Multiagent decision-making and control

Dynamic games

Maryam Kamgarpour

Professor of Engineering (IGM, STI), EPFL

ME-429, Spring 2025, EPFL

Course topics

1. Static games
2. Zero-sum games
3. Potential games
4. Extensive form games
5. Randomized strategies in extensive form games
6. Dynamic games, dynamic programming for games
7. Dynamic games, linear quadratic games
8. Convex games, Nash equilibria characterization
9. Convex games, Nash equilibria computation
10. Auctions
11. Bayesian games
12. Learning in games
13. Final project presentations

Admin matter

- Timeline: project update: April 16, 13:15-14:00, 10%
- Template for presentation was posted on last week's Moodle

Review - loop model

Dynamic game (first zero-sum setting for simplicity)

There exists a **state** that evolves at each stage

$$x_{k+1} = f(x_k, u_k, v_k).$$

The **outcome** of the game can be expressed as

$$\sum_{k=1}^{K} g_k(x_k, u_k, v_k).$$

- player 1 minimizer, player 2 maximizer

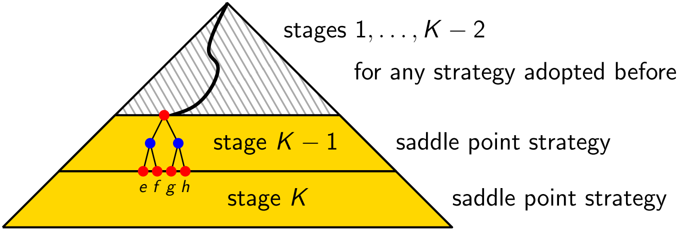
Last time, we motivated and defined subgame perfect equilibrium.
Today's goals:

- more intuition into subgame perfect equilibrium
- computation for specific class of dynamic games: linear quadratic games

Backward induction to compute subgame perfect equilibria

Subgame-perfect Nash Equilibrium

A strategy is a **subgame perfect** equilibrium if it represents a NE of every subgame of the original game.



At every stage, we have a static **simultaneous game**

Stage game at time K

$$V_K(x) = \underbrace{\inf_{u \in U} \sup_{v \in V} g_K(x, u, v)}_{\text{equilibrium for } g_K(x,u,v)}$$

Backward induction in 2-player zero-sum setting

STEP K**:** Consider all the infinite subgames rooted in x_K:

(Notice the abuse of notation: P1 node / state)

$$x_{K+1} = f(x_K, u_K, v_K)$$

with outcome

$$g_K(x_K, u_K, v_K)$$

Determine γ_K^*, σ_K^* (functions of x_K) such that

$$g_K(x_K, \gamma_K^*(x_K), \sigma_K(x_K)) \ \leq \ g_K(x_K, \gamma_K^*(x_K), \sigma_K^*(x_K)) \ \leq \ g_K(x_K, \gamma_K(x_K), \sigma_K^*(x_K))$$

Backward induction in 2-player zero-sum setting continued

Value function $V_K(x_K)$: value of the subgame rooted in x_K, that is

$$V_K(x_K) = g_K(x_K, \gamma_K^*(x_K), \sigma_K^*(x_K))$$

STEP $K-1$**:** Consider all the infinite subgames rooted in x_{K-1}

$$x_K = f(x_{K-1}, u_{K-1}, v_{K-1})$$

K-1

with outcome

$$\sum_{k=K-1}^{K} g_k(x_k, u_k, v_k)$$

which we rewrite as

$$g_{K-1}(x_{K-1}, u_{K-1}, v_{K-1}) + V_K(x_K)$$

and therefore

$$g_{K-1}(x_{K-1}, u_{K-1}, v_{K-1}) + V_K(f(x_{K-1}, u_{K-1}, v_{K-1}))$$

K-1

Backward induction

$$J_{K-1}(x_{K-1}, u_{K-1}, v_{K-1})$$

$$\| $$

$$\overbrace{g_{K-1}(x_{K-1}, u_{K-1}, v_{K-1}) + V_K(f(x_{K-1}, u_{K-1}, v_{K-1}))}$$

Determine $\gamma^*_{K-1}, \sigma^*_{K-1}$ (functions of x_{K-1}) that are saddle-points for

$$g_{K-1}(x_{K-1}, \gamma^*_{K-1}(x_{K-1}), \sigma^*_{K-1}(x_{K-1})) + V_K(f(x_{K-1}, \gamma^*_{K-1}(x_{K-1}), \sigma^*_{K-1}(x_{K-1})))$$

and so on, backward until **stage 1**.

