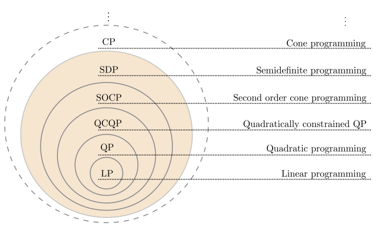
Motion Planning for Autonomous Vehicles

Introduction to Optimization

Geesara Kulathunga

March 8, 2023



Introduction to Optimization

Contents

Task 01

$$\begin{aligned} \max_{\mathbf{x}} \quad & x_1 + x_2 \\ \text{s.t.} \quad & 9x_1 + 3x_2 \leq 56, \\ & -7x_1 + 9x_2 \leq 56, \\ & -1 \leq \mathbf{x} \leq 1 \end{aligned} \tag{1}$$

Formulate the problem using CVXPY and scipy.optimize.linprog
`https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html`

In least-squares, given the measurements: $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$, seek a vector $x \in \mathbb{R}^m$ that project Ax on b (or such that Ax close to b). Such closeness is defined as the residual sum of squares (RSS):

$$J(\mathbf{x}) = (A\mathbf{x} - \mathbf{b})^T (A\mathbf{x} - \mathbf{b}) = \sum_{i=1}^{m} (a_i^T \mathbf{x} - b_i)^2 = \min_{\mathbf{x}} \|A\mathbf{x} - b\|_2^2 \qquad (2)$$

Using CVXPY to formulate the (2), you may generate some random matrix and vector for A and b, respectively.

If $\mathbf{x}^*$ is the optimal solution you obtained, comment on $A\mathbf{x}^* - b$

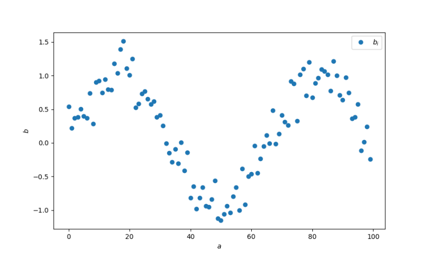
Task 03

Let's try adding a regularization term in the objective (2) as follows:

$$\min_x \sum_{i=1}^{m} (a_i^T x - b_i)^2 + \lambda \|x\|_1 \tag{3}$$

1. Using CVXPY formulate the (3), use the same matrix and vector you used in task 02 for A and b, respectively. λ is a regularization parameter, e.g., $1/\sqrt{n}$
2. Compare the x^* in both cases

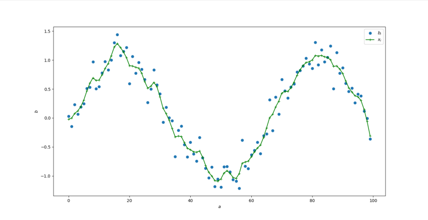
The smoothness of a curve, in general, is defined as if the slope of the curve does not change much, a curve is said to be a smooth curve. Thus, the objective is to minimize the rate of changes in the slope while smoothing the curve as much as closer to the original dataset. Considering these two conditions, smoothness can be formulated as an optimization problem.



Task 04

$$\min_{\mathbf{x}} \|\mathbf{x}-\mathbf{b}\|_2^2 + \rho\|D\mathbf{x}\|_2^2, D = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & & 0 \\ 0 & 0 & 1 & -2 & -1 & 0 & \dots & 0 \\ \vdots & & & & & & & \end{pmatrix}, \quad (4)$$

where ρ is the penalizing parameter between fitting and smoothness. Try to solve this optimization problem and see you will able to get a smoothed curve as follows:



Huber function `https://en.wikipedia.org/wiki/Huber_loss`, in general, is better for handling aforementioned outliers (Task 04)

$$h_M(x) = \begin{cases} x^2 & |x| \leq M \\ 2M|x| - M^2 & |x| > M \end{cases}, \tag{5}$$

where M is a scalar constant and quadratic loss in interval $[-M, M]$ and linear loss for $|x| > M$. Formulate Task 04 using Huber cost as follows and compare your results

$$\min \quad \sum_{i=1}^{n} h_M(x_i - y_i) + \rho \|Dx\|_2^2 \ , \tag{6}$$

where $D = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & & 0 \\ 0 & 0 & 1 & -2 & -1 & 0 & \dots & 0 \\ \vdots & & & & & & & \end{pmatrix}$

