Problem 1. LQ games for affine dynamics

Consider the N-person affine quadratic game, described by the state equation

$$x_{k+1} = Ax_k + \sum_{i \in \mathbf{N}} B^i u_k^i + c$$

(where $\mathbf{N}$ is the set of players) with cost function for each player i

$$J^i = \sum_{k=1}^{K} c_k^i(x_{k+1}, u_k^i), \qquad c_k^i(x_{k+1}, u_k^i) = x_{k+1}^\top Q^i x_{k+1} + u_k^{i\,\top} R^i u_k^i.$$

with $x_k, c \in \mathbb{R}^n$, $u_k^i \in \mathbb{R}^{d_i}$, $A \in \mathbb{R}^{n \times n}$, $B^i \in \mathbb{R}^{n \times d_i}$, $Q^i \in \mathbb{R}^{n \times n}$, and $R^i \in \mathbb{R}^{d_i \times d_i}$. Assume that the matrices have been chosen such that $Q^i \succeq 0$ and $R^i \succ 0$.

a) Consider the last stage K, and write the first-order necessary conditions for the optimal feedback strategies that minimise the remaining cost-to-go $c_K^i(x_{K+1}, u_K^i)$.

b) Prove that the Nash equilibrium strategies at time K must be affine in x_K, i.e. they must have the form

$$\gamma_K^{i*} = -P_K^i x_K - \alpha_K^i \ , \ i \in \mathbf{N}$$

and explain in detail how you can compute the parameters P_K^i and α_K^i.

c) Show that the value function at time K is a function of the state x of the form

$$V_K^i = x_K^\top S_K^i x_K + (r_K^i)^\top x_K + q_K^i \ , \ i \in \mathbf{N}.$$

d) Using an induction argument, provide recursive expression for the computation of the sub-game perfect Nash equilibrium strategy for the entire game (stages $1, \ldots, K$).

e) Consider a scalar LQ game with two agents, where $n = 1$ and $d_i = 1 \ \forall i \in \{1, 2\}$. Is the Nash equilibrium unique?

f) Provide an algorithm to compute a Nash equilibrium strategy for this game which is NOT a sub-game perfect Nash equilibrium.

Solution:

a) At the last stage K the remaining cost to go for player i is

$$\begin{aligned} c_K^i(x_{K+1}, u_K^i) &= x_{K+1}^\top Q^i x_{K+1} + u_K^{i\,\top} R^i u_K^i \\ &= \left(Ax_K + \sum_{j \in \mathbf{N}} B^j u_K^j + c\right)^\top Q^i \left(Ax_K + \sum_{j \in \mathbf{N}} B^j u_K^j + c\right) + u_K^{i\,\top} R^i u_K^i \end{aligned} \tag{0.1}$$

which is a quadratic function of u_K^i. The first order necessary condition for optimality of the feedback is therefore obtained by taking the first derivative with respect to u_K^i, obtaining

$$\left((B^i)^\top Q^i B^i + R^i\right) u_K^i + (B^i)^\top Q^i \left(Ax_K + \sum_{j \neq i} B^j u_K^j + c\right) = 0. \tag{0.2}$$

b) The optimal control $u_K^i = \gamma_K^{i*}$ is obtained by solving (0.2) simultaneously by all agents. As in (0.2) all the u_K^i and x_K appear linearly, the solution γ_K^{i*} must be an affine function of x_K.

In fact, one can compute γ_K^{i*} by solving the following system of linear equations, which is just a reformulation of (0.2) in which terms have been simply reordered.

$$\underbrace{\begin{bmatrix} (B^1)^\top Q^1 B^1 + R^1 & \dots & (B^1)^\top Q^1 B^i & \dots & (B^i)^\top Q^1 B^N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ (B^i)^\top Q^i B^1 & \dots & (B^i)^\top Q^i B^i + R^i & \dots & (B^i)^\top Q^i B^N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ (B^N)^\top Q^N B^1 & \dots & (B^N)^\top Q^N B^i & \dots & (B^N)^\top Q^N B^N + R^N \end{bmatrix}}_{\Delta} \gamma_K^* + \underbrace{\begin{bmatrix} \vdots \\ (B^i)^\top Q^i A \\ \vdots \end{bmatrix}}_{\Lambda} x_K + \underbrace{\begin{bmatrix} \vdots \\ (B^i)^\top Q^i c \\ \vdots \end{bmatrix}}_{\eta} = 0 \tag{0.3}$$