$$J_{K-1}(x_{K-1}, \gamma^*_{K-1}(x_{K-1}), \sigma_{K-1}(x_{K-1})) \leq J_{K-1}(x_{K-1}, \gamma^*_{K-1}(x_{K-1}), \sigma^*_{K-1}(x_{K-1})) \leq J_{K-1}(x_{K-1}, \gamma_{K-1}(x_{K-1}), \sigma^*_{K-1}(x_{K-1}))$$

$$\forall\, \sigma_{K-1} : X \to V \qquad \forall\, \gamma_{K-1} : X \to U$$

Dynamic games with N players, Nash equilibrium

We can generalize what we saw for two players in last lecture to N players

- Dynamics $x_{k+1} = f_k(x_k, u_k^1, \dots, u_k^N)$, $x_k \in X$
- Player i's input set: U^i—can be time and state dependent, namely: $U_k^i(x)$
- Player i's cost function: $\sum_{k=1}^{K} g_k^i(x_k, u_k^1, \dots, u_k^N)$
- State-feedback policy: $\pi_k^i : X \to U^i$, $\pi_k^i(x_k) = u_k^i$
- Open-loop policy: $\pi_k^i : X \to U^i$, $\pi_k^i(x_1) = u_k^i$

at $k = K$

final equilibrium

$\{g_K^i(x, u^1, u^2, \dots, u^N)\}_{i=1}^N$

Finding subgame perfect equilibrium by backward induction:
Set $V_{K+1}^i(x) = 0$, $i \in \{1, 2, \dots, N\}$. For $k = K, K-1, \dots, 1$,

- find a Nash equilibrium policy at stage k: $\{\pi_k^{*,i}(x)\}_{i=1}^N$
- compute $V_k^i(x)$ corresponding to the above polices, $i \in \{1, 2, \dots, N\}$

An important extension: infinite horizon

Let us present it for simplicity in notation in single-player setting
player's cost: $\lim_{K\to\infty}\sum_{k=1}^{K} g(x_k, u_k)$
ensure convergence of the sum, one approach is geometrically discounting costs

- infinite horizon discounted cost

$$\sum_{k=1}^{\infty} \alpha^k g(x_k, u_k)$$

  - $\alpha \in [0, 1)$ and $\exists M > 0$ such that $|g(x, u)| \leq M$ for all $x \in X, u \in U$
  $\implies \sum_{k=1}^{\infty} \alpha^k g(x_k, u_k) < \infty$ $\leq \sum_{k=0}^{\infty} \alpha^k M = \frac{M}{1-\alpha}$

- Bellman equation

$$V^*(x) = \min_{u\in U}\{g(x, u) + \alpha V^*(f(x, u))\},\ \forall x \in X$$

- Optimal stationary policy

$$\pi^*(x) = \arg\min_{u\in U} g(x, u) + \alpha V^*(f(x, u)),\ \forall x \in X$$

Computation of subgame perfect equilibrium policies

Dynamic game (2-player zero-sum)

There exists a **state** that evolves at each stage

$$x_{k+1} = f(x_k, u_k, v_k).$$

The **outcome** of the game can be expressed as

$$\sum_{k=1}^{K} g_k(x_k, u_k, v_k).$$

Finding the NE strategy for a dynamic game corresponds to solving an optimization problem for every stage.
These problems are in general difficult to solve.

One special case:

- **Linear** state update f
- **Quadratic** cost function g.

One-player LQR

Consider the one-player case first.

One-player LQR

Consider the one-player case first.

Update equation

$$x_{k+1} = Ax_k + Bu_k, \; x \in \mathbb{R}^n, u \in \mathbb{R}^m$$

One-player LQR

Consider the one-player case first.

Update equation

$$x_{k+1} = Ax_k + Bu_k,\ x \in \mathbb{R}^n, u \in \mathbb{R}^m$$

Cost function

$$\sum_{k=0}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_K^T S x_K,$$

R, Q are symmetric and $R \succ 0$, $Q \succeq 0$

One-player LQR

Consider the one-player case first.

