ECE 586 – Markov Decision Processes and Reinforcement Learning

Hidden Markov Models, Partially Observable Markov Decision Processes and Linear Quadratic Regulation

Instructor: Dimitrios Katselis *Acknowledgment: R. Srikant's Notes and Discussions*

1 Hidden Markov Models

Before giving some basic material on Hidden Markov Models (HMM), we look into Markov models. Markov models describe the evolution of randomly varying systems based on an underlying Markov assumption, which establishes that future system states given the current state are independent of any past events. Depending on whether the state is fully observable or not and if the system is controlled or not the Markov models are:

- **Autonomous Systems**: Markov chains (if the state is fully observed) and HMMs (if the state is partially observed).
- **Controlled Systems**: Markov Decision Processes (if the state is fully observed) and Partially Observable MDPs or POMDPs (if the system state is partially observable).

HMMs: In full generality, HMMs correspond to partially observable Markov processes of the form $(X, Y) = \{(x_n, y_n)\}_{n\in\mathbb{Z}_+}$ for which only one of the two components is (fully) observable[1]. The main purpose of these models is to formalize statistical inference setups for the hidden (unobserved) component of the given Markov process based on fully observing the remaining component. The hidden (unobserved) component X is called **signal** (or **state** or **plant**) and the statistical analysis and inference rely on the observed realization of Y, which corresponds to the observable component of the Markov process (X, Y). HMMs correspond to a paradigm of **dynamic Bayesian networks (DBN)**. DBNs relate temporally ordered random variables, i.e., they describe their statistical dependencies encoding a natural inference mechanism. They can be used to represent system dynamics in steady-state. A DBN consists of a series of *time slices* that represent the state of all the variables at a certain time t. For each temporal slice, a dependency structure between the variables at that time is defined, called the *base network*. Additionally, there are edges between variables from different slices, with their directions following the direction of time, defining the *transition network*. DBNs can model complex multivariate time series, i.e., they can encode the relationships between multiple time series in the same model.

HMMs in control theory: Let (X, Y) be a random process on a probability space $(\Omega, \mathcal{F}, P)$ taking values in the measurable state space $(E_1 \times E_2, \mathcal{E}_1 \times \mathcal{E}_2)$ ($E_1 \times E_2$ is the *state space*, E_1 is the *signal state space* and E_2 is the *observation state space*) and generated by the stochastic recursion:

$$\begin{aligned} x_{n+1} &= g(x_n, y_n, \eta_{n+1}) \\ y_{n+1} &= h(x_n, y_n, \eta_{n+1}), \quad n \geq 0 \end{aligned} \tag{1}$$

[1]Although usually in statistical literature uppercase letters are used to represent random variables, here we use lowercase letters to be in accordance with our notation in previous files.

with initial condition (x_0, y_0) independent of $\{\eta_n\}_{n\in\mathbb{Z}_+}$ and g, h measurable functions. The sequence $\{\eta_n\}_{n\in\mathbb{Z}_+}$ is assumed i.i.d. and is called **innovation process** or **driving noise** of the system. Moreover, the pair of equations given by (1) is called **state space representation** of (X, Y) and often defines the time evolution of a physical system. It can be easily seen that the process (X, Y) generated by (1) satisfies the Markov property.

Note: Neither X nor Y have to be Markov processes individually.

The most typical scenario of (1) corresponds to the case where the dynamics of X are not affected by Y, i.e,

$$\begin{aligned} x_{n+1} &= \tilde{g}(x_n, \xi_{n+1}) \\ y_{n+1} &= \tilde{h}(x_n) + \epsilon_{n+1} \quad \text{or more often} \quad y_n = \tilde{h}(x_n) + \epsilon_n, \end{aligned} \tag{2}$$

where $\{\xi_n\}, \{\epsilon_n\}$ are independent i.i.d. sequences (consider $\eta_n = [\xi_n, \epsilon_n]$ in this setup and appropriate definitions of g, h in (1) in terms of $\tilde{g}$ and $\tilde{h}$, respectively). The most notable example of such systems is the **linear Gaussian system**, where $\tilde{g}(x_n, \xi_{n+1}) = Ax_n + \xi_{n+1}$ and $\tilde{h}(x_n) = Bx_n$ (A, B are either scalars or matrices) with $\{\xi_n\}, \{\epsilon_n\}$ being independent i.i.d. Gaussian sequences, independent of (x_0, y_0). Moreover, (x_0, y_0) is itself a Gaussian vector. We recall here that the Gaussian distribution is stable under affine transformations and therefore, (X, Y) is a Gaussian process in this setup.

