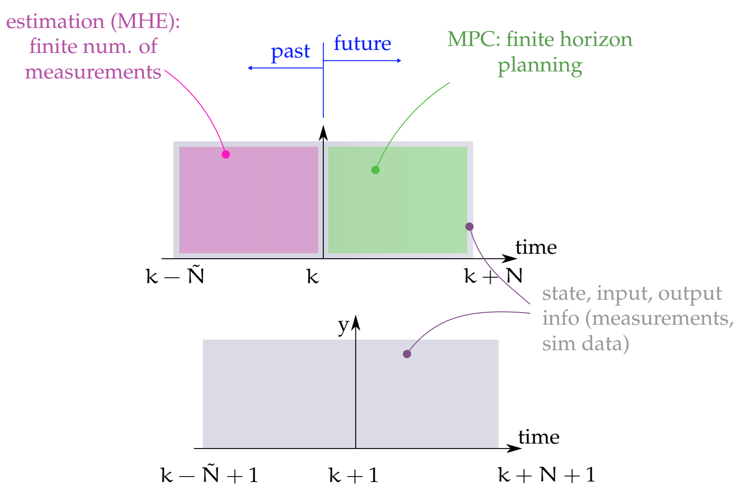
solve PDE, fixpoint equation, NLP $:\to \infty$ many decision variables (constraints)

State estimation use all past information (measurements) for estimation $\to$ unbounded data

4.1 Receding Horizon Online Decision Making

Decision making $\triangleq$ find feedback $u(x)$ or find state estimate $\hat{x}$; "principle of chapter 4"

RH decision making (real-time, online) decision making based on a moving (receding) finite-time window of past and future information



(RH) online decision making := solve (compute) at each/some time instances a decision problem/optimisation problem (based on the receding horizon information)

History

- Economics (1950s), rolling (horizon) plans
- 1963, Propoi: MPC
- 1970, Jazwinski: MHE
- $\geqslant$ 1980s: process control (MPC)[13]

4.2 MPC

- "online (repeated) open loop computations $\triangleq$ feedback"
- repeated open-loop finite-horizon OC policy (implemented in a receding horizon fashion)

[OP] Foundation of OC (book)

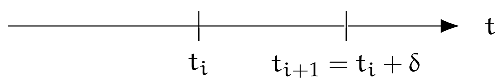
(1967, Marcus, Lee)

"One technique for obtaining a feedback controller/synthesis from knowledge of open-loop controllers is to measure the current process control state and then compute **very rapidly** for the open loop control.

"The first portion of this [open-loop] function is used during a short time interval, after which a new measurement of the state is made and a a new open loop control function is computed for this measurement.

"Then the procedure is repeated."

Model Predictive Control Scheme



T_p prediction/planning horizon
δ sampling interval

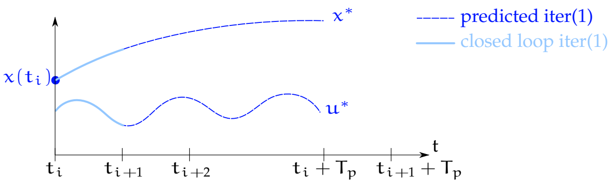
Typical scheme:

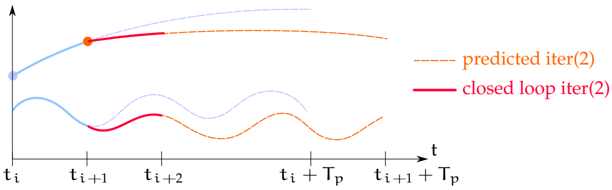
1. Measure the state $x(t_i)$ and initialise it to be the current sate
2. Predict the solution

$$\begin{aligned} u_{\text{MPC}}(., x(t_i)) = \arg\min & \int_{t_i}^{t_i+T_p} f_0(t,x,u)\ dt \\ & + \phi(t_i + T_p, x(t_i+T_p)) \\ \text{s.t.}\quad & \dot{x} = f(t,u,x) \\ & x_0 = x(t_i) \quad \text{IC} \\ & x \in \mathcal{X} \\ & u \in \mathcal{U} \end{aligned}$$