Therefore the optimal solution for player i has the form

$$\gamma_K^{i*} = -P_K^i x_K - \alpha_K^i \tag{0.4}$$

where

$$P_K^i = \Delta^{-1}\Lambda \qquad \alpha_K^i = \Delta^{-1}\eta.$$

c) The value function at time K can be obtained by simply plugging (0.4) into (0.1). One obtains

$$V_K^i = \left(A x_K + \sum_{j\in\mathbf{N}} B^j(-P_K^j x_K - \alpha_K^j) + c \right)^\top Q^i \left(A x_K + \sum_{j\in\mathbf{N}} B^j(-P_K^j x_K - \alpha_K^j) + c \right)$$
$$+ (-P_K^i x_K - \alpha_K^i)^\top R^i (-P_K^i x_K - \alpha_K^i)$$

which is clearly a quadratic form in x_K.

Namely we have

$$\begin{aligned} S_K &= (A - \sum_j B^j P_K^j)^\top Q^i (A - \sum_j B^j P_K^j) + (P_K^i)^\top R^i P_K^i \\ (r_K^i)^\top &= (-\sum_j B^j \alpha_K^j + c)^\top Q^i (A - \sum_j B^j P_K^j) + (\alpha_K^i)^\top R^i P_K^i \\ q_K^i &= (\sum_j B^j \alpha_K^j + c)^\top Q^i (\sum_j B^j \alpha_K^j + c) + (\alpha_K^i)^\top R^i \alpha_K^i. \end{aligned}$$

d) Let's consider stage $K-1$. The remaining cost to go is given by

$$\begin{aligned} & c_{K-1}^i(x_K, u_{K-1}^i) + V_K(x_K) \\ &= x_K^\top Q^i x_K + (u_{K-1}^i)^\top R^i u_{K-1}^i + x_K^\top S_K^i x_K + (r_K^i)^\top x_K + q_K^i \\ &= x_K^\top (Q^i + S_K^i) x_K + (u_{K-1}^i)^\top R^i u_{K-1}^i + (r_K^i)^\top x_K + q_K^i \\ &= (A x_{K-1} + \sum_j B^j u_{K-1}^j + c)^\top (Q^i + S_K^i)(A x_{K-1} + \sum_j B^j u_{K-1}^j + c) \\ &\quad + (u_{K-1}^i)^\top R^i u_{K-1}^i + (r_K^i)^\top (A x_{K-1} + \sum_j B^j u_{K-1}^j + c) + q_K^i \end{aligned}$$

which is a quadratic form in u_{K-1}^i. Proceeding as before, we find the optimal u_{K-1}^i as the zero of the gradient:

$$((B^i)^\top (Q^i + S_K^i) B^i + R^i) u_{K-1}^i + (B^i)^\top (Q^i + S_K^i)(A x_{K-1} + \sum_{j\neq i} B^j u_{K-1}^j + c) + (B^i)^\top r_K^i = 0$$

As before, in order to solve the optimality condition for all the inputs at time $K-1$, we can solve a system of linear equations that will return an affine form in x_{K-1}. The expressions for Δ and Γ are the same, with the only difference that Q^i is replaced by $Q^i + S^i$. The term η is a bit different, as its terms have now the form $(B^i)^\top (Q^i + S_K^i) c + (B^i)^\top r_K^i$.

To conclude the inductive reasoning, we need to prove that the value function at step $K-1$ is again a quadratic function of the state.

