

# Contents

<b>1</b>	<b>Introduction to optimal control</b>	<b>3</b>
1.1	Optimal control problem formulation . . . . .	3
1.1.1	Optimization . . . . .	4
1.1.2	Discrete-time optimal control . . . . .	4
<b>2</b>	<b>Nonlinear Optimization</b>	<b>7</b>
2.1	Unconstrained Optimization . . . . .	7
2.1.1	Conditions of optimality . . . . .	8
2.1.2	Minimization of convex functions . . . . .	8
2.1.3	Quadratic programming . . . . .	9
2.2	Unconstrained Optimization Algorithms . . . . .	9
2.2.1	Iterative descent methods . . . . .	9
2.2.2	Gradient methods . . . . .	10
2.2.3	gradient method . . . . .	10
2.2.4	Newton's method for root finding . . . . .	11
2.2.5	Newton's method for unconstrained optimization . . . . .	11
2.2.6	Gradient methods via quadratic optimization . . . . .	12
2.2.7	step-size selection rules . . . . .	12
2.3	Constrained optimization over convex sets . . . . .	13
2.3.1	Projected gradient method . . . . .	14
2.3.2	Feasible direction method . . . . .	14
2.4	Constrained optimization (equality and inequality constraints): optimality conditions . . . . .	14
2.4.1	Quadratic programming (constrained) . . . . .	15
2.5	Constrained optimization (equality and inequality constraints): optimization algorithms . . . . .	16
2.5.1	Newton's method for equality constrained problems . . . . .	16
2.5.2	Sequential Quadratic Programming (SQP) . . . . .	17
2.5.3	Barrier function strategy for inequality constraints . . . . .	17
<b>3</b>	<b>Optimality conditions for optimal control</b>	<b>19</b>
3.1	boh . . . . .	19
3.1.1	Dynamics as equality constraints . . . . .	19
3.1.2	system trajectories and trajectory manifold . . . . .	19
3.2	Unconstrained optimal control problem (d-t) . . . . .	19
3.3	KKT conditions for unconstrained optimal control . . . . .	20
3.3.1	Indirect methods for optimal control . . . . .	21
3.4	KKT conditions for constrained optimal control . . . . .	21
<b>4</b>	<b>Linear Quadratic (LQ) optimal control</b>	<b>23</b>
4.1	First order optimality condition . . . . .	23
4.2	Infinite horizon LQ optimal control . . . . .	25

<b>5</b>	<b>Optimality Conditions for Unconstrained Optimal Control via Shooting</b>	<b>27</b>
5.1	Reduced optimal control problem . . . . .	27
5.2	Algorithms for optimal control problem solution . . . . .	28
5.2.1	First order necessary condition for optimality . . . . .	29
5.2.2	explicit computation of . . . . .	29
<b>6</b>	<b>Optimal Control based trajectory generation and tracking</b>	<b>31</b>
6.1	main strategy idea over a finite horizon . . . . .	31
6.2	LQR based trajectory tracking . . . . .	31
6.3	Affine LQR for trajectory tracking . . . . .	32
<b>7</b>	<b>Dynamic Programming</b>	<b>33</b>
7.1	Dynamic programming Recursion . . . . .	33
7.1.1	Optimal Control Policy and Trajectory . . . . .	33
7.1.2	DP Advantages and Limitations . . . . .	34
7.1.3	Linear Quadratic Optimal Control via DP . . . . .	34
7.1.4	DP for LQP – Summary . . . . .	35
7.1.5	LQ for time-invariant systems and cost . . . . .	35
7.2	Infinite Horizon Linear Quadratic Problems . . . . .	35
<b>8</b>	<b>Numerical methods for nonlinear optimal control</b>	<b>37</b>
<b>9</b>	<b>Optimization-based predictive control</b>	<b>39</b>

# Chapter 1

## Introduction to optimal control

### 1.1 Optimal control problem formulation

Consider the continuous-time system ( $t \in \mathbb{R}$ )

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= h(x(t), u(t), t)\end{aligned}\tag{1.1}$$

- $x(t) \in \mathbb{R}^n$  state of the system at time  $t$
- $u(t) \in \mathbb{R}^m$  input of the system at time  $t$
- $y(t) \in \mathbb{R}^p$  output of the system at time  $t$

We will mainly work with time invariant systems,  $\dot{x}(t) = f(x(t), u(t))$ .

We consider nonlinear, discrete-time systems described by

$$x(t+1) = f_t(x(t), u(t)) \quad t \in \mathbb{N}_0$$

but from now on we will use the compact notation

$$x_{t+1} = f_t(x_t, u_t) \quad t \in \mathbb{N}_0$$

where  $x_t \in \mathbb{R}^n$  and  $u_t \in \mathbb{R}^m$  are the state and the input of the system at time  $t$ .

Consider a nonlinear, discrete-time system on a finite time horizon

$$x_{t+1} = f_t(x_t, u_t) \quad t = 0, \dots, T-1$$

We use  $\mathbf{x} \in \mathbb{R}^{nT}$  and  $\mathbf{u} \in \mathbb{R}^{mT}$  to denote, respectively, the stack of the states  $x_t$  for all  $t \in \{1, \dots, T\}$  and the inputs  $u_t$  for all  $t \in \{0, \dots, T-1\}$ , that is:

$$\mathbf{x} := \begin{bmatrix} x_1 \\ \vdots \\ x_T \end{bmatrix} \quad \mathbf{u} := \begin{bmatrix} u_0 \\ \vdots \\ u_{T-1} \end{bmatrix}$$

#### Trajectory of a system

Definition: A pair  $(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT}$  is called a trajectory of system (1.1) if  $\bar{x}_{t+1} = f_t(\bar{x}_t, \bar{u}_t)$  for all  $t \in \{0, \dots, T-1\}$ . That is, if  $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$  satisfies the system dynamics (the same holds for continuous time systems with proper adjustments). In particular,  $\bar{\mathbf{x}}$  is the state trajectory, while  $\bar{\mathbf{u}}$  is the input trajectory.

#### Equilibrium

Definition: A state-input pair  $(x_e, u_e) \in \mathbb{R}^n \times \mathbb{R}^m$  is called an equilibrium pair of (1.1) if  $(x_t, u_t) = (x_e, u_e), t \in \mathbb{N}_0$  is a trajectory of the system.

Equilibria of time-invariant systems satisfy  $x_e = f(x_e, u_e)$

### Linearization of a system about a trajectory

Given the dynamics (1.1) and a trajectory  $(\bar{x}, \bar{u})$ , the linearization of (1) about  $(\bar{x}, \bar{u})$  is given by the linear (possibly) time-varying system

$$\Delta x_{t+1} = A_t \Delta x_t + B_t \Delta u_t \quad t \in \mathbb{N}_0$$

with  $A_t$  and  $B_t$  the Jacobians of  $f_t$ , with respect to state and input respectively, evaluated at  $(\bar{x}, \bar{u})$

$$A_t = \left. \frac{\partial}{\partial x} f(\bar{x}_t, \bar{u}_t) \right|_{(\bar{x}, \bar{u})} \quad B_t = \left. \frac{\partial}{\partial u} f(\bar{x}_t, \bar{u}_t) \right|_{(\bar{x}, \bar{u})}$$

### 1.1.1 Optimization

#### Main ingredients

- Decision variable:  $x \in \mathbb{R}^n$
- Cost function:  $\ell(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  cost associated to decision  $x$
- Constraints (constraint sets): for some given functions  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ , and  $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ , the decision vector  $x \in \mathbb{R}^n$  needs to satisfy

$$h_i(x) = 0 \quad i = 1, \dots, m$$

$$g_j(x) = 0 \quad j = 1, \dots, r$$

equivalently we can say that we require  $x \in X$  with

$$X = \{x \in \mathbb{R}^n | h(x) = 0, g(x) \leq 0\},$$

where we compactly denoted  $h(x) = \text{col}(h_1(x), \dots, h_m(x))$  and  $g(x) = \text{col}(g_1(x), \dots, g_r(x))$

#### Minimization

We can write our optimization problem as

$$\min_{x \in \mathbb{R}^n} \ell(x) \tag{1.2}$$

$$\text{subj. to } h_i(x) = 0 \quad i = 1, \dots, m \tag{1.3}$$

$$g_j(x) \leq 0 \quad j = 1, \dots, r \tag{1.4}$$

where  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$

We can write it more compactly as

$$\min_{x \in \mathbb{R}^n} \ell(x)$$

$$\text{subj. to } h(x) = 0, g(x) \leq 0$$

where  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^r$

### 1.1.2 Discrete-time optimal control

#### main ingredients

- Dynamics: a discrete-time system in state space form

$$x_{t+1} = f_t(x_t, u_t) \quad t = 0, 1, \dots, T-1$$

- the dynamics introduce  $T$  equality constraints

$$\begin{array}{ll} x_1 = f(x_0, u_0) & \text{i.e.} \quad x_1 - f(x_0, u_0) = 0 \\ x_2 = f(x_1, u_1) & \text{i.e.} \quad x_2 - f(x_1, u_1) = 0 \\ \vdots & \\ x_T = f(x_{T-1}, u_{T-1}) & \text{i.e.} \quad x_T - f(x_{T-1}, u_{T-1}) = 0 \end{array}$$

This is equivalent to  $nT$  scalar constraints

- Cost function: a cost "to be payed" for a chosen trajectory. We consider an additive structure in time

$$\ell(\mathbf{x}, \mathbf{u}) = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$$

where  $\ell_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  is called stage-cost, while  $\ell_T : \mathbb{R}^n \rightarrow \mathbb{R}$  is the terminal cost.

