5. Optimization problems

- optimization problems
- solving optimization problems
- problem transformations
- control example

Optimization problem

$$
\begin{array}{ll}
\text{minimize} & f(x) \\
\text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\
& h_j(x) = 0, \quad j = 1, \ldots, p
\end{array} \tag{5.1}
$$

- $x = (x_1, \ldots, x_n)$ is the *optimization* or *decision variable*
- $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective* or *cost* function
- $g_i : \mathbb{R}^n \to \mathbb{R}$ are the *inequality constraint* functions
- $h_j : \mathbb{R}^n \to \mathbb{R}$ are the *equality constraint* functions
- can be compactly written as

$$
\begin{array}{ll}
\text{minimize} & f(x) \\
\text{subject to} & g(x) \leq 0 \\
& h(x) = 0
\end{array} \tag{5.2}
$$

where $g(x) = (g_1(x), \ldots, g_m(x))$ and $h(x) = (h_1(x), \ldots, h_p(x))$

Domain and implicit constraints

the *domain* of an optimization problem is

$$\mathcal{D} = \mathrm{dom}\, f \;\cap\; \bigcap_{i=1}^{m} \mathrm{dom}\, g_i \;\cap\; \bigcap_{j=1}^{p} \mathrm{dom}\, h_j$$

- the standard from problem (5.1) has an *implicit constraint* $x \in \mathcal{D}$
- *the explicit constraints* are $g(x) \leq 0 \leq 0$ and $h(x) = 0$
- a problem is *unconstrained* if it has no explicit constraints ($m = p = 0$)
- for example, the unconstrained problem

$$\text{minimize} \quad -\log x_1 + \log(x_2 - x_1)$$

has implicit constraints $x_1 > 0$, $x_2 - x_1 > 0$, which defines the domain $\mathcal{D}$

Feasible and optimal points

Feasible point: $\hat{x}$ is *feasible* if $\hat{x} \in \mathcal{D}$ and it satisfies the constraints

Solution: a point $x^\star$ is an *optimal* or a *solution* if it is feasible and

$$f(x^\star) \le f(x) \quad \text{for any feasible } x$$

Optimal value

greatest ρ such that $\rho \le f(x)$ for all feasible x, denoted by $p^\star$

- if there exists an optimal point $x^\star$, then $p^\star = f(x^\star)$
  - we say the optimal value is *attained* or *achieved* and the problem is *solvable*
- a minimization problem is *unbounded below* if $p^\star = -\infty$
- if a minimization problem is infeasible, then we let $p^\star = +\infty$

Examples

- the unconstrained problem

$$\text{minimize} \quad (x_1 - 1)^2 + (x_2 - 1)^2 = \|x - \mathbf{1}\|^2$$

  has optimal value $p^\star = 0$, which is attained at the optimal point $x^\star = (1, 1) = \mathbf{1}$

- the problem

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 \\ \text{subject to} & -x_1 \leq 10 \\ & \quad x_2 \geq 0 \end{array}$$

  has solution $x^\star = (-10, 0)$ and $p^\star = -10$

- the problem

$$\text{minimize} \quad x_1^2 - x_2^2$$

  is unbounded below since $f(x) \to -\infty = p^\star$ as $|x_2| \to \infty$

- consider the problem

$$\text{minimize} \quad f(x) = e^{-x}$$

  the optimal value is $p^\star = 0$, but it is not attained since it only holds as $x \longrightarrow \infty$

- for the problem

$$\text{minimize} \quad f(x) = 1/x, \qquad \mathbf{dom}\, f = \{x \mid x > 0\}$$

  for this problem $p^\star = 0$ but is not attained by any feasible x

- the problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 + x_2 \le 1 \\ & x_1 + x_2 \ge 2 \end{array}$$

  is infeasible; hence, $p^\star = \infty$

Maximization problems

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

- f is often called *utility function* instead of cost
- note that $\max f(x) = -\min -f(x)$
- thus, maximization problems can be written as minimization problems