We can do that by simply plugging the optimal control

$$\gamma_{K-1}^{i*} = -P_{K-1}^i x_{K-1} - \alpha_{K-1}^i$$

into the cost $c_{K-1}^i(x_K, u_{K-1}^i) + V_K(x_K)$ and obtain

$$V_{K-1}^i = x_{K-1}^\top S_{K-1}^i x_{K-1} + (r_{K-1}^i)^\top x_{K-1} + q_{K-1}^i$$

with

$$\begin{aligned}
S_{K-1} &= (A - \sum_j B^j P_{K-1}^j)^\top (S_K^i + Q^i)(A - \sum_j B^j P_{K-1}^j) + (P_{K-1}^i)^\top R^i P_{K-1}^i \\
(r_{K-1}^i)^\top &= (-\sum_j B^j \alpha_K^j + c)^\top (S_K^i + Q^i)(A - \sum_j B^j P_{K-1}^j) + \\
&\quad (r_K^i)^\top (A - \sum_j B^j P_{K-1}^j) + (\alpha_{K-1}^i)^\top R^i P_{K-1}^i \\
q_{K-1}^i &= \text{...not needed for the iterative algorithm}
\end{aligned}$$

e) The Nash equilibrium is unique if and only if the linear system 0.3 has a unique solution. System 0.3 is an equation of the form $Ax = b$. There exists a solution if $b \in range(A)$ and the solution is unique if and only if A has full column rank, and since in this case A is square, this is equivalent to invertibility of A, which is equivalent to eigenvalues being non-zero.

Thus, we can see if the system has a unique solution by checking if the matrix Δ is invertible. For the two agents scalar LQ game, the matrix Δ at the last time step is:

$$\Delta = \begin{bmatrix} r_1 + q_1 b_1^2 & q_1 b_1 b_2 \\ q_2 b_1 b_2 & r_2 + q_2 b_2^2 \end{bmatrix}.$$

A matrix is invertible if and only if its determinant is different from zero. We can thus check the determinant of Δ:

$$\begin{aligned}
\det(\Delta) &= (r_1 + q_1 b_1^2)(r_2 + q_2 b_2^2) - q_1 q_2 b_1^2 b_2^2 \\
&= r_1 r_2 + r_1 q_2 b_2^2 + r_2 q_1 b_1^2 > 0.
\end{aligned}$$

Thus the matrix is always invertible. For the previous time steps, the expression for matrix Δ is the same, thus the same proof holds.

f) There are different ways to compute a NE strategy which is not subgame perfect. For example, one could use the subgame perfect NE strategies γ_k^{i*} that we computed (that are functions of the state x_k!) and evaluate them along the optimal trajectory x_k^*. The resulting control actions $\hat{u}_k^i = \gamma_k^{i*}(x_k^*)$ are not a function of x_k any more, and if applied in open loop to the system they are a non-subgame-perfect NE. Another possible way is to construct a quadratic optimization problem with the total cost and the system dynamics as constraints. The solution of this optimizaton problem are a NE, but when applied in open loop they are not subgame perfect.

Problem 2. A three-truck platoon

Consider the three-truck platoon example described at slide 12 of the Dynamic games lecture (Lecture 8) as an LQ game. The leading truck is moving at a constant speed, $s^{(1)}$. The second truck aims to maintain a distance, D, both from the first truck ahead and the third truck behind. Similarly, the last truck aims to keep the same distance D from the second truck. Consider $x^{(i)}$ and $s^{(i)}$ as respectively the positions and speeds of trucks, for $i = 1, 2, 3$. The derivative of the position is equal to the speed

$$\dot{x}^{(i)} = s^{(i)},$$

and the derivative of the speed is equal to the acceleration

$$\dot{s}^{(i)} = u^{(i)}.$$

Using Euler discretization, the dynamics can then be written as

$$\begin{aligned} x_{k+1}^{(i)} &= x_k^{(i)} + \Delta_T s_k^{(i)} \\ s_{k+1}^{(i)} &= s_k^{(i)} + \Delta_T u_k^{(i)} \end{aligned}$$

where Δ_T is the discretization time.
Let the relative position of trucks i, j be denoted by $d_k^{(ij)} = x_k^{(i)} - x_k^{(j)}$.

a) Write the discrete-time equation for the relative positions.

b) Identify the state matrix A, the control matrices B_i, and the cost matrices Q_i and R_i.

c) Write the equations corresponding to the subgame perfect Nash equilibrium.