- End-point constraints: function of the state variable prescribed at initial and/or final point

$$r(x_0, x_T) = 0$$

- Path constraints: point-wise (in time) constraints representing possible limits on states and inputs at each time  $t$

$$g_t(x_t, u_t) \leq 0, \quad t \in \{0, \dots, T-1\}$$

A discrete-time optimal control problem can be written as

$$\begin{aligned} & \min_{\substack{x_0, x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \\ \text{subj. to } & x_{t+1} = f_t(x_t, u_t), \quad t \in \{0, \dots, T-1\} \\ & r(x_0, x_T) = 0 \\ & g_t(x_t, u_t) \leq 0, \quad t \in \{0, \dots, T-1\} \end{aligned}$$

### Optimal control for trajectory generation

We can pose a trajectory generation problem as

$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m} \sum_{t=0}^{T-1} \frac{1}{2} \|x_t - x_t^{\text{des}}\|_Q^2 + \frac{1}{2} \|u_t - u_t^{\text{des}}\|_R^2 + \frac{1}{2} \|x_T - x_T^{\text{des}}\|_{P_f}^2$$

### Continuous-time Optimal Control problem

A continuous-time optimal control problem, i.e.,  $t \in \mathbb{R}$  can be written as

$$\begin{aligned} & \min_{(x(\cdot), u(\cdot)) \in \mathcal{F}} \int_0^T \ell_\tau(x(\tau), u(\tau)) d\tau + \ell_T(x(T)) \\ \text{subj. to } & \dot{x}(t) = f_t(x(t), u(t)) \quad t \in [0, T] \\ & r(x(0), x(T)) = 0 \\ & g_t(x(t), u(t)) \leq 0 \quad t \in [0, T] \end{aligned}$$

Note that  $\mathcal{F}$  is a space of functions (function space). This is an infinite dimensional optimization problem

- Cost functional  $\ell : \mathcal{F} \rightarrow \mathbb{R}$

$$\ell(x(\cdot), u(\cdot)) = \int_0^T \ell_\tau(x(\tau), u(\tau)) d\tau + \ell_T(x(T))$$

- Space of trajectories (or trajectory manifold)

$$\mathcal{T} = \{(x(\cdot), u(\cdot)) \in \mathcal{F} | \dot{x}(t) = f_t(x(t), u(t)), t \geq 0\}$$



## Chapter 2

# Nonlinear Optimization

### 2.1 Unconstrained Optimization

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} \ell(x)$$

with  $\ell : \mathbb{R}^n \rightarrow \mathbb{R}$  a cost function to be minimized and  $x$  a decision vector

We say that  $x^*$  is a

- global minimum if  $\ell(x^*) \leq \ell(x)$  for all  $x \in \mathbb{R}^n$
- strict global minimum if  $\ell(x^*) < \ell(x)$  for all  $x \neq x^*$
- local minimum if there exists  $\epsilon > 0$  such that  $\ell(x^*) \leq \ell(x)$  for all  $x \in B(x^*, \epsilon) = \{x \in \mathbb{R}^n \mid \|x - x^*\| < \epsilon\}$
- strict local minimum if there exists  $\epsilon > 0$  such that  $\ell(x^*) < \ell(x)$  for all  $x \in B(x^*, \epsilon)$

#### Notation

We denote  $\ell(x^*)$  the optimal (minimum) value of a generic optimization problem, i.e.

$$\ell(x^*) = \min_{x \in \mathbb{R}^n} \ell(x)$$

where  $x^*$  is the minimum point (optimal value for the optimization variable) i.e.

$$x^* = \arg \min_{x \in \mathbb{R}^n} \ell(x)$$

#### Gradient and Hessian

Gradient of a function: for a function  $r : \mathbb{R}^n \rightarrow \mathbb{R}$  the gradient is denoted as

$$\nabla r(x) = \begin{bmatrix} \frac{\partial r(x)}{\partial x_1} \\ \vdots \\ \frac{\partial r(x)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

Hessian matrix of a function: for a function  $r : \mathbb{R}^n \rightarrow \mathbb{R}$  the Hessian matrix is denoted as

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

Gradient of a vector-valued function: for a vector field  $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , the gradient is denoted as

$$\nabla r(x) = \begin{bmatrix} \nabla r_1(x) & \cdots & \nabla r_m(x) \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1(x)}{\partial x_1} & \cdots & \frac{\partial r_m(x)}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_1(x)}{\partial x_n} & \cdots & \frac{\partial r_m(x)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

which is the transpose of the Jacobian matrix of  $r$

### 2.1.1 Conditions of optimality

#### First order necessary condition (FNC) of optimality (unconstrained)

Let  $x^*$  be an unconstrained local minimum of  $\ell : \mathbb{R}^n \rightarrow \mathbb{R}$  and assume that  $\ell$  is continuously differentiable ( $\mathcal{C}^1$ ) in  $B(x^*, \epsilon)$  for some  $\epsilon > 0$ . Then  $\nabla \ell(x^*) = 0$

#### Second order necessary condition (FNC) of optimality (unconstrained)

If additionally  $\ell$  is twice continuously differentiable ( $\mathcal{C}^2$ ) in  $B(x^*, \epsilon)$ , then  $\nabla^2 \ell(x^*) \geq 0$  (The Hessian of  $\ell$  is positive semidefinite)

#### Second order sufficient conditions of optimality (unconstrained)

Let  $\ell : \mathbb{R}^n \rightarrow \mathbb{R} \in \mathcal{C}^2$  in  $B(x^*, \epsilon)$  for some  $\epsilon > 0$ . Suppose that  $x^* \in \mathbb{R}^n$  satisfies

$$\nabla \ell(x^*) = 0 \text{ and } \nabla^2 \ell(x^*) > 0$$

Then  $x^*$  is a strict (unconstrained) local minimum of  $\ell$

#### Convex set

A set  $X \subset \mathbb{R}^n$  is convex if for any two points  $x_A$  and  $x_B$  in  $X$  and for all  $\lambda \in [0, 1]$ , then

$$\lambda x_A + (1 - \lambda)x_B \in X$$

#### Convex functions

Let  $X \subset \mathbb{R}^n$  be a convex set. A function  $\ell : X \rightarrow \mathbb{R}$  is convex if for any two points  $x_A$  and  $x_B$  in  $X$  and for all  $\lambda \in [0, 1]$ , then

$$\ell(\lambda x_A + (1 - \lambda)x_B) \leq \lambda \ell(x_A) + (1 - \lambda)\ell(x_B)$$

### 2.1.2 Minimization of convex functions

#### Proposition

Let  $X \subset \mathbb{R}^n$  be a convex set and  $\ell : X \rightarrow \mathbb{R}$  a convex function. Then a local minimum of  $\ell$  is also a global minimum

Proof: not done in class but present in slides for funsies

#### Necessary and sufficient condition of optimality (unconstrained)

For the unconstrained minimization of a convex function it can be shown that the first order necessary condition of optimality is also sufficient (for a global minimum).

#### Proposition

Let  $\ell_{\mathbb{R}}^n \rightarrow \mathbb{R}$  be a convex function. Then  $x^*$  is a global minimum if and only if  $\nabla \ell(x^*) = 0$

Proof: not done in class but present in slides for funsies



### 2.1.3 Quadratic programming

Let us consider a special class of optimization problems, namely quadratic optimization problems or quadratic programs:

$$\min_{x \in \mathbb{R}^n} x^T Q x + b^T x$$

with  $Q = Q^T \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$

#### optimality conditions

First-order necessary condition for optimality: if  $x^*$  is a minimum then

$$\nabla \ell(x^*) = 0 \implies 2Qx^* + b = 0$$

Second-order necessary condition for optimality: if  $x^*$  is a minimum then

$$\nabla^2 \ell(x^*) \geq 0 \implies 2Q \geq 0$$

A necessary condition for the existence of minima for a quadratic program is that  $Q \geq 0$ . Thus, quadratic programs admitting at least a minimum are convex optimization problems.

#### properties

Since quadratic programs are convex programs ( $Q \geq 0$  is necessary to have a local minimum), then the following holds:

- For a quadratic program necessary conditions of optimality are also sufficient and minima are global

If  $Q > 0$ , then there exists a unique global minimum given by

$$x^* = -\frac{1}{2}Q^{-1}b$$

## 2.2 Unconstrained Optimization Algorithms

### 2.2.1 Iterative descent methods

We consider optimization algorithms relying on the iterative descent idea. We denote  $x^k \in \mathbb{R}^n$  an estimate of a local minimum at iteration  $k \in \mathbb{N}$ . The algorithm starts at a given initial guess  $x^0$  and iteratively generates vectors  $x^1, x^2, \dots$  such that  $\ell$  is decreased at each iteration, i.e.

$$\ell(x^{k+1}) < \ell(x^k) \quad k = 1, 2, \dots$$

#### two-step procedure

We consider a general two-step procedure that reads as follows

$$x^{k+1} = x^k + \gamma^k d^k, \quad k = 1, 2, \dots$$

in which

1. each  $\gamma^k > 0$  is a "step-size"
2.  $d^k \in \mathbb{R}^n$  is a "direction"

The goal is to

1. choose a direction  $d^k$  along which the cost decreases for  $\gamma^k$  sufficiently small;
2. select a step-size  $\gamma^k$  guaranteeing a sufficient decrease.

In other references these are called line-search methods.

### 2.2.2 Gradient methods

Let  $x^k$  be such that  $\nabla\ell(x^k) \neq 0$ . We start by considering the update rule

$$x^{k+1} = x^k - \gamma^k \nabla\ell(x^k)$$

i.e., we choose  $d^k = \nabla\ell(x^k)$

From the first order Taylor expansion of  $\ell$  at  $x$  we have

$$\begin{aligned} \ell(x^{k+1}) &= \ell(x^k) + \nabla\ell(x^k)^T(x^{k+1} - x^k) + o(\|x^{k+1} - x^k\|) \\ &= \ell(x^k) - \gamma^k \|\nabla\ell(x^k)\|^2 + o(\gamma^k) \end{aligned}$$

Thus, for  $\gamma^k > 0$  sufficiently small it can be shown that  $\ell(x^{k+1}) < \ell(x^k)$

The update rule

$$x^{k+1} = x^k - \gamma^k \nabla\ell(x^k)$$

can be generalized to so called *gradient methods*

$$x^{k+1} = x^k + \gamma^k d^k$$

with  $d^k$  such that

$$\nabla\ell(x^k)^T d^k < 0$$

Also,  $d^k$  must be gradient related, i.e.  $d^k$  must not asymptotically become perpendicular to  $\nabla\ell$

#### selecting the descent direction

Several gradient methods can be written as

$$x^{k+1} = x^k - \gamma^k D^k \nabla\ell(x^k) \quad k = 1, 2, \dots$$

where  $D^k \in \mathbb{R}^{n \times n}$  is a symmetric positive definite matrix. It can be immediately seen that

$$-\nabla\ell(x^k)^T D^k \nabla\ell(x^k) < 0$$

i.e.  $d^k = -D^k \nabla\ell(x^k)$  is a descent direction. The choice of  $D^k$  must be made such that there exist  $d_1, d_2$  positive real, such that  $d_1 I \leq D^k \leq d_2 I$

Some choices for  $D^k$ :

- Steepest descent  $D^k = I_n$
- Newton's method  $D^k = (\nabla^2\ell(x^k))^{-1}$   
It can be used when  $\nabla^2\ell(x^k) > 0$ . It typically converges very fast asymptotically. For  $\gamma^k = 1$  pure Newton's method
- Discretized Newton's method  $D^k = (H(x^k))^{-1}$ , where  $H(x^k)$  is a positive definite symmetric approximation of  $\nabla^2\ell(x^k)$  obtained by using finite difference approximations of the second derivatives
- Some regularized version of the Hessian

### 2.2.3 gradient method

The update rule obtained for  $D^k = I$  is called steepest descent. The name steepest descent is due to the following property: the normalized negative gradient direction

$$d^k = -\frac{\nabla\ell(x^k)}{\|\nabla\ell(x^k)\|}$$

minimizes the slope  $\nabla\ell(x^k)^T d^k$  among all normalized directions, i.e. it gives the steepest descent.