[13] processes are slow $\therefore$ plans can be recomputed every minute

3. Implement $u(t, x(t_i))$ within the sampling interval, $t \in [t_i, t_{i+1}] \to$ 'closed loop'
4. Set value at end of sampling period t_{i+1} to be the new initial value of the next iteration $\to$ step 1 $i \leftarrow i+1$





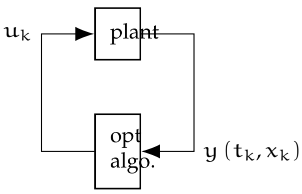
$\vdots$

$\Rightarrow$ predicted solution $\neq$ closed loop solution

$u_{MPC}(t, x(t_i))$

Discussion / Challenges

MPC $\subseteq$ (online) optimisation-based control



Fast and reliable (feasible) algorithms needed[14]!

Sequential/recursive feasibility

"if solution at t_i is feasible, guarantee that there exists a solution at t_{i+1}"

$$\min \int_{t_i}^{t_i+T_p} \ldots \, dt + \phi \quad \text{is feasible}$$

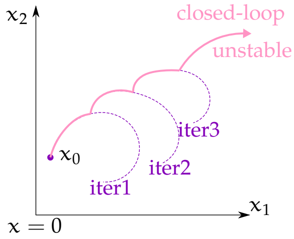
$$\Downarrow$$

$$\min \int_{t_{i+1}}^{t_{i+1}+T_p} \ldots \, dt + \phi \quad \text{feasible?}$$

Stability of MPC

Problem: predicted trajectory $\neq$ closed-loop trajectory

[14] fast: opt algo has to be faster than the plant; reliable: algo not allowed to crash

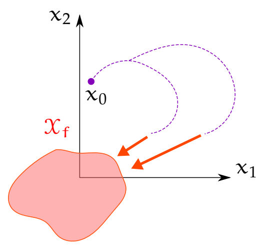


Mo. 10/12/18 exercise

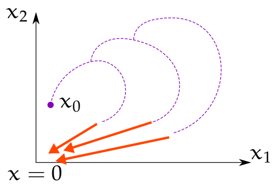
Solution: "change/modify objective function, constraints, prediction horizons ..."

Fr. 14/12/18

- Terminal set based conditions/MPC schemes
  1. Terminal set constraint
     $x(t_i + T_p) \in \mathcal{X}_f$



  2. Zero terminal state constraint
     Special case of the above, where 0 is the desired equilibrium point, $x(t_i + T_p) = 0$



- Terminal set free approaches
  e.g. choose T_p **sufficiently large**
  Disadvantage: no. of decision variables increases, more computation power needed, T_p difficult to quantise

THEOREM: A stability theorem

Consider

$$\begin{aligned} V(x(t_i)) = \min &\int_{t_i}^{t_i+T_p} f_0(x(t), u(t)) \, dt \\ &+ \phi(x(t_i + T_p)) \\ \text{s.t.} \quad &\dot{x}(t) = f(x(t), u(t)), \\ &\quad t \in [t_i, t_i + T_p] \\ &u(t) \in \mathcal{U} \\ &x(t) \in \mathcal{X} \\ &\text{IC } x(t_i) \\ &x(t_i + T_p) \in \mathcal{X}_f \end{aligned}$$

which is solved at each sampling time t_i, $t_{i+1} = t_i + \delta$ and implemented in a receding horizon fashion.

THEOREM: (cont.)