Task 06

Consider the discretized dynamics of a system is given as follows:

$$x_{t+1} = x_t + \left(1 - \frac{\gamma \Delta t}{2}\right) v_t \Delta t + \frac{1}{2} w_t \Delta t^2 \qquad (7)$$
$$v_{t+1} = \left(1 - \gamma \Delta\right) v_t + w_t \Delta t.$$

A Kalman filter can be formulated as an optimization problem though it can be solved analytically.

$$\begin{array}{ll} \text{min} & \sum_{t=0}^{N-1} \left(\|w_t\|_2^2 + \tau \|\eta_t\|_2^2 \right) \\ \text{subject to} & x_{t+1} = A x_t + B w_t, \quad t = 0, \ldots, N-1 \\ & z_t = H x_t + \eta_t, \quad t = 0, \ldots, N-1, \end{array} \qquad (8)$$

where system transition, design, and input matrices are denoted by $A \in \mathbb{R}^{n_x \cdot n_x}$, $H \in \mathbb{R}^{n_z \cdot n_x}$, and $B \in \mathbb{R}^{n_x \cdot n_u}$, respectively.

The number of system states, control inputs, and sensor readings are given by n_x, n_u, and n_z. τ is a regularization parameter. The control inputs and measurement noises are denoted by w_t and η_t. However, the large outliers are not properly handled by quadratic objectives. Thus, we are going to introduce Huber cost, which is also convex, that handles the outliers robustly. Let's reformulate the Kalman filter with Huber cost.

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{N-1} \left(\|w_t\|_2^2 + \tau \phi_\rho(\eta_t) \right) \\ \text{subject to} & x_{t+1} = Ax_t + Bw_t, \quad t = 0, \ldots, N-1 \\ & z_t = Hx_t + \eta_t, \quad t = 0, \ldots, N-1, \end{array} \tag{9}$$

where ϕ_ρ is the Huber function

$$\phi_\rho(a) = \begin{cases} \|a\|_2^2 & \|a\|_2 \le \rho \\ 2\rho\|a\|_2 - \rho^2 & \|a\|_2 > \rho. \end{cases}$$

Let's say we are going to use this modified filter for trajectory tracking. Let $\mathbf{x}_t \in \mathbb{R}^{n_x}, n_x = 4$ where $\mathbf{x}_t = \{x_0, x_1, v_0, v_1\}$ position and velocity of the model x and y direction, respectively. Assume that we have a measurement that gives position information only. Formulate this as an optimization problem.

What is the intersection point of these lines if they are?

$$\begin{aligned} 4x + 0.5y &= 23 \\ -4x + 8y &= 34 \end{aligned} \tag{10}$$

Are these formulations giving the same solution?

$$\begin{aligned} \min_{x} \quad & f(\mathbf{x}) = \tfrac{1}{2}(\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b}; C\mathbf{x} = \mathbf{d}; \mathbf{s} \leq \mathbf{x} \leq \mathbf{t}, \end{aligned} \tag{11}$$

where Q is symmetric.

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}|Q\mathbf{x} - \mathbf{c}|_2^2 \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b}; C\mathbf{x} = \mathbf{d}; \mathbf{s} \leq \mathbf{x} \leq \mathbf{t}, \end{aligned} \tag{12}$$

Now consider the following minimization problem

$$\min_x (Ax-b)^\top (Ax-b) \tag{13}$$

$$\begin{aligned} \min_x \quad & (Ax-b)^\top (Ax-b) \\ \text{s.t.} \quad & Gx \leq h, \end{aligned} \tag{14}$$

1. Let's consider with no constraints, trying to solve analytically ($x^* = (A^T A)^{-1} A^T$) and as well as a QP problem and compare your answers? Take $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$
2. Constraint x within $-0.9 \leq x \leq 0.9$ and solve it again,

Task 09

Now we are going to consider trajectory state x_k prediction at each time instance k, in terms of control input sequence $\mathbf{u_k}$ for a given initial condition, i.e., $x_{0|k}$. An optimal control sequence (or control policy) has to be calculated to estimate the optimal state prediction. Such a control policy can be estimated as minimizing the following quadratic cost:

$$J(x_k, \mathbf{u_k}) = \sum_{i=0}^{N-1} \|x_{k+i}\|_Q^2 + \|u_{k+i}\|_R^2 + \|x_{k+N}\|_P^2 \tag{15}$$

How do you determine the weight matrices: Q, R, and P?