### 2.2.4 Newton's method for root finding

Consider the nonlinear root finding problem

$$r(x) = 0$$

Idea: iteratively refine the solution such that the improved guess  $x^{k+1}$  represents a root of the linear approximation of  $r$  about the current tentative solution  $x^k$ . Consider the linear approximation of  $r$  about  $x^k$ , we have

$$r^k(x^k + \Delta x^k) = r(x^k) + \nabla r(x^k)^T \Delta x^k$$

then, finding the zeros of the approximation, we have

$$\Delta x^k = -(\nabla r(x^k)^T)^{-1} r(x^k)$$

Thus, the solution is improved as

$$x^{k+1} = x^k - (\nabla r(x^k)^T)^{-1} r(x^k)$$

### 2.2.5 Newton's method for unconstrained optimization

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} \ell(x)$$

stationary points  $\bar{x}$  satisfy the first order optimality condition

$$\nabla \ell(\bar{x}) = 0$$

We can look at it as a root finding problem, with  $r(x) = \nabla \ell(x)$ , and solve it via Newton's method. Therefore, we can compute  $\Delta x^k$  as the solution of the linearization of  $r(x) = \nabla \ell(x)$  at  $x^k$ , i.e.

$$\nabla \ell(x^k) + \nabla^2 \ell(x^k) \Delta x^k = 0$$

and run the update

$$x^{k+1} = x^k - (\nabla^2 \ell(x^k))^{-1} \nabla \ell(x^k)$$

We can introduce a variable step-size

$$x^{k+1} = x^k - \gamma^k (\nabla^2 \ell(x^k))^{-1} \nabla \ell(x^k)$$

This is called generalized Newton's method

#### Newton's method via Quadratic Optimization

Observe that

$$\nabla \ell(x^k) + \nabla^2 \ell(x^k) \Delta x^k = 0$$

is the first-order necessary and sufficient condition of optimality for the quadratic program

$$\Delta x^k = \arg \min_{\Delta x} \nabla \ell(x^k)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 \ell(x^k) \Delta x \quad (2.1)$$

Thus, the  $k$ -th iteration of Newton's method can be seen as

$$x^{k+1} = x^k + \Delta x^k$$

with  $\Delta x^k$  solution of the quadratic problem (2.1). Generalized version:

$$x^{k+1} = x^k + \gamma^k \Delta x^k$$

### 2.2.6 Gradient methods via quadratic optimization

Similarly to Newton's method, a descent direction  $\Delta x^k = D^k \nabla \ell(x^k)$  can be seen as the direction that minimizes at each iteration a different quadratic approximation of  $\ell$  about  $x^k$ . In fact, consider the quadratic approximation  $\ell^k(x)$  about  $x^k$  given by

$$\ell^k(x) = \ell(x^k) + \nabla \ell(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T (D^k)^{-1} (x - x^k)$$

By setting the derivative to zero, we have

$$\nabla \ell(x^k) + (D^k)^{-1} (x - x^k) = 0$$

we can calculate the minimum of  $\ell^k(x)$  and set it as the next iterate  $x^{k+1}$

$$\Delta x^k = -D^k \nabla \ell(x^k)$$

### 2.2.7 step-size selection rules

- Constant step-size:  $\gamma^k = \gamma > 0$
- Diminishing step-size:  $\gamma^k \rightarrow 0$  as  $k \rightarrow \infty$ . It must hold that

$$\sum_{k=0}^{\infty} \gamma^k = +\infty \quad \text{and} \quad \sum_{k=0}^{\infty} (\gamma^k)^2 < +\infty$$

The above conditions avoid pathological choices of  $\gamma^k$

- minimization rule
- Armijo rule

#### Armijo rule

Step-size is selected following the procedure:

1. set  $\bar{\gamma}^0 > 0$ ,  $\beta \in (0, 1)$ ,  $c \in (0, 1)$

given  $d^k$  descent direction we can consider

$$g^k(\gamma) = \ell(x^k + \gamma d^k), \quad g : \mathbb{R} \rightarrow \mathbb{R}$$

The value of  $g^k(\gamma)$  for  $\gamma = 0$  is  $\ell(x^k)$ . The minimization rule chooses as the value for  $\gamma$  the value that minimizes  $g^k(\gamma)$ . The partial minimization rule would search for a minimum in a restricted set of values for  $\gamma$ . Let us differentiate  $g$  wrt  $\gamma$ :

$$\begin{aligned} g'(\gamma) &= \frac{d}{d\gamma} g(\gamma) = \frac{d}{d\gamma} \ell(x^k + \gamma d^k) \\ g'(0) &= \frac{d}{d\gamma} \ell(x^k + \gamma d^k) \big|_{\gamma=0} = \nabla \ell(x^k)^T d^k \end{aligned}$$

We compute a linear approximation of  $g(\gamma)$ :

$$\begin{aligned} g(\gamma) &= g(0) + g'(0)\gamma + o(\gamma) \\ \ell(x^k + \gamma d^k) &= \ell(x^k) + \nabla \ell(x^k)^T d^k \gamma + o(\gamma) \end{aligned}$$

This is the tangent to the  $g(\gamma)$  curve at  $\gamma = 0$ . We also consider the line

$$\ell(x^k) + c\gamma \nabla \ell(x^k)^T d^k$$

which is a line with a slightly less negative slope given that  $c \in (0, 1)$  The Armijo rule is applied as follows:

1. Set  $\bar{\gamma}^0 > 0$ ,  $\beta \in (0, 1)$ ,  $c \in (0, 1)$
2. While  $\ell(x^k + \bar{\gamma}^1 d^k) \geq \ell(x^k) + c\bar{\gamma}^1 \nabla \ell(x^k)^T d^k$ :

$$\bar{\gamma}^{i+1} = \beta \bar{\gamma}^i$$

3. Set  $\gamma^k = \bar{\gamma}^i$

Typical values are  $\beta = 0.7$  and  $c = 0.5$

**Proposition: convergence with Armijo step-size**

Let  $\{x^k\}$  be a sequence generated by a gradient method  $x^{k+1} = x^k - \gamma^k D^k \nabla \ell(x^k)$  with  $d_1 I \leq D^k \leq d_2 I$ ,  $d_1, d_2 > 0$ . Assume that  $\gamma^k$  is chosen by the Armijo rule and  $\ell(x) \in \mathcal{C}^1$ . Then, every limit point  $\bar{x}$  of the sequence  $\{x^k\}$  is a stationary point, i.e.  $\nabla \ell(\bar{x}) = 0$

Recall that a vector  $x \in \mathbb{R}^n$  is a limit point of a sequence  $\{x^k\}$  in  $\mathbb{R}^n$  if there exists a subsequence of  $\{x^k\}$  that converges to  $x$ .

**convergene with constant or diminishing step-size**

Let  $\{x^k\}$  be a sequence generated by a gradient method  $x^{k+1} = x^k - \gamma^k D^k \nabla \ell(x^k)$  with  $d_1 I \leq D^k \leq d_2 I$ ,  $d_1, d_2 > 0$ . Assume that for some  $L > 0$

$$\|\nabla \ell(x) - \nabla \ell(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n$$

i.e. The gradient is a Lipschitz continuous function. Assume either

1.  $\gamma^k = \gamma > 0$  sufficiently small, or

2.  $\gamma^k \rightarrow 0$  and  $\sum_{t=0}^{\infty} \gamma^k = \infty$

Then, every limit point  $\bar{x}$  of the sequence  $\{x^k\}$  is a stationary point, i.e.  $\nabla \ell(\bar{x}) = 0$

**Remarks on gradient methods**

- The propositions do not guarantee that the sequence converges and not even existence of limit points. Then either  $\ell(x^k) \rightarrow -\infty$  or  $\ell(x^k)$  converges to a finite value and  $\nabla \ell(x^k) \rightarrow 0$ . In the second case, one can show that any subsequence  $\{x^{k_p}\}$  converges to some stationary point  $\bar{x}$  satisfying  $\nabla \ell(\bar{x}) = 0$
- Existence of minima can be guaranteed by excluding  $\ell(x^k) \rightarrow -\infty$  via suitable assumptions. Assume, e.g.,  $\ell$  coercive (radially unboundend)
- For general (nonconvex) problems, assuming coecivity, only convergence (of subsequences) to stationary points can be proven.
- for convex programs, assuming coercivity, convergence to global minima is guardanteed since necessary conditions of optimality are also sufficient.

## 2.3 Constrained optimization over convex sets

consider the optimization problem

$$\min_{x \in X} \ell(x)$$

where  $X \subset \mathbb{R}^n$  is nonempty, convex, and closed, and  $\ell$  is continuously differentiable on  $X$ .