Suppose
- $f(0,0) = 0$
  f_0, ϕ are positive definite
  $\phi \in C^1$
- V well-defined in a set $\mathcal{X}_0 \subseteq \mathbb{R}^n$ of initial conditions, where the problem is feasible.
  $V \in C^1$ in (interior) of $\mathcal{X}_0$
  V positive definite
- $0 \in \mathcal{X}_f \subseteq \mathcal{X}_0 \subseteq \mathcal{X}$
- There exists a 'local feedback controller' $k(x)$ such that
  A1. $k(x) \in \mathcal{U}, x \in \mathcal{X}$
  A2. $\dot{x} = f(x, k(x))$ renders $\mathcal{X}_f$ invariant, i.e.

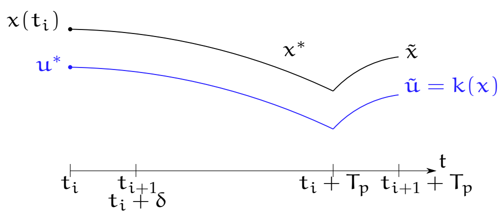
$$\begin{aligned} x(0) &\in \mathcal{X}_f \\ x(t) &\in \mathcal{X}_f, t > 0 \end{aligned}$$

  A3.

$$\begin{aligned} \dot{\phi}(x) &= \nabla\phi^T(x) f(x, k(x)) \\ &\leqslant -f_0(x, k(x)) \end{aligned}$$

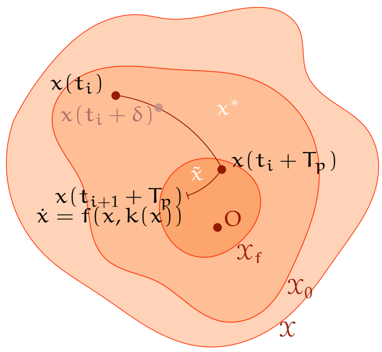
$\Rightarrow$ Then the MPC scheme is recursively feasible and converges $x(t_i) \to 0$, $i \to \infty$ (and the MPC closed loop is asymptotically stable for $\delta \to \infty$)

Proof



a) Recursive feasibility
b) Convergence

a) Recursive feasibility OCP is feasible for $x(t_i) =$ IC
$\Rightarrow$ OCP is feasible for $x(t_{i+1}) =$ IC.



Candidate solution (suboptimal, feasible solution at t_{i+1}:

$$v(t, x(t_{i+1})) = \begin{cases} u^*(t) & t \in [t_i + \delta, t_i + T_p] \\ \tilde{u} = k(x(t)) & t \in [t_i + T_p, t_{i+1} + T_p] \end{cases}$$

We have:

$$\begin{aligned} x(t_{i+1} + T_p) &\in \mathcal{X}_f && \because \text{(A2)} \\ v(t_i, x(t_{i+1})) &\in \mathcal{U} && \because \text{(A1)} \end{aligned}$$

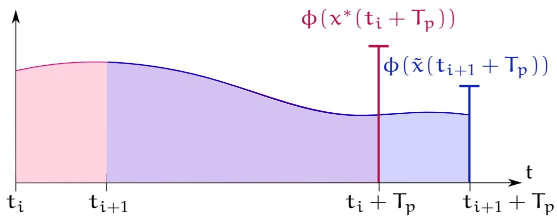
$\Rightarrow$ recursive feasibility
(state constraints ok, since $\mathcal{X}_f$ is invariant and $\mathcal{X}_f \subseteq \mathcal{X}$)

b) Convergence $x(t_i) \to 0$

$$V(x(t_{i+1})) \overset{\text{by def./opt.}}{\leqslant} V(x(t_{i+1}, v)) \overset{?}{<} V(x(t_i))$$

$$\begin{aligned} V(x(t_i)) = &\int_{t_i}^{t_i+T_p} f_0(x^*, u^*)\, dt \\ &+ \phi(x^*(t_i + T_p)) \end{aligned}$$

$$\begin{aligned} V(x(t_{i+1}), v) = &\int_{t_{i+1}}^{t_i+T_p} f_0(x^*, u^*)\, dt \\ &+ \int_{t_i+T_p}^{t_{i+1}+T_p} f_0(\tilde{x}, \tilde{u})\, dt \\ &+ \phi(\tilde{x}(t_{i+1} + T_p)) \end{aligned}$$