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & h(x) = 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \text{minimize} & -f(x) \\ \text{subject to} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

both problems have the same solutions

Standard form

we refer to problem (5.1) as an optimization problem in *standard form*

- equality $r_j(x) = \tilde{r}_j(x)$ is same as $h_j(x) = 0$ with $h_j(x) = r_j(x) - \tilde{r}_j(x)$
- inequality $\tilde{g}_i(x) \geq 0$ is same as $g_i(x) \leq 0$ with $g_i(x) = -\tilde{g}_i(x)$
- maximization can be represented as minimization by changing the objective sign

Example: the problem

$$\begin{array}{ll} \text{maximize} & -x_1^2 + x_2^2 \\ \text{subject to} & -x_1 + x_2 \geq 10 \\ & x_2 = 2 - x_1 \end{array}$$

can be expressed in standard form:

$$\begin{array}{ll} \text{minimize} & x_1^2 - x_2^2 \\ \text{subject to} & x_1 - x_2 + 10 \leq 0 \\ & x_1 + x_2 - 2 = 0 \end{array}$$

Outline

- optimization problems
- **solving optimization problems**
- problem transformations
- control example

Set-constrained problems

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(x)$$

- find x that minimizes $f(x)$ among all points in the *constraint set* $\mathcal{X} \subseteq \mathbb{R}^n$
- for problem (5.1), the constraint set is described by *functional constraints*

$$\mathcal{X} = \{x \mid g_i(x) \leq 0,\ h_j(x) = 0,\ i = 1, \ldots, p,\ j = 1, \ldots, m\}$$

- this is not always the case, for example consider the integer set

$$\mathcal{X} = \{1, 2, 3\} \subset \mathbb{R}$$

Existence of a solution

- existence of a solution is not always guaranteed
- can be guaranteed to exists under some conditions

Solution existence: a continuous f over $\mathcal{X}$ has an optimal point over $\mathcal{X}$ if either

- $\mathcal{X}$ is nonempty and compact (closed and bounded)
- f is *coercive*:

  $$\lim_{\|x\| \to \infty} f(x) = \infty$$

  and $\mathcal{X} \subseteq \mathbb{R}^n$ is a nonempty closed set

Simple problems solution

- general optimization problems require sophisticated methods to solve them that utilize derivatives, linear equations, nonlinear operators,...etc
- that said, there are some simple optimization problems that can be solved by inspection or using some basic inequalities such as Cauchy-Schwarz

Example

$$\begin{array}{ll} \text{minimize} & \|x - \mathbf{1}\| \\ \text{subject to} & -\mathbf{1} \leq x \leq 0 \end{array}$$

- we seek to find a feasible x that is closest in distance to $\mathbf{1}$
- hence

$$x^\star = 0 \quad \text{and} \quad p^\star = \| - \mathbf{1}\| = \sqrt{n}$$

Example

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 \leq 1 \end{array}$$

- using Cauchy-Schwarz, we can lower bound the objective by

$$x_1 + x_2 = \mathbf{1}^T x \geq -\|\mathbf{1}\|\|x\| \geq -\sqrt{2}$$

  for all $x_1^2 + x_2^2 \leq 1$

- the minimum value is attained at $x = (-1/\sqrt{2}, -1/\sqrt{2})$, which is feasible
- hence, the optimal point is $x = (-1/\sqrt{2}, -1/\sqrt{2})$

Optimization methods

- after formulating the problem, a suitable algorithm is applied to solve it
- an optimization *algorithm* is a set of calculations and rules that are followed to find a solution or an approximate solution to an optimization problem

Iterative algorithms: start from an initial guess $x^{(0)}$ and computes

$$x^{(k+1)} = F(x^{(k)}), \quad k = 0, 1, \ldots$$

- F depends on $f(x), g_i(x), h_j(x)$ to generate a new estimate $x^{(k+1)}$
- moving from $x^{(k)}$ to $x^{(k+1)}$ is called an *iteration* of the algorithm
- stops when a good estimate of a solution is reached or $k = k^{\max}$ for some $k^{\max}$