For a linear system, the state prediction sequence can be written in a compact sequence as follows:

$$\mathbf{x_k} = Mx_k + C\mathbf{u_k}, \quad M = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & \dots & 0 \\ B & & & \\ AB & B & & \\ \vdots & \vdots & \ddots & \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \tag{16}$$

The defined quadratic cost (15) can be written in terms of $\mathbf{x_k}$ and $\mathbf{u_k}$ as

$$J = \mathbf{x}_k^T \tilde{Q} \mathbf{x_k} + \mathbf{u_k}^T \tilde{R} \mathbf{u_k} = \mathbf{u_k}^T H \mathbf{u_k} + 2x_k^T F^T \mathbf{u_k} + x_k^T G x_k \tag{17}$$

Can you define the $\tilde{Q}$ and $\tilde{R}$? as well as prove that H, F, and G are given by $C^T\tilde{Q}C + \tilde{R}$, $C^T\tilde{Q}M$, and $M^T\tilde{Q}M$, respectively.

Task 09

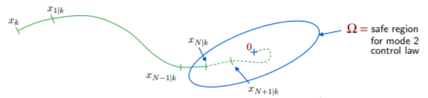
Let's say there are no additional constraints given, eq.17 has a closed-form solution which can be derived by minimizing the J with respect to $\mathbf{u}$. Show that $\mathbf{u}^* = -H^{-1}Fx_k$. What can you say about when H is singular (i.e., positive semi-definite rather than positive definite); this implies there are multiple optimal solutions can be exits. Since H and F are constant matrices, $\mathbf{u_k} = Lx_k$, where $L = -H^{-1}F$.
Now let's try to find out the feedback control law, namely L, considering following second order system with

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \tag{18}$$

for horizon N = 4, you may assume $Q = C^T C$, $R = 0.01$, and $P = Q$.

Task 10

Here we are interested in the terminal constraints set which helps to guarantee the recursive feasibility. In details description about recursive feasibility [1,2]. Let u_{min} and u_{max} be the minimum and maximum values for u, respectively. Let's consider N horizon state prediction as we did before. To ensure u stays within its boundary constraints, i.e., u_{min}, and u_{max}, u always within the Ω for all $i = 0, ..., N$



The terminal constraint set
$\Omega = \{x : u_{min} \le K(A+BK)^i x \le u_{max},\ i = 0, ..., N\}$, where K is the LQ optimal gain.

[1]. `https://markcannon.github.io/assets/downloads/teaching/C21_Model_Predictive_Control/mpc_notes.pdf`, [2]. Rakovic, S. V. (2016). Model predictive control: classical, robust, and stochastic [bookshelf]. IEEE Control Systems Magazine, 36(6), 102-105.

Task 10

Consider the system we examined in (25) and assume $K = [-1.19 \;\; -7.88]$. The terminal constraint set can be calculated as follows:

$$
\begin{aligned}
\Omega_0 &= \{x : -1 \le [-1.19 \;\; -7.88]x \le 1\} \\
\Omega_1 &= \Omega_0 \cap \{x : -1 \le [-0.5702 \;\; -4.9572]x \le 1\} \\
\Omega_2 &= \Omega_1 \cap \{x : -1 \le [-0.1621 \;\; -2.7826]x \le 1\} \\
&\;\;\vdots
\end{aligned}
\tag{19}
$$

