Optimal Control M

Giorgio Medico based on Dante Piotto's notes

fall semester 2024

Contents

Course Overview

This course focuses on optimization-based control of dynamical systems. It provides theoretical and numerical methods to compute control system trajectories that minimize a performance index, with particular emphasis on their application to trajectory optimization and maneuvering of Autonomous Systems.

Learning Objectives

Upon completion of this course, students will be able to:

1. Set up optimal control problems and characterize optimality conditions
2. Develop numerical optimization methods to compute optimal, feasible trajectories
3. Design optimization-based receding-horizon control schemes for nonlinear systems

Course Applications

To bridge the gap between theory and application, students will apply the proposed techniques to trajectory optimization and maneuvering of Autonomous Systems across various domains including:

- Autonomous Vehicles
- Robotic Systems (e.g., Aerial Robots)
- Other Mechatronic Systems

Chapter 1

Introduction to optimal control

1.1 Optimal control problem formulation

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

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

- $x(t) \in \mathbb{R}^n$ 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 Finite-Difference Equations (FDE):

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

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

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

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 $t \in \{1, \ldots, T\}$ and the inputs u_t for all $t \in \{0, \ldots, T-1\}$, that is:

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

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 $t \in \{0, \ldots, T-1\}$. 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 $(x_t, u_t) = (x_e, u_e), \forall t \in \mathbb{N}_0$ 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 $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$, the linearization of (1.1) about $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ 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 $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$

$$A_t = \frac{\partial}{\partial x} f(\bar{x}_t, \bar{u}_t)\Big|_{(\bar{\mathbf{x}}, \bar{\mathbf{u}})} \qquad B_t = \frac{\partial}{\partial u} f(\bar{x}_t, \bar{u}_t)\Big|_{(\bar{\mathbf{x}}, \bar{\mathbf{u}})}$$

where $x_t \approx \bar{x}_t + \Delta x_t$ and $u_t \approx \bar{u}_t + \Delta u_t$.
($\bar{x_t}$ and $\bar{u_t}$ are the nominal trajectory values of $x_{t+1} = f_t(x_t, u_t)$ and Δx_t, Δu_t are the trajectory of the linearized system.)

1.1.1 Optimization

Main ingredients

- Decision variable: $z \in \mathbb{R}^d$
- Cost function: $\ell(z) : \mathbb{R}^d \to \mathbb{R}$ cost associated to decision z
- Constraints (constraint sets): for some given functions $h_i : \mathbb{R}^d \to \mathbb{R},$ and $g_j : \mathbb{R}^d \to \mathbb{R}$, the decision vector $z \in \mathbb{R}^d$ needs to satisfy

  $$\begin{aligned} h_i(z) &= 0 \quad i = 1, \dots, m \\ g_j(z) &\le 0 \quad j = 1, \dots, r \end{aligned}$$

  equivalently we can say that we require $z \in Z$ with

  $$Z = \{z \in \mathbb{R}^d | h(z) = 0, g(z) \le 0\},$$

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

Minimization

We can write our optimization problem as

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

where $h_i : \mathbb{R}^d \to \mathbb{R}$ and $g_j : \mathbb{R}^d \to \mathbb{R}$
We can write it more compactly as

$$\begin{aligned} &\min_{z \in \mathbb{R}^d} \ell(z) \\ \text{subj. to } & h(z) = 0 \\ & g(z) \le 0 \end{aligned}$$

where $h : \mathbb{R}^d \to \mathbb{R}^m$ and $g : \mathbb{R}^d \to \mathbb{R}^r$

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

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

  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 $\ell_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is called stage-cost, while $\ell_T : \mathbb{R}^n \to \mathbb{R}$ 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

$$\min_{\substack{x_0, x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$$

$$\begin{aligned} \text{subj. to} \quad & x_{t+1} = f_t(x_t, u_t), \quad t \in \{0, \dots, T-1\} \\ & r(x_0, x_T) = 0 \\ & g_t(x_t, u_t) \leq 0, \quad t \in \{0, \dots, T-1\} \end{aligned}$$

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

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

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

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

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

- Space of trajectories (or trajectory manifold)

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

Chapter 2

Nonlinear Optimization

2.1 Overview

1. **Motivating Examples**
    - Drone trajectory generation
    - Pendulum control
    - Trajectory manifold concept
2. **Theoretical Foundations**
    - Types of minima
        - Global/Local
        - Strict/Non-strict
    - Optimality conditions
        - First-order necessary
        - Second-order necessary
        - Second-order sufficient
3. **Convex Optimization**
    - Convex sets and functions
    - Constraint definitions
    - Properties
        - Local = Global minima
        - First-order sufficiency
    - Quadratic programs
4. **Solution Methods**
    - Iterative descent methods
        - Gradient methods
        - Quadratic minimization
    - Newton's method
        - Zero-finding
        - Unconstrained optimization
    - Step-size selection
        - Constant
        - Diminishing

    - Armijo rule

5. **Constrained Optimization**
    - Convex set optimization
        - Projection methods
        - Feasible directions
    - Equality/Inequality constraints
        - Active constraints
        - KKT conditions
        - Complementary slackness
    - Advanced methods
        - Sequential Quadratic Programming
        - Barrier functions

6. **Implementation**
    - Python implementation
    - Convergence visualization
    - Performance analysis

Nonlinear programming (NLP) is the process of solving an optimization problem where the objective function or some of the constraints are Non-Linear.

2.2 Unconstrained Optimization

Consider the unconstrained optimization problem

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

with $\ell : \mathbb{R}^d \to \mathbb{R}$ a cost function to be minimized and z a decision vector

We say that z^* is a

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

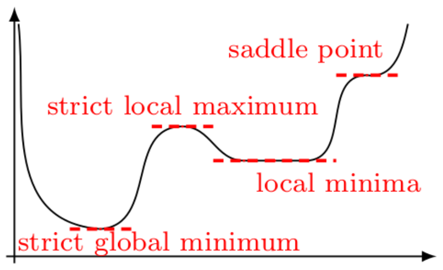


Figure 2.1: Illustration of different types of minima in an optimization problem

Notation

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

$$\ell(z^*) = \min_{z \in \mathbb{R}^d} \ell(z)$$

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

$$z^* = \arg\min_{z \in \mathbb{R}^d} \ell(z)$$

Gradient and Hessian

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

$$\nabla r(z) = \begin{bmatrix} \frac{\partial r(z)}{\partial z_1} \\ \vdots \\ \frac{\partial r(z)}{\partial z_d} \end{bmatrix} \in \mathbb{R}^{d \times 1}$$

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

$$\nabla^2(r(z)) = \begin{bmatrix} \frac{\partial^2 r(z)}{\partial z_1^2} & \cdots & \frac{\partial^2 r(z)}{\partial z_1 z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 r(z)}{\partial z_d z_1} & \cdots & \frac{\partial^2 r(z)}{\partial z_d^2} \end{bmatrix} \in \mathbb{R}^{d \times d}$$

The Hessian matrix is a symmetric matrix, since the assumption of continuity of the second derivatives implies that the order of differentiation does not matter.

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

$$\nabla r(z) = \begin{bmatrix} \nabla r_1(z) & \cdots & \nabla r_m(z) \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1(z)}{\partial z_1} & \cdots & \frac{\partial r_m(z)}{\partial z_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_1(z)}{\partial z_d} & \cdots & \frac{\partial r_m(z)}{\partial z_d} \end{bmatrix} \in \mathbb{R}^{d \times m}$$

which is the transpose of the Jacobian matrix of r

2.2.1 Conditions of optimality

First order Necessary condition (FNC) of optimality (unconstrained)

Let z^* be an unconstrained local minimum of $\ell : \mathbb{R}^d \to \mathbb{R}$ and assume that ℓ is continuously differentiable ($\mathcal{C}^1$) in $B(z^*, \varepsilon)$[1] for some $\varepsilon > 0$. Then $\nabla \ell(z^*) = 0$ (the gradient of ℓ at z^* is zero).

Second order Necessary condition (FNC) of optimality (unconstrained)

If additionally ℓ is twice continuously differentiable ($\mathcal{C}^2$) in $B(z^*, \varepsilon)$, then $\nabla^2 \ell(z^*) \geq 0$ (The Hessian of ℓ is positive semidifinite).

Remark. Points $\bar{z}$ satisfying $\nabla \ell(\bar{z}) = 0$ are called stationary points. They include minima, maxima and saddle points.

[1]Ball of radius ε centered in z^*

Second order Sufficient conditions of optimality (unconstrained)

Let $\ell : \mathbb{R}^d \to \mathbb{R} \in \mathcal{C}^2$ (twice differentiable) in $B(z^*, \varepsilon)$ for some $\varepsilon > 0$. Suppose that $z^* \in \mathbb{R}^d$ satisfies

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

Then z^* is a strict (unconstrained) local minimum of ℓ

Convex set

A set $Z \subset \mathbb{R}^d$ is convex if for any two points z_A and z_B in Z and for all $\theta \in [0,1]$, then

$$\theta z_A + (1-\theta) z_B \in Z$$

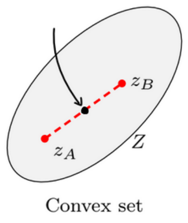


Figure 2.2: Convex set

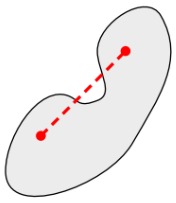


Figure 2.3: Non convex set

Convex functions

Let $Z \subset \mathbb{R}^d$ be a convex set. A function $\ell : Z \to \mathbb{R}$ is convex if for any two points z_A and z_B in Z and for all $\theta \in [0,1]$, then

$$\ell(\theta z_A + (1-\theta) z_B) \leq \theta \ell(z_A) + (1-\theta)\ell(z_B)$$

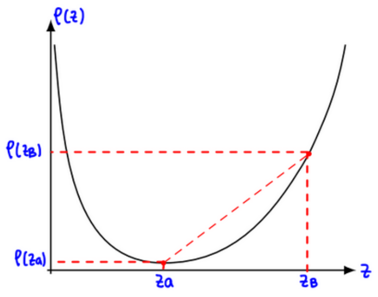


Figure 2.4: Convex function

Remark. A function ℓ is *concave* if $-\ell$ is convex. A function ℓ is strictly convex if the inequality holds strictly for $z_A \neq z_B$ and $\theta \in (0,1)$.

