SSY281 Model Predictive Control

Assignment 3

Yongzhao Chen

February 21, 2023

1 Q1: Constrained optimization

1.1 a

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if, for any two points x_1, x_2 in the domain of f and any scalar λ between 0 and 1, the following inequality holds:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

A function f is strictly convex if the inequality above is strict, i.e., $f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2)$ for all x_1, x_2 in the domain of f and all λ in $(0, 1)$.

1.2 b

A set $S \subseteq \mathbb{R}^n$ is said to be convex if for any two points x_1, x_2 in S and any scalar λ between 0 and 1, the point $\lambda x_1 + (1 - \lambda)x_2$ also belongs to S. In other words, a set is convex if the line segment connecting any two points in the set is entirely contained in the set.

1.3 c

To determine whether the optimization problem (1) is convex, we need to check whether the objective function f and the constraint functions g and h satisfy certain conditions. Specifically:
f must be a convex function.
g must be a convex function, and its feasible set $x \mid g(x) \leq 0$ must be a convex set.
h must be an affine function, i.e., $h(x) = a^T x + b$ for some vector a and scalar b, and its feasible set $x \mid h(x) = 0$ must be an affine subspace.
If all of these conditions are satisfied, then the optimization problem (1) is convex. In this case, any local minimum of (1) is also a global minimum. If f is strictly convex, then the global minimum is unique.

2 Question 2:Convexity

2.1 a

The set $S1$ is a convex set. To prove this, let x_1 and x_2 be two arbitrary points in $S1$ such that $a^T x_1 \geq \alpha$ and $a^T x_2 \geq \alpha$, and let $\lambda \in [0,1]$ be a scalar. Then we have:

$$a^T(\lambda x_1 + (1-\lambda)x_2) = \lambda(a^T x_1) + (1-\lambda)(a^T x_2) \geq \lambda\alpha + (1-\lambda)\alpha = \alpha$$

Similarly, we can show that $a^T(\lambda x_1 + (1-\lambda)x_2) \leq \beta$. Therefore, $\lambda x_1 + (1-\lambda)x_2$ also belongs to $S1$, and thus $S1$ is convex.

2.2 b

The set $S2$ is also a convex set. To prove this, let x_1 and x_2 be two arbitrary points in $S2$, and let $\lambda \in [0,1]$ be a scalar. Then for any $y \in S$, we have:

$$|(\lambda x_1 + (1-\lambda)x_2) - y| \leq \lambda|x_1 - y| + (1-\lambda)|x_2 - y| \leq \lambda f(y) + (1-\lambda)f(y) = f(y)$$

where we used the triangle inequality and the fact that $f(y) \geq 0$. Therefore, $\lambda x_1 + (1-\lambda)x_2$ also belongs to $S2$ for any $y \in S$, and thus $S2$ is convex.

2.3 c

The set $S3$ is a convex set. To prove this, let (x_1, y_1) and (x_2, y_2) be two arbitrary points in $S3$, and let $\lambda \in [0,1]$ be a scalar. Then we have:

$$y_1 \leq 2x_1 \quad \text{and} \quad y_2 \leq 2x_2$$

Multiplying the first inequality by λ and the second by $(1-\lambda)$ and adding them together, we get:

$$\lambda y_1 + (1-\lambda)y_2 \leq 2(\lambda x_1 + (1-\lambda)x_2)$$

Therefore, $\lambda x_1 + (1-\lambda)x_2$ also belongs to $S3$, and thus $S3$ is convex.

3 Q3:Norm problems as linear programs

3.1 a

The reason why (5) and (3) yield the same results is that the constraints in (5) ensure that each element of the difference vector $Ax - b$ is bounded by ϵ, which is equivalent to saying that the maximum absolute difference between any element of Ax and the corresponding element of b is less than or equal to ϵ. This is precisely the definition of the infinity norm in (4), and hence the optimal value of (5) is equal to the optimal value of (3).