Update equation

$$x_{k+1} = Ax_k + Bu_k, \; x \in \mathbb{R}^n, u \in \mathbb{R}^m$$

Cost function

$$\sum_{k=0}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_K^T S x_K,$$

R, Q are symmetric and $R \succ 0$, $Q \succeq 0$
Value function (minimum cost to go, starting from x at time k)

$$V_k(x) = \min_{u_k, \ldots u_{K-1}} \sum_{s=k}^{K-1} \left(x_s^T Q x_s + u_s^T R u_s \right) + x_K^T S x_K$$

$V_0(x_0)$ is the value of the game.

One-player LQR

$x^T S x$

Key ideas in the derivation of the NE feedback control law.

1. $V_K(x) = x^T Q_f x$ (quadratic function)
2. We will show that $V_k(x)$ is also quadratic: $V_k(x) = x^T P_k x$
3. P_k can be found recursively, working backward from K
4. The NE u_k is a function of P_k and can be found easily because P_k is positive definite.

We proceed by induction at stage k, where the value function is

$$V_k(x) = \min_u x^T Qx + u^T Ru + V_{k+1}(Ax + Bu)$$

optimal input (since single player)

Induction

recall

$$V_k(x) = \min_u x^T Q x + u^T R u + V_{k+1}(f(x,u))$$

We proceed by induction, assuming that $V_{k+1}(x) = x^T P_{k+1} x$. Then $V_k(x)$ becomes

$$V_k(x) = x^T Q x + \min_u u^T R u + (Ax + Bu)^T P_{k+1} (Ax + Bu) =$$

$$x^T Q x + \min_u u^T R u + x^T A^T P_{k+1} A x + 2u^T B^T P_{k+1} A x + u^T B^T P_{k+1} B u$$

$$= x^T Q x + \min_u u^T (R + B^T P_{k+1} B) u + x^T A^T P_{k+1} A x + 2u^T B^T P_{k+1} A x$$

To find the minimizer, we notice that $R + B^T P_{k+1} B \succ 0$. Hence, this is a strongly convex objective in $u \in \mathbb{R}^m$ and by setting the gradient with respect to u to zero, we can find the optimizer.

positive definite $\Leftrightarrow$ all eigen values strictly > 0 $\Rightarrow$ invertible & function is strongly convex

$$(R + B^T P_{k+1} B) u + B^T P_{k+1} A x = 0$$

which gives the optimal feedback control

$$u^* = -(R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x$$

Induction

Now we need to check that $V_k(x)$ is also quadratic in x, i.e. $V_k(x) = x^T P_k x$, and to find how to compute P_k.
We plug the minimizer u^* into $V_k(x)$, and obtain the expression

$$\begin{aligned} V_k(x) = x^T Q x + x^T A P_{k+1} B (R + B^T P_{k+1} B)^{-1} (R + B^T P_{k+1} B) \cdot \\ \cdot (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x + x^T A^T P_{k+1} A x \\ - 2 x^T A P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x. \end{aligned}$$

Just collecting the different terms we get $V_k(x) = x^T P_k x$ where P_k is defined as

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

which completes the proof.

One-player LQR

Optimal feedback policy is linear:

- Value function $V_k(x) = x^T P_k x$, $P_k \succeq 0$

One-player LQR

Optimal feedback policy is linear:

- Value function $V_k(x) = x^T P_k x$, $P_k \succeq 0$
- Subgame-perfect NE input is given by the linear state feedback

$$u_k = \Gamma_k x_k$$

where for all $k = 0, \ldots, K-1$ we define

$$\Gamma_k = -(R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

One-player LQR

Optimal feedback policy is linear:

- Value function $V_k(x) = x^T P_k x$, $P_k \succeq 0$
- Subgame-perfect NE input is given by the linear state feedback

$$u_k = \Gamma_k x_k$$

where for all $k = 0, \dots, K-1$ we define

$$\Gamma_k = -(R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

- For $k = K, \dots, 1$,

$$P_{k-1} = Q + A^T P_k A - A^T P_k B (R + B^T P_k B)^{-1} B^T P_k A$$

starting from $P_K = S$.

Comments on one-player LQR

Symmetry and positive semi-definitess of P_k

Also this can be verified by induction.

Comments on one-player LQR

Symmetry and positive semi-definitess of P_k

Also this can be verified by induction.