Inequality constraints and convex sets

Let $g : \mathbb{R}^d \to \mathbb{R}^p$, we can define a set $Z_{\text{ineq}} \subset \mathbb{R}^d$ as

$$Z_{\text{ineq}} = \{z \in \mathbb{R}^d | g(z) \leq 0\}$$

The set Z_{ineq} is convex iff g is a quasi-convex function (e.g., monotone functions on the axis)

Corollary. if g is convex then Z_{ineq} is convex and vice versa.

z_2
$az_1^2 + bz_2^2 - c \leq 0$
z_1
z_2
$a^\top z - c \leq 0$
z_1

Figure 2.5: Convex set identified through inequality constraints

Equality constraints and convex sets

Let $h : \mathbb{R}^d \to \mathbb{R}^p$, we can define a set $Z_{\text{eq}} \subset \mathbb{R}^d$ as

$$Z_{\text{eq}} = \{z \in \mathbb{R}^d | h(z) = 0\}$$

The set Z_{eq} is convex iff h is an affine function. Convex sets identified through equality constraints are linear spaces (hyperplanes).

Convexity and gradient monotonicity

If ℓ is differentiable and convex, then for all $z_A, z_B \in Z$ it holds

$$\ell(z_B) \geq \ell(z_A) + \nabla\ell(z_A)^T(z_B - z_A)$$

If ℓ is differentiable and convex, then its gradient $\nabla\ell : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$(\nabla\ell(z_A) - \nabla\ell(z_B))^T(z_A - z_B) \geq 0$$

for all z_A, z_B. That is, the gradient $\nabla\ell$ is a monotone operator.

Strict convexity and gradient monotonicity

A function ℓ is strictly convex if for $z_A \neq z_B$ and $\theta \in (0, 1)$

$$\ell(\theta z_A + (1-\theta)z_B) < \theta\ell(z_A) + (1-\theta)\ell(z_B)$$

If the strictly convex function ℓ is also differentiable, then its gradient satisfies

$$(\nabla\ell(z_A) - \nabla\ell(z_B))^T(z_A - z_B) > 0$$

for all z_A, z_B. That is, the gradient $\nabla\ell$ is a strictly monotone operator.

Strong convexity and gradient monotonicity

A function ℓ is strongly convex with parameter $\mu > 0$ if for $z_A \neq z_B$ and $\theta \in (0, 1)$

$$\ell(\theta z_A + (1-\theta)z_B) < \theta\ell(z_A) + (1-\theta)\ell(z_B) - \mu\theta(1-\theta)\|z_A - z_B\|^2$$

The gradient of a differentiable strongly convex function satisfies

$$(\nabla\ell(z_A) - \nabla\ell(z_B))^T(z_A - z_B) \geq \mu\|z_A - z_B\|^2$$

for all z_A, z_B. That is, the gradient $\nabla\ell$ is a strongly monotone operator.

Convexity and Lipschitz continuity of the gradient

Consider a differentiable convex function ℓ with a Lipschitz continuous gradient with parameter $L > 0$, i.e.,

$$\|\nabla\ell(z_A) - \nabla\ell(z_B)\| \leq L\|z_A - z_B\|$$

for all z_A, z_B.

Then, the following characterization holds

$$(\nabla\ell(z_A) - \nabla\ell(z_B))^T(z_A - z_B) \geq \frac{1}{L}\|\nabla\ell(z_A) - \nabla\ell(z_B)\|^2$$

for all z_A, z_B. That is, the gradient $\nabla\ell$ is a co-coercive operator.

Strong convexity and Lipschitz continuity of the gradient

Consider a strongly convex (with parameter $\mu > 0$) function ℓ with Lipschitz continuous gradient (with parameter $L > 0$)

Then the following characterization holds

$$(\nabla\ell(z_A) - \nabla\ell(z_B))^T(z_A - z_B) \geq \frac{\mu L}{\mu + L}\|z_A - z_B\|^2 + \frac{1}{\mu + L}\|\nabla\ell(z_A) - \nabla\ell(z_B)\|^2$$

for all z_A, z_B

Graphical interpretation of convexity

If ℓ is differentiable and convex, then

$$\ell(z_B) \geq \ell(z_A) + \nabla\ell(z_A)^T(z_B - z_A)$$

for all $z_A, z_B \in Z$

If ℓ is differentiable and μ-strongly convex, then

$$\ell(z_B) \geq \ell(z_A) + \nabla\ell(z_A)^T(z_B - z_A) + \frac{\mu}{2}\|z_B - z_A\|^2$$

for all $z_A, z_B \in Z$

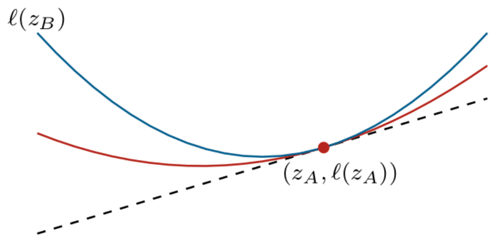


Figure 2.6: Strong convexity imposes a quadratic lower bound on ℓ

Graphical interpretation of Lipschitz continuity of the gradient

If ℓ is differentiable with Lipschitz continuous gradient, then

$$\ell(z_B) \leq \ell(z_A) + \nabla\ell(z_A)^T(z_B - z_A) + \frac{L}{2}\|z_B - z_A\|^2$$

for all $z_A, z_B \in Z$

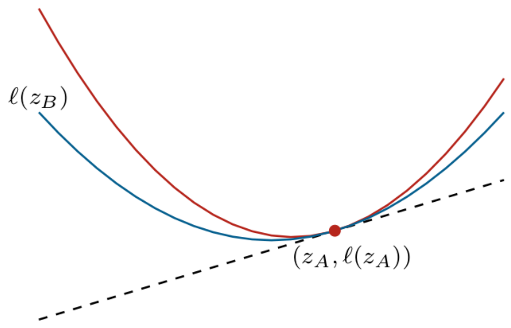


Figure 2.7: Lipschitz continuity of $\nabla\ell$ imposes a quadratic upper bound on ℓ

2.2.2 Minimization of convex functions

Proposition

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

2.3 Quadratic programming (unconstrained)

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

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

2.4 Unconstrained Optimization Algorithms

2.4.1 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 ℓ is decreased at each iteration, i.e.

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

two-step procedure

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

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

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 γ^k sufficiently small;
2. select a step-size γ^k guaranteeing a sufficient decrease.

In other references these are called line-search methods.

2.4.2 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 ℓ 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

2.4.3 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.

2.4.4 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

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

2.4.5 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 Δx^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

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

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

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

with Δx^k solution of the quadratic problem (2.1). Generalized version:

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

2.4.6 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 ℓ about x^k. In fact, consider the quadratic approximation $\ell^k(x)$ of ℓ 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}

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

2.4.7 step-size selection rules

- Constant step-size: $\gamma^k = \gamma > 0$
- Diminishing step-size: $\gamma^k \to 0$ as $k \to \infty$. 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 γ^k
- minimization rule
- Armijo rule

2.4.8 Armijo rule

Given the descent direction d^k we can consider

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

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

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

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

$$g(\gamma) = g(0) + g'(0)\gamma + o(\gamma)$$
$$\ell(x^k + \gamma d^k) = \ell(x^k) + \gamma \nabla \ell(x^k)^T d^k + o(\gamma)$$

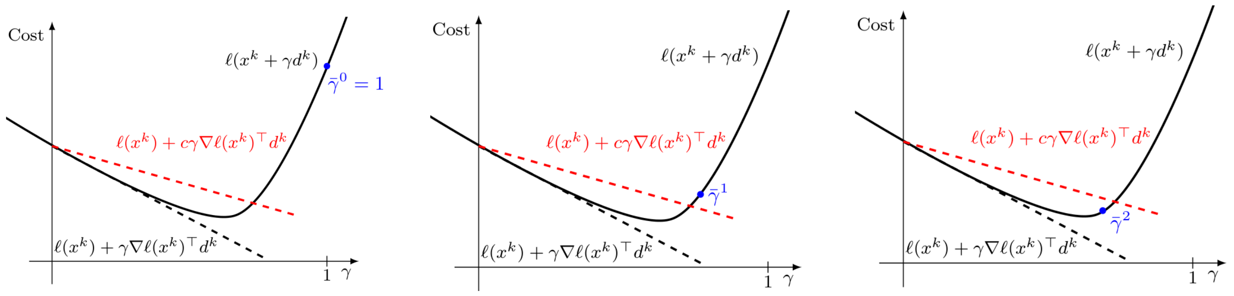
This is the tangent to the $g(\gamma)$ curve at $\gamma = 0$. We also consider the line

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

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

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



Cost
$\ell(x^k + \gamma d^k)$
$\bar{\gamma}^0 = 1$
$\ell(x^k) + c\gamma\nabla\ell(x^k)^\top d^k$
$\ell(x^k) + \gamma\nabla\ell(x^k)^\top d^k$
1
γ
Cost
$\ell(x^k + \gamma d^k)$
$\bar{\gamma}^1$
$\ell(x^k) + c\gamma\nabla\ell(x^k)^\top d^k$
$\ell(x^k) + \gamma\nabla\ell(x^k)^\top d^k$
1
γ
Cost
$\ell(x^k + \gamma d^k)$
$\bar{\gamma}^2$
$\ell(x^k) + c\gamma\nabla\ell(x^k)^\top d^k$
$\ell(x^k) + \gamma\nabla\ell(x^k)^\top d^k$
1
γ

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 $d_1 I \leq D^k \leq d_2 I, \quad d_1, d_2 > 0$. Assume that γ^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. $\nabla \ell(\bar{x}) = 0$

Remark. 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.