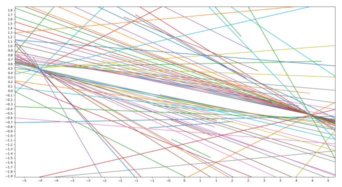


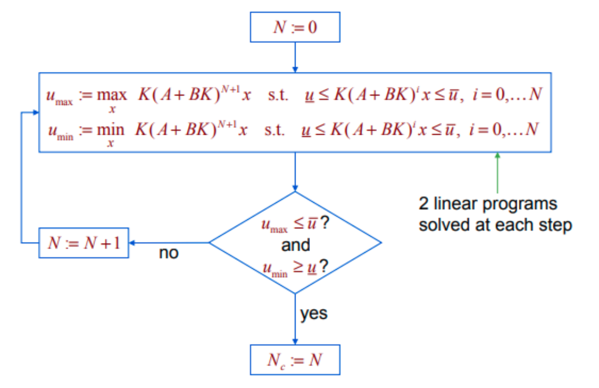
Figure: $\Omega_N = \Omega_{N+1}, \ldots, = \Omega_\infty$

Task 10

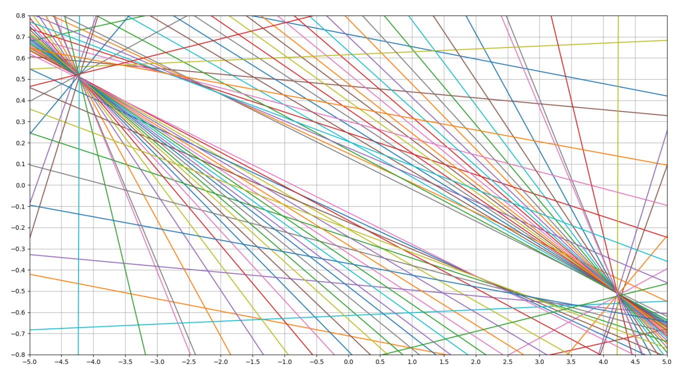
If the input u has n_u dimension, how can we estimate $u_{max,j}$ and $u_{min,j}$ for $j = 0,..,n_u$?

$$
\begin{aligned}
u_{max,j} &= \max_x K_j(A+BK)^{N+1}x \\
s.t.\ u_{min} \le K(A+BK)^i x &\le u_{max},\ i = 0,...,N, \\
u_{min,j} &= \min_x K_j(A+BK)^{N+1}x \\
s.t.\ u_{min} \le K(A+BK)^i x &\le u_{max},\ i = 0,...,N
\end{aligned}
\tag{20}
$$

Hence, terminal constraints set finding can be formulated in the following way:



The terminal set after applying the mentioned algorithm



Assume a system dynamics is described in terms of an LTI (linear time-invariant) state-space model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k, \end{aligned} \tag{21}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, and $y_k \in \mathbb{R}^{n_y}$. Such a control system is assumed to be enforced by a set of linear constraints, i.e., may involve both state and inputs. In general, those can be expressed as

$$Fx + Gu \leq 1, \tag{22}$$

where $F \in \mathbb{R}^{n_c \times n_x}$ and $G \in \mathbb{R}^{n_c \times n_u}$.

Theorem

The MPI (Maximum Positive Invariant) set for the system with dynamics (21) and the system constraints set (22) can be defined as:

$$X^{MPI} \doteq \{x : (F + GK)\Phi^i x \leq 1,\ i = 0, \ldots, v\}, \tag{23}$$

where v is the smallest positive integer such that $(F + GK)\Phi^{v+1}x \leq 1$, $\forall x$ satisfying $(F + GK)\Phi^i x \leq 1,\ i = 0, \ldots, v$. Φ is determined as $A + B \cdot K$. The value of v can be computed by solving the following LPs, namely

$$\begin{aligned} \max_x \quad & (F + GK)_j \Phi^{n+1} x \\ \text{s.t.} \quad & (F + GK)\Phi^i x \leq 1,\ i = 0, \ldots, n \end{aligned} \tag{24}$$

for $j = 1, \ldots, n_c$, n=1,...,v, where $(F + GK)_j$ denotes the jth row of F+GK.

Now considering following second order system with

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \tag{25}$$

whose constraints are given by $-1 \leq x/8 \leq 1$ and $-1 \leq u \leq 1$. With listed constraints we can define the F and G as follows:

$$F = \begin{bmatrix} 0 & 1/8 \\ 1/8 & 0 \\ 0 & -1/8 \\ -1/8 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \tag{26}$$

Calculate the MPI set for this system where assume the value of v some value in between 5 to 20.

The expected output is something similar to this: