Optimal Control Notes

SAR

Contents

1 Introduction **1**
- 1.1 Optimal control problems 1
  - 1.1.1 Ingredients of an optimal control problem 1
  - 1.1.2 Optimisation problem (calculus) 1
- 1.2 Topics/Tools 2
- 1.3 Examples 2
  - 1.3.1 Shortest path problem . . . 2
  - 1.3.2 Time-optimal control problem 2
  - 1.3.3 Linear quadratic regulator problem (LQR) 2

2 Nonlinear programming approach **3**
- 2.1 Nonlinear programming and optimal control 3
- 2.2 Problem setup 3
- 2.3 Unconstrained optimisation 3
- 2.4 Constrained optimisation 3
- 2.5 NLP approach to optimal control . 3

3 Dynamic programming **5**
- 3.1 Principle of optimality 5
- 3.2 Discrete-time, finite-dimension DP 6
- 3.3 Discrete time, infinite horizon DP 8
  - 3.3.1 Problem formulation 8
  - 3.3.2 Policy iteration algorithm . 10
  - 3.3.3 Summary 11
- 3.4 Continuous time, finite horizon DP (HJB theory) 11
  - 3.4.1 Problem formulation 11
  - 3.4.2 Principle of optimality . . . 12
  - 3.4.3 Hamilton-Jacobi-Bellman equation 12
  - 3.4.4 Verification theorem 12
- 3.5 Continuous time infinite horizon DP 13

4 Receding Horizon Optimal Control **17**
- 4.1 Receding Horizon Online Decision Making 17
- 4.2 MPC 17
- 4.3 Moving Horizon Estimation (MHE) 20

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

1.1.1 Ingredients of an optimal control problem

1. Control system
2. Cost functional

Control system

$$\begin{matrix} u(t) \\ u(k) \end{matrix} \longrightarrow \boxed{\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t)\right) \\ x\left(k+1\right) &= f\left(x(k), u(k)\right) \end{aligned}} \longrightarrow \begin{matrix} x(t) \\ x(k) \end{matrix}$$

- Generates possible behaviours
- Described by ODEs of the form:

$$\dot{x}(t) = f\left(x(t), u(t)\right), \quad x(t_0) = x_0 \tag{1}$$

$$x\left(k+1\right) = f\left(x(k), u(k)\right) \tag{2}$$

$x = x(t)$	state	$x \in \mathbb{R}^n$
$u = u(t)$	control input	$u \in \mathbb{R}^q$
t	time	
t_0	initial time	
x_0	initial state	

Cost functional [1]

- Associates a cost with each possible behaviour
- For a given initial data (t_0, x_0), the behaviours are parametrised by control functions $u(t)$
- Assigns a cost value to each admissible control

$$J(u) := \int_{t_0}^{t_f} f_0\left(t, x(t), u(t)\right)\, dt + \phi\left(t_f, x_f\right) \tag{3}$$

J	- objective/cost functional
f_0	- running/infinitesimal/
	- incremental cost
ϕ	- terminal cost
t_f	- final/terminal time
$x_f := x(t_f)$	- final/terminal state

[1] Functional: maps a function to another function. In this case: J maps $u(t) \mapsto \int f_0(\ldots$

1.1.2 Optimisation problem (calculus)

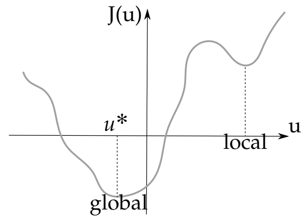
Optimisation problem

$$\min_u J(u) \triangleq \max_u -J(u) \tag{4}$$

$$\begin{aligned} \text{s.t.} \quad h_i(u) &= 0 & i &= 1 \ldots m \\ g_i &\leqslant 0 & j &= 1 \ldots M \end{aligned}$$

J	objective/cost function	$J : \mathbb{R}^n \to \mathbb{R}$
h_i	equality constraint	$h_i : \mathbb{R}^n \to \mathbb{R}$
	$h = \left[h_1 \quad \cdots h_m\right]^T$	
g_i	inequality constraint	$g_i : \mathbb{R}^n \to \mathbb{R}$

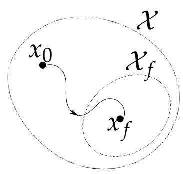
- Decision/optimisation variables (unknown):
  u, x, t_0, t_f
- Feasible/admissible set
  $\text{FS} = \{u \in \mathbb{R}^n : h_i(u) = 0, g_i(u) \leqslant 0\}$
  $$i = 1 \ldots m, \quad j = 1 \ldots M$$
- Solution: u^* is a (global) solution of
  $J^* = J(u^*) \leqslant J(u), \ \forall\, u \in \text{FS}$



Typical optimisation problem in conti. time

$$\min_{u(.),\, x(.)} \int_{t_0}^{t_f} f_0\left(t, x(t), u(t)\right)\, dt + \phi\left(t_f, x_f\right) \tag{5}$$

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



$\mathcal{X}_f$	terminal/target set	
$\mathcal{X}$	state constraints	$\mathcal{X} \subseteq \mathbb{R}^n$
$\mathcal{U}$	input constraints	$\mathcal{U} \subseteq \mathbb{R}^q$

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

$$\begin{aligned} J^* = J\left(x^*(.),\ u^*(.)\right) &\leqslant J\left(x(.),\ u(.)\right) \\ \Rightarrow u^*(.),\ x^*(.) &\text{ is a solution of } J^* \end{aligned}$$

Typical optimal control problem in discrete time

$$\min \sum_{k=0}^{N} f_0\left(k, x(k), u(k)\right) + \phi\left(x(N)\right) \qquad (6)$$

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

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

1.2 Topics/Tools

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

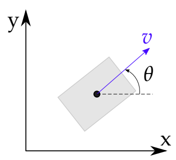
- Infinite vs. finite horizon
  $t_f = \infty \quad t_f < \infty$
  $N = \infty \quad N < \infty$
- Open loop vs. feedback
- Continuous time vs. discrete time
- Online (realtime) vs. offline computation

1.3 Examples

1.3.1 Shortest path problem

Robot unicycle model with constant velocity v

$$\begin{aligned} \dot{x} &= v \cos\theta \\ \dot{y} &= v \sin\theta \\ \dot{\theta} &= u \end{aligned}$$



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

1.3.2 Time-optimal control problem

t_f is the decision variable.