Convergence 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 $d_1 I \leq D^k \leq d_2 I, \quad d_1, d_2 > 0$. Assume that for some $L > 0$

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

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

1. $\gamma^k = \gamma > 0$ sufficiently small, or
2. $\gamma^k \to 0$ and $\displaystyle\sum_{t=0}^{\infty} \gamma^k = \infty$

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 $\ell(x^k) \to -\infty$ or $\ell(x^k)$ converges to a finite value and $\nabla \ell(x^k) \to 0$. In the second case, one can show that any subsequnce $\{x^{k_p}\}$ converges to some stationary point $\bar{x}$ satisfying $\nabla \ell(\bar{x}) = 0$
- Existence of minima can be guaranteed by excluding $\ell(x^k) \to -\infty$ via suitable assumptions. Assume, e.g., ℓ coercive (radially unboundend)
- For general (nonconvex) problems, assuming coercivity, only convergence (of subsequences) to stationary points can be proven.
- for convex programs, assuming coercivity, convergence to global minima is guaranteed since necessary conditions of optimality are also sufficient.

2.5 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 ℓ is continuously differentiable on X.

Optimality conditions

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

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

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.

2.5.1 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.

2.5.2 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 γ^k sufficiently small, $x^{k+1} \in X$

Barrier function strategy for inequality constraints

Consider the inequality constrained optimization problem

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

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

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

2.6 Constrained optimization: optimality conditions

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

Definition 2.6.1 (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 2.6.2 (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

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

Theorem 2.6.1 (Karush-Kuhn-Tucker necessary conditions)**.** Let x^* be a regular local minimum of

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

where ℓ, g_j and h_i are $\mathcal{C}^1$.
Then $\exists!$ μ_j^* and λ_i^*, called *Lagrange multipliers*, s.t.

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

Moreover, if ℓ, 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

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

Remark. The condition $\mu^* g_j^*(x^*) = 0, j \in \{1, \dots, r\}$, 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. ∇_1 denotes the gradient wrt the first variable of the function

Notation. ∇_{11} denotes the hessian of a function wrt the first variable

2.6.1 Quadratic programming (constrained)

Let us consider quadratic optimization problems with linear equality constraints

$$\min_{x \in \mathbb{R}^n} x^T Q x + q^T x$$
$$\text{subj. to } \; Ax = b$$

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:

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

And the gradient computes as

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

The equality constraints must also be enforced:

$$Ax^* - b = 0$$

We can note that

$$\nabla_2 \mathcal{L}(x^*, \lambda^*) = Ax - b$$

Therefore, first order conditions of optimality may be written as

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

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

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

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

$$\nabla h_i(x)^T y = 0 \qquad i \in \{1, \dots, m\} \qquad \Longrightarrow \qquad A^T y = 0$$

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

2.7 Constrained optimization: optimization algorithms

2.7.1 Newton's method for equality constrained problems

KKT points can be found by solving a root finding problem in variables x, λ wrt $r(x, \lambda) = \nabla \mathcal{L}(x, \lambda)$. 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

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

where

$$\nabla^2 \mathcal{L}(x^k, \lambda^k) = \begin{bmatrix} \nabla_{11} \mathcal{L}(x^*, \lambda^*) & \nabla_{12} \mathcal{L}(x^*, \lambda^*) \\ \nabla_{21} \mathcal{L}(x^*, \lambda^*) & \nabla_{22} \mathcal{L}(x^*, \lambda^*) \end{bmatrix} = \begin{bmatrix} H^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix}$$

$$\nabla \mathcal{L}(x^k, \lambda^k) = \begin{bmatrix} \nabla \ell(x^k) + \nabla h(x^k) \lambda^k \\ h(x^k) \end{bmatrix}$$

$$H^k = \nabla_{11}^2 \mathcal{L}(x^k, \lambda^k) \qquad \nabla_{11} \mathcal{L}(x, \lambda) = \nabla^2 \ell(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 h_i(x)$$

We can write

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

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

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

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

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

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

2.7.2 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 $(\Delta x^k, \Delta\lambda^k_{QP})$ from

$$\Delta x^k = \arg\min_{\Delta x} \nabla \ell(x^k)^T \Delta x + \frac{1}{2}\Delta x^T H^k \Delta x \tag{2.2}$$

$$\text{subj. to } \; h(x^k) + \nabla h(x^k)^T \Delta x = 0 \tag{2.3}$$

   with $\Delta\lambda^k_{QP}$ the Lagrange multiplier associated to the optimal solution of (2.2)

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

4. Update

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

2.7.3 Barrier function strategy for inequality constraints

Consider the inequality optimization problem

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

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

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

Implementation: every few iterations shrink the barrier parameters ε

Chapter 3

Optimality conditions for optimal control

3.1 boh

3.1.1 Dynamics as equality constraints

We consider nonlinear, discrete-time systems described by

$$x_{t+1} = f_t(x_t, u_t) \qquad t \in \mathbb{N}_0 \tag{3.1}$$

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

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

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 (3.1)

$$\begin{aligned} \mathcal{T} &:= \left\{((x), (u)) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | h\left((x), (u)\right) = 0\right\} \\ &= \left\{((x), (u)) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | x_{t+1} = f_t\left(x_t, u_t\right), t = 0, \dots, T-1\right\} \end{aligned}$$

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, $T_{(\bar{\mathbf{x}}, \bar{\mathbf{u}})}\mathcal{T} = \{(\mathbf{\Delta x}, \mathbf{\Delta u}) \in \mathbb{R}^{nT} \times \mathbb{R}^{mT} | \nabla_1 h(\mathbf{x}, \mathbf{u})^T \mathbf{\Delta x} + \nabla_2 h(\mathbf{x}, \mathbf{u})^T \mathbf{\Delta u} = 0\}$ is the set of trajectories $(\mathbf{\Delta x}, \mathbf{\Delta u})$ of

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

with

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

3.2 Unconstrained optimal control problem (d-t)

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

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

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 continuosly differentiable, i.e. they are class $\mathcal{C}^2$ Consider the discrete-time system (3.1). We can introduce the compact notation

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

which allows us to write the cost function compactly as

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

and the equality constraint represented by the dynamics

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

In light of this compact notation, we can rewrite the optimal control law problem as

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{nT},\mathbf{u}\in\mathbb{R}^{mT}} & \ell(\mathbf{x},\mathbf{u}) \\ \text{subj. to } & h(\mathbf{x},\mathbf{u}) = 0 \end{aligned}$$

where $\ell : \mathbb{R}^{nT} \times \mathbb{R}^{mT} \to \mathbb{R}$ and $h : \mathbb{R}^{nT} \times \mathbb{R}^{mT} \to \mathbb{R}^{nT}$. This is a constrained nonlinear optimization problem with decision variable $(\mathbf{x},\mathbf{u})$

3.3 KKT conditions for unconstrained optimal control

the Lagrangian function has the form

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

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

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

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

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

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

Let us note that

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

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

$$\begin{aligned} \nabla_2 \mathcal{L}_0(x_0, u_0, \boldsymbol{\lambda}) = 0 \qquad & \nabla_2 \ell_0(x_0, u_0) + \nabla_2 f_0(x_0, u_0)\lambda_1 \\ \begin{bmatrix} \nabla_1 \mathcal{L}_t(x_t, u_t, \boldsymbol{\lambda}) \\ \nabla_2 \mathcal{L}_t(x_t, u_t, \boldsymbol{\lambda}) \end{bmatrix} = 0 \qquad & \begin{bmatrix} \nabla_1 \ell_t(x_t, u_t) + \nabla_1 f_t(x_t, u_t)\lambda_{t+1} - \lambda_t \\ \nabla_2 \ell_t(x_t, u_t) + \nabla_2 f_t(x_t, u_t)\lambda_{t+1} \end{bmatrix} = 0 \quad t = 1, \dots, T-1 \\ \nabla_1 \mathcal{L}_t(x_t, \boldsymbol{\lambda}) = 0 \qquad & \nabla \ell_T(x_T) - \lambda_T = 0 \end{aligned}$$

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:

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

3.3.1 Indirect methods for optimal control

Based on solving the optimality conditions:

- Guess some $u_t^0, \quad t = 0, \dots, T-1 \quad k = 0$
- run "forward"
  $$x_{t+1}^0 = f_t(x_t^0, u_t^0) \quad x_0 = x_{\text{init}}$$
- run "backward"
  $$\lambda_t^0 = A_t^T \lambda_{t+1}^0 + a_t^0$$
  starting from $\lambda_T^0 = \nabla \ell(x_T^0)$
- 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

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

where

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

The Lagrangian function has the form

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

where

$$\begin{aligned}\mathcal{L}_0(x_0,u_0,\boldsymbol{\lambda},\boldsymbol{\mu}) &= \ell_0(x_0,u_0)+\lambda_{d,1}^T f_0(x_0,u_0)+\lambda_{b,0}r_0(x_0)\\ \mathcal{L}_t(x_t,u_t,\boldsymbol{\lambda}) &= \ell_t(x_t,u_t)+\lambda_1^T f_t(x_t,u_t)-\lambda_t x_t+\mu_t^T g_t(x_t,u_t)\\ \mathcal{L}_T(x_T,\boldsymbol{\lambda}) &= \ell_T(x_T)-\lambda_T^T x_T+\lambda_{b,T}^T r_T(x_T)\end{aligned}$$

and we assumed $r(x_0,x_T)=\text{col}(r_0(x_0),r_T(x_T))$ and $\lambda_b=\text{col}(\lambda_{b,0},\lambda_{b,T})$.