**Optimality conditions**

If a point  $x^* \in X$  is a local minimum of  $\ell(x)$  over  $X$ , then

$$\nabla \ell(x^*)^T (\bar{x} - x^*) \geq 0 \quad \forall \bar{x} \in X$$

### Projection over a convex set

Given a point  $x \in \mathbb{R}^n$  and a closed convex set  $X$ , it can be shown that

$$P_X(x) := \arg \min_{z \in X} \|z - x\|^2$$

exists and is unique. The point  $P_X(x)$  is called the projection of  $x$  on  $X$ .

### 2.3.1 Projected gradient method

Gradient methods can be generalized to optimization over convex sets

$$x^{k+1} = P_X(x^k - \gamma^k \nabla \ell(x^k))$$

The algorithm is based on the idea of generating at each  $t$  feasible points (i.e. belonging to  $X$ ) that give a descent in the cost. The analysis follows similar arguments to the one of unconstrained gradient methods.

### 2.3.2 Feasible direction method

Find  $\tilde{x} \in \mathbb{R}^n$  such that

$$\tilde{x} = \arg \min_{x \in X} \ell(x^k) + \nabla \ell(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T (x - x^k)$$

Update the solution

$$x^{k+1} = x^k + \gamma^k (\tilde{x} - x^k)$$

where  $(\tilde{x} - x^k)$  is a feasible direction as it is contained in the set by construction. For  $\gamma^k$  sufficiently small,  $x^{k+1} \in X$

### Barrier function strategy for inequality constraints

Consider the inequality constrained optimization problem

$$\min_{x \in \mathbb{R}^d} \ell(x) \text{ subj. to } g_j(x) \leq 0 \quad j \in \{1, \dots, r\}$$

inequality constraints can be relaxed and embedded in the cost function by means of a barrier function  $-\varepsilon \log(x)$ . The resulting unconstrained problem reads as

$$\min_{x \in \mathbb{R}^d} \ell(x) + \varepsilon \sum_{j=1}^r -\log(-g_j(x))$$

Implementation: every few iterations shrink the barrier parameters  $\varepsilon$

Methods such as this go by the name of *interior point methods*

## 2.4 Constrained optimization (equality and inequality constraints): optimality conditions

$$\begin{aligned} & \min_{x \in X} \ell(x) \\ \text{subj. to } & g_j(x) \leq 0 \quad j \in \{1, \dots, r\} \\ & h_i(x) = 0 \quad i \in \{1, \dots, m\} \end{aligned}$$

**Definition 2.4.1** (Set of active inequality constraints). For a point  $x$ , the set of active inequality constraints at  $x$  is  $A(x) = \{j \in \{1, \dots, r\} | g_j(x) = 0\}$

**Definition 2.4.2** (Regular point). A point  $x$  is regular if the vectors  $\nabla h_i(x), i \in \{1, \dots, m\}$  and  $\nabla g_j(x), j \in A(x)$ , are linearly independent

### Lagrangian function

In order to state the first-order necessary conditions of optimality for (equality and inequality) constrained problems it is useful to introduce the Lagrangian function

$$L(x, \mu, \lambda) = \ell(x) + \sum_{j=1}^r \mu_j g_j(x) + \sum_{i=1}^m \lambda_i h_i(x)$$

**Theorem 2.4.1** (Karush-Kuhn-Tucker necessary conditions). Let  $x^*$  be a regular local minimum of

$$\begin{aligned} & \min_{x \in \mathbb{R}^d} \ell(x) \\ \text{subj. to } & g_j(x) \leq 0 \quad j \in \{1, \dots, r\} \\ & h_i(x) = 0 \quad i \in \{1, \dots, m\} \end{aligned}$$

where  $\ell, g_j$  and  $h_i$  are  $\mathcal{C}^1$ .

Then  $\exists!$   $\mu_j^*$  and  $\lambda_i^*$ , called Lagrange multipliers, s.t.<sup>1</sup>

$$\begin{aligned} \nabla_1 L(x^*, \mu^*, \lambda^*) &= 0 \\ \mu_j^* &\geq 0 \\ \mu_j^* g_j(x^*) &= 0 \quad j \in \{1, \dots, r\} \end{aligned}$$

Moreover, if  $\ell, g_j$  and  $h_i$  are  $\mathcal{C}^2$  it holds

$$y^T \nabla_{11}^2 \mathcal{L}(x^*, \mu^*, \lambda^*) y \geq 0$$

for all  $y \in \mathbb{R}^n$  such that

$$\nabla h_i(x)^T y = 0, \quad i \in \{1, \dots, m\}, \quad \nabla g_j(x)^T y = 0, \quad j \in A(x) \quad (\text{i.e. } j \in \{1, \dots, r\} \text{ s.t. } g_j(x) = 0)$$

*Remark.* The condition  $\mu_j^* g_j^*(x^*) = 0, j \in \{1, \dots, r\}$ , is called complementary slackness

*Notation.* Points satisfying the KKT necessary conditions of optimality are referred to as KKT points. They are the counterpart of stationary points in constrained optimization.

*Notation.* note that  $\nabla_{11}$  denotes the hessian of a function wrt the first variable

#### 2.4.1 Quadratic programming (constrained)

Let us consider quadratic optimization problems with linear equality constraints

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} x^T Q x + q^T x \\ \text{subj. to } & A x = b \end{aligned}$$

with  $q = Q^T \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, a \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$  The Lagrangian function is

$$L(x, \lambda) = x^T Q x + q^T x + \sum_{i=1}^m \lambda_i (A_i x + b_i) = x^T Q x + q^T x + \lambda^T (A x - b)$$

And the gradient computes as

$$\nabla_1 L(x^*, \lambda^*) = 2Qx^* + q + \sum_{i=1}^m \lambda_i^* A_i^T = 2Qx^* + q + A^T \lambda^*$$

The equality constraints must also be enforced:

$$A x^* - b = 0$$

We can note that

$$\nabla_2 L(x^*, \lambda^*) = A x - b$$

---

<sup>1</sup> $\nabla_1$  denotes the gradient wrt the first variable of the function

Therefore, first order conditions of optimality may be written as

$$\begin{bmatrix} \nabla_1 L(x^*, \lambda^*) \\ \nabla_2 L(x^*, \lambda^*) \end{bmatrix} = 0$$

This is always the case when only equality constraints are present. Second order necessary conditions for optimality impose that, if  $x^*$  is a minimum then

$$y^T \nabla^2 \mathcal{L}(x^*, \lambda^*) y = y^T Q y \geq 0$$

for all  $y \in \mathbb{R}^n$  such that

$$\nabla h_i(x)^T y = 0 \quad i \in \{1, \dots, p\} \implies A^T y = 0$$

namely, for all  $y \in \mathbb{R}^n$  in the null-space of  $A^t$

## 2.5 Constrained optimization (equality and inequality constraints): optimization algorithms

### 2.5.1 Newton's method for equality constrained problems

KKT points can be found by solving a root finding problem in variables  $x, \lambda$  wrt  $r(x, \lambda) = \nabla L(x, \lambda)$ . Newton's method for this root finding problem reads as

$$\begin{bmatrix} x^{k+1} \\ \lambda^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} + \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \end{bmatrix}$$

with

$$\begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \end{bmatrix} = -(\nabla^2 L(x^k, \lambda^k))^{-1} \nabla L(x^k, \lambda^k)$$

where

$$\begin{aligned} \nabla^2 L(x^k, \lambda^k) &= \begin{bmatrix} \nabla_{11} L(x^*, \lambda^*) & \nabla_{12} L(x^*, \lambda^*) \\ \nabla_{21} L(x^*, \lambda^*) & \nabla_{22} L(x^*, \lambda^*) \end{bmatrix} = \begin{bmatrix} H^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix} \\ \nabla L(x^k, \lambda^k) &= \begin{bmatrix} \nabla \ell(x^k) + \nabla h(x^k) \lambda^k \\ h(x^k) \end{bmatrix} \\ H^k &= \nabla_{11}^2 L(x^k, \lambda^k) \quad \nabla_{11} L(x, \lambda) = \nabla^2 \ell(x) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x) \end{aligned}$$

We can write

$$\nabla^2 L(x^k, \lambda^k) \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \end{bmatrix} = -\nabla L(x^k, \lambda^k)$$

namely

$$\begin{bmatrix} H^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \end{bmatrix} = -\begin{bmatrix} \nabla \ell(x^k) + \nabla h(x^k) \lambda^k \\ h(x^k) \end{bmatrix}$$

thus,  $\Delta x^k, \Delta \lambda^k$  can be obtained as solution of a linear system of equations in the variables  $\Delta x, \Delta \lambda$ . The linear system of equations can be rewritten as

$$\begin{aligned} H^k \Delta x^k + \nabla H(x^k) \Delta \lambda^k &= -\nabla \ell(x^k) - \nabla h(x^k) \lambda^k \\ \nabla h(x^k)^T \Delta x^k &= -h(x^k) \end{aligned}$$

and equivalently as

$$\begin{aligned} \nabla \ell(x^k) + H^k \Delta x^k + \nabla H(x^k) \Delta \lambda^{k+1} \\ h(x^k) + \nabla h(x^k)^T \Delta x^k &= 0 \end{aligned}$$

We can observe that the above equations are the necessary and sufficient optimality conditions for the Quadratic Program (QP)

$$\min_{\Delta x} \nabla \ell(x^k)^T \Delta x + \frac{1}{2} \Delta x^T H^k \Delta x \quad \text{subj. to} \quad h(x^k) + \nabla h(x^k)^T \Delta x = 0$$

Therefore, in the Newton's update, we can obtain  $(\Delta x^k, \lambda^{k+1})$  by solving this QP.