Time optimal control problem formulation

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

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

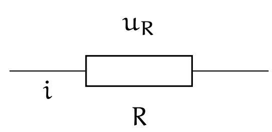
Finite horizon, time-optimal, open-loop OC problem

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
    (infinite horizon, closed loop)
    $$\min \int_0^{\infty} x^T(t) Q x(t) + u^T(t) R u(t) \, dt$$
  - The closed loop is asymptotically stable with respect to $x = 0$
- Q and R are positive definite matrices, i.e. their eigenvalues are positive
  $$\begin{aligned} Q &= Q^T > 0 \\ R &= R^T > 0 \end{aligned}$$

- 1D:
  $x \in \mathbb{R} \quad u \in \mathbb{R}$
  $Q \to q \quad R \to r$
  $q > 0 \quad r > 0$
  $$J = \int_0^{\infty} q x^2 + r u^2 \, dt$$

  1. Control energy
     $$\int_0^{\infty} r u^2 \, dt$$
     e.g.

     Input $u = i$
     $$\Rightarrow \quad P = Ri^2 = Ru^2$$
     $$\Rightarrow \quad E = \int Ru^2 dt$$
  2. Penalisation of deviation from x_0
     $$\int_0^{\infty} q x^2 \, dt$$

Mo. 29/10/18

2 Nonlinear programming approach

2.1 Nonlinear programming and optimal control

2.2 Problem setup

2.3 Unconstrained optimisation

a) Optimality condition

Fr. 02/11/18 no class

b) Algorithms

Mo. 05/11/18

2.4 Constrained optimisation

Fr. 09/11/18

- Equality constrains
- Inequality constraints

Mo. 12/11/18

2.5 NLP approach to optimal control

3 Dynamic programming

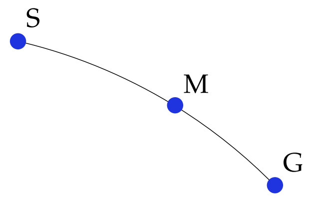
NLP	DP
finite horizon	(in)finite horizon
discrete time	discrete/continuous
open loop	feedback

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 → M, M → G
- recursive solution

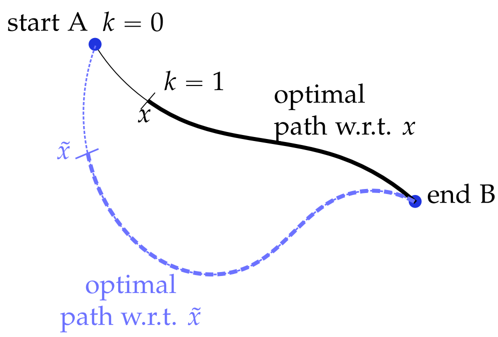
2. Find optimal route M → G given the/an optimal route S $\overset{M}{\to}$ G

Solution: "endpiece of optimal routes/trajectories are optimal" → **principle of optimality**

- endpiece of S $\overset{M}{\to}$ G is M → 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

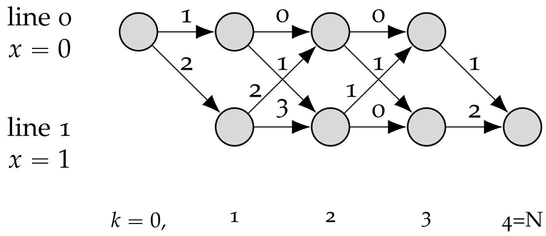
Assumption: cost sums up

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

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): $J(x, k)$

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

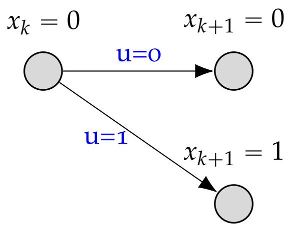
Complexity (count no. of +, -, min, max)

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

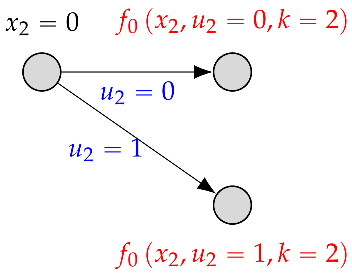
$\Sigma : \mathcal{O}\left(M^2 N\right)$

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

$$\min \sum_{k=k_0}^{N-1} f_0\left(k, x_k, u_k\right) + \phi\left(N, x_N\right) \tag{7}$$

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

$$J^*\left(k,x\right) = \min_{u_j \in \mathcal{U}\left(j, x_j\right)} \sum_{j=k}^{N-1} f_0\left(j, x_j, u_j\right) + \phi\left(N, x_N\right) \tag{8}$$

"optimal cost of endpiece"