General iterative algorithm

given a starting point $x^{(0)}$, tolerance ϵ, stopping criteria, and $k^{\max}$

for $k \geq 1$

1. determine a search direction $v^{(k)}$
2. **quit** if stopping criterion is met (depends on ϵ), and output $x^{(k)}$
3. determine scalar α_k
4. update:

$$x^{(k+1)} = x^{(k)} + \alpha_k v^{(k)}$$

until $k = k^{\max}$

- α_k is called the *stepsize* or *learning rate*
- $v^{(k)}$ depends on f, g_i, h_j and their derivatives if differentiable

Local minimum point

a point $x^\circ \in \mathcal{X}$ is a *local minimizer* or *local minimum point* (locally optimal) of

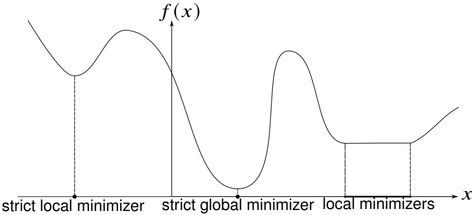
$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(x)$$

if there exists a scalar $r > 0$ such that:

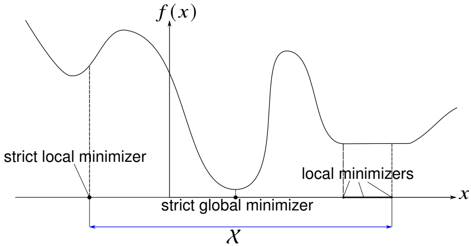
$$f(x^\circ) \leq f(x) \quad \text{for all} \quad x \in \mathcal{X} \quad \text{and} \quad \|x - x^\circ\| \leq r$$

- if $f(x^\circ) < f(x)$, then the point x° is called a *strict local minimizer*
- $x^\star \in \mathcal{X}$ is a *global minimizer* (*minimum point*) if $f(x^\star) \leq f(x)$ for all $x \in \mathcal{X}$
- 'globally optimal' is used for 'optimal' to distinguish from 'locally optimal'
- a point is a maximum point of f if it is a minimum point of $-f$

Global and local optima

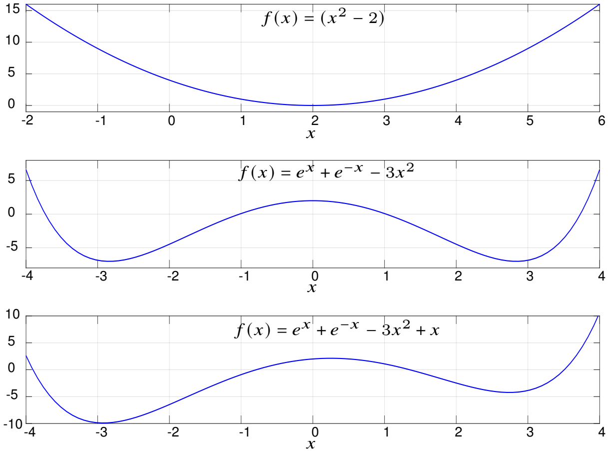


unconstrained case $\mathcal{X} = \mathbb{R}^n$



constrained case $x \in \mathcal{X}$

Example



- $(x-2)^2$; $p^\star = \min f(x) = 0$; global minimizer $x^\star = 2$
- $e^x + e^{-x} - 3x^2$; $p^\star = -7.02$; two global minima: $x^\star = \pm 2.84$
- $e^x + e^{-x} - 3x^2 + x$; $p^\star = -9.9$; global min. $x^\star = -2.92$; local min. at $x = 2.74$

Nonlinear optimization methods

Local optimization methods

- find a locally optimal solution with no global optimality guarantees
- fast, can handle large-scale problems, and are widely applicable
- can be used to improve the performance of an engineering design obtained by manual, or other, design methods