$V(x(t_{i+1}, v) < V(x(t_i))$ if
endpiece area $<$ cost difference

$$\begin{aligned} \int_{t_i+T_p}^{t_{i+1}+T_p} f_0(\tilde{x}, \tilde{u})\, dt < &\ \phi(x^*(t_i + T_p)) \\ &- \phi(\tilde{x}(t_{i+1} + T_p)) \end{aligned}$$

$V(x(t_{i+1}), v) = V(x(t_i)) +$ Rest

$$\begin{aligned} &V(x(t_{i+1}), v) \\ = &\int_{t_{i+1}}^{t_i+T_p} f_0(x^*, u^*)\, dt + \int_{t_i+T_p}^{t_{i+1}+T_p} f_0(\tilde{x}, \tilde{u})\, dt + \phi(\tilde{x}(t_{i+1} + T_p)) \\ &+ \int_{t_i}^{t_{i+1}} f_0(x^*, u^*)\, dt - \int_{t_i}^{t_{i+1}} f_0(x^*, u^*)\, dt \\ &+ \phi(x^*(t_i + T_p)) - \phi(x^*(t_i + T_p)) \end{aligned}$$

$$\begin{aligned} V(x(t_{i+1}, v) \leqslant V(x(t_i)) &+ \phi(\tilde{x}(t_{i+1} + T_p)) \\ &- \phi(x^*(t_i + T_p)) \\ &+ \int_{t_i+T_p}^{t_{i+1}+T_p} f_0(\tilde{x}, \tilde{u})\, dt \end{aligned}$$

If $\phi(\tilde{x} \ldots) - \phi(x^* \ldots) + \int_{t_i+T_p}^{t_{i+1}+T_p} f_0 \ldots\, dt \leqslant 0$ (A3)

Remark

$$
\begin{aligned}
\min \quad & \sum_{j=k}^{k+N_p-1} f_0\left(x(j), u(j)\right) + \phi\left(x\left(k+N_p\right)\right) \\
\text{s.t.} \quad & x(j+1) = f(x(j), u(j)), \\
& \qquad j = k, \ldots, k+N_p-1 \\
& u(j) \in \mathcal{U} \\
& x(j) \in \mathcal{X} \\
& \text{IC} = x(k) \\
& \qquad \text{current state of the plant} \\
& x(k+N_p) \in \mathcal{X}
\end{aligned}
$$

Stability proof see exercise

$$x_k \in \mathcal{X}_f \Rightarrow x_{k+1} \in \mathcal{X}_f \tag{4.1}$$

$$x_{k+1} = f(x_k, \overline{k}(x_k))$$

$$x_k \in \mathcal{X}_f \Rightarrow \overline{k}(x_k) \in \mathcal{U} \tag{4.2}$$

$$\phi(x_{k+1}) - \phi(x_k) \leqslant -f_0(x_k, \overline{k}(x_k)) \tag{4.3}$$

$\Rightarrow$ parametrised, time-varying NLP

$$
\begin{aligned}
\min \quad & f(u, p) \qquad p = x(k) \\
\text{s.t.} \quad & g(u, p) \leqslant 0
\end{aligned}
$$

Pros and cons of MPC
- easy to handle state and input constraints
- good model needed
- fast and reliable algorithm needed

Mo. 17/12/18 exercise

Fr. 21/12/18

4.3 Moving Horizon Estimation (MHE)

MHE = (online) optimisation-based, state estimation approach (algo) based on a receding horizon idea, i.e.

Goal least squares parameter estimation $\leftrightarrow$ recursive estimation (Kalman) $\leftrightarrow$ MHE