### 2.5.2 Sequential Quadratic Programming (SQP)

Start from a tentative solution  $x^0$ . For  $k = 0, 1, \dots$  (up to convergence)

1. Compute  $\nabla \ell(x^k), H^k, \nabla h(x^k)$
2. Obtain  $(\Delta x^k, \Delta \lambda_{QP}^+)$  from

$$\begin{aligned} \Delta x^k = \arg \min_{\Delta x} \quad & \nabla \ell(x^k)^T \Delta x + \frac{1}{2} \Delta x^T H^k \Delta x \\ \text{subject to} \quad & h(x^k) + \nabla h(x^k)^T \Delta x = 0 \end{aligned} \quad (2.2)$$

with  $\Delta_{QP}^*$  the Lagrange multiplier associated to the optimal solution of (2.2)

3. Choose  $\gamma^k$  using Armijo's rule on *merit function*  $M_1(x^k + \gamma \Delta x^k)$
4. Update

$$\begin{aligned} x^{k+1} &= x^k + \gamma^k \Delta x^k \\ \lambda^{k+1} &= \Delta \lambda_{QP}^* \end{aligned}$$

### 2.5.3 Barrier function strategy for inequality constraints

Consider the inequality optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \ell(x) \\ \text{subject to} \quad & g_j(x) \leq 0 \quad j \in \{1, \dots, r\} \\ & h(x) = 0 \end{aligned}$$

Inequality constraints can be embedded in the cost function by means of a barrier function  $-\varepsilon \log(x)$ . The resulting unconstrained problem reads as

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \ell(x) + \varepsilon \sum_{j=1}^r -\log(-g_j(x)) \\ & h(x) = 0 \end{aligned}$$

Implementation: every few iterations shrink the barrier parameters  $\varepsilon$



## Chapter 3

# Optimality conditions for optimal control

### 3.1 boh

#### 3.1.1 Dynamics as equality constraints

Let us rerwrite the nonlinear dynamics of a dt system as an implicit equality constraint  $h : \mathbb{R}^{nT} \times \mathbb{R}^{mT} \rightarrow \mathbb{R}^{nT}$

$$h(\bar{\mathbf{x}}, \bar{\mathbf{u}}) := \begin{bmatrix} f_0(x_0, u_0) - x_1 \\ \vdots \\ f_{T-1}(x_{T-1}, u_{T-1}) - x_T \end{bmatrix}$$

so that a curve  $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$  is a trajectory of the system if it satisfies the (possibly nonlinear) equality constraint

$$h(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = 0$$

#### 3.1.2 system trajectories and trajectory manifold

We can now define the trajectory manifold  $\mathcal{T} \subset \mathbb{R}^{nT} \times \mathbb{R}^{mT}$  of (ref)

$$\mathcal{T} := \{((x), (u)) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | h((x), (u)) = 0\} = \{((x), (u)) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | x_{t+1} = f_y(x_t, u_t), t = 0, \dots, T-1\}$$

Let  $(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \in \mathcal{T}$  be a trajectory of the system, i.e. a point on the trajectory manifold  $\mathcal{T}$ . The tangent space to  $\mathcal{T}$  at a given trajectory (point)  $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ , denoted as  $T_{(\bar{\mathbf{x}}, \bar{\mathbf{u}})}\mathcal{T}$ , is the set of trajectories satisfying the linearization of  $x_{t+1} = f_t(x_t, u_t)$  about the trajectory  $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ . That is,  $T_{(\bar{\mathbf{x}}, \bar{\mathbf{u}})}\mathcal{T} = \{(\Delta \mathbf{x}, \Delta \mathbf{u}) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | \nabla_1 h(\bar{\mathbf{x}}, \bar{\mathbf{u}})^T \Delta \mathbf{x} + \nabla_2 h(\bar{\mathbf{x}}, \bar{\mathbf{u}})^T \Delta \mathbf{u} = 0\}$  is the set of trajectories  $(\Delta \mathbf{x}, \Delta \mathbf{u})$  of

$$\Delta x_{t+1} = A_t \Delta x_t + B_t \Delta u_t$$

with

$$A_t = \nabla_1 f_t(\bar{x}_t, \bar{u}_t)^T B_t = \nabla_2 f_t(\bar{x}_t, \bar{u}_t)^T$$

### 3.2 Unconstrained optimal control problem (d-t)

We look for a solution of the discrete-time optimal control problemm

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m} \quad & \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \\ \text{subj. to} \quad & x_{t+1} = f_t(x_t, u_t), \quad t \in \{0, \dots, T-1\} \end{aligned}$$

with given initial condition  $x_0 = x_{\text{init}} \in \mathbb{R}^n$ .

From now on, we will assume that functions  $\ell_t(\cdot, \cdot), \ell_T(\cdot), f_t(\cdot, \cdot)$  are twice continuously differentiable, i.e. the are  $\mathcal{C}^2$

### 3.3 KKT conditions for unconstrained optimal control

the Lagrangian function has the form

$$\begin{aligned}
 \mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda) &= \ell(\mathbf{x}, \mathbf{u}) + \lambda^T h(\mathbf{x}, \mathbf{u}) = \\
 &= \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) + \sum_{t=0}^{T-1} \lambda_{t+1}^T (f_t(x_t, u_t) - x_{t+1}) \\
 &= \sum_{t=0}^{T-1} (\ell_t(x_t, u_t) + \lambda_{t+1}^T (f_t(x_t, u_t) - x_{t+1}) + \ell_T(x_T)) \\
 &= \sum_{t=0}^T \mathcal{L}_t(x_t, u_t, \lambda)
 \end{aligned}$$

where  $\lambda \in \mathbb{R}^{nT}$  and

$$\begin{aligned}
 \mathcal{L}_0(x_0, u_0, \lambda) &= \ell_0(x_0, u_0) + \lambda_1^T f_0(x_0, u_0) \\
 \mathcal{L}_t(x_t, u_t, \lambda) &= \ell_t(x_t, u_t) + \lambda_1^T f_t(x_t, u_t) - \lambda_t x_t \\
 \mathcal{L}_T(x_T, \lambda) &= \ell_T(x_T) - \lambda_T^T x_T
 \end{aligned}$$

Let  $(\mathbf{x}^*, \mathbf{u}^*)$  be a regular point for the dynamics constraints and an optimale (state-input) trajectory. Then there exists  $\lambda^*$  such that  $\nabla \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) = 0$

Let us explicitly write condition  $\nabla_{(1,2)} \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) = 0$

$$\nabla_{(1,2)} \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) = \begin{bmatrix} \nabla_1 \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) \\ \nabla_2 \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) \end{bmatrix} = 0$$

Let us note that

$$\nabla_1 \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) = \begin{bmatrix} \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda)}{\partial (x_1)_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda)}{\partial (x_1)_n} \\ \vdots \\ \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda)}{\partial (x_T)_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda)}{\partial (x_T)_n} \end{bmatrix} \bigg|_{\mathbf{x}=\mathbf{x}^*}$$

Since  $\mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda) = \sum_{t=0}^T \mathcal{L}_t(x_t, u_t, \lambda)$ , we can exploit this sparsity and write

$$\begin{aligned}
 \nabla_2 \mathcal{L}_0(x_0, u_0, \lambda) &= 0 & \nabla_2 \ell_0(x_0, u_0) \nabla_2 f_0(x_0, u_0) \lambda_0 \\
 \begin{bmatrix} \nabla_1 \mathcal{L}_t(x_t, u_t, \lambda) \\ \nabla_2 \mathcal{L}_t(x_t, u_t, \lambda) \end{bmatrix} &= 0 & \begin{bmatrix} \nabla_1 \ell_t(x_t, u_t) + \nabla_1 f_t(x_t, u_t) \lambda_{t+1} - \lambda_t \\ \nabla_2 \ell_t(x_t, u_t) + \nabla_2 f_t(x_t, u_t) \lambda_{t+1} \end{bmatrix} = 0 \quad t = 1, \dots, T-1 \\
 \nabla_1 \mathcal{L}_T(x_T, \lambda) &= 0 & \nabla \ell_T(x_T) - \lambda_T = 0
 \end{aligned}$$

Let us introduce some notation:

$$\begin{aligned}
 \nabla_1 \ell_t(x_t^*, u_t^*) &= a_t \in \mathbb{R}^n \\
 \nabla_1 f_t(x_t^*, u_t^*) &= A_t^T \\
 \nabla_2 \ell_t(x_t^*, u_t^*) &= b_t \in \mathbb{R}^n \\
 \nabla_2 f_t(x_t^*, u_t^*) &= B_t^T
 \end{aligned}$$

So we can rewrite the KKT conditions for unconstrained optimal control as:

$$\begin{aligned}\lambda_t^* &= A_t^T \lambda_{t+1}^* + a_t & t = T-1, \dots, 1 \\ \lambda_T^* &= \nabla \ell(x_T^*) \\ B_t^T \lambda_{t+1}^* + b_t &= 0 & t = 0, \dots, T-1\end{aligned}$$

### 3.3.1 Indirect methods for optimal control

Solving the optimality conditions:

- Guess some  $u_t^0, \quad t = 0, \dots, T-1 \quad k = 0$

- run "forward"

$$x_{t+1}^0 = f - t(x_t^0, u_t^0) \quad x_0$$

- run "backward"

$$a$$

- given  $\lambda_t^0 \quad t = 1, \dots, T$  solve:

$$\nabla_2 \ell(x_t^0, u_t) + \nabla_2 f(x_t^0, u_t) \lambda_{t+1}^0 = 0 \quad t = 0, \dots, T-1$$

to get  $u_t^1 \quad t = 0, \dots, T-1$

## 3.4 KKT conditions for constrained optimal control

We look for a solution of the discrete-time optimal control problem

$$\begin{aligned}\min_{x_0 \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^{nT}, \mathbf{u} \in \mathbb{R}^{mT}} \quad & \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \\ \text{subj. to} \quad & x_{t+1} = f_t(x_t, u_t), \quad t = 0, \dots, T-1 \\ & r(x_0, x_T) = 0 \\ & g_t(x_t, u_t) \leq 0, \quad t = 0, \dots, T-1\end{aligned}$$

where

- $\ell_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  is the stage cost,
- $\ell_T : \mathbb{R}^n \rightarrow \mathbb{R}$  is the terminal cost,
- $r : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{p_0}$  identifies a boundary constraint on initial and final states,
- $g_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$  for each  $t$  identifies point-wise constraints on state and input at some time  $t$

The Lagrangian function has the form

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda, \mu) &= \ell(\mathbf{x}, \mathbf{u}) + \lambda_d^T h(\mathbf{x}, \mathbf{u}) + \lambda_b^T r(x_0, x_T) + \mu^T g(\mathbf{x}, \mathbf{u}) \\ &= \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) + \sum_{t=0}^{T-1} \lambda_{d,t+1} (f_t(x_t, u_t) - x_{t+1}) + \lambda_b^T r(x_0, x_T) + \sum_{t=0}^{T-1} \mu_t^T g_t(x_t, u_t) \\ &= \sum_{t=0}^T \mathcal{L}_t(x_t, u_t, \lambda, \mu)\end{aligned}$$



