Optimization Models in Engineering—Homework 3

1. Consider an aerial system that moves in $\mathbb{R}^3$ according to the dynamics

$$x(k+1) = Ax(k) + Bu(k), \quad k = 0, 1, 2, 3 \tag{1}$$

where $x(k) \in \mathbb{R}^3$ is the position of the system at time $k \in \{0, 1, 2, 3, 4\}$ and $u(k) \in \mathbb{R}$ is the scalar input applied to the system at time k. Assume that the initial position $x(0)$ is equal to $[0 \;\; 0 \;\; 0]^T$. Given a target position $x_d \in \mathbb{R}^3$, the goal is to design the input sequence $u(0), u(1), u(2), u(3)$ to take the system to the target position x_d at time $k = 4$, i.e., $x(4) = x_d$.

   i) Find a matrix $H \in \mathbb{R}^{3\times 4}$ in terms of A and B with the property that

$$x(4) = H \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \end{bmatrix} \tag{2}$$

   ii) Assume that

$$A = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \tag{3}$$

   Show that the vector $[1 \;\; 1 \;\; 0]^T$ belongs to $\mathcal{N}(H^T)$ (note: you are allowed to use a calculator to compute H, but you cannot use a calculator or a computer code to study the null space of H^T and the analysis should be done by hand).

   iii) Again, consider the system parameters given in (3). By studying the relationship between $\mathcal{N}(H^T)$ and $\mathcal{R}(H)$, prove that there is no sequence of inputs that can take the system to the position $x_d = [1 \;\; 1 \;\; 0]^T$ at time 4.

   iv) Again, consider the system parameters given in (3). By finding $\mathcal{N}(H^T)$ and using the relations $\mathcal{N}(H^T) \perp \mathcal{R}(H)$ and $\mathcal{N}(H^T) \oplus \mathcal{R}(H) = \mathbb{R}^3$, show that there exists a sequence of inputs to take the system to the position x_d at time 4 if and only if x_d belongs to the set

$$\{x \in \mathbb{R}^3 \mid x_1 + x_2 = 0\} \tag{4}$$

   v) Now, assume that

$$A = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \tag{5}$$

   The goal is to find a sequence of inputs such that the total energy $u(0)^2 + u(1)^2 + u(2)^2 + u(3)^2$ is minimized and yet the system arrives at the target position $x_d = [3 \;\; 2 \;\; 2]^T$ at time 4. Formulate this as an optimization problem and write a code in CVX to solve the problem numerically. Plot the optimal trajectory (i.e., plot the optimal values of the points $x(0), ..., x(4)$ in $\mathbb{R}^3$ and then connect each point to the next one (such as $x(1)$ to $x(2)$)).

   vi) Consider the safety set

$$\mathcal{S} = \{x \in \mathbb{R}^3 \mid -3.3 \leq x_i \leq 3.3, \quad i = 1, 2, 3\} \tag{6}$$

   Assume that the state $x(k)$ must always stay in the safety set $\mathcal{S}$ for $k = 0, 1, ..., 4$. Redo Part (v) under this additional constraint and find the optimal input sequence. Compares the optimal trajectories and optimal energies (objective values) obtained in Parts (v) and (vi).

Solution.

i) Given the initial condition $x(0) = [0 \;\; 0 \;\; 0]^T$, we can express the positions at subsequent time steps as follows:

(a) $k = 1$:

$$x(1) = Ax(0) + Bu(0)$$

Since $x(0) = [0 \;\; 0 \;\; 0]^T$, $x(1) = Bu(0)$

(b) $k = 2$:

$$x(2) = A(Bu(0)) + Bu(1) = ABu(0) + Bu(1).$$

(c) $k = 3$:

$$x(3) = A(ABu(0) + Bu(1)) + Bu(2) = A^2Bu(0) + ABu(1) + Bu(2).$$

(d) $k = 4$:

$$x(4) = A(A^2Bu(0) + ABu(1) + Bu(2)) + Bu(3) = A^3Bu(0) + A^2Bu(1) + ABu(2) + Bu(3).$$

We can represent $x(4)$ in the matrix form:

$$x(4) = \begin{bmatrix} A^3B & A^2B & AB & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \end{bmatrix} = H \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \end{bmatrix}.$$