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

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

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

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

$$\begin{aligned}\begin{bmatrix}\nabla_1\mathcal{L}_0(x_0,u_0,\boldsymbol{\lambda},\boldsymbol{\mu})\\ \nabla_2\mathcal{L}_0(x_0,u_0,\boldsymbol{\lambda},\boldsymbol{\mu})\end{bmatrix}=0 &\qquad \begin{bmatrix}\nabla_1\ell(x_0,u_0)\nabla_1 f_0(x_0,u_0)\lambda_1+\nabla r_0(x_0)\lambda_{b,0}+\nabla_1 g_t(x_0,\mu_0)\\ \nabla_2\ell_0(x_0,u_0)\nabla_2 f_0(x_0,u_0)\lambda_1+\nabla_2 g_t(x_0,u_0)\mu_0\end{bmatrix}=0\\ \begin{bmatrix}\nabla_1\mathcal{L}_t(x_t,u_t,\boldsymbol{\lambda})\\ \nabla_2\mathcal{L}_t(x_t,u_t,\boldsymbol{\lambda})\end{bmatrix}=0 &\qquad \begin{bmatrix}\nabla_1\ell_t(x_t,u_t)+\nabla_1 f_t(x_t,u_t)\lambda_{t+1}-\lambda_t+\nabla_1 g_t(x_t,u_t)\mu_t\\ \nabla_2\ell_t(x_t,u_t)+\nabla_2 f_t(x_t,u_t)\lambda_{t+1}+\nabla_2 g_t(x_t,u_t)\mu_t\end{bmatrix}=0 \quad t=1,\dots,T-1\\ \nabla_1\mathcal{L}_t(x_t,\boldsymbol{\lambda})=0 &\qquad \nabla\ell_T(x_T)-\lambda_T+\nabla r_T(x_T)\lambda_{b,T}=0\end{aligned}$$

for $(x_t,u_t,\lambda_t,\mu_t)=(x_t^*,u_t^*,\lambda_t^*,\mu_t^*)$

Chapter 4

Linear Quadratic (LQ) optimal control

Consider a linear quadratic optimal control problem as:

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

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

4.1 First order optimality condition

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

therefore

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

Remark. second order optimality conditions

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

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, 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

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

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 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

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

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

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

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

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

and because

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

then

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

so

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

from which

$$P_t = A_t^T P_{t+1} A_t - A_t^T P_{t+1} B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t + Q_t \tag{4.1}$$

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

Other formulations of the Riccati equation

The usual Riccati recursion reads:

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

by exploiting the matrix inversion lemma[1], we can write:

$$\begin{aligned}
P_t &= Q_t + A_t^T P_{t+1} A_t - A_t^T P_{t+1} B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t \\
P_t &= Q_t + A_t^T P_{t+1} (I - B_t (R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1}) A_t \\
P_t &= Q_t + A_t^T P_{t+1} (I - B_t ((I + B_t^T P_{t+1} B_t R_t^{-1}) R_t)^{-1} B_t^T P_{t+1}) A_t \\
P_t &= Q_t + A_t^T P_{t+1} (I - B_t R_t^{-1} (I + B_t^T P_{t+1} B_t R_t^{-1})^{-1} B_t^T P_{t+1}) A_t \\
P_t &= Q_t + A_t^T P_{t+1} (I - B_t R_t^{-1} B_t^T P_{t+1}) A_t \\
P_t &= Q_t + A_t^T (I - B_t R_t^{-1} B_t^T P_{t+1}) P_{t+1} A_t
\end{aligned}$$

4.2 Infinite horizion LQ optimal control

Consider the infinite-horizon optimal control problem

$$\begin{aligned}
\min_{\substack{x_1, x_2, \dots \\ u_0, u_1, \dots}} \; & \sum_{t=0}^{\infty} \frac{1}{2} \left[x_t^T Q x_t + u_t^T R u_t \right] \\
\text{subj. to } & x_{t+1} = A x_t + B u_t \quad t = 0, 1, \dots \\
& x_0 = x_{\text{init}}
\end{aligned}$$

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 guardantees 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.

[1] $(A + BC)^{-1} = A^{-1} - A^{-1} B (I + C A^{-1} B)^{-1} C A^{-1}$

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_∞ equilibrium solution of the Difference Riccati Equation. That is, P_∞ 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:
$$\begin{aligned} K^* &= -(R + B^T P_\infty B)^{-1}(B^T P_\infty A) \\ u_t^* &= K^* x_t^* \\ x_{t+1}^* &= Ax_t^* + Bu_t^* \quad t = 1, 2, \ldots \quad x_0^* = X_{\text{init}} \end{aligned}$$

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

4.3 Affine LQR

Consider a LQR problem with affine cost and affine dynamics
$$\begin{aligned} \min_{\substack{x_1,\ldots,x_T \\ u_0,\ldots,u_{T-1}}} & \sum_{t=0}^{T-1} \begin{bmatrix} q_t \\ r_t \end{bmatrix}^T \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T \begin{bmatrix} Q_t & S_t^T \\ S_t & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + q_t^T x_T + x_T^T Q_T x_T \\ \text{subj. to } & x_{t+1} = A_t x_t + B_t u_t + c_t \quad t = 0, \ldots, T-1 \end{aligned}$$
where $Q_t = Q_t^T \geq 0$, $R_t = R_t^T > 0$, S_t such that the problem is convex, i.e. $Q_t - S_t^T R_t^{-1} S_t \geq 0$

This problem can be conveniently solved by augmenting the state as
$$\tilde{x}_t := \begin{bmatrix} 1 \\ x_t \end{bmatrix}$$
we can rewrite the cost and system matrices as
$$\tilde{Q}_t := \begin{bmatrix} 0 & q_t^T \\ q_t & Q_t \end{bmatrix} \quad \tilde{S}_t := \begin{bmatrix} r_t & S_t \end{bmatrix} \quad \tilde{R}_t := R_t \quad \tilde{A}_t := \begin{bmatrix} 1 & 0 \\ c_t & A_t \end{bmatrix} \quad \tilde{B}_t := \begin{bmatrix} 0 \\ B_t \end{bmatrix}$$
Then, we solve the associated LQR problem
$$\begin{aligned} \min_{\substack{x_1,\ldots,x_T \\ u_0,\ldots,u_{T-1}}} & \sum_{t=0}^{T-1} \begin{bmatrix} \tilde{x}_t \\ u_t \end{bmatrix}^T \begin{bmatrix} \tilde{Q}_t & \tilde{S}_t^T \\ \tilde{S}_t & \tilde{R}_t \end{bmatrix} \begin{bmatrix} \tilde{x}_t \\ u_t \end{bmatrix} + \tilde{x}_T^T Q_T \tilde{x}_T \\ \text{subj. to } & \tilde{x}_{t+1} = \tilde{A}_t \tilde{x}_t + \tilde{B}_t u_t \quad t = 0, \ldots, T-1 \\ & x_0 = x_{\text{init}} \end{aligned}$$
The optimal solution of the problem reads
$$\begin{aligned} u_t^* &= K_t^* x_t^* + \sigma_t^* \\ x_{t+1}^* &= A_t x_t^* + B_t u_t^* \end{aligned}$$
where
$$\begin{aligned} K_t^* &= -(R_t + B_t^T P_{t+1} B_t)^{-1}(S_t + B_t^T P_{t+1} A_t) \\ \sigma_t^* &= -(R_t + B_t^T P_{t+1} B_t)^{-1}(r_t + B_t^T p_{t+1} + B_t^T P_{t+1} c_t) \end{aligned}$$

$$\begin{aligned} p_t &= q_t + A_t^T P_{t+1} + A_t^T P_{t+1} C_t + K_t^{*T}(R_t + B_t^T P_{t+1} B_t)\sigma_t^* \\ P_T &= Q_t + A_t^T P_{t+1} A_t - K_t^{**} T(R_t + B_t^T P_{t+1} B_t) K_t^* \end{aligned}$$
with $P_T = Q_T$ and $p_T = q_T$

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:

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

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

$\vdots$

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

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

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

$$\begin{aligned} g_0(u_0) &\leq 0 \\ &\vdots \\ g_{T-1}(u_{T-1}) &\leq 0 \end{aligned}$$

the problem becomes

$$\begin{aligned} &\min_{\mathbf{u}\in\mathbb{R}^{mT}} J(\mathbf{u}) \\ &\text{subj to } g_0(u_0) \leq 0 \\ &\qquad\vdots \\ &\qquad g_{T-1}(u_{T-1}) \leq 0 \end{aligned}$$

Remark. if we have constraints of the type

$$\begin{aligned} &g_0(x_0, u_0) \leq 0 \\ &\vdots \\ &g_{T-1}(x_{T-1}, u_{T-1}) \leq 0 \end{aligned}$$

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. Consider

$$J(\mathbf{u}) = \ell(\phi(\mathbf{u}), \mathbf{u})$$

Suppose $\mathbf{u} \in \mathbb{R}$, we have:

$$\frac{d}{d\mathbf{u}} J(\bar{\mathbf{u}}) = \left.\frac{\partial \ell(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\phi(\bar{\mathbf{u}}) \\ \mathbf{u}=\bar{\mathbf{u}}}} \left.\frac{\partial \phi(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\bar{\mathbf{u}}} + \left.\frac{\partial \ell(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\phi(\bar{\mathbf{u}}) \\ \mathbf{u}=\bar{\mathbf{u}}}}$$

In general, we have $\mathbf{u} \in \mathbb{R}^{mT}$, Therefore

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

However, notice that the calculation of $\nabla J(\mathbf{u})$ requires $\nabla\phi(\mathbf{u})$, which may be difficult to compute.