## Chapter 4

# Linear Quadratic (LQ) optimal control

Consider a linear quadratic optimal control problem as:

$$\begin{aligned} \min_{\substack{x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} & \sum_{t=0}^{T-1} \frac{1}{2} [x_t^T q_t x_t + u_t^T R_t u_t] + \frac{1}{2} x_T^T q_T x_T \\ \text{subj. to} & \quad x_{t+1} = A_t x_t + B_t u_t \quad t = 0, \dots, T-1 \\ & \quad x_0 = x_{\text{init}} \end{aligned}$$

We assume  $Q_t = Q_t^T \geq 0 \forall t = 0, \dots, T-1$ ,  $Q_t = Q_t^T \geq 0$ , and  $R_t = R_t^T > 0 \forall t = 0, \dots, T-1$

### 4.1 First order optimality condition

$$\begin{aligned} \nabla_1 f_t(x_t, u_t) &= A_t^T \\ \nabla_1 \ell(x_t, u_t) &= \nabla_1 \left( \frac{1}{2} x_t^T Q_t x_t + \frac{1}{2} u_t^T R_t u_t \right) = Q_t x_t \\ \nabla_2 f_t(x_t, u_t) &= B_t^T \\ \nabla_2 \ell_t(x_t, u_t) &= R_t u_t \end{aligned}$$

therefore

$$\begin{aligned} \lambda_t^* &= A_t^T \lambda_t + 1^* + Q_t x_t^* \quad t = T-1, \dots, 0 \\ \lambda_T^* &= Q_T x_T^* \\ B_t^T \lambda_{t+1}^* + R_t u_t^* &= 0 \quad t = 0, \dots, T-1 \end{aligned}$$

Remark: second order optimality conditions

$$y^T \nabla_{(1,2)(1,2)}^2 \mathcal{L}(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) y \geq 0$$

For vectors  $y$  satisfying the "linear approximation of the constraint". The hessian turns out as

$$\begin{bmatrix} Q_1 & & 0 \\ & \ddots & \\ 0 & & Q_n \end{bmatrix}$$

Because  $R_t > 0$  it is invertible. Therefore, we can write

$$u_t^* = -R_t^{-1} B_t^T \lambda_{t+1}^*$$

Introducing a matrix  $P_t = P_t^T \geq 0$ , it can be proven that

$$\lambda_t^* = P_t x_t^*$$

Assuming that it holds for some  $t \leq T - 1$ , then we have

$$u_t^* = -R_t^{-1} B_t^T P_{t+1} x_{t+1}^*$$

Now, considering the constraint represented by the dynamics

$$u_t^* = -R_t^{-1} N_t^T P_{t+1} (A_t x_t^* + B_t u_t^*)$$

Solving by  $u_t^*$  yields

$$u_t^* = -(R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t x_t^* \quad t = 0, \dots, T - 1$$

we now get

$$\begin{aligned} u_t^* &= -R_t^{-1} B_t^T p_{t+1} x_{t+1}^* \\ &= -R_t^{-1} B_t^T p_{t+1} (A_t x_t^* + B_t u_t^*) \end{aligned}$$

we multiply both sides by  $R_t$ :

$$\begin{aligned} R_t u_t^* &= -B_t^T P_{t+1} (A_t x_t^* + B_t u_t^*) \\ R_t u_t^* &= -B_t^T P_{t+1} A_t x_t^* - B_t^T P_{t+1} B_t u_t^* \\ (R_t + B_t^T P_{t+1} B_t) u_t^* &= -B_t^T P_{t+1} A_t x_t^* \end{aligned}$$

The matrix on the left is clearly positive definite, therefore:

$$u_t^* = -(R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t x_t^*$$

which we can write as

$$u_t^* = K_t^* x_t^*$$

that is, the optimal control is a state feedback with gain  $-(R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1}$

$$x_{t+1} = A_t x_t^* - B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t x_t^*$$

which we can rewrite as

$$x_{t+1}^* = (A_t - B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t) x_t^*$$

which is a closed loop system. We multiply both sides by  $P_{t+1}$  and obtain

$$P_{t+1} x_{t+1}^* = P_{t+1} (\dots) x_t^*$$

On the left side of the equation we have obtained  $\lambda_{t+1}^*$

$$\lambda_{t+1}^* = P_{t+1} (\dots) x_t^*$$

Remembering that  $\lambda_t^* = A_t^T \lambda_{t+1}^* + Q_t x_t^*$  we multiply both sides by  $A_t^T$  and then add  $Q_t x_t^*$  and obtain

$$A_t^T \lambda_{t+1}^* + Q_t x_t^* = A_t^T P_{t+1} (\dots) x_t^* + Q_t x_t^*$$

and because

$$\lambda_t^* = P_t x_t^*$$

then

$$P_t x_t^* = [A_t^T P_{t+1} (\dots) + Q_t] x_t^*$$

so

$$P_t x_t^* = [A_t^T P_{t+1} A_t - A_t^T P_{t+1} B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t + Q_t] x_t^*$$

from which

$$P_T = [A_T^T P_{T+1} A_T - A_T^T P_{T+1} B_T (R_T + B_T^T P_{T+1} B_T)^{-1} B_T^T P_{T+1} A_T + Q_T] \quad (4.1)$$

because  $\lambda_T^* = Q_T x_T^*$  we have that

$$P_T = Q_T$$

Therefore, by propagating equation (4.1) back in time,  $P_t$  can be calculated. equation (4.1) is called difference Riccati equation

- gains  $K_t^*$  can be precomputed offline and the used for different  $x_0$
- It can be shown that if  $T \rightarrow \infty$  the gains  $K_t^*$  converge and asymptotically stabilize the system



## 4.2 Infinite horizon LQ optimal control

Consider the infinite-horizon optimal control problem

$$\begin{aligned} \min_{\substack{x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} & \sum_{t=0}^{\infty} \frac{1}{2} [x_t^T Q x_t + u_t^T R u_t] \\ \text{subj. to} & \quad x_{t+1} = A x_t + B u_t \quad t = 0, \dots, T-1 \\ & \quad x_0 = x_{\text{init}} \end{aligned}$$

where

- $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$
- $A \in \mathbb{R}^{n \times n}$
- $B \in \mathbb{R}^{n \times m}$
- $Q \in \mathbb{R}^{n \times n}$  and  $Q = Q^T \geq 0$
- $R \in \mathbb{R}^{m \times m}$  and  $R = R^T > 0$

We assume the pair  $(A, B)$  is controllable and the pair  $(A, C)$  with  $Q = C^T C$  is observable. Let us write

$$y_t = C x_t$$

which leads to

$$\frac{1}{2} x_t^T Q x_t = \frac{1}{2} x_t^T C^T C x_t = \frac{1}{2} y_t^T y_t$$

The controllability assumption guarantees that an optimal controller exists: if  $(A, B)$  controllable, then  $\exists \bar{u}_0, \dots, \bar{u}_{T-1}$  for  $T$  sufficiently large ( $T = n$ ) such that  $\forall x_0 \in \mathbb{R}^n \implies x_T = 0$ . Consider the input

$$\bar{u}_0, \dots, \bar{u}_{T-1}, 0, \dots, 0, \dots$$

Let us compute the cost associated to this input

$$\sum_{t=0}^{\infty} \frac{1}{2} [x_t^T Q x_t + u_t^T R u_t] = \sum_{t=0}^{T-1} \frac{1}{2} \bar{x}_t^T Q \bar{x}_t + \frac{1}{2} \bar{u}_t^T R \bar{u}_t$$

We can note that the cost is a finite quantity. Because the cost is finite, There must exist a solution which minimizes the cost.

**Proposition 4.2.1.** Let the pair  $(A, B)$  be controllable and the pair  $(A, C)$  with  $Q = C^T C$  be observable. Then the following holds:

- there exists a unique positive definite  $P_{\infty}$  equilibrium solution of the Difference Riccati Equation. That is,  $P_{\infty}$  is a solution of

$$P_{\infty} = Q + A^T P_{\infty} A - A^T P_{\infty} B (R + B^T P_{\infty} B)^{-1} B^T P_{\infty} A$$

which is called Algebraic Riccati Equation

- the optimal control is a feedback of the state given by:

$$\begin{aligned} K^* &= -(R + B^T P_{\infty} B)^{-1} (B^T P_{\infty} A) \\ u_t^* &= K^* x_t^* \\ x_{t+1}^* &= A x_t^* + B u_t^* \quad t = 1, 2, \dots \quad x_0^* = x_{\text{init}} \end{aligned}$$

*Remark.* The observability of  $(A, C)$  guarantees that if the stage cost goes to zero, then the state trajectory goes to zero.



## Chapter 5

# Optimality Conditions for Unconstrained Optimal Control via Shooting

Let us consider the system dynamics

$$x_{t+1} = f_t(x_t, u_t) \quad t = 0, \dots, T-1 \quad x_0 \text{ given}$$

and let us suppose we have an input sequence  $u_0, \dots, u_{T-1}$ . We have:

$$x_1 = f_0(x_0, u_0) = \tilde{\Phi}_1(\mathbf{u})$$

$$x_2 = f_1(x_1, u_1) = f_1(f_0(x_0, u_0), u_1) = \tilde{\Phi}_2(\mathbf{u})$$

$\vdots$

$$x_t = \tilde{\Phi}_t(\mathbf{u}) \quad t = 0, \dots, T-1$$

$$x_T = \tilde{\Phi}_T(\mathbf{u}) \quad t = 0, \dots, T-1$$

$\vdots$

Idea: express the state  $x_t$  at each  $t = 1, \dots, T$  as a function of the input sequence  $\mathbf{u}$  only. For all  $t$  we can introduce a map  $\Phi_t : \mathbb{R}^m \rightarrow \mathbb{R}^n$  such that

$$x_t := \Phi_t(\mathbf{u})$$

compact notation

$$\Phi(\mathbf{u}) = \text{col}(\Phi_1(\mathbf{u}), \dots, \Phi_T(\mathbf{u}))$$

so that

$$\mathbf{x} = \Phi(\mathbf{u})$$

Note: Given any arbitrary  $\bar{u}_0, \dots, \bar{u}_{T-1}$ , we have that  $\Phi_{t+1}(\bar{\mathbf{u}}) = f_t(\Phi_t(\bar{\mathbf{u}}), u_t)$  by construction. This is equivalent to the equality constraint for the optimal control problem.