Thus,

$$H = \begin{bmatrix} A^3B & A^2B & AB & B \end{bmatrix}. \tag{7}$$

ii) Plug in A and B in (7) and calculate using a calculator, we get

$$H = \begin{bmatrix} 8 & 4 & 2 & 1 \\ -8 & -4 & -2 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Since

$$H^T \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 8 & -8 & 1 \\ 4 & -4 & 1 \\ 2 & -2 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 0,$$

we have $[1 \;\; 1 \;\; 0]^T \in \mathcal{N}(H^T)$.

iii) Since $\mathcal{N}(H^T) \perp \mathcal{R}(H)$ and $x_d \in \mathcal{N}(H^T)$, $x_d \notin \mathcal{R}(H)$. That is, there exists no $u = [u(0) \;\; u(1) \;\; u(2) \;\; u(3)]^T$ such that $x_d = Hu$.

iv) To find $\mathcal{N}(H^T)$, we start with the augmented matrix for the system $Ax = 0$:

$$\left[\begin{array}{ccc|c} 8 & -8 & 1 & 0 \\ 4 & -4 & 1 & 0 \\ 2 & -2 & 1 & 0 \\ 1 & -1 & 1 & 0 \end{array}\right].$$

Divide the first row by 8:

$$\left[\begin{array}{ccc|c} 1 & -1 & \frac{1}{8} & 0 \\ 4 & -4 & 1 & 0 \\ 2 & -2 & 1 & 0 \\ 1 & -1 & 1 & 0 \end{array}\right].$$

Subtract 4 times the first row from the second row, 2 times the first row from the third row, and the first row from the fourth row:

$$\left[\begin{array}{ccc|c} 1 & -1 & \frac{1}{8} & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{7}{8} & 0 \end{array}\right].$$

Multiply the second row by 2:

$$\left[\begin{array}{ccc|c} 1 & -1 & \frac{1}{8} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{7}{8} & 0 \end{array}\right].$$

Subtract $\frac{3}{4}$ times the second row from the third row and $\frac{7}{8}$ times the second row from the fourth row:

$$\left[\begin{array}{ccc|c} 1 & -1 & \frac{1}{8} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Now we have the row-reduced form of the augmented matrix. From this matrix, we can write the system of equations as:

$$\begin{aligned} x_1 - x_2 + \frac{1}{8}x_3 &= 0, \\ x_3 &= 0. \end{aligned}$$

Since $x_3 = 0$, substituting into the first equation gives:

$$x_1 - x_2 = 0 \implies x_1 = x_2.$$

Letting $x_2 = t$ (where t is a free parameter), the solution is:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t \\ t \\ 0 \end{bmatrix} = t \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Thus,

$$\mathcal{N}(H^T) = \text{span}\left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}.$$

There exists a sequence of inputs to take the system to the position x_d at time 4 if and only if $x_d \in \mathcal{R}(H)$.
Necessary condition:
We know that $\mathcal{R}(H) \perp \mathcal{N}(H^T)$. If $x_d \in \mathcal{R}(H)$, it is orthogonal to $[1\ 1\ 0]^T$, i.e., $x_d(1) + x_d(2) = 0$. It belongs to the set $\{x \in \mathbb{R}^3 \mid x_1 + x_2 = 0\}$.
Sufficient condition:
Since $\mathcal{N}(H^T) \oplus \mathcal{R}(H) = \mathbb{R}^3$, $x_d \in \mathbb{R}^3$ can be decomposed to components in $\mathcal{N}(H^T)$ and $\mathcal{R}(H)$. If x_d belongs to the set $\{x \in \mathbb{R}^3 \mid x_1 + x_2 = 0\}$, i.e., it is orthogonal to $\mathcal{N}(H^T)$ and has no component in $\mathcal{N}(H^T)$. Thus, x_d is in $\mathcal{R}(H)$.

v) The objective value is 32.7323 and the trajectory is

$$x_1 = \begin{bmatrix} 1.4076 \\ 1.4076 \\ -1.4076 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -0.0652 \\ -2.8804 \\ -1.3424 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -0.8424 \\ -3.5272 \\ 2.3152 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 3.0000 \\ 2.0000 \\ 2.0000 \end{bmatrix}.$$