3.2 b

Assuming $z^T = [x^T \ \epsilon], x \in \mathbb{R}^n, z = \begin{bmatrix} x, \\ \epsilon \end{bmatrix}, n = 2$, we can write the optimization problem in (5) as a linear program in the form of (2) by defining the following:

$$\begin{aligned} c &= [0 \ 0 \ \cdots \ 0 \ 1], c \in \mathbb{R}^{n+1} \\ F &= \begin{bmatrix} A & \textit{-1} \\ \text{-}A & \textit{-1} \end{bmatrix} \\ g &= \begin{bmatrix} b \\ \text{-}b \end{bmatrix} \end{aligned} \tag{1}$$

where $\mathbf{1}, \mathbf{b} \in \mathbb{R}^4$ is a vector of ones. The matrix F has 8 rows(Because A has 4 rows). The vector g concatenates the original right-hand side vector b and its negation.
the first n rows represent the inequalities $Ax - b \leq \epsilon(4 \times 1)$ and the last n rows represent the inequalities $-Ax + b \leq \epsilon(4 \times 1)$. Finally finished in (2) edition.
Validation:
$z^T = [x^T \ \epsilon], x \in \mathbb{R}^2, c \in \mathbb{R}^3$, Calculate $c^T z$:

$$c^T z = [0 \ 0 \ \cdots \ 0 \ 1] \begin{bmatrix} x \\ \epsilon \end{bmatrix} = \epsilon \tag{2}$$

So the first equation of (2) appears.
Then we get the second. After F multiply with z , we get this:

$$Fz = \begin{bmatrix} A & \textit{-1} \\ \text{-}A & \textit{-1} \end{bmatrix} \begin{bmatrix} x \\ \epsilon \end{bmatrix} = \begin{bmatrix} Ax - \epsilon \\ -Ax - \epsilon \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix} = g \tag{3}$$

3.3 c

To solve the problem (3) using linear programming in MATLAB, we need to write it in the form given in equation (2). Here, we have $n = 2$, and the infinity norm of a vector $x \in \mathbb{R}^2$ is defined as $|x|_\infty = \max |x_1|, |x_2|$.
Using the formulation from question 3, we can write the problem (3) as a linear program in the following form:

$$\min_{x,\epsilon} \epsilon \text{s.t.} - Ax_i + b_i \leq \epsilon \text{ and } Ax_i - b_i \leq \epsilon \tag{4}$$

We can use the *linprog* function in MATLAB to solve this linear program. Here is the MATLAB code to solve the problem:

```
clear;
clc;
% Define A and b
A = [0.4889 0.2939; 1.0347 -0.7873; 0.7269 0.8884; -0.3034 -1.1471];
b = [-1.0689; -0.8095; -2.9443; 1.4384];

% Reshape A into a matrix of size 2n by n
n = size(A, 2);
F = [A, -ones(2*n,1); -A, -ones(2*n,1)];
g = [b; -b];
c = [0 0 1];

% Solve the linear program
x = linprog(c, F, g);

% Extract the solution for x
x_sol = x(1:n);

% Print the solution
disp(x_sol);
```

The optimal result is $x = \begin{bmatrix} -2.0674 \\ -1.1067 \end{bmatrix}$

3.4 d

To derive the dual of the given primal problem:

$$\min_z c^T z \quad \text{s.t. } Fz \leq g$$

We can introduce a set of Lagrange multipliers $\lambda \geq 0$ for each constraint in the primal problem:

$$L(z, \lambda) = c^T z + \lambda^T (Fz - g)$$

The Lagrangian dual function is defined as the minimum value of the Lagrangian over the primal variables:

$$g(y) = \min_z L(z, \lambda) = \min_z \left(c^T z + \lambda^T F z - \lambda^T g\right)$$

We can obtain the optimal value of z by setting the gradient of the Lagrangian with respect to z to zero:

$$\nabla_z L(z, \lambda) = c + F^T \lambda = 0 \implies q(\lambda) = -\lambda^T g, c + F^T = 0 \text{ or } q(\lambda) = \inf, \text{otherwise}$$

So we get the dual form:

$$\max -\lambda^T g \quad \text{s.t. } c^T + \lambda^T F \geq 0$$

3.5 e

3.6 f

4 Question 4:Quadratic programming

For this question, firstly need to reformulate the constraints into equality and inequality forms.It will result in the same with the lb, ub form, but the standard form of optimization problem requires this procedure.

$$
\begin{aligned}
&x_1 - u_0 = 1 \\
&-0.4x_1 + x_2 - u_1 = 0 \\
&x_1 \leq 5, -x1 \leq -2.5 \\
&x_2 \leq 0.5, -x2 \leq 0.5 \\
&u_0 \leq 2, -u_0 \leq 2 \\
&u_1 \leq 2, -u_1 \leq 2
\end{aligned}
\tag{5}
$$