Hint: the quadratic term in the cost function will not be $d^\top Qd$ but $(d - \bar{d})^\top Q(d - \bar{d})$. For this reason, the value function of the an agent is given by three terms: $V_{2,k}(d) = d^\top P_{2,k} d + (r_{2,k})^\top d + C_{2,k}$. Thus, for the second question of the problem, you can repeat the steps at slide 26 of Lecture 8 with the value function proposed. When an equation becomes too long, substitute some terms with new variables to simplify the writing.

Solution:

a) The state dynamic can thus be written as:

$$x_{k+1} = \begin{bmatrix} d_{k+1}^{(12)} \\ d_{k+1}^{(23)} \\ s_{k+1}^{(2)} \\ s_{k+1}^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T & 0 \\ 0 & 1 & T & -T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_k^{(12)} \\ d_k^{(23)} \\ s_k^{(2)} \\ s_k^{(3)} \end{bmatrix} + \begin{bmatrix} T \\ 0 \\ 0 \\ 0 \end{bmatrix} s^{(1)} + \begin{bmatrix} 0 \\ 0 \\ T \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix} v_k.$$

The cost function for the second truck is:

$$\begin{aligned} J_2(x, u, v) &= \sum_{k=0}^{T} \left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + (u_k)^2 \\ &= \sum_{k=0}^{T} \left(x_k - \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} D \right)^\top \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left(x_k - \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} D \right) + (u_k)^2 . \end{aligned}$$

The cost function for the third ruck is:

$$\begin{aligned} J_3(x, u, v) &= \sum_{k=0}^{T} \left(d_k^{(23)} - D\right)^2 + (v_k)^2 \\ &= \sum_{k=0}^{T} \left(x_k - \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} D \right)^\top \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left(x_k - \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} D \right) + (u_k)^2 . \end{aligned}$$

So the matrices are:

$$A = \begin{bmatrix} 1 & 0 & -T & 0 \\ 0 & 1 & T & -T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 \\ 0 \\ T \\ 0 \end{bmatrix},\ B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ R_2 = 1,$$

$$Q_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ R_3 = 1.$$

b) For the first agent

$$J_2(x,u,v) = \sum_{k=0}^{T} \left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + (u_k)^2 \tag{0.5}$$

Using the value function we have:

$$\begin{aligned}
V_{2,k}(x_k, v_k) &= \min_{u_k}\left[\left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + (u_k)^2 + V_{2,k+1}(x_{k+1})\right] \\
&= \left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + \min_{u_k}\left[(u_k)^2 + x_{k+1}^\top (S_{2,k+1})x_{k+1} + (r_{2,k+1})^\top x_{k+1} + q_{2,k+1}\right] \\
&= \left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + \min_{u_k}\left[(u_k)^2 + \right. \\
&\quad (Ax_k + Cs_k^{(1)} + B_2u_k + B_3v_k)^\top (S_{2,k+1})x_{k+1}(Ax_k + Cs_k^{(1)} + B_2u_k + B_3v_k) + \\
&\quad \left.(r_{2,k+1})^\top (Ax_k + Cs_k^{(1)} + B_2u_k + B_3v_k) + q_{2,k+1}\right] \\
&= \left(d_k^{(12)} - D\right)^2 + \left(d_k^{(23)} - D\right)^2 + \min_{u_k}\left[u_k^\top (1 + B_2^\top (S_{2,k+1})B_2)u_k + \right. \\
&\quad 2u_kB_2^\top\left[(S_{2,k+1})(Ax_k + Cs_k^{(1)} + B_3v_k) + (r_{2,k+1})\right] + \\
&\quad (Ax_k + Cs_k^{(1)} + B_3v_k)^\top (S_{2,k+1})(Ax_k + Cs_k^{(1)} + B_3v_k) + \\
&\quad \left.(r_{2,k+1})^\top (Ax_k + Cs_k^{(1)} + B_3v_k) + q_{2,k+1}\right]
\end{aligned}$$