Def: Optimal cost

$$J^*\left(k_0, x\right) = V\left(x\right) \tag{9}$$

"optimal value starting at point x"
optimal value function

Theorem Backward DP recursion
Let:

$$\bar{J}\left(N,x\right) = \phi\left(N,x\right) \quad \forall\, x \in \mathbb{R}^n \tag{10}$$
$$\bar{J}\left(k,x\right) = \min_{u \in \mathcal{U}\left(k,x\right)}\left\{f_0\left(k,x,u\right) + \bar{J}\left(k+1, f\left(x,u,k\right)\right)\right\} \tag{11}$$

"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

$$J^*\left(k,x\right) = \bar{J}\left(k,x\right) \tag{12}$$
$$u^*\left(k,x\right) = \bar{u}\left(k,x\right) \tag{13}$$

Proof By induction

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

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

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

$$\min \sum_{k=0}^{N-1} f_0(x_k,u_k,k) + \phi(x_N,N)$$
$$x_k \in \mathbb{R}^n$$
$$u_k \in \mathbb{R}^q$$
$$\text{s.t.} \quad x_{k+1} = f(k,x_k,u_k), \quad k = k_0 \dots (N-1)$$
$$u_k \in \mathcal{U}(x_k,k) \subseteq \mathbb{R}^q$$
$$x_0 = x$$

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

$$\bar{J}(x,N) = \phi(x,N) \quad \forall\ x \tag{14}$$
$$\bar{J}(x_k,k) = \min_{u_k \in \mathcal{U}(k,x_k)} \{f_0(x_k,u_k,k) + \bar{J}(f(k,x_k,u_k),k+1)\} \tag{15}$$
$$\bar{u}(x_k,k) = \arg\min \{f_0(x_k,u_k,k) + \bar{J}(f(k,x_k,u_k),k+1)\} \tag{16}$$

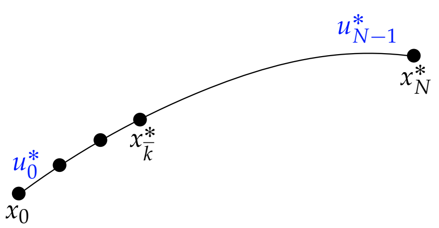
then

$$\bar{J}(k,x) = J^*(k,x) \,\forall\, X, k = k_0 \dots N-1 \tag{17}$$
$$\bar{u}(x,k) = \text{optimal feedback} \tag{18}$$

- (+) quite general, feedback, constructive (algorithm)
- (-) computational complexity (solve a NLP for each $x \in \mathbb{R}^n$)
  → discretisation → 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 ≙ 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
→ problem on a graph
(no. of states is finite, production line problem)

LQR – analytic solution

$$\min \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T S_N x_N,$$
$$R > 0, S_N \geqslant 0$$
$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k = f(x_k,u_k)$$

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: $(k = N-1)$

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

$$\frac{\partial}{\partial u} \{\} = 2\left(R + B^T S_N B\right) u + 2B^T S_N A x \stackrel{!}{=} 0$$
$$u^*(x) = -\left(R + R^T S_N B\right)^{-1} B^T S_N A x$$
$$u^*(x_{N-1}) = -K_{N-1} x_{N-1}$$

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

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:

$$\begin{aligned} S_{i-1} &= Q + K_{i-1}^T R K_{i-1} \\ &\quad + (A - BK_i)^T S_i (A - BK_i) \quad (19) \\ u^*(i-1) &= -K_{i-1} x_{i-1} \\ &= -\left(R + B^T S_i B\right)^{-1} B^T S_i A x_{i-1} \quad (20) \end{aligned}$$

$$J(i-1, x_{i-1}) = x_{i-1}^T S_{i-1} x_{i-1} \quad (21)$$

Ex. 3.1 Forward DP recursion

$$\min \sum_{k=0}^{N-1} f_0(x_{k+1}, u_k) + \underbrace{\phi(x_0)}_{\text{note}^2}$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

$$\begin{aligned} &J_a(k+1, x_{k+1}) \quad (22) \\ &= \min_u \{f_0(x_{k+1}, u_k) + J_a(k, x_k)\} \quad (23) \end{aligned}$$

J_a - arrival cost
- (summarises "past cost")

$J_a(0, x)$ - optimal cost

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

Why correct? time transformation

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

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

$$\sum_{k=0}^{N-1} f_0(x_{k+1}, u_k) + \phi(x_0)$$
$$= \sum_{\bar{k}=1}^{N} f_0\left(\xi_{\bar{k}-1}, \nu_{\bar{k}-1}\right) + \phi(\xi_N)$$

where $\xi_{\bar{k}-1} = f(\xi_{\bar{k}}, \nu_{\bar{k}})$

$\xi_0 = x_N = f(x_{N-1}, u_{N-1})$

$$= \sum_{\tilde{k}=0}^{N-1} f_0(\xi_{\tilde{k}}, \nu_{\tilde{k}}) + \phi(\xi_N)$$

where $\xi_{\tilde{k}-1} = f\left(\xi_{\tilde{k}+1}, \nu_{\tilde{k}}\right)$

$g(\xi_k, \xi_{k+1}, \nu_k) = 0$

Ex. 3.2 Backward DP recursion

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

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

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

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

...compare with (23)

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

3.3 Discrete time, infinite horizon DP

3.3.1 Problem formulation

Ex. 3.3 Problem setup

$$\min \sum_{k=0}^{N=\infty} f_0(x_k, u_k) \quad (24)$$

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

(Optimal) cost to go

$$J(k, x_k) = \min \sum_{j=k}^{\infty} f_0(x_j, u_j) \quad (25)$$

Value function

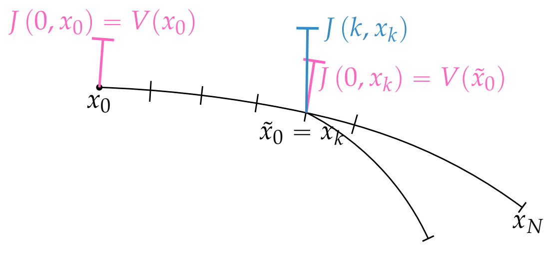
$$V(x) = J(0, x) \quad (26)$$

Remark: 'Cannot go backwards starting from infinity!'