### 5.1 Reduced optimal control problem

We can rewrite the optimal control problem as

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} \sum_{t=0}^{T-1} \ell_t(\Phi_t(\mathbf{u}), u_t) + \ell_T(\Phi_T(\mathbf{u}))$$

as noted before, the equality constraint is satisfied by construction, making this an unconstrained optimization problem. We can rewrite it compactly as

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} \ell(\Phi(\mathbf{u}), \mathbf{u})$$

and by defining  $J(\mathbf{u}) := \ell(\Phi(\mathbf{u}), \mathbf{u})$

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} J(\mathbf{u}) :$$

This goes by the name of *reduced* or *condensed optimal control problem*. The procedure of writing  $\mathbf{x}$  as a function of  $\mathbf{u}$  and then plugging it into the optimal control problem is called shooting.

*Remark.* if we consider path input constraints

$$g_0(u_0) \leq 0; g_{T-1}(u_{T-1}) \leq 0$$

the problem becomes

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} J(\mathbf{u}) \text{ subject to}$$

*Remark.* if we have constraints of the type

$$g_0(x_0, u_0) \leq 0; g_{T-1}(x_{T-1}, u_{T-1}) \leq 0$$

They can be rewritten as functions of  $x_0$  and  $\mathbf{u}$  only, however  $\Phi(\cdot)$  must be explicitly known

## 5.2 Algorithms for optimal control problem solution

We can apply the gradient method, i.e.

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \gamma \nabla J(\mathbf{u}^k)$$

We can formally write the expression of  $\nabla J(\mathbf{u}) = \nabla \ell(\Phi(\mathbf{u}), \mathbf{u})$  by using the chain rule of differentiation.

$$\nabla \Phi(\mathbf{u}) = \nabla \begin{bmatrix} \Phi_{1,1}(\mathbf{u}) \\ \Phi_{1,2}(\mathbf{u}) \\ \vdots \\ \Phi_{t,1}(\mathbf{u}) \\ \Phi_{t,2}(\mathbf{u}) \\ \vdots \end{bmatrix}$$

$$\nabla \Phi(\mathbf{u}) = \begin{bmatrix} \frac{\partial \Phi_{1,1}}{\partial u_0} & \frac{\partial \Phi_{1,2}}{\partial u_0} & \dots & \frac{\partial \Phi_{T,n}}{\partial u_0} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \Phi_{1,1}}{\partial u_{T-1}} & \frac{\partial \Phi_{1,2}}{\partial u_{T-1}} & \dots & \frac{\partial \Phi_{T,n}}{\partial u_{T-1}} \end{bmatrix}$$

where  $\Phi_{t,j} : \mathbb{R}^{mT} \rightarrow \mathbb{R}$ , therefore the above matrix is a matrix of scalars. Let us introduce an auxiliary function  $\mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda) : \mathbb{R}^{nT} \times \mathbb{R}^{mT} \times \mathbb{R}^{nT} \rightarrow \mathbb{R}$  given by

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda) = \ell(\mathbf{x}, \mathbf{u}) + h(\mathbf{x}, \mathbf{u})^T \lambda$$

where  $\lambda \in \mathbb{R}^{nT}$  is a "costate vector" and

$$h(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_0(x_0, u_0) - x_1 \\ \vdots \\ f_{T-1}(x_{T-1}, u_{T-1}) - x_T \end{bmatrix}$$

To compute  $\nabla J(\mathbf{u})$  let us evaluate  $\hat{\uparrow}(\cdot)$  for  $\mathbf{x} = \Phi(\mathbf{u})$ . Since  $h(\Phi(\mathbf{u}), \mathbf{u}) = 0$  it holds that

$$\mathcal{L}(\Phi(\mathbf{u}), \mathbf{u}, \lambda) = J(\mathbf{u}) \quad \forall \lambda \in \mathbb{R}^{nT}$$

Therefore

$$\nabla \mathcal{L}(\Phi(\mathbf{u}), \mathbf{u}, \lambda) = \nabla J(\mathbf{u}) \quad \forall \lambda$$

hence we can write

$$\nabla J(\mathbf{u}) = \nabla \Phi(\mathbf{u})(\nabla_1 \ell(\Phi(\mathbf{u}), \mathbf{u}) + \nabla_1 h(\Phi(\mathbf{u}), \mathbf{u})\lambda) + \nabla_2 \ell(\Phi(\mathbf{u}), \mathbf{u}) + \nabla_2 h(\Phi(\mathbf{u}), \mathbf{u})\lambda$$

which holds for every  $\lambda$ . Therefore, for a given  $\mathbf{u}$ , we can cleverly select  $\lambda = \lambda(\mathbf{u})$  such that:

$$\nabla_1 \ell(\Phi(\mathbf{u}), \mathbf{u}) + \nabla_1 h(\Phi(\mathbf{u}), \mathbf{u})\lambda(\mathbf{u})$$

which leads to

$$\nabla J(\mathbf{u}) = \nabla_2 \ell(\Phi(\mathbf{u}), \mathbf{u}) + \nabla_2 h(\Phi(\mathbf{u}), \mathbf{u})\lambda(\mathbf{u})$$

### 5.2.1 First order necessary condition for optimality

Let  $\mathbf{u}^*$  be a local minimum with  $\mathbf{x}^* = \Phi(\mathbf{u}^*)$ . Then

$$\nabla J(\mathbf{u}^*) = 0$$

that is, if there exists a  $\lambda^*$  such that

$$\nabla_1 \mathcal{L}(\Phi(\mathbf{u}^*), \mathbf{u}^*, \lambda^*) = 0$$

it holds

$$\nabla_2 \mathcal{L}(\Phi(\mathbf{u}^*), \mathbf{u}^*, \lambda^*) = 0$$

### 5.2.2 explicit computation of

$$\mathcal{L}(x, u, \lambda) = \sum_{t=0}^{T-1} [\ell_t(x_t, u_t) + \lambda_{t+1}^T f_t(x_t, u_t) - \lambda_{t+1}^T x_{t+1}] + \ell_T(x_T) = \sum_{t=0}^{T-1} (x_t, u_t)$$

$$\nabla_1 \ell_1(x_1, u_1) + \nabla_1 f_1(x_1, u_1)\lambda_2 - \lambda_1 = 0$$

$$\mathcal{L}(x, u, \lambda)\ell_0(x_0, u_0) + \lambda_1^T f_0(x_0, u_0) - \lambda_1 x_1 + \ell_1(x_1, u_1) + \lambda_2^T f_1(x_1, u_1) - \lambda_2 x_2 + \dots$$

Notice we can write

$$A_t^T = \nabla_1 f(x_t, u_t)$$

$$B_t^T = \nabla_2 f(x_t, u_t)$$

so that we obtain

$$\lambda_t = A_t^T \lambda_{t+1} + 1 + a_t$$

so given  $u_0, \dots, u_{T-1}$  and  $x_1, \dots, x_T$  such that  $x_{t+1} = f(x_t, u_t)$  we can compute  $\lambda_T, \dots, \lambda_1$  running backwards. We can also state that

$$(\nabla J(u))_t = \nabla_2 \ell_t(x_t, u_t) + \nabla_2 f_t(x_t, u_t)\lambda_{t+1}$$

which we can rewrite as

$$(\nabla J(u))_t = B_t^T \lambda_{t+1} + b_t$$



## Chapter 6

# Optimal Control based trajectory generation and tracking

Task request: We want to control a (discrete-time) nonlinear system

$$x_{t+1} = f_t(x_t, u_t)$$

along a (possibly aggressive) evolution to perform a task while satisfying some performance criteria.

Possible performance criteria:

- reduce energy consumption
- avoid excessive accelerations (due to e.g., a fragile payload)

### 6.1 main strategy idea over a finite horizon

First, a trajectory generation task is reformulated into an optimal control problem such as

$$\min \sum_{t=0}^{T-1} \frac{1}{2} \|x_t - x_t^{des}\|_{Q_t}^2 + \frac{1}{2} \|u_t - u_t^{des}\|_{R_t}^2 + \frac{1}{2} \|x_T - x_T^{des}\|_{P_f}^2 \text{ s.t. } x_{t+1} = f(x_t, u_t) \quad t = 0, \dots, T-1, x_0 = x_{\text{init}}$$

Where  $Q_t, R_t, P_f$  are suitably chosen cost matrices and  $(\mathbf{x}^{des}, \mathbf{u}^{des})$  is a "reference curve" describing a desired evolution.

Note:  $(\mathbf{x}^{des}, \mathbf{u}^{des})$  is NOT a trajectory. It is based, e.g., on geometric considerations

Idea: by using an optimal control algorithm, compute an open loop (optimal) state-input trajectory  $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$ , i.e., such that  $x_{t+1}^{opt} = f(x_t^{opt}, u_t^{opt}), t = 0, \dots, T-1$ . Then, a feedback controller can be used to track the system trajectory  $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$

### 6.2 LQR based trajectory tracking

Idea: track the generated (optimal) trajectory via a (stabilizing) feedback Linear Quadratic Regulator (LQR) on the linearization.

Step 1 - linearize the system

Linearize the dynamics about the (feasible) trajectory  $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$ , get the linear (time-varying) system

$$\Delta x_{t+1} = A_t^{opt} \Delta x_t + B_t^{opt} \Delta u_t$$

where  $A_t^{opt} \in \mathbb{R}^{n \times n}$  and  $B_t^{opt} \in \mathbb{R}^{n \times m}$  are defined as:

$$A_t^{opt} := \nabla_1 f_t(x_t^{opt}, u_t^{opt})^T \quad (6.1)$$

$$B_t^{opt} := \nabla_2 f_t(x_t^{opt}, u_t^{opt})^T \quad (6.2)$$

for all  $(x_t^{opt}, u_t^{opt})$  with  $t = 0, \dots, T$ , state-input pairs at time  $t$  of trajectory  $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$  with length  $T$ .