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

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

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

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

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

$$h(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_0(x_0, u_0) - x1 \\ \vdots \\ f_{T-1}(x_{T-1} u_{T-1}) - x_T \end{bmatrix}$$

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

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

Therefore

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

hence we can write

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

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

$$\nabla_1 \ell(\Phi(\mathbf{u}), \mathbf{u}) + \nabla_1 h(\Phi(\mathbf{u}), \mathbf{u})\boldsymbol{\lambda}(\mathbf{u}) = 0$$

which leads to

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

By recalling that

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

We have

$$\nabla J(\mathbf{u}) = \nabla\phi(\mathbf{u})\nabla_1 \mathcal{L}(\Phi(\mathbf{u}), \mathbf{u}, \boldsymbol{\lambda}) + \nabla_2 \mathcal{L}(\Phi(\mathbf{u}), \mathbf{u}, \boldsymbol{\lambda})$$

so that choosing $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{u})$ such that

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

it holds

$$\nabla J(\mathbf{u}) = \nabla_2 \mathcal{L}(\Phi(\mathbf{u}), \mathbf{u}, \boldsymbol{\lambda}(\mathbf{u}))$$

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 $\boldsymbol{\lambda}^*$ such that

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

it holds

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

5.2.2 Explicit computation of $\nabla J(\mathbf{u})$

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

By writing out the sum we can easily calculate $\nabla_1\mathcal{L}(\phi(\mathbf{u},\mathbf{u},\mathbf{x})) = 0$ for each x_t:

$$\begin{gathered}\mathcal{L}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \ell_0(x_0,u_0) + \lambda_1^T f_0(x_0,u_0) - \lambda_1 x_1 + \ell_1(x_1,u_1) + \lambda_2^T f_1(x_1,u_1) - \lambda_2 x_2 + \dots \\ \nabla_t\ell_t(x_t,u_t) + \nabla_t f_t(x_t,u_t)\lambda_{t+1} - \lambda_t = 0 \\ \nabla\ell_T(x_T) - \lambda_T = 0\end{gathered}$$

which leads to

$$\begin{aligned}\lambda_T &= \nabla\ell_T(x_T) \\ \lambda_t &= \nabla_t\ell_t(x_t,u_t) + \nabla_t f_t(x_t,u_t)\lambda_{t+1} \quad t = T-1,\dots,1\end{aligned}$$

then

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

Notice we can write

$$\begin{aligned}A_t^T &= \nabla_1 f(x_t,u_t) \\ B_t^T &= \nabla_2 f(x_t,u_t)\end{aligned}$$

so that we can rewrite

$$\begin{aligned}\lambda_t &= A_t^T\lambda_t + 1 + a_t \\ (\nabla J(u))_t &= B_t^T\lambda_{t+1} + b_t\end{aligned}$$

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.

Remark. $A_t \in \mathbb{R}^{n\times n}$ and $B_t \in \mathbb{R}^{n\times m}$ are the state and input matrices of the linearization of the system $x_{t+1} = f(x_t,u_t)$ about the trajectory $(\mathbf{x},\mathbf{u})$

5.3 Optimality conditions for unconstrained optimal control

Optimality conditions for the optimal control problem can then be written as follows: let $(\mathbf{x}^*,\mathbf{u}^*)$ be an optimal (minimum) state-input trajectory. Then there exists $\boldsymbol{\lambda}$ such that

$$\begin{aligned}x_{t+1}^* &= f_t(x_t^*,u_t^*) && t = 0,\dots,T-1 \\ \lambda_t^* &= \nabla_1\ell_t(x_t^*,u_t^*) + \nabla_1 f_t(x_t^*,u_t^*)\lambda_{t+1}^* && t = T-1,\dots,1 \\ 0 &= \nabla_2\ell_t(x_t^*,u_t^*) + \nabla_2 f_t(x_t^*,u_t^*)\lambda_{t+1}^* && t = 0,\dots,T-1\end{aligned}$$

with $x_0* = x_{\text{init}}$ and $\lambda_T^* = \nabla\ell_T(x_T^*)$. In order to find "stationary trajectories" we must solve the system of equations above in $(\mathbf{x},\mathbf{u},\boldsymbol{\lambda})$

5.3.1 Hamiltonian function and optimality conditions

It is customary to write these conditions in terms of the *Hamiltonian function*:

$$H_t(x_t,u_t,\lambda_{t+1}) = \ell_t(x_t,u_t) + \lambda_{t+1}^T f(x_t,u_t)$$

The first-order necessary conditions of optimality can then be written in terms of the Hamiltonian, so that we get the following proposition:

Proposition 5.3.1 (First order conditions of optimality for an unconstrained optimal control problem). Let $(\mathbf{x}^*, \mathbf{u}^*)$ be a local (minimum) optimal trajectory. Then

$$\nabla_2 H_t(x_t^*, u_t^*, \lambda_{t+1}^*) = 0 \qquad\qquad t = 0, \dots, T-1$$

where the costate vector $\boldsymbol{\lambda}^*$ is obtained from the adjoint equation

$$\lambda_t^* = \nabla_1 H_t(x_t^*, u_t^*, \lambda_{t+1}^*) t = 1, \dots, T-1$$

with terminal condition

$$\lambda_T^* = \nabla \ell_T(x_T^*)$$

Remark. Notice that we do note need to assume any "regularity" condition on the local optimal trajectory

It is worth noting that the constraint due to the dynamics can be written as

$$x_{t+1}^* = \nabla_3 H_t(x_t^*, u_t^*, \lambda_{t+1}^*)$$

Thus denoting

$$\nabla H_t(x_t^*, u_t^*, \lambda_{t+1}^*) = \begin{bmatrix} \nabla_1 H_t(x_t^*, u_t^*, \lambda_{t+1}^*) \\ \nabla_2 H_t(x_t^*, u_t^*, \lambda_{t+1}^*) \\ \nabla_3 H_t(x_t^*, u_t^*, \lambda_{t+1}^*) \end{bmatrix}$$

The first-order necessary conditions of optimality can be compactly written as

$$\begin{bmatrix} x_{t+1}^* \\ 0 \\ \lambda_t^* \end{bmatrix} = \nabla H_t(x_t^*, u_t^*, \lambda_{t+1}^*) \quad t = 0, \dots, T-1$$

with the additional condition

$$x_1^* = f(x_{\text{init}}, u_0) \quad \text{and} \quad \lambda_T^* = \nabla \ell_T(x_T^*)$$

Chapter 6

Newton's method for Optimal Control

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

$$\min_{\substack{\mathbf{x}\in\mathbb{R}^{nT}\\ \mathbf{u}\in\mathbb{R}^{mT}}} \sum_{t=0}^{T-1} \ell_t(x_t,u_t) + \ell_T(x_T)$$

with given initial condition $x_0 = x_{\text{init}} \in \mathbb{R}^n$, where $\ell_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the so-called stage cost while $\ell_T : \mathbb{R}^n \to \mathbb{R}$ is the terminal cost. In this chapter, we will use the following assumption:
Assumption: Functions $\ell_t(\cdot,\cdot), \ell_T(\cdot), f_t(\cdot,\cdot)$ are $\mathcal{C}^2$

6.1 Newton's method for unconstrained optimization

Given an unconstrained problem:

$$\begin{gathered}\min_z \ell(z)\\ \text{subj. to } z \in \mathbb{R}^d\end{gathered}$$

where $\ell : \mathbb{R}^d \to \mathbb{R}$, Newton's method applied to this problem reads:

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

where $\gamma^k > 0$ chosen according to Arimjo, constant or diminishing rules. The descent direction Δz^k can be calculated as the solution of the QP

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

Newton's method applied to the reduced optimization problem (see section 5.1) reads as:

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \gamma^k \Delta\mathbf{u}^k$$

where

$$\Delta\mathbf{u}^k = \arg\min_{\Delta u} \nabla J(\mathbf{u}^k)^T \Delta\mathbf{u} + \frac{1}{2}\Delta\mathbf{u}^T \nabla^2 J(\mathbf{u}^k)\Delta\mathbf{u}$$

6.2 Calculation of $\nabla^2 J(u)$

In order to calculate $\nabla^2 J(\mathbf{u})$ we move from:

$$\nabla J(\mathbf{u}) = \nabla\phi(\mathbf{u})\nabla_1\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda}) + \nabla_2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})$$

differentiating again this equation we have:

$$\begin{aligned}\nabla^2 J(\mathbf{u}) =& \nabla^2\phi(\mathbf{u})\nabla_1\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda}) + \nabla\phi(\mathbf{u})\nabla_{11}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\nabla\phi(\mathbf{u})^T + \nabla\phi(\mathbf{u})\nabla_{12}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\\ &+ \nabla_{22}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda}) + \nabla_{21}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\nabla\phi(\mathbf{u})^T\\ =&\nabla\phi(\mathbf{u})\nabla_{11}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\nabla\phi(\mathbf{u})^T + \nabla\phi(\mathbf{u})\nabla_{12}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\\ &+ \nabla_{22}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda}) + \nabla_{21}^2\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda})\nabla\phi(\mathbf{u})^T\end{aligned}$$

since $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{u})$ such that $\nabla_1\mathcal{L}(\phi(\mathbf{u}),\mathbf{u},\boldsymbol{\lambda}) = 0$

In this case, we also need to take into account $\nabla\phi(\mathbf{u})$.

6.3 Algorithm

Step 1 : Descent direction calculation

Evaluate $\nabla_1 f_t(x_t^k,u_t^k), \nabla_2 f_t(x_t^k,u_t^k), \nabla_1\ell_t(x_t^k,u_t^k), \nabla_2\ell_t(x_t^k,u_t^k), \nabla\ell_T(x_T^k)$
Solve backwards the co-state equation, with $\lambda_T^k = \nabla\ell_T(x_T^k)$

$$\lambda_t^k = \nabla_1 f_t(x_t^k,u_t^k)\lambda_{t+1}^k + \nabla_1\ell(x_t^k,u_t^k) \quad t = T-1,\dots,1$$

Compute for all $t = 0,\dots,T-1$

$$Q_t^k := \nabla_{11}^2\ell_t(x_t^k,u_t^k) + \nabla_{11}^2 f_t(x_t^k,u_t^k)\cdot\lambda_{t+1}^k, \quad R_t^k := \nabla_{22}^2\ell_t(x_t^k,u_t^k) + \nabla_{22}^2 f_t(x_t^k,u_t^k)\cdot\lambda_{t+1}^k$$
$$S_t^k := \nabla_{22}^2\ell_t(x_t^k,u_t^k) + \nabla_{22}^2 f_t(x_t^k,u_t^k)\cdot\lambda_{t+1}^k$$

and $Q_T^k := \nabla_{22}^2\ell_T(x_T^k)$. Compute the descent direction solving

$$\begin{aligned}\min_{\boldsymbol{\Delta x},\boldsymbol{\Delta u}} \sum_{t=0}^{T-1} &\begin{bmatrix}\nabla_1\ell_t(x_t^k,u_t^k)\\ \nabla_1\ell_t(x_t^k,u_t^k)\end{bmatrix}^T \begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix}^T \begin{bmatrix}Q_t^k & S_t^{k\,T}\\ S_t^k & R_t^k\end{bmatrix}\begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix} + \nabla\ell_T(x_T^k)^T\Delta x_T + \frac{1}{2}\Delta x_T^T Q_T^k\Delta x_T\\ \text{subj to } &\Delta x_{t+1} = \nabla_1 f_t(x_t^k,u_t^k)^T\Delta x_t + \nabla_2 f_t(x_t^k,u_t^k)^T\Delta u_t \quad t=0,\dots,T-1\\ &\Delta x_0 = 0\end{aligned}$$