$K \to \infty$

If P_0 converges to some $\bar{P}$ when $K \to +\infty$, then $\bar{P}$ can be evaluated via a **discrete Algebraic Riccati Equation**:

$$\bar{P} = Q + A^T \bar{P} A - A^T \bar{P} B (R + B^T \bar{P} B)^{-1} B^T \bar{P} A$$

Exercise

How quickly does P_k converge to $\bar{P}$ in a simple example,
and what kind of performance is given by $\bar{P}$

Comments on one-player LQR

Time-varying A, B, Q, R

The entire derivation (for finite horizons) works also in the case of time-varying system parameters (for example, linearization of a non-linear system along the optimal trajectory).

Comments on one-player LQR

Time-varying A, B, Q, R

The entire derivation (for finite horizons) works also in the case of time-varying system parameters (for example, linearization of a non-linear system along the optimal trajectory).

Alternative approach in finite horizon

If there are state and input constraints, the Riccati equation above will not hold. We can use numerical computation approaches to solve the following

$$
\begin{aligned}
\min_{u_0,\ldots,u_{K-1}} \quad & \sum_{k=0}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_K^T S x_K \\
\text{subject to} \quad & x_{k+1} = A x_k + B u_k \quad \text{for } k = 0, \ldots, K-1 \\
& x_k \in \mathcal{X}_k \\
& u_k \in \mathcal{U}_k
\end{aligned}
$$

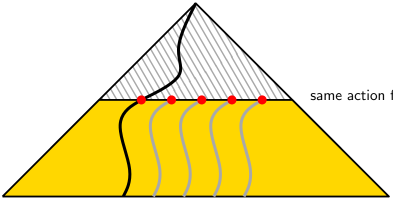
Open-loop NE strategy

By solving the optimization **in one shot** (no DP), we get

$$u_0, u_1, \ldots, u_{K-1} \in \mathbb{R}^m$$

instead of a state feedback

$$u_0(x_0), u_1(x_1), \ldots u_{K-1}(x_{K-1})$$



same action for all Player-1 nodes

with LQR approach we found optimal policy to be linear in state x

$$u_k = \Gamma_k x_k$$

$$\Gamma_k \in \mathbb{R}^{m \times n}$$

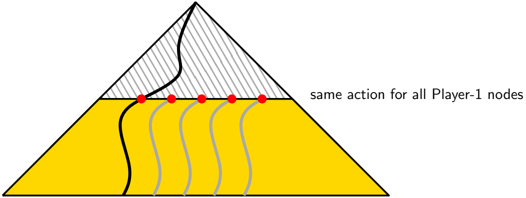
Open-loop NE strategy

By solving the optimization **in one shot** (no DP), we get

$$u_0, u_1, \ldots, u_{K-1} \in \mathbb{R}^m$$

instead of a state feedback

$$u_0(x), u_1(x), \ldots u_{K-1}(x)$$



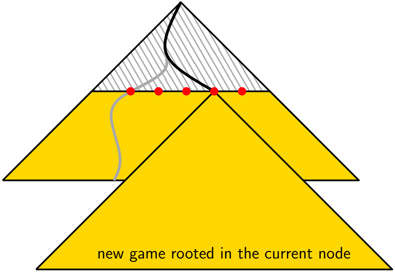
This is an **open-loop** (= non-subgame-perfect) NE strategy!
What to do if we get to a node of the tree which does not belong to the NE trajectory?

Receding horizon

Receding horizon

Re-calculate a NE strategy from the new starting point (i.e., for the subtree rooted in the current node).

For example: Model Predictive Control (MPC)



- Computationally not as efficient as explicit **backward induction**
- Optimization has to be solved **in real time**
- Still **better than solving the entire tree**

Two-player LQR

Two controllers, with their own objective.

Climate control

Player 1 goal: low humidity (byproduct: high temperature)
Player 2 goal: low temperature

Multi-robot control

Both players want to create a formation that achieves optimal coverage of an area.

Platoon of trucks

Fuel saving + safety objectives for all trucks.
Competitive (different companies) or collaborative (1-player LQR)

Two-player LQR

State update equation for $k = 0, \ldots, K-1$

$$x_{k+1} = Ax_k + B_1 u_k + B_2 v_k$$

Non-zero sum game setup

$$J_1(u_1, \ldots, u_{K-1}, v_1, \ldots, v_{K-1}) = \sum_{k=0}^{K-1} \left(x_k^T Q_1 x_k + u_k^T R_1 u_k \right) + x_K^T S_1 x_K$$

$$J_2(u_1, \ldots, u_{K-1}, v_1, \ldots, v_{K-1}) = \sum_{k=0}^{K-1} \left(x_k^T Q_2 x_k + v_k^T R_2 v_k \right) + x_K^T S_2 x_K$$

We try to apply backward induction to find a sub-game perfect NE.