Mo. 26/11/18

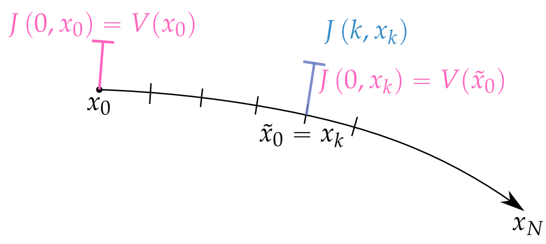
- N finite ($N < \infty$):

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



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

- 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

$$V(x) = \min_{u\in\mathcal{U}(x)} \{f_0(x,u) + V(f(x,u))\} \quad (27)$$

Theorem: (necessity)

Consider:

$$\overline{V}(x) = \min_{u\in\mathcal{U}(x)} \{f_0(x,u) + \overline{V}(f(x,u))\} \quad (28)$$

$$\overline{u}(x) = \arg\min \{f_0(x,u) + \overline{V}(f(x,u))\} \quad (29)$$

Suppose V in problem setup exists, then V solves $\overline{V}(x)$ in (28) (and the optimal feedback solves $\overline{u}(x)$ in (29))

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

$$\min \sum_{k=0}^{N=\infty} \alpha^k f_0(x_k,u_k), \quad \alpha \in (0,1) \quad (30)$$

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

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

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

Ex. 3.4 DP operator T

$$TV(.) := \min_{u\in\mathcal{U}(.)} \{f_0(.,u) + \alpha V(f(.,u))\} \quad (31)$$

$$\begin{array}{c}V(.)\\ \text{function}\end{array} \longrightarrow \boxed{T} \longrightarrow \begin{array}{c}TV(.)\\ \text{function}\end{array}$$

$$T_{w(x)}V(x) := \{f_0(x,w(x)) + \alpha V(f(x,w(x)))\} \quad (32)$$

Properties Let V_1, V_2 be some functions

contribution:

$$\|TV_1 - TV_2\|_\infty \leqslant \alpha \|V_1 - V_2\|_\infty \quad (33)$$

monotonity:

$$V_1 \geqslant V_2 \Rightarrow TV_1 \geqslant TV_2 \quad (34)$$

$$\exists \overline{V} : T\overline{V} = \overline{V} \quad (35)$$

iteration converges to unique fix point:

$$T\ldots T(TV_1) =: T^\infty V_1 = \overline{V} \quad (36)$$

Remark

$$\overline{V} = T\overline{V} \quad (\hat{=} \text{ DP equations})$$
$$T^{\infty} V_1 = \overline{V}$$
$$\left(V_2 := TV_1 \dots V_{k+1} = TV_k \overset{k\to\infty}{\longrightarrow} \overline{V}\right), \text{ note}^3$$

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

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

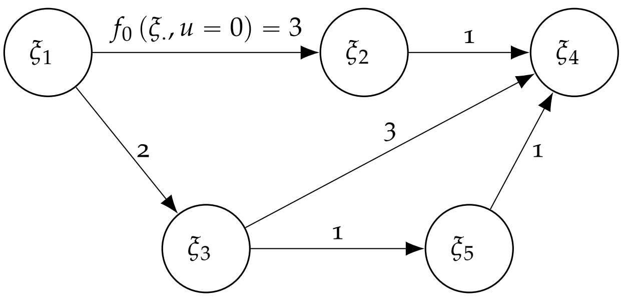
Value function iteration

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

Example

$\mathcal{X} = \{\xi_1, \dots \xi_5\}$ state space
$\mathcal{U} = \{0, 1\}$ input space

Dynamics:



$$V_1 = V(\xi_1)$$

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

Fr.
30/11/18

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

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

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

$$\begin{aligned} V(\xi_1^{\text{new}}) &= V_1^{\text{new }(k+1)} \\ &= \min \left\{ f_0(\xi_1, 0) + 0.9 \underbrace{V_2^{\text{old}}}_{\text{note}^4}, 2 + 0.9 V_3^{\text{old}} \right\} \\ &\vdots \\ V_2^{\text{new}} &= \\ &\vdots \\ V_5^{\text{new}} &= \end{aligned}$$

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

---

[3] value function iteration algorithm

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

3.3.3 Summary

Discrete time DP

$$\min \sum_{k=0}^{N-1} \alpha^k f_0(x_k, u_k) + \phi(x_N) \tag{37}$$

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

For $N < \infty$: DP recursion

$$J(k, x_k)$$
$$= \min_{u_k \in \mathcal{U}(x_k)} \left\{ \alpha^k f_0(x_k, u_k) + \underbrace{J(k+1, x_{k+1})}_{\text{cost to go}} \right\} \tag{38}$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \quad k = 0 \dots N$$

$$J(N, x) = \phi(x) \tag{39}$$

- Necessary and sufficient optimal condition

$$\text{opt FB } \overline{u}(k, x_k)$$
$$= \arg\min \left\{ \alpha^k f_0(x_k, u_k) + J(k+1, x_{k+1}) \right\} \tag{40}$$