Global optimization methods

- true global solution is found with optimality guarantees
- difficult to find in general; even small problems, with a few tens of variables, can take a very long time (*e.g.*, hours or days) to solve
- usually seek the global optimum by finding local solutions to a sequence of approximate subproblems

Efficiently solvable problem classes

(linear) Least squares

$$\text{minimize} \quad \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_{ij} x_j - b_i \Big)^2$$

where the coefficients a_{ij}, b_i are given constants

- reliable and efficient algorithms and software
- least-squares problems are easy to recognize
- has many applications such as data-fitting and linear estimation

Linear program (optimization)

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \\ & \sum_{j=1}^{n} g_{ij} x_j = h_i, \quad i = 1, \ldots, p \end{array}$$

the coefficients $c_j, a_{ij}, g_{ij}, h_i, b_i$ are given constants

- there exist robust and efficient algorithms and software for solving LPs
- LPs isn't as immediately recognizable as that of least-squares problems
- common techniques are available to transform various problems into LPs

Convex optimization

$$\begin{array}{ll} \text{minimize} & f(x) = g_0(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, p \end{array}$$

the coefficients a_{ij}, b_i are given constants

- the objective and constraints functions are *convex*:

$$g_i(\theta x + (1-\theta) y) \leq \theta g_i(x) + (1-\theta) g_i(y), \quad 0 \leq \theta \leq 1$$

- if objective or one constraint is nonconvex, problem is called *nonconvex*

Convex optimization

- include least-squares problems and linear programs as special cases
- has many of applications
- reliable and efficient algorithms
- difficult to recognize
- many tricks can be used to transform nonconvex problems into convex form
- basis for several heuristics for solving nonconvex problems

Outline

- optimization problems
- solving optimization problems
- **problem transformations**
- control example

Equivalent optimization problems

two optimization problems are *equivalent* if from a solution of one, we can find a solution of the other, and vice versa

- for example, maximization problems are equivalent to minimization problems
- many optimization problems can be transformed into equivalent ones
- can be very useful if the equivalent problem is easier to solve

Scaling and slack variables

Scaling: problem (5.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & \alpha f(x) \\ \text{subject to} & \beta_i g_i(x) \leq 0,\ i = 1, \ldots, m \\ & \gamma_j h_j(x) = 0,\ j = 1, \ldots, p \end{array}$$

- $\alpha > 0$, $\beta_i > 0$ and $\gamma_j \neq 0$
- 'scaling' does not alter solutions

Slack variables: problem (5.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & s_i \geq 0,\ i = 1, \ldots, m \\ & g_i(x) + s_i = 0,\ i = 1, \ldots, m \\ & h_j(x) = 0,\ j = 1, \ldots, p \end{array}$$

- the variables are $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^m$
- we replaced $g_i(x) \leq 0$ with $g_i(x) + s_i = 0$ for some $s_i \geq 0$
- variable s_i is called *slack variable* associated with inequality constraint $g_i(x) \leq 0$

Monotone transformations

$$
\begin{array}{ll}
\text{minimize} & \phi(f(x)) \\
\text{subject to} & g_i(x) \leq 0, \ i = 1, \ldots, m \\
& h_j(x) = 0, \ j = 1, \ldots, p
\end{array}
$$

- $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a continuous and monotone increasing function, *i.e.*,

  $$\phi(a) > \phi(b) \quad \text{for all } a > b \text{ over the optimization domain}$$

  this implies that ϕ is one-to-one and its inverse ϕ^{-1} is well defined

- this problem is equivalent to (5.1)

Constraint transformation: find $\psi_i : \mathbb{R} \rightarrow \mathbb{R}$ and $\varphi_j : \mathbb{R} \rightarrow \mathbb{R}$ so that

- $\psi_i(g_i(x)) \leq 0$ if and only if $g_i(x) \leq 0$
- $\varphi_j(h_j(x)) = 0$ if and only if $h_j(x) = 0$

Example

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & g(x) \leq 0 \end{array}$$

- norm is non-differentiable and we prefer differentiable objectives
- norm is nonnegative and $\phi(\cdot) = (\cdot)^2$ is monotone increasing over nonneg. no.
- hence, we can transform the problem into