Then is simple to put that into standard format and solve it using Matlab.

```
 % expand x to [x1 x2 u0 u1]
H = eye(4);
f = zeros(4,1);
% xk+1=0.4xk+uk can be translate into 2 euqations:
% x1 - u0 = 1.5
% x2 - 0.4x1 -u1 = 0
Aeq = [1 0 -1 0;
       -0.4 1 0 -1];
beq = [1.5;0];

% dispate the inequality to single side for the standard form
Aleq= [eye(4);
       -eye(4)];
bleq= [5 0.5 2 2   -2.5 0.5 2 2]';

% lb=[2.5 -0.5 -2 -2]’;
% ub=[5 0.5 2 2]’

options = optimoptions('quadprog','Display','iter');
[x,fval,exitflag,output,lambda] = quadprog(H,f,Aleq,bleq,Aeq,beq
    ,[],[],[],options);
```

4.1 a

The result is :

$$
\begin{aligned}
x_1 &= 2.5000\\
x_2 &= 0.5000\\
u_0 &= 1.0000\\
u1 &= -0.7500
\end{aligned}
\tag{6}
$$

And the value of the min cost function f is fval $= 4.0313$.

4.2 b

The μ^* can be got from the *quadprog* output.
Follow the textbook, these conditions needs to be tell:

$$
\begin{aligned}
\mu^* &\geq 0\\
h\left(x^*\right) &= 0\\
g\left(x^*\right) &\leq 0\\
\mu_i g_i\left(x^*\right) &= 0\\
\nabla f(x^*) + \nabla g(x^*)\mu^* + \nabla h(x^*)\lambda^* &= 0
\end{aligned}
\tag{7}
$$

Calculating these condition values and verify:

```
mu = lambda.ineqlin
if all(mu >= 0)
disp('mu >= 0 is satisfied');
else
    disp('the constraint for mu is not satisfied');
end

gx=Aleq*x-bleq
if all(gx <= 0)
    disp('gx <= 0 is satisfied');
else
    disp('the constraint for gx is not satisfied');
end

hx=Aeq*x-beq
if all( abs(hx) <= 1e-5)
    disp('hx == 0 is satisfied');
else
```

```
    disp('the constraint for hx is not satisfied');
end

muigi=mu.*gx
if all(abs(muigi) <= 1e-5)
    disp('muigi == 0 is satisfied');
else
    disp('the constraint for muigi is not satisfied');
end

kkt_residual=x + Aleq.'*lambda.ineqlin + Aeq'*lambda.eqlin

if norm(kkt_residual) <= 1e-5
    disp('KKT residual is satisfied');
else
    disp('KKT residual is not satisfied');
end

Output:
mu = ×81
         0
    0.0001
    0.0000
         0
    3.7000
         0
         0
    0.0000
mu >= 0 is satisfied

gx = ×81
   -2.5000
   -0.0000
   -1.0000
   -2.5000
   -0.0000
   -1.0000
   -3.0000
   -1.5000
gx <= 0 is satisfied

hx = ×21
1.0e-15 *

```

```
          0
     0.2220
hx == 0 is satisfied

muigi = ×81
1.0e-08 *

          0
    -0.3313
    -0.0039
          0
    -0.0039
          0
          0
    -0.0039

muigi == 0 is satisfied

KKT residual is satisfied
```

So from the verification output, all the 5 KKT conditions are hold.
To determine which constraints are active, we need to look at the complementary slackness conditions, which state that if constraint g_i is inactive at x^*, then $\mu_i^* = 0$. From the values of μ I calculated, we can see that $\mu_1^* = \mu_4^* = \mu_6^* = \mu_7^* = \mu_8^* = 0$. This means that constraints 1, 4, 6, 7, and 8 are inactive at x^*.
The only two non-zero value in μ is μ_2^* *and* μ_5^*, which means that constraint 2 and 5 are active at x^*.
Therefore, the constraints 2 and 5 are is $-x_1 \leq -2.5, x_2 \leq 0.5$, which matches output x^*

4.3 c

Remove the lower bound:

```
Aleq1 = Aleq([1:4],:);
bleq1 = bleq(1:4);

Output:
x = ×41
    0.7212
```

```
    0.1442
   -0.7788
   -0.1442

fval=
0.5841
```

Remove the upper bound:

```
Aleq2 = Aleq([5:8],:);
bleq2 = bleq(5:8);

Output:
x = ×41
    2.5000
    0.5000
    1.0000
   -0.5000

fval=
3.8750
```

From the output, if removing the upper bound, this optimal cost can be reduced a lot to 0.5841. That is because after removing the upper bound, the $x2$ can go up to the optimal position. And now μ^* is all zero, which means no constraint is active.

If removing the lower bound, nothing happen because the optimal value does not lies in the lower bound zone.