- Algorithms: discretisation (Approximated Dynamic Programming / ADP algorithm)
  → curse of dimensionality

For $N = \infty$ ($\phi = 0$): 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

$$\text{opt FB } \overline{u}(x_k)$$
$$= \arg\min \{ f_0(x_k, u_k) + \alpha V(x_{k+1}) \} \tag{41}$$

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

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

$$x_{k+1} = (1 - \lambda_k) x_k + \lambda_k T(x_k) \xrightarrow{k \to \infty} x^* \tag{42}$$

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

3.4 Continuous time, finite horizon DP (HJB theory)

HJB: Hamilton-Jacobi-Bellman

3.4.1 Problem formulation

Ex. 3.5 Problem setup

$$\min_{u(.)} \int_{t_0}^{t_f} f_0(t, x, u)\ dt + \phi\left(t_f, x_f\right) \tag{43}$$

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

- Optimal cost to go

$$J^*(t_1, x_1) = \min_{u(.)} \int_{t_1}^{t_f} f_0(t, x, u)\ dt + \phi\left(x_f\right) \tag{44}$$

- Value function

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

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

PoO as a formula:

$$J(t_0, x_0) = \min_u \left\{ \underbrace{\int_{t_0}^{t_1} f_0(t, x, u)\ dt}_{\text{note}^5} + J(t_1, x_1) \right\}$$

[5] '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:

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

$$\begin{aligned} & J(t_0, x_0) \\ &= \min_{u(.)} \left\{ \int_{t_0}^{t_0+\Delta t} f_0(t,x,u)\ dt + \underbrace{J(t_0+\Delta t, x(t_0+\Delta t))}_{g(t_0+\Delta t),\ \text{note}^6} \right\} \\ &= \min_{u(.)} \{ f_0(t_0,x_0,u_0)\Delta t + J(t_0,x_0,u_0) \\ &\quad + \left[\frac{\partial J(t_0,x_0,u_0)}{\partial t} + \frac{\partial J(t_0,x_0,u_0)}{\partial x} f(t_0,x_0,u_0) \right] \Delta t \\ &\quad + \mathcal{O}(\Delta t^2) \Big\} \end{aligned}$$

Divide by Δt:

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

$\Delta t \to 0$:

$$\begin{aligned} & -\frac{\partial J(t_0,x_0)}{\partial t} \\ &= \min_{u(.)} \left\{ f_0(t_0,x_0,u_0) + \frac{\partial J(t_0,x_0,u_0)}{\partial x} f(t_0,x_0,u_0) \right\} \end{aligned} \tag{45}$$

3.4.3 Hamilton-Jacobi-Bellman equation

PDE, nonlinear[7]

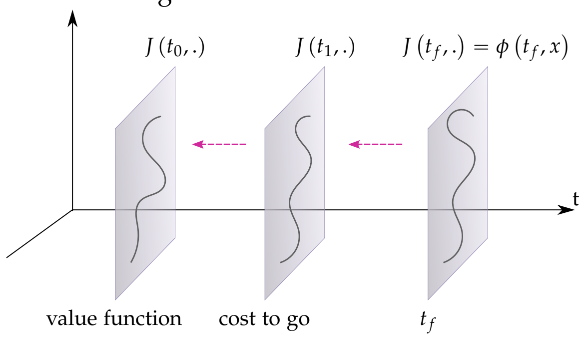
$$\begin{aligned} & -\frac{\partial J(t_0,x_0)}{\partial t} \\ &= \min_{u(.)} \left\{ f_0(t_0,x_0,u_0) + \frac{\partial J(t_0,x_0,u_0)}{\partial x} f(t_0,x_0,u_0) \right\} \end{aligned} \tag{45}$$

$$J(t_f, x) = \phi(t_f, x) \tag{46}$$

[6] $g(t_0 + \Delta t) = g(t_0) + \dot{g}(t_0)\Delta t + \mathcal{O}(\Delta t^2)$

[7] nonlinear due to min ...

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 (45).

THEOREM: sufficiency, verification theorem

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

$$-\frac{\partial \bar{J}(t,x)}{\partial t} = \min_{u \in \mathcal{U}(x)} \left\{ f_0(t,x,u) + \frac{\partial \bar{J}(t,x,u)}{\partial x} f(t,x,u) \right\} \tag{45}$$

$$J(t_f, x) = \phi(t_f, x) \tag{46}$$

$$\bar{u}(t,x) = \arg\min \left\{ f_0(t,x,u) + \frac{\partial \bar{J}(t,x,u)}{\partial x} f(t,x,u) \right\} \tag{47}$$

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 $\bar{J} \geqslant J$

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

Integrate from t_0 to t_f:

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

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

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

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

Integrate from t_0 to t_f:

$$\begin{aligned}\bar{J}(t_0, x_0) &\leqslant \int_{t_0}^{t_f} f_0(t,x,u)\ dt + \bar{J}\left(t_f, x_f\right) \\ \bar{J}(t_0, x_0) &\leqslant \int_{t_0}^{t_f} f_0(t,x,u)\ dt + \phi\left(t_f, x_f\right) \\ &\qquad \forall\ u(t,x) \in \mathcal{U}(t,x), \text{note}^8 \\ \bar{J}(t_0, x_0) &\leqslant J(t_0, x_0)\end{aligned}$$

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

Remarks

- Solving HJB (45) is in general hard (analytically + computationally[9])
- Sometimes helpful:

$$\text{opt FB } = \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right) \tag{48}$$

$$\bar{u}(t,x,\lambda) = \arg\min\left\{f_0(t,x,u) + \lambda^T f(t,x,u)\right\} \tag{49}$$

$$\begin{aligned}-\frac{\partial \bar{J}}{\partial t} &= f_0\left(t, x, \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right)\right) \\ &\quad + \underbrace{\frac{\partial \bar{J}(t,x)}{\partial x}}_{\lambda^T} f\left(t, x, \bar{u}\left(t, x, \frac{\partial \bar{J}}{\partial x}\right)\right)\end{aligned} \tag{50}$$

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 $\bar{u}(t,x)$

$$\bar{u}(t,x) = \arg\min\left\{x^T Q X + u^T R u + \lambda^T (Ax + Bu)\right\} \tag{49}$$

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

$$\begin{aligned}-\frac{\partial \bar{J}}{\partial t} &= x^T Q x + \frac{1}{4}\lambda^T B R^{-1} B^T \lambda + \lambda^T A x \\ &\quad + \lambda^T B(-\frac{1}{2} R^{-1} B^T \lambda) \qquad (50) \\ &= x^T Q x - \frac{1}{4}\lambda^T B R^{-1} B^T \lambda + \lambda^T A x \qquad (51)\end{aligned}$$

$$\bar{J}\left(t_f, x\right) = x^T S_F x \tag{46}$$

$$\begin{aligned}\text{We try:} \quad \bar{J}(t,x) &= x^T S(t) x, \text{ with} \\ \frac{\partial \bar{J}}{\partial t} &= x^T \dot{S}(t) x \\ \frac{\partial \bar{J}}{\partial x} &= 2x^T S(t) = \lambda^T \\ &\to \text{ plug into } -\frac{\partial \bar{J}}{\partial t} \text{ in (51)}\end{aligned}$$

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

Ricatti equation (ODE)[10]:

$$0 = \dot{S}(t) + Q - S(t) B R^{-1} B^T S(t) + 2S(t) A \tag{52}$$

$$S(t_f) = S_f \tag{53}$$

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

Mo. 03/12/18

3.5 Continuous time infinite horizon DP

Ex. 3.6 Problem setup

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

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

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

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

[10] PDE is converted into an ODE

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., $y = f_0(x(t),0) \to 0 \Rightarrow x(t) = 0$
- $f_0(0,0) = 0$

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 $\hat{=}$ 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

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

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

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

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

$$\overline{V}(x_0) \leqslant \int_0^{\infty} f_0(x,u)\, d\tau \quad \forall\ u(x) \in \mathcal{U}(x), \text{n}^{11}$$

- $\overline{V}(x_0)$ 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:

$$0 = \min \Big\{ q(x) + u^T R(x) u + \frac{\partial \overline{V}(x)}{\partial x} f(x) + \frac{\partial \overline{V}(x)}{\partial x} G(x) u \Big\} \tag{54}$$

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

$$\overline{u}(x) = -\frac{1}{2} R^{-1}(x) G^T(x) \frac{\partial \overline{V}^T(x)}{\partial x} \tag{55}$$

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

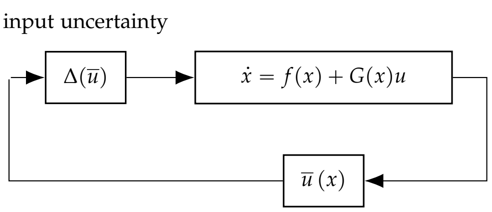
'$L_g V$' control law $\to$ nonlinear control

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

Substitute (55) in (54)

$$
\begin{aligned}
0 &= q(x) + \frac{1}{4}\frac{\partial V(x)}{\partial x}G(x)R(x)G^T(x)\frac{\partial V^T(x)}{\partial x} \\
&\quad + \frac{\partial V(x)}{\partial x}f(x) - \frac{1}{2}\frac{\partial V(x)}{\partial x}G(x)R^{-1}(x)G^T(x)\frac{\partial V^T(x)}{\partial x} \\
0 &= q(x) - \frac{1}{4}\frac{\partial V(x)}{\partial x}G(x)R^{-1}(x)G^T(x)\frac{\partial V^T(x)}{\partial x} \\
&\quad + \frac{\partial V(x)}{\partial x}f(x) \\
0 &= q(x) + \frac{1}{2}\frac{\partial V(x)}{\partial x}G(x)\overline{u}(x) + \frac{\partial V(x)}{\partial x}f(x)
\end{aligned}
$$

Uncertainties e.g. actuator nonlinearities (unmodelled)



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

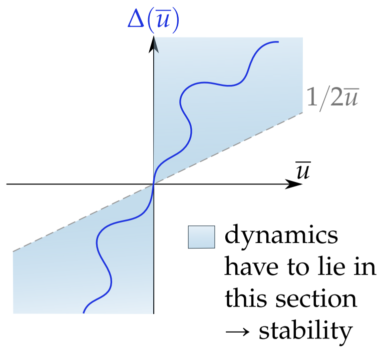
$$
\begin{aligned}
(\dot{\overline{V}} =)\dot{V}(x) &= \frac{\partial V(x)}{\partial x}\left(f(x) + G(x\Delta(\overline{u})\right) \overset{!}{\leqslant} 0 \\
\dot{V}(x) &= \frac{\partial V(x)}{\partial x}f(x) + \frac{1}{2}\frac{\partial V(x)}{\partial x}G(x)\overline{u}(x) \\
&\quad - \frac{1}{2}\frac{\partial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

Asymptotically stable if $* \leqslant 0$:

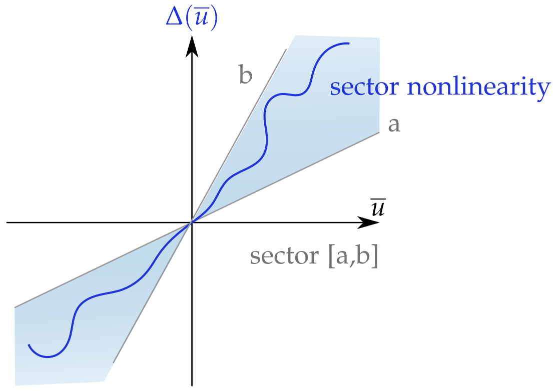
$$
\begin{aligned}
\underbrace{\frac{\partial V(x)}{\partial x}G(x)}_{-2\overline{u}^T R}\left(\Delta(\overline{u}) - \frac{1}{2}\overline{u}(x)\right) &\leqslant 0 \\
-2\overline{u}^T(x)R(x)\left(\Delta(\overline{u}) - \frac{1}{2}\overline{u}(x)\right) &\overset{?}{\leqslant} 0 \\
\overline{u}^T(x)R(x)\left(\Delta(\overline{u}) - \frac{1}{2}\overline{u}(x)\right) &\overset{?}{\geqslant} 0
\end{aligned}
$$

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

$$
\begin{aligned}
\overline{u}\cdot\left(\Delta(\overline{u}) - \frac{1}{2}\overline{u}(x)\right) &\geqslant 0 \\
\overline{u}\Delta(\overline{u}) - \frac{1}{2}\overline{u}^2 &\geqslant 0
\end{aligned}
$$



Remark:



$q > 1$

THEOREM: (similar to $q = 1$)

Suppose

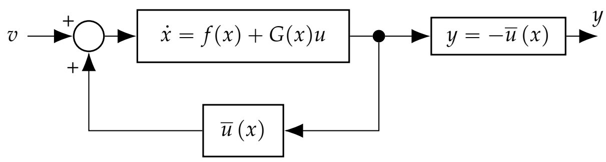
$$
R(x) = \begin{bmatrix} r_x(1) & & 0 \\ & \ddots & \\ 0 & & r_q(x) \end{bmatrix} \tag{56}
$$

$$
\Delta(\overline{u}) = \begin{bmatrix} \Delta_1(\overline{u}_1 \\ \vdots \\ \Delta_q(\overline{u}_q \end{bmatrix} \tag{57}
$$

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

Remark:

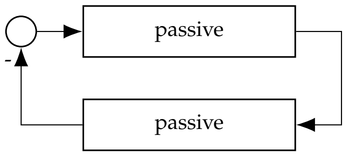
- LQR gain margin $[1/2, \infty]$
- disc margin (as a generalisation of phase margin)
  → nonlinear control



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

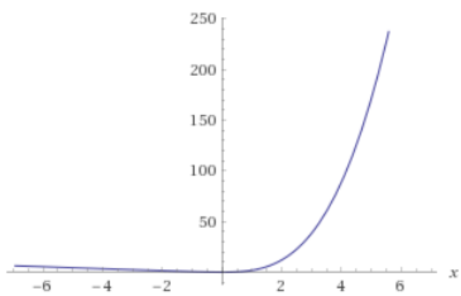
$$\begin{aligned} \dot{x} &= x^2 + u \\ u &= -x^2 - x \Rightarrow \text{ closed loop} \\ \Rightarrow \dot{x} &= -x \quad \text{FB linearisation} \end{aligned}$$

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

$\rightarrow$ closed loop is not globally stable – even worse, shows finite escape behaviour[12], even for small uncertainties.

Integrating the HJBE:

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



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

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) $\rightarrow$ $\max_w$

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

HJBIE[13]:

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

(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

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

[13] 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

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

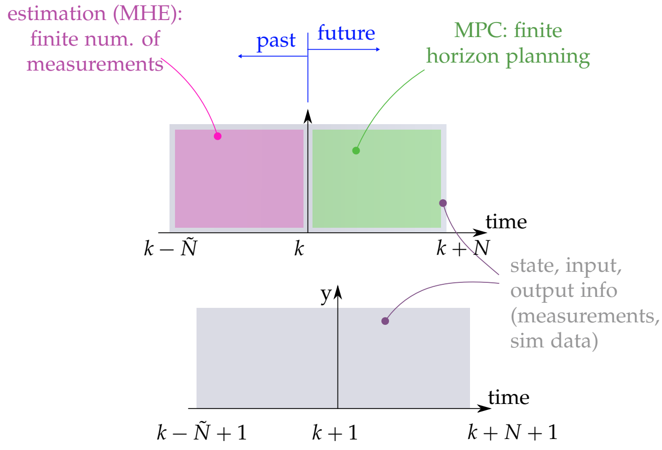
solve PDE, fixpoint equation, NLP $:\rightarrow \infty$ many decision variables (constraints)

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

4.1 Receding Horizon Online Decision Making

Decision making $\mathrel{\hat{=}}$ 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)[14]

4.2 MPC

- "online (repeated) open loop computations $\mathrel{\hat{=}}$ feedback"
- repeated open-loop finite-horizon OC policy (implemented in a receding horizon fashion)

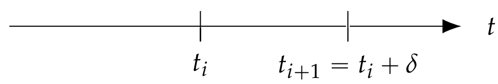
Ex. 4.7 1967, Marcus, Lee: Foundation of OC

"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



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

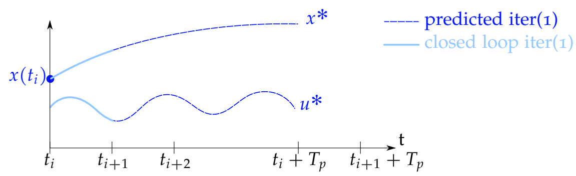
T_p - prediction/planning horizon
δ - sampling interval

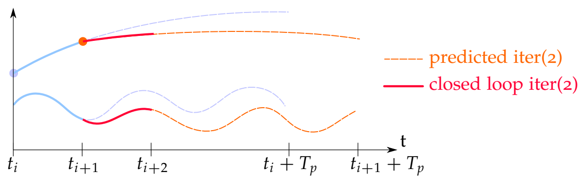
Typical scheme:

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

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

3. Implement $u\left(t, x\left(t_i\right)\right)$ 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 $i \leftarrow i+1$

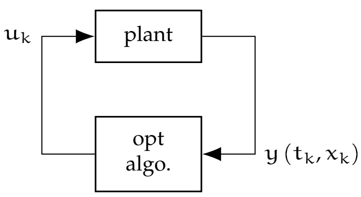




$\vdots$

$\Rightarrow$ predicted solution $\neq$ closed loop solution
$u_{\text{MPC}}\left(t, x\left(t_i\right)\right)$

Discussion / Challenges
MPC $\subseteq$ (online) optimisation-based control



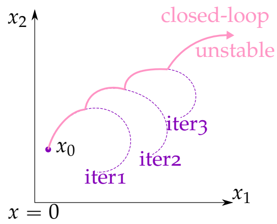
Fast and reliable (feasible) algorithms needed[15]!

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

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

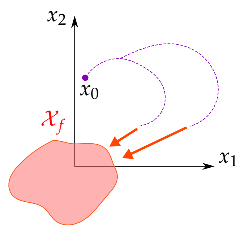


Mo. 10/12/18 exercise

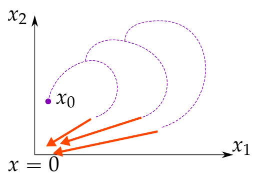
Solution: "change/modify objective function, constraints, prediction horizons ..."

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

$$V\left(x(t_i)\right) = \min \int_{t_i}^{t_i+T_p} f_0\left(x(t), u(t)\right)\, dt + \phi\left(x\left(t_i+T_p\right)\right)$$

$$\begin{aligned} \text{s.t.}\quad & \dot{x}(t) = f\left(x(t), u(t)\right), \\ & \quad t \in [t_i, t_i + T_p] \\ & u(t) \in \mathcal{U} \\ & x(t) \in \mathcal{X} \\ & \text{IC } x(t_i) \\ & x\left(t_i + T_p\right) \in \mathcal{X}_f \end{aligned}$$

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. $\dot{x} = f\left(x, k(x)\right)$ renders $\mathcal{X}_f$ invariant, i.e.

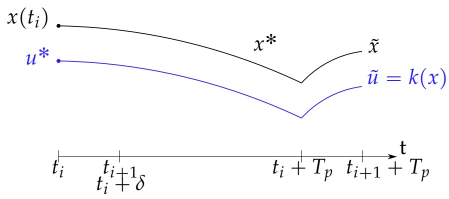
$$\begin{aligned} x(0) &\in \mathcal{X}_f \\ x(t) &\in \mathcal{X}_f, t > 0 \end{aligned}$$

A3.

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

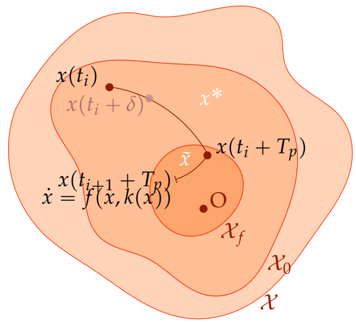
$\Rightarrow$ Then the MPC scheme is recursively feasible and converges $x(t_i) \to 0$, $i \to \infty$ (and the MPC closed loop is asymptotically stable for $\delta \to \infty$)

Proof



a) Recursive feasibility
b) Convergence

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



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

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

We have:

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

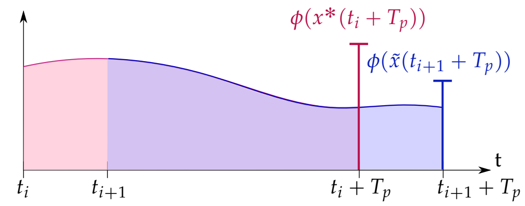
$\Rightarrow$ recursive feasibility
(state constraints ok, since $\mathcal{X}_f$ is invariant and $\mathcal{X}_f \subseteq \mathcal{X}$)

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

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

$$V\left(x\left(t_i\right)\right) = \int_{t_i}^{t_i+T_p} f_0\left(x^*, u^*\right)\, dt + \phi\left(x^*\left(t_i + T_p\right)\right)$$

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



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

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

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

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

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

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

Remark

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

Stability proof see exercise

$$x_k \in \mathcal{X}_f \Rightarrow x_{k+1} \in \mathcal{X}_f \tag{58}$$
$$x_{k+1} = f(x_k, \bar{k}(x_k))$$
$$x_k \in \mathcal{X}_f \Rightarrow \bar{k}(x_k) \in \mathcal{U} \tag{59}$$
$$\phi(x_{k+1}) - \phi(x_k) \leqslant -f_0(x_k, \bar{k}(x_k)) \tag{60}$$

$\Rightarrow$ 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