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & g(x) \leq 0 \end{array}$$

- the new objective function is differentiable, which simplifies the problem

Change of variables

$$\begin{array}{ll} \text{minimize} & f(F(y)) \\ \text{subject to} & g_i(F(y)) \leq 0, \quad i = 1, \ldots, m \\ & h_j(F(y)) = 0, \quad j = 1, \ldots, p \end{array}$$

- $F : \mathbb{R}^n \to \mathbb{R}^n$ is 1-to-1 with image covering problem domain ($\mathcal{D} \subseteq F(\mathrm{dom}\, F)$)
- this implies for each $x \in \mathcal{D}$, there's a unique $y \in \mathrm{dom}\, F$ such that

$$x = F(y) \iff y = F^{-1}(x)$$

- if x solves (5.1), then $y = F^{-1}(x)$ solves the above problem
- if y solves the above problem, then $x = F(y)$ solves (5.1).

Example

$$\begin{array}{ll} \text{minimize} & x_1 x_2 x_3^2 \\ \text{subject to} & x_1 x_2 \leq 2 \\ & x_1, x_2, x_3 > 0 \end{array}$$

- $\log(\cdot)$ is strictly increasing (for non-negative argument),
- hence, we can use monotone transformations on objective and constraints

$$\log(x_1 x_2 x_3^2) \quad \text{and} \log(x_1 x_2) \leq \log(2)$$

- also use the change of variable $z_i = \log x_i$ to transform the problem into

$$\begin{array}{ll} \text{minimize} & z_1 + z_2 + 2z_3 \\ \text{subject to} & z_1 + z_2 \leq \log 2 \end{array}$$

this is a linear program, which is easier to solve

Example

$$
\begin{array}{ll}
\text{minimize} & x_1 x_2 - x_3^2 \\
\text{subject to} & x_1 + x_2 + x_3 \leq 20 \\
& x_2 \geq 10
\end{array}
$$

- let $y_1 = (x_1 + x_2)/2$ and $y_2 = (x_1 - x_2)/2$ so that

$$x_1 = y_1 + y_2, \quad x_2 = y_1 - y_2$$

- thus, we can transform the problem into

$$
\begin{array}{ll}
\text{minimize} & y_1^2 - y_2^2 - x_3^2 \\
\text{subject to} & 2y_1 + x_3 \leq 20 \\
& y_1 - y_2 \geq 10
\end{array}
$$

- the objective is now separable in the new variables
- for separable problems, there exist efficient specialized optimization methods

Eliminating equality constraints

suppose $\phi : \mathbb{R}^k \to \mathbb{R}^n$ and $h(x) = 0$ iff there is some z such that $x = \phi(z)$

then problem (5.1) is equivalent to

$$\begin{array}{ll} \text{minimize} & f(\phi(z)) \\ \text{subject to} & g_i(\phi(z)) \leq 0,\ i = 1, \ldots, m \end{array}$$

- for optimal $z^\star$, $x^\star = \phi(z^\star)$ is optimal for the original problem
- for optimal $x^\star$, any z such that $x = \phi(z)$ is optimal for the transformed problem

Example

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 + x_3^2 \\ \text{subject to} & x_1 - x_2 x_3 = 1 \end{array}$$

we use $x_1 = 1 + x_2 x_3$ to remove the equality constraint and get

$$\text{minimize} \quad 1 + x_2 x_3 + x_2 + x_3^2$$

in this case, we have $\phi(z_1, z_2) = (1 + z_1 z_2, z_1, z_2)$

Example

$$
\begin{array}{ll}
\text{minimize} & x_1 + 4x_2 + x_3 \\
\text{subject to} & 2x_1 - 2x_2 + x_3 = 4 \\
& x_1 - x_3 = 1 \\
& x_2 \geq 0,\ x_3 \geq 0
\end{array}
$$