Two-player LQR

cost-to-go for each player

Let us define the value of the game at state x in stage k as

player 1 →

$$V_{1k}(x) = \min_{u_k,\ldots,u_{K-1}} \sum_{s=k}^{K-1} \left(x_s^T Q_1 x_s + u_s^T R_1 u_s \right) + x_K^T S_1 x_K$$

$$V_{2k}(x) = \min_{u_k,\ldots,u_{K-1}} \sum_{s=k}^{K-1} \left(x_s^T Q_2 x_s + v_s^T R_2 v_s \right) + x_K^T S_2 x_K$$

player 2 →

Two-player LQR

Let us define the value of the game at state x in stage k as

$$V_{1k}(x) = \min_{u_k,\ldots,u_{K-1}} \sum_{s=k}^{K-1} \left(x_s^T Q_1 x_s + u_s^T R_1 u_s \right) + x_K^T S_1 x_K$$

$$V_{2k}(x) = \min_{u_k,\ldots,u_{K-1}} \sum_{s=k}^{K-1} \left(x_s^T Q_2 x_s + v_s^T R_2 v_s \right) + x_K^T S_2 x_K$$

STAGE K: For $k = K$ (last stage) we have the quadratic forms

$$V_{1K}(x) = x^T S_1 x$$
$$V_{2K}(x) = x^T S_2 x$$

Two-player LQR

As before, we conjecture that the value function is quadratic for all k.

$$V_{1k}(x) = x^T P_{1k} x$$
$$V_{2k}(x) = x^T P_{2k} x$$

If this is true, then it is enough to find the Nash equilibrium of the **static** game at a generic stage k, and iterate backwards.

Two-player LQR

As before, we conjecture that the value function is quadratic for all k.

$$V_{1k}(x) = x^T P_{1k} x$$
$$V_{2k}(x) = x^T P_{2k} x$$

If this is true, then it is enough to find the Nash equilibrium of the **static** game at a generic stage k, and iterate backwards.

Stage k **game:** two coupled problems

$$\begin{aligned}
J_1 &= x_k^T Q_1 x_k + u_k^T R_1 u_k + V_{1,k+1}(x_{k+1}) \\
&= x_k^T Q_1 x_k + u_k^T R_1 u_k + V_{1,k+1}(A x_k + B_1 u_k + B_2 v_k) \\
J_2 &= x_k^T Q_2 x_k + v_k^T R_2 v_k + V_{2,k+1}(x_{k+1}) \\
&= x_k^T Q_2 x_k + v_k^T R_2 v_k + V_{2,k+1}(A x_k + B_1 u_k + B_2 v_k)
\end{aligned}$$

Two-player LQR

$$(Ax + B_1 u_k + B_2 v_k)^T P_{k+1} (Ax + B_1 u_k + B_2 v_k)$$

$$J_1 = x_k^T Q_1 x_k + u_k^T R_1 u_k + V_{1,k+1}(Ax_k + B_1 u_k + B_2 v_k)$$

$$J_2 = x_k^T Q_2 x_k + v_k^T R_2 v_k + V_{2,k+1}(Ax_k + B_1 u_k + B_2 v_k)$$

For each player, we construct the **best response strategy**.

Best response strategies

$$\hat{u}_k(v_k) = -\left(R_1 + B_1^T P_{1,k+1} B_1\right)^{-1} B_1^T P_{1,k+1}\left(Ax_k + B_2 v_k\right)$$

$$\hat{v}_k(u_k) = -\left(R_2 + B_2^T P_{2,k+1} B_2\right)^{-1} B_2^T P_{2,k+1}\left(Ax_k + B_1 u_k\right)$$

$$\left.\begin{aligned} \frac{\partial J_1}{\partial u_1} &= 0 \\ \frac{\partial J_2}{\partial u_2} &= 0 \end{aligned}\right\} \rightarrow A$$

$$M \begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} c_k^1 \\ c_k^2 \end{bmatrix}$$