To find the minimizer, since $1 + B_2^\top (S_{2,k+1})B_2 \succ 0$, we can find the best response by setting the gradient with respect to u_k to zero:

$$(1 + B_2^\top (S_{2,k+1})B_2)u_k + B_2^\top\left[(S_{2,k+1})(Ax_k + Cs_k^{(1)} + B_3v_k) + (r_{2,k+1})\right] = 0,$$

which gives us:

$$u_k^*(v_k) = -\frac{B_2^\top\left[(S_{2,k+1})(Ax_k + Cs_k^{(1)} + B_3v_k) + (r_{2,k+1})\right]}{(1 + B_2^\top (S_{2,k+1})B_2)} = -\Gamma_{1,x,k}x_k - \Gamma_{1,v,k}v_k - \Gamma_{1,k}.$$

Similarly, we can find that the optimal v_k^*, given a certain u_k, is:

$$v_k^*(u_k) = -\frac{B_3^\top\left[S_{3,k+1}(Ax_k + Cs_k^{(1)} + B_2u_k) + (r_{3,k+1})^\top\right]}{(1 + B_3^\top S_{3,k+1}B_3)} = -\Gamma_{2,x,k}x_k - \Gamma_{2,u,k}u_k - \Gamma_{2,k}.$$

The Nash equilibrium is given by the following system:

$$\begin{cases} u_k^*\left(v_k^{NE}\right) &= u_k^{NE} \\ v_k^*\left(u_k^{NE}\right) &= v_k^{NE} \end{cases}$$

Substituting the equations we found, we obtain:

$$\begin{cases} u_k^{NE} & = -\Gamma_{1,x,k} x_k - \Gamma_{1,v,k} v_k^{NE} - \Gamma_{1,k} \\ v_k^{NE} & = -\Gamma_{2,x,k} x_k - \Gamma_{2,u,k} u_k^{NE} - \Gamma_{2,k} \end{cases}$$

By substituting v_k^{NE} in u_k^{NE}, we obtain:

$$\begin{aligned} u_k^{NE} &= -\Gamma_{1,x,k} x_k - \Gamma_{1,v,k}\left(-\Gamma_{2,x,k} x_k - \Gamma_{2,u,k} u_k^{NE} - \Gamma_{2,k}\right) - \Gamma_{1,k} \\ &= \left(-\Gamma_{1,x,k} + \Gamma_{1,v,k}\Gamma_{2,x,k}\right) x_k + \Gamma_{1,v,k}\Gamma_{2,x,k} u_k^{NE} + \left(-\Gamma_{1,k} + \Gamma_{1,v,k}\Gamma_{2,k}\right) \end{aligned}$$

and thus, with some simple computation:

$$\begin{aligned} u_k^{NE} &= \frac{-\Gamma_{1,x,k} + \Gamma_{1,v,k}\Gamma_{2,x,k}}{1 - \Gamma_{1,v,k}\Gamma_{2,x,k}} x_k + \frac{-\Gamma_{1,k} + \Gamma_{1,v,k}\Gamma_{2,k}}{1 - \Gamma_{1,v,k}\Gamma_{2,x,k}} \\ &= K_{1,k} x_k + \gamma_{1,k}. \end{aligned}$$

For v_k^{NE} we will obtain a similar result:

$$v_k^{NE} = K_{2,k} x_k + \gamma_{2,k}.$$

Now that we have the values of u_k^{NE} and v_k^{NE} as a function of x_k, we can substitute them in the expression for $V_{2,k}(x_k, v_k^{NE})$ to compute $S_{2,k}$, $r_{2,k}$ and $q_{2,k}$ (and the same for $V_{3,k}(x_k, u_k^{NE})$).