Step 2 - calculate the LQ optimal controller Solve the optimal control problem

for some cost matrices  $Q_t^{reg} \geq 0 \in \mathbb{R}^{n \times R}$ ,  $Q_t^{reg} \geq 0 \in \mathbb{R}^{n \times m}$  and  $Q_T^{reg} \geq 0 \in \mathbb{R}^{n \times n}$  (DoF). Set  $P_T = Q_T^{reg}$  and backward iterate  $t = T - 1, \dots, 0$ :

$$P_t = Q_t^{reg} + A_t^{optT} P_{t+1} A_t^{opt} - (A_t^{optT} P_{t+1} B_t^{opt})$$

and define for all  $t = 0, \dots, T - 1$ , the feedback gain  $K_t^{reg} \in \mathbb{R}^{m \times n}$

Step 3 - track the generated (optimal) trajectory

Apply the feedback controller designed on the linearization to the nonlinear system to track  $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$ . Namely, for all  $t = 0, \dots, T - 1$ , we apply

$$u_t = u_t^{opt} + K_t^{reg}(x_t - x_t^{opt}) \quad (6.3)$$

$$x_{t+1} = f_t(x_t, u_t) \quad (6.4)$$

with  $x_0$  given

Remark: Under suitable assumptions, it can be shown that an infinite horizon trajectory of a nonlinear system,  $(x_t, u_t)$  with  $t = 0, \dots$  is (locally) exponentially stable if and only if the system linearization about the trajectory is exponentially stable. (this can be viewed as a time-varying version of the Lyapunov indirect theorem)

### 6.3 Affine LQR for trajectory tracking

The general trajectory tracking problem for a linear system can be recast into an affine LQR problem, with the affine part being generated by the trajectory.



# Chapter 7

## Dynamic Programming

Consider the optimal control problem Dynamic programming aims at solving optimal control problems by exploiting Bellman's principle of optimality: Each subtrajectory of an optimal trajectory is an optimal trajectory as well

The optimal value function (or *cost go-to function*)

$$V_t^*(\bar{x}) = \min_{\substack{x_{t+1}, x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \ell_t(x_t, u_t) + \sum_{\tau=t}^{T-1} \ell_\tau(x_\tau, u_\tau) + \ell_T(x_T) \\ \text{subj. to } \begin{aligned} x_{\tau+1} &= f_\tau(x_\tau, u_\tau), & \tau \in \{0, \dots, T-1\} \\ x_t &= \bar{x}_t \end{aligned}$$

It is the cost incurred starting from  $x_t = \bar{x}$  in the horizon  $[t, T]$  when the optimal policy is applied. Notice that  $V_T^*(\bar{x}) = \ell_T(\bar{x})$

### 7.1 Dynamic programming Recursion

By isolating the first contribution in the cost, we have:

$$V_t^*(\bar{x}) = \min_{\substack{x_{t+1}, x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \ell_t(x_t, u_t) + \sum_{\tau=t+1}^{T-1} \ell_\tau(x_\tau, u_\tau) + \ell_T(x_T) \\ \text{subj. to } \begin{aligned} x_{\tau+1} &= f_\tau(x_\tau, u_\tau), & \tau = t, t+1, \dots, T-1 \\ x_t &= \bar{x}_t \end{aligned}$$

Take  $\bar{x}_{t+1} = f_t(\bar{x}_t, u_t^*)$  with  $u_t^*$  solution of the previous problem we can write:

$$V_{t+1}^*(f_t(\bar{x}_t, u_t^*)) = \min_{\substack{x_{t+2}, x_1, \dots, x_T \\ u_{t+1}, \dots, u_{T-1}}} \sum_{\tau=t+1}^{T-1} \ell_\tau(x_\tau, u_\tau) + \ell_T(x_T) \\ \text{subj. to } \begin{aligned} x_{\tau+1} &= f_\tau(x_\tau, u_\tau), & \tau = t+1, \dots, T-1 \\ x_{t+1} &= \bar{x}_{t+1} \end{aligned}$$

for  $t = 0, \dots, T-1$  the optimal value function satisfies:

$$V_t^*(\bar{x}) = \min_{u \in \mathbb{R}^m} \ell_t(\bar{x}, u) + V_{t+1}^*(f_t(\bar{x}, u))$$

for any  $\bar{x} \in \mathbb{R}^n$ . This equation is known as *Bellman's Equation* remark: The optimal cost for the original optimal control problem is  $V_0^*(x_{init})$

#### 7.1.1 Optimal Control Policy and Trajectory

Policy: a policy is a feedback control law  $\pi_t(x)$  that associates, at time  $t$ , to each state  $x$  an input  $u$ , i.e.,  $\pi_t : \mathbb{R}^n \rightarrow \mathbb{R}^m$ .

The optimal policy to apply at time  $t$  when in a given state  $x_t$  can be computed as:

$$\pi_t^*(x_t) = \arg \min_u \ell_t(x_t, u) + V_{t+1}^*(f_t(x_t, u))$$

Given this policy, an optimal trajectory can be computed by forward simulation as:

$$\begin{aligned} u_t^* &= \pi_t^*(x_t^*) \\ x_{t+1}^* &= f_t(x_t^*, u_t^*) \quad t = 0, \dots, T-1 \\ x_0^* &= x_{init} \end{aligned}$$

### 7.1.2 DP Advantages and Limitations

Advantages:

- no need for differentiability or convexity assumptions on  $\ell_t(\cdot)$ ,  $\ell_T(\cdot)$ ,  $f_t(\cdot)$
- works well on discrete state-control spaces

Disadvantages:

- analytical solution not available on continuous spaces (e.g.  $\mathbb{R}^n$ ) – *curse of dimensionality*

Remark: A special case where DP can be performed exactly is Linear Quadratic optimal control.

### 7.1.3 Linear Quadratic Optimal Control via DP

Consider a linear quadratic control problem as: Let us write Bellman's equation for this problem:

$$V_t^*(x_t) = \min_{u \in \mathbb{R}^m} \frac{1}{2} \begin{bmatrix} x_t \\ u \end{bmatrix}^T \begin{bmatrix} Q_t & S_t^T \\ S_t & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u \end{bmatrix} + V_{t+1}^*(A_t x_t + B_t u)$$

The optimal input policy is the minimizer, i.e.,

$$\pi_t^*(x_t) = \arg \min_{u \in \mathbb{R}^m} \frac{1}{2} \begin{bmatrix} x_t \\ u \end{bmatrix}^T \begin{bmatrix} Q_t & S_t^T \\ S_t & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u \end{bmatrix} + V_{t+1}^*(A_t x_t + B_t u)$$

by considering

$$V_{t+1}^*(z) = \frac{1}{2} z^T P_{t+1} z$$

we obtain

$$\pi_t^*(x_t) = \arg \min_{u \in \mathbb{R}^m} \frac{1}{2} \begin{bmatrix} x_t \\ u \end{bmatrix}^T \begin{bmatrix} Q_t & S_t^T \\ S_t & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u \end{bmatrix} + (A_t x_t + B_t u)^T P_{t+1} (A_t x_t + B_t u)$$

because The optimization is wrt  $u \in \mathbb{R}^m$ , the terms that do not depend on  $u$  need not be considered as they do not affect the minimization problem. The problem can be rewritten as:

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} u^T (R_t + B_t^T P_{t+1} B_t) u + x_t^T (S_t^T + A_t^T P_{t+1} B_t) u + \text{const}$$

This is a Quadratic Program. Because  $R_t + B_t^T P_{t+1} B_t$  is positive definite, the second order sufficient optimality conditions are satisfied, therefore there exists a unique minimum. Let us take the gradient and set it to zero:

$$(R_t + B_t^T P_{t+1} B_t) u + (S_t + B_t^T P_{t+1} A_t) x_t = 0$$

which leads to

$$u^* = \pi^*(x_t) = -(R_t + B_t^T P_{t+1} B_t)^{-1} (S_t + B_t^T P_{t+1} A_t) x_t = K^* x_t$$

It is therefore possible to write

$$V_t^*(x_t) = \frac{1}{2} \begin{bmatrix} x_t \\ K_t^* x_t \end{bmatrix}^T \begin{bmatrix} Q_t + A_t^T P_{t+1} A_t & S_t^T + A_t^T P_{t+1} B_t \\ S_t + B_t^T P_{t+1} A_t & R_t + B_t^T P_{t+1} B_t \end{bmatrix} \begin{bmatrix} x_t \\ K_t^* x_t \end{bmatrix}$$

### 7.1.4 DP for LQP – Summary

### 7.1.5 LQ for time-invariant systems and cost

The solution turns out to be:

$$P_T = Q_T \quad (7.1)$$

$$P_t = Q + A^T P_{t+1} A + (S^T + A^T P_{t+1} B)(R + B^T P_{t+1} B)^{-1}(S^T + B^T P_{t+1} A) \quad (7.2)$$

$$K_t^* = -(R + B^T P_{t+1} B)^{-1}(S + B^T P_{t+1} A) \quad (7.3)$$

we can notice that even though the system is not time-varying, the optimal gain is. We can consider this to be a dynamical system with  $P_t$  as the state, that has as an equilibrium the solution to the algebraic Riccati equation

## 7.2 Infinite Horizon Linear Quadratic Problems

Consider a linear quadratic optimal control problem as: The optimal value function can be defined as: which does not depend on time  $t$  (the horizon is always  $\infty$ ), i.e.

$$V_{t_1}^*(\bar{x}) = V_{t_2}^*(\bar{x}) \quad \forall t_1 \neq t_2, \forall \bar{x}$$

therefore, we say that infinite horizon LQ is *shift invariant* and we can then drop the subscript  $t$  in the definition of the optimal value function, namely, for all  $t$ :

$$V^*(\bar{x}) = V_t^*(\bar{x})$$

If we suppose  $V^*$  to be a positive semi-definite quadratic function, we can write

$$V^*(\bar{x}) = \frac{1}{2} \bar{x}^T P \bar{x}$$

where  $P = P^T \geq 0$



## Chapter 8

# Numerical methods for nonlinear optimal control



## Chapter 9

# Optimization-based predictive control