The visualization is in Fig.1.
We show the MATLAB code as follows

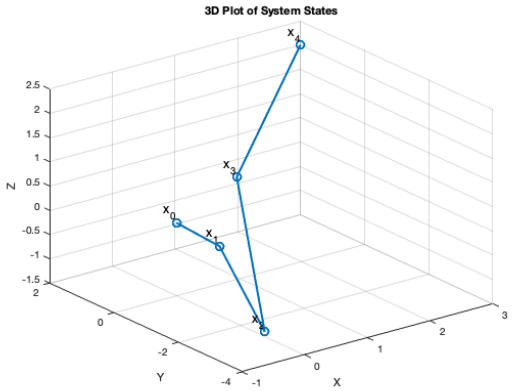

Figure 1: subpart (v) trajectory

```
clear;
A = [2 0 0; -1  1 0 ; -1 -1  1];
B = [-1; -1; 1];
x_d = [3; 2; 2];
H = [A^3*B A^2*B A*B B];
x0 = [0;0;0];
cvx_begin
    variable u(4);
    variable x1(3);
    variable x2(3);
    variable x3(3);
    variable x4(3);
    minimize transpose(u)*u;
    subject to
        x_d == H * u;
        x1 == B*u(1);
        x2 == A*x1 + B*u(2);
        x3 == A*x2 + B*u(3);
        x4 == A*x3 + B*u(4);
cvx_end
% You can also simply do
% cvx_begin
%     variable u(4);
%     minimize transpose(u)*u;
%     subject to
```



```
%         x_d == H * u;
% cvx_end
% and calculate the states x after the optimal u values are found.

X = [x0(1), x1(1), x2(1), x3(1), x4(1)]; % X coordinates
Y = [x0(2), x1(2), x2(2), x3(2), x4(2)]; % Y coordinates
Z = [x0(3), x1(3), x2(3), x3(3), x4(3)]; % Z coordinates

% Plot 3D line
figure;
plot3(X, Y, Z, '-o', 'LineWidth', 2, 'MarkerSize', 8);
grid on;
xlabel('X');
ylabel('Y');
zlabel('Z');
title('3D Plot of System States');
text(x0(1), x0(2), x0(3), 'x_0', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x1(1), x1(2), x1(3), 'x_1', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x2(1), x2(2), x2(3), 'x_2', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x3(1), x3(2), x3(3), 'x_3', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x4(1), x4(2), x4(3), 'x_4', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
```

vi) The objective value is 32.9961 and the trajectory is

$$x_1 = \begin{bmatrix} 1.4833 \\ 1.4833 \\ -1.4833 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -0.2167 \\ -3.1833 \\ -1.2667 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -0.7667 \\ -3.3000 \\ 2.4667 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 3.0000 \\ 2.0000 \\ 2.0000 \end{bmatrix}.$$

The objective function is larger than that in part (v). $x_3(1)$ was out of the safety set in part (v), whereas all points are within the safety set in part (vi). The visualization is in Fig.2.

We show the MATLAB code as follows.

```
clear;
A = [2 0 0; -1  1 0 ; -1 -1  1];
B = [-1; -1; 1];
x_d = [3; 2; 2];
H = [A^3*B A^2*B A*B B];
x0 = [0;0;0];
cvx_begin
    variable u(4);
    variable x1(3);
    variable x2(3);
    variable x3(3);
    variable x4(3);
    minimize transpose(u)*u;
    subject to
        x_d == H * u;
        x1 == B*u(1);
        x2 == A*x1 + B*u(2);
        x3 == A*x2 + B*u(3);
```