Step 2: compute new input sequence

Implement setp-size selection rule

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

Step 3: compute new state trajectory

Forward integrate (open-loop), for all $t = 0,\dots,T-1$, with $x_0^{k+1} = x_{\text{init}}$

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

6.3.1 Regularization

In many cases the matrices Q_t^k, R_t^k and S_t^k may not be positive definite. A viable choice is to implement a "first order" update solving the affine LQR problem using, in their place, the matrices

$$\begin{aligned}\tilde{Q}_t^k &:= \nabla_{11}^2\ell_t(x_t^k,u_t^k), & \tilde{R}_t^k &:= \nabla_{22}^2\ell_t(x_t^k,u_t^k)\\ \tilde{S}_t^k &:= \nabla_{12}^2\ell_t(x_t^k,u_t^k), & \tilde{Q}_T^k &:= \nabla^2\ell_T(x_T^k)\end{aligned}$$

6.3.2 Robustified step 2 update: closed-loop state-input trajectory update

Idea: approximate $\Delta x_t \approx x_t^{k+1} - x_t^k$
Initialization: consider an initial guess trajectory $(\mathbf{x}^0, \mathbf{u}^0)$
For each iter k,
Step 1: solve the affine LQR

$$\begin{aligned}\min_{\boldsymbol{\Delta}\mathbf{x},\boldsymbol{\Delta}\mathbf{u}} \sum_{t=0}^{T-1} \begin{bmatrix}\nabla_1 \ell_t(x_t^k, u_t^k)\\ \nabla_1 \ell_t(x_t^k, u_t^k)\end{bmatrix}^T \begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix}^T \begin{bmatrix}Q_t^k & S_t^{k\,T}\\ S_t^k & R_t^k\end{bmatrix}\begin{bmatrix}\Delta x_t\\ \Delta u_t\end{bmatrix} + \nabla \ell_T(x_T^k)^T \Delta x_T + \frac{1}{2}\Delta x_T^T Q_T^k \Delta x_T\\ \text{subj to } \Delta x_{t+1} = \nabla_1 f_t(x_t^k, u_t^k)^T \Delta x_t + \nabla_2 f_t(x_t^k, u_t^k)^T \Delta u_t \quad t = 0, \dots, T-1\\ \Delta x_0 = 0\end{aligned}$$

and obtain gain matrices K_t^k and feed-forward actions σ_t^k for all $t = 0, \dots, T-1$
Step 2: compute new state-input trajectory
Forward integrate(closed-loop), for all $t = 0, \dots, T-1$ with $x_0^{k+1} = x_{\text{init}}$

$$\begin{aligned}u_t^{k+1} &= u_t^k + \gamma^k(\sigma_t^k + K_t^k \Delta x_t^k) + K_t^k(x_t^{k+1} - x_t^k - \gamma^k \Delta x_t)\\ x_{t+1}^{k+1} &= f_t(x_t^{k+1}, u_t^{k+1})\end{aligned}$$

which results as

$$\begin{aligned}u_t^{k+1} &= u_t^k + K_t^k(x_t^{k+1} - x_t^k) + \gamma^k \sigma_t^k\\ x_{t+1}^{k+1} &= f_t(x_t^{k+1}, u_t^{k+1})\end{aligned}$$

Chapter 7

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)

7.1 Main strategy idea over a finite horizon

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

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

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 $(\mathbf{x}^{opt}, \mathbf{u}^{opt})$.

7.2 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:

$$A_t^{opt} := \nabla_1 f_t(x_t^{opt}, u_t^{opt})^T \tag{7.1}$$

$$B_t^{opt} := \nabla_2 f_t(x_t^{opt}, u_t^{opt})^T \tag{7.2}$$

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

$$\begin{aligned}
\min_{\substack{\Delta x_1,\dots\Delta,x_T\\ \Delta u_0 u,\dots,\Delta u_{T-1}}} \sum_{t=0}^{\infty} & \Delta x_t^T Q_t^{\text{reg}} \Delta x_t + \Delta u_t^T R_t^{\text{reg}} \Delta u_t + \Delta x_T^T Q_T^{\text{reg}} \Delta x_T \\
\text{subj. to } & \Delta x_{t+1} = A_t^{opt} \Delta x_t + B_t^{opt} \Delta u_t \quad t = 0, 1, \dots \\
& \Delta x_0 = 0
\end{aligned}$$

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

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

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

$$K_t^{reg} := -(R_t^{reg} + B_t^{opt^T} P_{t+1} B_t^{opt})^{-1} (B_t^{opt^T} P_{t+1} A_t^{opt})$$

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

$$\begin{aligned}
u_t &= u_t^{opt} + K_t^{reg}(x_t - x_t^{opt}) \\
x_{t+1} &= f_t(x_t, u_t)
\end{aligned}$$

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)

7.3 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 8

Continuous time Optimal Control

A continuous-time optimal control problem can be written as

$$\begin{aligned}\min_{(x(\cdot),u(\cdot))\in\mathcal{F}_x\times\mathcal{F}_u} &\int_0^T \ell_\tau(x(\tau),u(\tau))d\tau + \ell_T(x(T))\\ \text{subj to } &\dot{x}(t) = f_t(x(t),u(t)) \quad \forall t\in[0,T]\\ &u(t)\in\mathcal{U}\\ &x(0) = x_{init}\end{aligned}$$

Remark. $\mathcal{F}_x, \mathcal{F}_u$ are function spaces

8.1 Hamiltonian function

Let us introduce the *Hamiltonian function* $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$

$$H(x(t),u(t),\lambda(t),t) = \ell_t(x(t),u(t)) + \lambda(t)^T f_t(x(t),u(t))$$

where $\lambda : [0,T] \to \mathbb{R}$. We denote the partial derivatives of $H(\cdot)$ wrt its arguments evaluated at a certain t as

$$\begin{aligned}H_x(x(t),u(t),\lambda(t),t) &:= \frac{\partial}{\partial x}H(x(t),u(t),\lambda(t),t)\\ H_u(x(t),u(t),\lambda(t),t) &:= \frac{\partial}{\partial u}H(x(t),u(t),\lambda(t),t)\\ H_\lambda(x(t),u(t),\lambda(t),t) &:= \frac{\partial}{\partial \lambda}H(x(t),u(t),\lambda(t),t)\end{aligned}$$

8.2 Pontryagin Minimum Principle

Let $(x^*(t),u^*(t)), t\in[0,T]$ be an optimal trajectory where $u(t)\in\mathcal{U}$, for all t, then

1. There exists a $\lambda^*(t)$ such that, for all $t\in[0,T]$:

   $$\dot{\lambda}^*(t) = -H_x(x^*(t),u^*(t),\lambda^*(t),t)^T$$

   where $\lambda^*(T) = \nabla\ell_T(x_T^*)$

2. The Hamiltonian function assumes its minimum in $\mathcal{U}$ when evaluated along $u^*(t)$, i.e.

   $$H(x^*(t),u^*(t),\lambda^*(t),t) \leq H(x^*(t),u(t),\lambda^*(t),t) \qquad \forall u(t)\in\mathcal{U},\ \forall t\in[0,T]$$

3. The optimal trajectory $(x^*(t),u^*(t)), t\in[0,T]$ satisfies, $\forall t\in[0,T]$

   $$\dot{x}(t) = H_\lambda(x^*(t),u^*(t),\lambda(t)^*,t)^T$$

   where $x^*(0) = x_{\text{init}}$

8.3 Continuous-time LQ Optimal Control Problem

Consider a linear quadratic optimal control problem as:

$$\begin{aligned}\min_{(x(\cdot),u(\cdot))\in\mathcal{F}_x\times\mathcal{F}_u} & \int_0^T \left(\frac{1}{2}x(\tau)^TQ(\tau)x(\tau)+\frac{1}{2}u(\tau)^TR(\tau)u(\tau)\right)d\tau+\frac{1}{2}x(T)^TQ_Tx(T)\\ \text{subj to } & \dot{x}(t)=A(t)x(t)+B(t)u(t)\quad \forall t\in[0,T]\\ & x(0)=x_{init}\end{aligned}$$

- $Q(t)=Q(t)^T\geq 0 \quad \forall t\in[0,T]$
- $Q(T)=Q(T)^T\geq 0 \quad \forall$
- $R(t)=R(t)^T>0 \quad \forall t\in[0,T]$

For the LQ optimal control problem we can define the Hamiltonian function as

$$H(x(t),u(t),\lambda(t),t)=\frac{1}{2}x(t)^TQ(t)x(t)+\frac{1}{2}u(t)^TR(t)u(t)+\lambda(t)^T(A(t)x(t)+B(t)u(t))$$

Imposing the necessary conditions (PMP) for optimality we obtain that, for $(x^*(t),u^*(t))$ to be optimal, we need to satisfy for all $t\in[0,T]$

- from PMP 1)
$$\dot{\lambda}^*(t)=-A(t)^T\lambda^*(t)-Q(t)x^*(t)$$
with $\lambda^*(T)=Q_Tx^*(T)$
- from PMP 2), imposing $H_u(x^*(t),u^*(t),\lambda^*(t),t)=0$
$$R(t)u^*(t)+B(t)^T\lambda^*(t)=0$$
- from PMP 3)
$$\dot{x}^*(t)=A(t)x^*(t)+B(t)u^*(t)$$
with $x^*(0)=x_{\text{init}}$