One-stage Nash equilibrium

Given the two best responses, the NE strategy (u_k^*, v_k^*) satisfies

$$\hat{u}_k(v_k^*) = u_k^*$$
$$\hat{v}_k(u_k^*) = v_k^*$$

that is

$$u_k^* = -\left(R_1 + B_1^T P_{1,k+1} B_1\right)^{-1} B_1^T P_{1,k+1} \left(A x_k + B_2 v_k^*\right)$$
$$v_k^* = -\left(R_2 + B_2^T P_{2,k+1} B_2\right)^{-1} B_2^T P_{2,k+1} \left(A x_k + B_1 u_k^*\right)$$

One-stage Nash equilibrium

Given the two best responses, the NE strategy (u_k^*, v_k^*) satisfies

$$\hat{u}_k(v_k^*) = u_k^*$$
$$\hat{v}_k(u_k^*) = v_k^*$$

that is

$$u_k^* = -\left(R_1 + B_1^T P_{1,k+1} B_1\right)^{-1} B_1^T P_{1,k+1} \left(A x_k + B_2 v_k^*\right)$$
$$v_k^* = -\left(R_2 + B_2^T P_{2,k+1} B_2\right)^{-1} B_2^T P_{2,k+1} \left(A x_k + B_1 u_k^*\right)$$

NE feedback

The NE strategy is a linear feedback of the state x_k

$$u_k^* = H_{1k} x_k, \qquad v_k^* = H_{2k} x_k$$

that satisfies the best-response coupled conditions above.

Value of the single-stage game

Given the NE strategy

$$u_k^* = H_{1k} x_k, \qquad v_k^* = H_{2k} x_k.$$

we have the following **quadratic** value of the single-stage game.

*k*th stage subgame value

$$V_{1k} = x_k^T P_{1k} x_k, \qquad V_{2k} = x_k^T P_{2k} x_k$$

where

$$\begin{aligned} P_{1k} &= Q_1 + H_{1k}^T R_1 H_{1k} + \\ &\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{1,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \\ P_{2k} &= Q_2 + H_{2k}^T R_2 H_{2k} + \\ &\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{2,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \end{aligned}$$

From single stage to multi stage

- The value function is **quadratic**: $V_{1k} = x_k^T P_{1k} x_k$, $V_{2k} = x_k^T P_{2k} x_k$
  - we can use iterate backward on the stages

From single stage to multi stage

- The value function is **quadratic**: $V_{1k} = x_k^T P_{1k} x_k$, $V_{2k} = x_k^T P_{2k} x_k$
  - we can use iterate backward on the stages
- Iterative computation via **two coupled Riccati equations**
  - how are they coupled?

$$
\begin{aligned}
P_{1k} &= Q_1 + H_{1k}^T R_1 H_{1k} + \\
&\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{1,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \\
P_{2k} &= Q_2 + H_{2k}^T R_2 H_{2k} + \\
&\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{2,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \\
P_{1K} &= S_1 \\
P_{2K} &= S_2
\end{aligned}
$$

From single stage to multi stage

- The value function is **quadratic**: $V_{1k} = x_k^T P_{1k} x_k$, $V_{2k} = x_k^T P_{2k} x_k$
  - we can use iterate backward on the stages
- Iterative computation via **two coupled Riccati equations**
  - how are they coupled?

$$
\begin{aligned}
P_{1k} &= Q_1 + H_{1k}^T R_1 H_{1k} + \\
&\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{1,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \\
P_{2k} &= Q_2 + H_{2k}^T R_2 H_{2k} + \\
&\qquad + (A + B_1 H_{1k} + B_2 H_{2k})^T P_{2,k+1} (A + B_1 H_{1k} + B_2 H_{2k}) \\
P_{1K} &= S_1 \\
P_{2K} &= S_2
\end{aligned}
$$

- The NE control u_k^*, v_k^* can be computed from P_{ik}
  - System of linear equations (or explicit form)

Summary and further reading

- We can find subgame perfect equilibria of dynamic feedback games using backward induction

- Computation:
  - at each stage, need to find a Nash equilibrium for every state $x \in X$: generally difficult
  - tractable for certain classes of games, such as linear quadratic games

- in linear quadratic games, there exists a linear state-feedback policy Nash equilibrium

- Reading: Chapter 17 of Hespanha