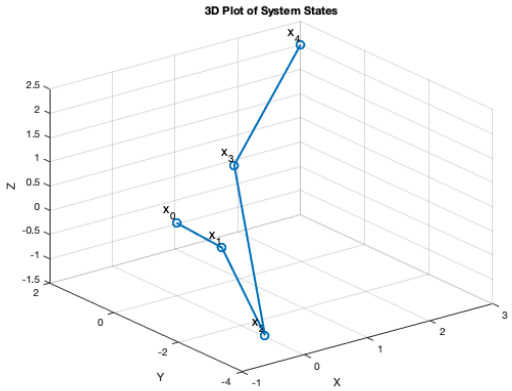

Figure 2: subpart (vi) trajectory

```
        x4 == A*x3 + B*u(4);
        -3.3 <= x1 <= 3.3;
        -3.3 <= x2 <= 3.3;
        -3.3 <= x3 <= 3.3;
        -3.3 <= x4 <= 3.3;
cvx_end

X = [x0(1), x1(1), x2(1), x3(1), x4(1)]; % X coordinates
Y = [x0(2), x1(2), x2(2), x3(2), x4(2)]; % Y coordinates
Z = [x0(3), x1(3), x2(3), x3(3), x4(3)]; % Z coordinates

% Plot 3D line
figure;
plot3(X, Y, Z, '-o', 'LineWidth', 2, 'MarkerSize', 8);
grid on;
xlabel('X');
ylabel('Y');
zlabel('Z');
title('3D Plot of System States');
text(x0(1), x0(2), x0(3), 'x_0', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x1(1), x1(2), x1(3), 'x_1', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x2(1), x2(2), x2(3), 'x_2', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
```



```
text(x3(1), x3(2), x3(3), 'x_3', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
text(x4(1), x4(2), x4(3), 'x_4', 'FontSize', 12, 'VerticalAlignment',
    'bottom', 'HorizontalAlignment', 'right');
```

2. Consider the matrix A and vector x^* defined as

$$A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad x^* = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

Define $b = Ax^* + v$ where $v \in \mathbb{R}^6$ is some measurement noise. Assume that the user has no access to x^* and aims to learn x^* from the measurement vector b. We consider two different estimators to learn x^*:

$$l_1 \text{ estimator:} \qquad \min_x \|Ax - b\|_1, \tag{9a}$$

$$l_2 \text{ estimator:} \qquad \min_x \|Ax - b\|_2 \tag{9b}$$

Given a solution $\hat{x}$ obtained from any of the above estimators, we define the estimation error $e = \|\hat{x} - x^*\|_2$ (note that the error is always computed with respect to the l_2-norm no matter which estimator is used for obtaining $\hat{x}^*$). Assume that the noise v is in the form

$$v = \begin{bmatrix} t_1 \\ 0 \\ 0 \\ 0 \\ t_2 \\ 0 \end{bmatrix} \tag{10}$$

where t_1 and t_2 are constants that belong to the discrete set $\{-2, -1.95, -1.9, ..., -0.05, 0, 0.05, ..., 1.9, 1.95, 2\}$ (the increment is 0.05).

i) For each possible value of the pair (t_1, t_2), solve the l_1 and l_2 estimators in CVX and record the corresponding estimation errors (note: there are 81 possibilities for (t_1, t_2)).

ii) Draw a grid in $\mathbb{R}^2$ obtained as follows: For each possible value of the pair (t_1, t_2), we put a symbol in the location (t_1, t_2) in $\mathbb{R}^2$, where the symbol is a small red circle if the l_1 estimator gives the lowest estimation error and is a small blue circle if the l_2 estimator gives the lowest estimation error (note: if the estimation errors for both estimators are the same, use the blue circle). Analyze the plot and report your observations.

Solution.

i) See the MATLAB code below

```
clear;
A = [1 1 1; -1 1 0; -1 -1 1; 1 0 1; -1 1 1; 0 -1 1];
x_star = [1; 1; 1];

t1_values = -2:0.05:2;
t2_values = -2:0.05:2;

% Initialize matrices to store the errors
norm_1_errors = zeros(length(t1_values), length(t2_values));
norm_2_errors = zeros(length(t1_values), length(t2_values));

% Loop over all values of t1 and t2
```



```
for i = 1:length(t1_values)
    for j = 1:length(t2_values)
        t1 = t1_values(i);
        t2 = t2_values(j);
        v = [t1; 0; 0; 0; t2; 0];
        b = A * x_star + v;

        % Solve the optimization problem using CVX for l1 norm
        cvx_begin quiet
            variable x_hat_norm_1(3)
            minimize(norm(A * x_hat_norm_1 - b, 1))
        cvx_end

        % Solve the optimization problem using CVX for l2 norm
        cvx_begin quiet
            variable x_hat_norm_2(3)
            minimize(norm(A * x_hat_norm_2 - b, 2))
        cvx_end

        % Calculate estimation errors
        norm_1_error = norm(x_hat_norm_1 - x_star, 2);
        norm_2_error = norm(x_hat_norm_2 - x_star, 2);

        % Record the errors
        norm_1_errors(i, j) = norm_1_error;
        norm_2_errors(i, j) = norm_2_error;
    end
end
```

ii) The figure is shown in Fig.3. Analyze the plot we can make the following observations