Rewriting the necessary conditions for optimality we have:

$$\begin{aligned}\dot{x}^*(t)&=A(t)x^*(t)-B(t)R^{-1}(t)B(t)^T\lambda^*(t)\\ \dot{\lambda}(t)&=-A(t)^T\lambda^*(t)-Q(t)x^*(t)\end{aligned}$$

which can be re-written in matrix form as

$$\begin{bmatrix}\dot{x}^*(t)\\ \dot{\lambda}^*(t)\end{bmatrix}=\begin{bmatrix}A(t) & -B(t)R(t)^{-1}B(t)^T\\ -Q(t) & -A(t)^T\end{bmatrix}\begin{bmatrix}x^*(t)\\ \lambda^*(t)\end{bmatrix}$$

with $\lambda^*(T)=Q_Tx^*(T)$ and $x^*(0)=x_{\text{init}}$. This is a *two point boundary value problem*
Let us assume that $\lambda^*(t)=P(t)x^*(t)$ so that the optimal (feedback) input is

$$u^*(t)=K(t)x^*(t)$$

where

$$K(t)=-R(t)^{-1}B(t)^TP(t)$$

To show that this assumption holds, let us expand the costate equation:

$$\begin{aligned}\dot{\lambda}^*(t)&=\dot{P}(t)x^*(t)+P(t)\dot{x}^*(t)\\ &=\dot{P}(t)x^*(t)+P(t)A(t)x^*(t)-P(t)B(t)R(t)^{-1}B(t)^TP(t)x^*(t)\\ Q(t)x^*(t)+A(t)^T\lambda^*(t)&=Q(t)x^*(t)+A(t)^TP(t)x^*(t)\end{aligned}$$

hence

$$\dot{P}(t)x^*(t)=-\left(P(t)A(t)+A(t)^TP(t)-P(t)B(t)R(t)^{-1}B(t)^TP(t)+Q(t)\right)x^*(t)$$

This equation is satisfied by the solution $P(\cdot)$ of the following *Riccati differential equation*:

$$\dot{P}(t)=-P(t)A(t)-A(t)^TP(t)+P(t)B(t)R^(t)-1B(t)^TP(t)-Q(t)$$

with terminal condition $P(T)=Q_T$ to be consistent with $\lambda^*(T)$

Chapter 9

Dynamic Programming

Consider the optimal control problem

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

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-to-go function*)

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

It is the cost incurred starting from $x_t = \bar{x}$ in the horizon $[t,T]$ when the optimal poilcy is applied.

Remark. Notice that $V_T^*(\bar{x}) = \ell_T(\bar{x})$

9.1 Dynamic programming Recursion

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

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

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

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

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 $V_0^*(x_{init})$

9.1.1 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., $\pi_t : \mathbb{R}^n \to \mathbb{R}^m$.

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:

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

9.1.2 DP Advantages and Limitations

Advantages:

- no need for differentiability or convexity assumptios 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.

9.1.3 Linear Quadratic Optimal Control via DP

Consider a linear quadratic control problem as:

$$\begin{aligned} \min_{\substack{x_1,\ldots,x_T \\ u_0,\ldots,u_{T-1}}} & \sum_{t=0}^{T-1} \frac{1}{2} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T \begin{bmatrix} Q_t & S_t^T \\ S_t & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \frac{1}{2} x_T^T Q_T x_T \\ \text{subj. to } & x_{t+1} = A_t x_t + B_t u_t \quad t = 0, \ldots, T-1 \\ & x_0 = x_{\text{init}} \end{aligned}$$

where $Q_t = Q_t^T \geq 0 \; \forall t = 0, \ldots, T$, $R_t = R_t^T > 0 \; \forall t = 0, \ldots, T-1$ and S_t such that the problem is convex, i.e. $Q_t - S_t^T R_t^{-1} S_t \geq 0$

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 policiy 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

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

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:

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

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

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

It is therefore possible to write

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

which should hold for all x_t, therefore we obtain the so-called *Difference Riccati Equation*:

$$P_t = Q_t + A_t^T P_{t+1} A_t - (S_t^T + A_t^T P_{t+1} B_t)(R_t + B_t^T P_{t+1} B_t^{-1})(S_t^T + B_t^T P_{t+1} A_t)$$

Setting $P_T = Q_T$, it is possible to otbain P_t via a backward integration for $t = T-1, \dots, =$

9.1.4 DP for LQP – Summary

Consider a linear quadratic control problem

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

1. Set $P_T = Q_T$ and backward iterate $t = T-1, \dots, 0$:
$$P_t = Q_t + A_t^T P_{t+1} A_t - (S_t^T + A_t^T P_{t+1} B_t)(R_t + B_t^T P_{t+1} B_t^{-1})(S_t^T + B_t^T P_{t+1} A_t)$$
2. Define the feedback gain:
$$K_t^* = -(R_t + B_t^T P_{t+1} B_t)^{-1}(S_t + B_t^T P_{t+1} A_t)$$
3. Compute the trajectory by forward integration $t = 0, \dots, T-1$:
$$\begin{aligned} u_t^* &= K_t^* x_t^* \\ x_{t+1}^* &= A_t x_t^* + B_t u_t^* \qquad x_0^* = x_{\text{init}} \end{aligned}$$

9.1.5 LQ for time-invariant systems and cost

The solution turns out to be:

$$\begin{aligned} P_T &= Q_T \\ P_t &= Q + A^T P_{t+1} A + (S^T + A^T P_{t+1} B)(R + B^T P_{t+1} B)^{-1}(S^T + B^T P_{t+1} A) \\ K_t^* &= -(R + B^T P_{t+1} B)^{-1}(S + B^T P_{t+1} A) \end{aligned}$$

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

9.2 Infinite Horizon Linear Quadratic Problems

Consider a linear quadratic optimal control problem as:

$$\begin{aligned}\min_{\substack{x_1,\dots\\u_0,\dots}} \sum_{t=0}^{\infty} \frac{1}{2}\begin{bmatrix}x_t\\u_t\end{bmatrix}^T \begin{bmatrix}Q & S^T\\ S & R\end{bmatrix}\begin{bmatrix}x_t\\u_t\end{bmatrix}&\\ \text{subj to } x_{t+1} = Ax_t + Bu_t \qquad & t = 0,\dots,T-1\\ x_0 = x_{\text{init}}&\end{aligned}$$

With the usual assumptions on simmetricity and positive definitiveness, assuming also that the pair (A, B) is controllable and the pair (A, C) with $Q = C^T C$ is observable
The optimal value function can be defined as:

$$\begin{aligned}V_t^*(\bar{x}) = \min_{\substack{x_{t+1},\dots\\u_t,\dots}} \sum_{\tau=t}^{\infty} \frac{1}{2}\begin{bmatrix}x_\tau\\u_\tau\end{bmatrix}^T \begin{bmatrix}Q & S^T\\ S & R\end{bmatrix}\begin{bmatrix}x_\tau\\u_\tau\end{bmatrix}&\\ \text{subj to } x_{\tau+1} = Ax_\tau + Bu_\tau \qquad & t \in [t,\infty]\\ x_t = \bar{x}&\end{aligned}$$

which does not depend on time t (the horizon is always ∞), i.e.

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

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-difinite quadratic function, we can write

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

where $P = P^T \geq 0$

9.2.1 DP Recursion in Infinite Horizon LQ Problem

Consider the dynamic programming recursion:

$$V^*(x_t) = \min_u \frac{1}{2}\begin{bmatrix}x_t\\u\end{bmatrix}^T \begin{bmatrix}Q & S^T\\ S & R\end{bmatrix}\begin{bmatrix}x_t\\u\end{bmatrix} + V^*(Ax_t + Bu)$$

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

$$\pi_t^*(x_t) = \arg\min_u \frac{1}{2}\begin{bmatrix}x_t\\u\end{bmatrix}^T \begin{bmatrix}Q & S^T\\ S & R\end{bmatrix}\begin{bmatrix}x_t\\u\end{bmatrix} + V^*(Ax_t + Bu)$$

by considering

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

we obtain

$$\begin{aligned}\pi^*(x_t) &= \arg\min_u \frac{1}{2}\begin{bmatrix}x_t\\u\end{bmatrix}^T \begin{bmatrix}Q & S^T\\ S & R\end{bmatrix}\begin{bmatrix}x_t\\u\end{bmatrix} + (Ax_t + Bu)^T P (Ax_t + Bu)\\ &= \arg\min_{u\in\mathbb{R}^m} \frac{1}{2}\begin{bmatrix}x_t\\u\end{bmatrix}^T \begin{bmatrix}Q + A^TPA & S^T + A^TPB\\ S + B^TPA & R + B^TPB\end{bmatrix}\begin{bmatrix}x_t\\u\end{bmatrix}\\ &= -(R + B^TPB)^{-1}(S + B^TPA)x_t\end{aligned}$$

Since:

$$\pi^*(x_t) = K^* x_t \qquad \text{with } K^* := -(R + B^T P B)^{-1}(S + B^T P A)$$

It is possible to write

$$\begin{aligned} V^*(x_t) &= \frac{1}{2}\begin{bmatrix} x_t \\ K^* x_t \end{bmatrix}^T \begin{bmatrix} Q + A^T P A & S^T + A^T P B \\ S + B^T P A & R + B^T P B \end{bmatrix} \begin{bmatrix} x_t \\ K^* x_t \end{bmatrix} \\ \frac{1}{2} x_t^T P x_t &= \frac{1}{2} x_t^T \left(Q + A^T P A - (S^T + A^T P B)(R + B^T P B^{-1})(S^T + B^T P A) \right) \end{aligned}$$

which should hold for all x_t, therefore we obtain the so-called *Algebraic Riccati Equation*:

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

It can be shown that K^* is exponentially stabilizing under the assumption of (A, B) controllable and (A, C) observable

Time invariant cost and dynamics

$$V^*(\bar{x}) = \min \sum_{\tau=t}^{\infty} \ell(\bar{x}, u)$$

does not depend explicitly on time

Chapter 10

Model Predictive Control

10.1 Introductionn

Motivations

We want to control a system

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

via a *stabilizing* controller, which

- minimizes a certain cost function
$$\sum_{t=0}^{\infty} \ell_t(x_t, u_t)$$
- enforces some constraints for all t
$$x_t \in \mathcal{X}, u_t \in \mathcal{U}$$
- works *online*

Idea: at each sampling time t solve an optimal control problem and apply the first optimal input.

Idea

For each t

1. Measure the current state x_t
2. Compute the optimal trajectory $x^*_{t|t}, \dots, x^*_{t+T|t}, u^*_{t|t}, \dots, u^*_{t+T-1|t}$[1] with initial state x_t
3. Apply the first control input $u^*_{t|t}$
4. Measure x_{t+1} and repeat

prediction horizon vs time horizon

Two time-scales:

- time $t = 0, \dots, \infty$ time instants in the real world
- prediction iteration $\tau = t, \dots, t+T$, samples evaulated by the mpc algorithm at each time instant t

[1] $x^*_{\tau|t}$ signifies the optimal state trajectory at instant τ for the optimal control problem starting at instant t, similarly for $u^*_{\tau|t}$

optimal control problem to be solved at each t

At each time instant t, solve

$$\begin{aligned}
\min_{\substack{x_0,x_1,\dots,x_T\\u_0,\dots,u_{T-1}}} \quad & \sum_{\tau=t}^{t+T-1} \ell_t(x_\tau,u_\tau)+\ell_{t+T}(x_{t+T})\\
\text{subj. to}\quad & x_{\tau+1}=f(x_\tau,u_\tau) \quad \tau=t,\dots,t+T-1\\
& x_\tau\in\mathcal{X}, u_\tau\in\mathcal{U} \qquad \tau=t,\dots,t+T\\
& x_t=x_t^{\text{meas}2}
\end{aligned}$$

10.2 MPC with Zero Terminal Constraint

At each time instant t, solve

$$\begin{aligned}
\min_{\substack{x_0,x_1,\dots,x_T\\u_0,\dots,u_{T-1}}} \quad & \sum_{\tau=t}^{t+T-1} \ell_t(x_\tau,u_\tau)\\
\text{subj. to}\quad & x_{\tau+1}=f(x_\tau,u_\tau) \quad \tau=t,\dots,t+T-1\\
& x_\tau\in\mathcal{X}, u_\tau\in\mathcal{U} \qquad \tau=t,\dots,t+T-1\\
& x_t=x_t^{\text{meas}}\\
& x_{t+T}=0
\end{aligned}$$

where

- x_τ and u_τ state and input predictions at future time τ computed at current time t
- x_t^{meas} state (of the real system) measured at t
- $x=0$ equilibrium point for the system we want to stabilize
- $\mathcal{X}$ and $\mathcal{U}$ state and input constraint sets, which satisfy $(0,0)\in\text{int}\{\mathcal{X}\times\mathcal{U}\}$.
- $\ell_\tau:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{Z}^+\to\mathbb{R}$ is a positive definite continous stage cost $\forall\tau\in\mathbb{Z}^+$.

Remark: in the more general case in which $(x^{eq},u^{eq})\neq(0,0)$, we can always perform a global change of coordinates $\psi:(x,u)\to(\bar{x},\bar{u})$ such that $(x^{eq},u^{eq})\to(0,0)$ which brings us back to the previous case

Theorem 10.2.1. Consider the discrete-time system

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

with $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ Lipschitz continuous wrt x. If the stage cost is continuous, bounded and positive definite for all τ, then the Zero Terminal Constraint MPC scheme is recursively feasible and the origin is asymptotically stable for the resulting closed-loop system. Assumptions: the optimal control problem at $t=0$ is feasible and some regularity on the constraint funcitinos $(g_t(x_t,u_t)\leq 0)$

Proof (sketch of).

- Recursive Feasibility
  - Assume the problem is feasible at generic time t and $\{u^*_{\tau|t}\}_{\tau=t}^{t+T-1}$ is the corresponding optimal input sequence, and assume $x_t^{\text{meas}}=x^*_{t|t},\dots,x^*_{t+T|t}$
  - At time $t+1$ consider the following candidate input trajectory:

$$u_{\tau|t+1}:=\begin{cases} u^*_{\tau|t}, & \tau=t+1,\dots,t+T-1\\ 0, & \tau=t+T\end{cases}$$

[2] x_t^{meas} is the measured state at time isntant t

  - As it can be easily verified, this new trajectory is still feasible for the MPC problem (though in general suboptimal).

- Asympototic Stability
  - The idea is to use the optimal cost $J^*(x_t^{\text{meas}})$ as a Lyapunov function
  - First note that $J^*(x) \geq 0, \forall x \in \mathcal{X}$ and $J^*(x) = 0 \iff x = 0$
  - Then observe that

$$\begin{aligned} J^*(x_{t+1}^{\text{meas}}) &= \sum_{\tau=t+1}^{t+T} \ell_\tau(x^*_{\tau|t+1}, u_{\tau|t+1}) \\ &\leq \sum_{\tau=t+1}^{t+T-1} \ell_\tau(x^*_{\tau|t}, u^*_{\tau|t}) + \ell_{t+T}(0,0) \\ &= J^*(x_t^{\text{meas}}) - \ell_t(x^*_{\tau|t}, u^*_{\tau|t}) = J^*(x_t^{\text{meas}}) - \ell_t(x_t^{\text{meas}}, u_t^{\text{MPC}}) \end{aligned}$$

  Thus

$$J^*(x_{t+1}^{\text{meas}}) - J^*(x_t^{\text{meas}}) < 0, \quad \forall x_t^{\text{meas}} \neq 0$$

□

10.3 Quasi-Infinte Horizon MPC

Main idea: Since the terminal constraint introduces nuerical instabilities, relax the terminal condition by introducing a proper terminal cost ℓ_{t+T} and constraining the terminal state to be in a proper region $\mathcal{X}^f$ around the origin

$$\begin{aligned} \min_{\substack{x_0,x_1,\dots,x_T \\ u_0,\dots,u_{T-1}}} \quad & \sum_{\tau=t}^{t+T-1} \ell_t(x_\tau, u_\tau) + \ell_{t+T}(x_{t+T}) \\ \text{subj. to } \quad & x_{\tau+1} = f(x_\tau, u_\tau) \quad \tau = t, \dots, t+T-1 \\ & x_\tau \in \mathcal{X}, u_\tau \in \mathcal{U} \qquad \tau = t, \dots, t+T-1 \\ & x_t = x_t^{\text{meas}} \\ & x_{t+T} \in \mathcal{X}^f \end{aligned}$$

Assumption: The terminal region is forward invariant wrt a local controller, i.e. there exists $k^{\text{loc}} : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$x \in \mathcal{X}^f \implies f(x, k^{\text{loc}}(x)) \in \mathcal{X}^f$$

Result: Under this assumption it can be proven that for a proper terminal cost and terminal set $\mathcal{X}^f$ the MPC scheme is recursively feasible and asymptotically stable.

10.3.1 Practical framework

- Quadratic stage cost and terminal cost:

$$\begin{aligned} \ell_\tau(x,u) &= x^T Q x + u^T R u \\ \ell_{t+T}(x) &= x^T P x \end{aligned}$$

  where P is computed in a way that $x^T P x$ is an upper bound for the optimal infinite-horizon cost-to-go

- Ellipsoidal terminal region:

$$\mathcal{X}^f = \{x \in \mathbb{R}^n | x^T P x \leq \alpha\}$$

  for a proper α which is forward invariant wrt the LQR controller K^{lqr}

- Hence, at each sampling time t, we solve the following problem:

$$\begin{aligned}
\min_{\substack{x_0, x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \; & \sum_{\tau=t}^{t+T-1} x_\tau^T Q x_\tau + u_\tau^T R u_\tau + x_{t+T}^T P x_{t+T} \\
\text{subj. to } & x_{\tau+1} = f(x_\tau, u_\tau) \quad \tau = t, \dots, t+T-1 \\
& x_t = x_t^{\text{meas}} \\
& u_\tau \in \mathcal{U} \qquad \tau = t, \dots, t+T-1 \\
& x_{t+T}^T P x_{t+T} \leq \alpha
\end{aligned}$$

  and we apply to the real system the first optimal input $u_{t|t}^*(x_t^{\text{meas}})$

10.4 Optimal control with constraints

when dealing with constrained optimal control, with constraints $g(x_t, u_t) \leq 0$ barrier functions can be used:

$$\begin{aligned}
\min_{\substack{x_1, \dots, x_T \\ u_0, \dots, u_{T-1}}} \; & \sum_{t=0}^{T-1} [\ell_t(x_t, u_t) - \varepsilon \log(-g(x_t, u_t))] + \ell_T(x_T) \\
\text{subj to } & x_{t+1} = f_t(x_t, u_t) \quad t = 0, \dots, T-1
\end{aligned}$$

start with $\varepsilon = \varepsilon^0$, solve the relaxed problem above, decrease $\varepsilon > 0$
Remarks:

- need to initialize with feasible initial trajectory
- alternatively extend the $-\log(\cdot)$ function on the entire domain, i.e. use some function

$$b(z) = \begin{cases} -\log(z) & z \leq -\delta \\ \text{smooth function} & z \geq -\delta \end{cases} \qquad \delta > 0$$

- use some heuristics to remain within the constraints when

Chapter 11

Reinforcement Learning

11.1 Notation

- States and actions:
$$(s, a) \in \mathcal{S} \times \mathcal{A}$$
- Transition Probability:
$$s^+ \sim p(\cdot|s, a)$$
- Reward function at time t:
$$R_t(s_t, a_t) \text{ or } R_t(s_{t-1}, a_{t-1}, s_t)$$
- Discounted Return:
$$\begin{aligned} &G_t(s_t) := \sum_{\tau=t}^{\infty} \gamma^\tau R_\tau(s_\tau, a_\tau) \\ &\text{subj to } s_{\tau+1} \sim p(\cdot|s_\tau, a_\tau), a_\tau \sim \pi(\cdot|s_\tau), \forall \tau \geq 0 \end{aligned}$$

Reinforcement learning is generally data-driven, whereas optimal control is generally model-based.

problem formulation

The Reinforcement Learning (RL) problem can be written in optimal control language as