- using the equality constraints, we have $x_1 = 1 + x_3$ and

$$2x_1 - 2x_2 + x_3 = 2(1 + x_3) - 2x_2 + x_3 = 4 \Rightarrow x_2 = \frac{3}{2}x_3 - 1$$

- hence, the problem can be simplified to

$$
\begin{array}{ll}
\text{minimize} & 8x_3 + 3 \\
\text{subject to} & x_3 \geq 2/3
\end{array}
$$

with solution $x_3 = 2/3$ from which we can find $x^\star = (5/3, 0, 2/3)$

Eliminating linear constraints

let $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, and consider the constraints

$$h(x) = Ax - b = 0$$

- solution of $Ax = b$ can parametrized by (see page 2.41)

$$x = \hat{x} + Fz \quad \text{for any arbitrary } z \in \mathbb{R}^{(n-p)}$$

  where columns of F form a basis for the nullspace of A ($\mathrm{range}(F) = \mathrm{null}(A)$)

- we can use change of variable $x = \hat{x} + Fz$, to transform (5.1) into

$$\begin{array}{ll} \text{minimize} & f(\hat{x} + Fz) \\ \text{subject to} & g_i(\hat{x} + Fz) \le 0, \quad i = 1, \ldots, m \end{array}$$

  with variable z

Example

$$
\begin{array}{ll}
\text{minimize} & f(x_1, \ldots, x_n) \\
\text{subject to} & x_1 + \cdots + x_n = b
\end{array}
$$

- we can eliminate any x_i, we choose x_n:

$$x_n = b - x_1 - \cdots - x_{n-1}$$

- the above corresponds to the choice

$$\hat{x} = be_n, \quad F = \begin{bmatrix} I \\ -\mathbf{1}^T \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$$

- the transformed problem is

$$\text{minimize} \quad f(x_1, x_2, \ldots, b - x_1 - \cdots - x_{n-1})$$

Adding equality constraints

- sometimes it is useful to introduce equality constraints
- for example, consider

$$\text{minimize} \quad f(Ax + b)$$

  where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$

- the above problem is equivalent to

$$\begin{array}{ll} \text{minimize} & f(z) \\ \text{subject to} & z = Ax + b \end{array}$$

  with variables $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$

Optimizing over some variables

it holds that (with abuse of $\inf$ notation):

$$\min_{x,y} f(x,y) = \min_x \tilde{f}(x)$$

where $\tilde{f}(x) = \min_y f(x,y)$

Example

$$\begin{array}{ll} \text{minimize} & x_1^T Q_{11} x_1 + 2x_1^T Q_{12} x_2 + x_2^T Q_{22} x_2 \\ \text{subject to} & g_i(x_1) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where Q_{11} and Q_{22} are symmetric; we can analytically minimize over x_2:

$$\min_{x_2} \left(x_1^T Q_{11} x_1 + 2x_1^T Q_{12} x_2 + x_2^T Q_{22} x_2 \right) = x_1^T (Q_{11} - Q_{12} Q_{22}^{-1} Q_{12}^T) x_1$$

thus, the original problem is equivalent to

$$\begin{array}{ll} \text{minimize} & x_1^T (Q_{11} - Q_{12} Q_{22}^{-1} Q_{12}^T) x_1 \\ \text{subject to} & g_i(x_1) \leq 0, \quad i = 1, \ldots, m \end{array}$$

Outline

- optimization problems
- solving optimization problems
- problem transformations
- **control example**

Dynamical system

a nonlinear dynamical system has the form

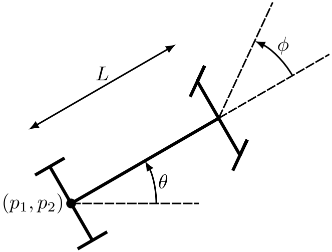
$$x_{k+1} = h(x_k, u_k), \quad k = 0, 1, \ldots, K$$

- $x_k \in \mathbb{R}^n$ is the *state vector* at instant k
- $u_k \in \mathbb{R}^m$ is the *input* or *control* at instant k
- $h : \mathbb{R}^{n+m} \to \mathbb{R}^n$ describes evolution of the system (system dynamics)
- examples: vehicle dynamics, robots, chemical plants evolution...