(a) There are more red circles than blue circles. Since l_1-norm encourages sparsity, which aligns with the form the noise vector is in, most of the time, it has smaller error than l_2-norm.

(b) The plot is symmetric along $t_1 = t_2$. The comparison result is identical for $v = [t_1\ 0\ 0\ 0\ t_2\ 0]^T$ and $v = [t_2\ 0\ 0\ 0\ t_1\ 0]^T$

(c) The plot is symmetric along $t_1 = -t_2$. The comparison result is identical for $v = [t_1\ 0\ 0\ 0\ t_2\ 0]^T$ and $v = [-t_1\ 0\ 0\ 0\ -t_2\ 0]^T$

We show the MATLAB code as follows.

```
clear;
A = [1 1 1; -1 1 0; -1 -1 1; 1 0 1; -1 1 1; 0 -1 1];
x_star = [1; 1; 1];

% Initialize vectors to store points and colors
t1_values = -2:0.05:2;
t2_values = -2:0.05:2;
[X, Y] = meshgrid(t1_values, t2_values);
color_matrix = zeros(size(X)); % To store the color: 1 for red, 2 for
    blue

% Total number of iterations for progress calculation
total_iterations = length(t1_values) * length(t2_values);
progress = 0;

for i = 1:length(t1_values)
```

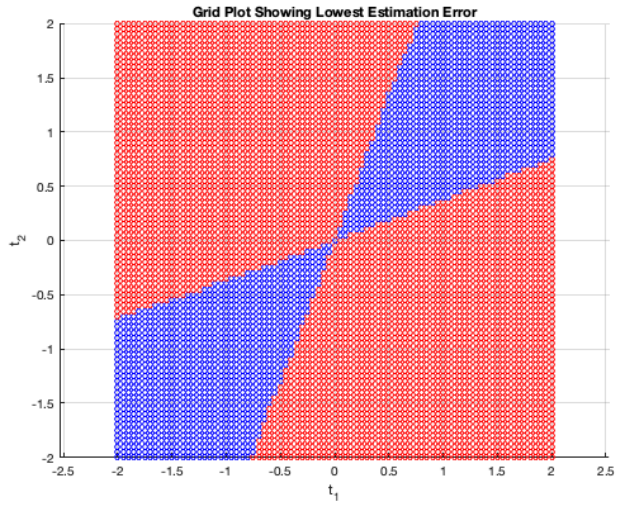

Figure 3: problem 2 grid

```
for j = 1:length(t2_values)
    t1 = t1_values(i);
    t2 = t2_values(j);
    v = [t1; 0; 0; 0; t2; 0];
    b = A * x_star + v;

    % Solve the optimization problem using CVX for l1 norm
    cvx_begin quiet
        variable x_hat_norm_1(3)
        minimize(norm(A * x_hat_norm_1 - b, 1))
    cvx_end

    % Solve the optimization problem using CVX for l2 norm
    cvx_begin quiet
        variable x_hat_norm_2(3)
        minimize(norm(A * x_hat_norm_2 - b, 2))
    cvx_end

    % Calculate estimation errors
    norm_1_error = norm(x_hat_norm_1 - x_star, 2);
    norm_2_error = norm(x_hat_norm_2 - x_star, 2);

    % Determine the color based on the lowest error
    if norm_1_error < norm_2_error
        color_matrix(j, i) = 1; % Red for l1
    else
        color_matrix(j, i) = 2; % Blue for l2 or equal
    end

    % Update and display progress
    progress = progress + 1;
    fprintf('Progress: %.2f%%\n', (progress / total_iterations) *
       100);
end
```



```
end

% Plotting the grid
figure;
hold on;
for i = 1:length(t1_values)
    for j = 1:length(t2_values)
        if color_matrix(j, i) == 1
            plot(t1_values(i), t2_values(j), 'ro', 'MarkerSize', 4); %
                Red circle
        else
            plot(t1_values(i), t2_values(j), 'bo', 'MarkerSize', 4); %
                Blue circle
        end
    end
end
hold off;
xlabel('t_1');
ylabel('t_2');
title('Grid Plot Showing Lowest Estimation Error');
grid on;
axis equal;
```

3. Consider the optimization problem

$$\min_{x\in\mathbb{R}^2} \quad e^{x_1+x_2} + 2x_1^2 + 2x_2^2 - x_1x_2 - \sin(x_1+x_2) \tag{11}$$

By analyzing the gradient and Hessian of the objective function, prove that $x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is a global minimum of the optimization problem (Hint: A matrix $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is positive definite if and only if $a > 0$ and $ac - b^2 > 0$).

Solution. Let $f(x_1, x_2) = e^{x_1+x_2} + 2x_1^2 + 2x_2^2 - x_1x_2 - \sin(x_1+x_2)$. We check that $\nabla f(x^*) = 0$ and that $\nabla^2 f(x)$ is positive definite for all $x \in \mathbb{R}^2$.

First, we calculate the gradient $\nabla f = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}]^T$, where

$$\frac{\partial f}{\partial x_1} = e^{x_1+x_2} + 4x_1 - x_2 - \cos(x_1+x_2),$$

$$\frac{\partial f}{\partial x_2} = e^{x_1+x_2} + 4x_2 - x_1 - \cos(x_1+x_2).$$

Plug in x^*, we have

$$\left.\frac{\partial f}{\partial x_1}\right|_{x=x^*} = 1 + 0 - 0 - 1 = 0,$$

$$\left.\frac{\partial f}{\partial x_2}\right|_{x=x^*} = 1 + 0 - 0 - 1 = 0.$$

Therefore, $\nabla f(x^*) = [0, 0]^T$.

Second, we calculate the Hessian

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix},$$

where

$$\frac{\partial^2 f}{\partial x_1^2} = e^{x_1+x_2} + 4 + \sin(x_1 + x_2),$$

$$\frac{\partial^2 f}{\partial x_2^2} = e^{x_1+x_2} + 4 + \sin(x_1 + x_2),$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = e^{x_1+x_2} - 1 + \sin(x_1 + x_2),$$

Using the hint, we verify that $\frac{\partial^2 f}{\partial x_1^2} > 0$:

$e^{x_1+x_2} + 4 + \sin(x_1 + x_2) > 0$. Since $0 < e^{x_1+x_2}$, $-1 \leq \sin(x_1 + x_2) \leq 1 < 4$. Therefore, $\frac{\partial^2 f}{\partial x_1^2} > 0$.

We also verify that $\frac{\partial^2 f}{\partial x_1^2}\frac{\partial^2 f}{\partial x_2^2} - (\frac{\partial^2 f}{\partial x_1 \partial x_2})^2 > 0$:

$$\frac{\partial^2 f}{\partial x_1^2}\frac{\partial^2 f}{\partial x_2^2} = \left(e^{x_1+x_2} + 4 + \sin(x_1 + x_2)\right)^2.$$

$$(\frac{\partial^2 f}{\partial x_1 \partial x_2})^2 = \left(e^{x_1+x_2} - 1 + \sin(x_1 + x_2)\right)^2.$$

Let $y = e^{x_1+x_2} + \sin(x_1 + x_2)$. We want to show that

$$(y + 4)^2 - (y - 1)^2 > 0$$

Expanding the left hand side:

$$y^2 + 8y + 16 - (y^2 - 2y + 1) = 10y + 15.$$

Since $10e^{x_1+x_2} > 0$ and $10\sin(x_1 + x_2) + 15 > 0$ for all x, it follows that $\frac{\partial^2 f}{\partial x_1^2}\frac{\partial^2 f}{\partial x_2^2} - (\frac{\partial^2 f}{\partial x_1 \partial x_2})^2 > 0$.