Optimal control

- initial state $x_1 = x_{\mathrm{initial}}$ is known
- choose the inputs $u_1, \ldots, u_K$ to achieve some goal for the states/inputs

Car control example



$$\frac{dp_1}{dt}(t) = s(t)\cos\theta(t)$$
$$\frac{dp_2}{dt}(t) = s(t)\sin\theta(t)$$
$$\frac{d\theta}{dt}(t) = (s(t)/L)\tan\phi(t)$$

- L wheelbase (length)
- $p(t)$ position
- $\theta(t)$ orientation (angle)
- $\phi(t)$ steering angle
- $s(t)$ speed
- we control speed s and steering angle ϕ

Discretized car dynamics

$$
\begin{aligned}
p_1(t+\tau) &\approx p_1(t) + \tau s(t)\cos\theta(t) \\
p_2(t+\tau) &\approx p_2(t) + \tau s(t)\sin\theta(t \\
\theta(t+\tau) &\approx \theta(t) + \tau (s(t)/L)\tan\phi(t)
\end{aligned}
$$

- τ is a small time interval
- let state vector $x_k = (p_1(k\tau), p_2(k\tau), \theta(k\tau))$
- input vector $u_k = (s(k\tau), \phi(k\tau))$
- discretized model is

$$x_{k+1} = h(x_k, u_k)$$

with

$$h(x_k, u_k) = x_k + \tau\,(u_k)_1 \left[\begin{array}{c} \cos(x_k)_3 \\ \sin(x_k)_3 \\ (\tan(u_k)_2)/L \end{array}\right]$$

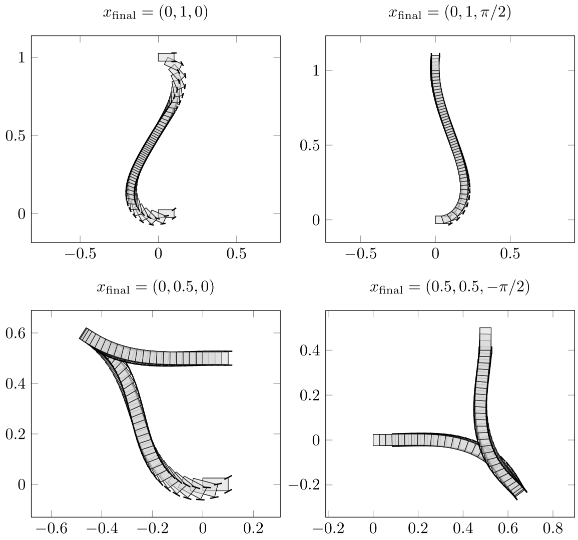
Car control problem

- move car from given initial to desired final position and orientation
- using a small and slowly varying input sequence

Problem formulation

$$
\begin{array}{ll}
\text{minimize} & \sum_{k=0}^{K} \|u_k\|^2 + \rho \sum_{k=0}^{K-1} \|u_{k+1} - u_k\|^2 \\
\text{subject to} & x_1 = h(0, u_0) \\
& x_{k+1} = h(x_k, u_k), \quad k = 1, \ldots, K-1 \\
& x_{\text{final}} = h(x_K, u_K)
\end{array}
$$

- variables $u_0, \ldots, u_N$, and $x_1, \ldots, x_N$
- the initial state is assumed to be zero
- the objective ensures the input is small with little variation
- $\rho > 0$ is an input variation trade-off parameter



solution trajectories with different final states; the outline of the car shows the position $(p_1(k\tau); p_2(k\tau))$, orientation $\theta(k\tau)$, and the steering angle $\phi(k\tau)$ at time $k\tau$

References and further readings

- S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004, chapter 4.1.
- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares.* Cambridge University Press, 2018 (ch 19.4, car control example).