Finally, in the most frequently encountered scenario (2), X is Markov on its own but not necessarily Y.

HMMs in Information Theory and Communications: An HMM (X, Y) corresponds to a basic abstraction of a communication channel by assuming a Markovian source represented by X and a discrete (memoryless) channel whose output process is represented by Y. The alphabet of X is assumed finite, while Y can either have a finite alphabet as well or it can be continuous-valued. The underlying statistical inference setup in this scenario consists of two stages: the first stage is called *system identification* and corresponds to inferring underlying system parameters such as the transition matrix of the Markov source and some channel related parameters which characterize the emitting probabilities of the channel. The second stage corresponds to using the inference outcome from the previous stage and the observation process Y to decide in the best possible way the transmitted symbols in X. More precisely, after a system identification phase in which the source transition probabilities and the statistical description of the channel are identified, the goal is typically to infer the transmitted symbol x_n given the observation path in Y up to time n.

HMMs in Statistical Signal Processing: In signal processing literature, signals are usually assumed to be stationary times series with rational power spectral densities. The typical model is the following autoregressive-moving–average (ARMA) model, denoted by ARMA(N, M):

$$x_n = \sum_{i=1}^{N} a_i x_{n-i} + \sum_{j=0}^{M} b_j \eta_{n-j}, \quad \forall n \geq \max\{M, N\}$$

with $\{\eta_n\}$ being an i.i.d. sequence and the coefficients being deterministic constants ($b_0 = 1$). Suppose that x_n is observed in white noise via $y_n = \tilde{h}(x_n) + \epsilon_n$. In signal processing parlance, y_n is a measurement of x_n, where x_n is distorted by a nonlinear memoryless sensor or channel $\tilde{h}(\cdot)$ and is corrupted by additive noise. Let X' be the random process with generic term

$$x'_n = [x_n, x_{n-1}, \ldots, x_{n-N+1}, \eta_n, \eta_{n-1}, \ldots, \eta_{n-M+1}]^T.$$

Then, X' is a Markov process and the pair (X', Y) forms an HMM.

Some notation: Let (X, Y) be an HMM. Suppose that X is a finite state Markov chain with state space E_1. We denote the corresponding transition matrix by $P = [P_{ij}]$ and the initial distribution by π_0. The probabilistic structure of Y is described either by an observation density function $O_{xy} = p(y_n = y|x_n = x)$ (continuous-valued observations) or by an observation mass function $O_{xy} = P(y_n = y|x_n = x)$ (discrete-valued observations). Often in applications, Y takes values in a finite observation space[2] E_2.

1.1 Filtering

Given an HMM (X, Y) defined on the measurable state space $(E_1 \times E_2, \mathcal{E}_1 \times \mathcal{E}_2)$, filtering is the problem of computing the conditional distribution

$$\pi_n(\mathcal{A}) = P\left(x_n \in \mathcal{A}|y_1, y_2, \ldots, y_n\right), \quad \mathcal{A} \in \mathcal{E}_1.$$

This conditional distribution is called *filter*. Computing the filter solves the problems of estimating the hidden process optimally in the mean square or maximum a posteriori senses.

Focusing on finite state HMMs specified by the triplet (π_0, P, O), the signal process X is assumed to be a Markov chain with state space $E_1 = \{1, 2, \ldots, |E_1|\}$, transition matrix $P = [P_{ij}]$ with $P_{ij} = P(x_{n+1} = j|x_n = i)$ and initial measure π_0. As mentioned earlier, the observation likelihoods are denoted by O_{xy}. Since E_1 is finite with cardinality $|E_1|$, π_n is the pmf

$$\pi_n(i) = P(x_n = i|y_1, y_2, \ldots, y_n), \quad i \in E_1.$$

Let $\pi_n = [\pi_n(1), \ldots, \pi_n(|E_1|)]^T$ and

$$O_{y_n} = \text{diag}(O_{1y_n}, O_{2y_n}, \ldots, O_{|E_1|y_n}),$$

where $\text{diag}(\cdot)$ denotes a diagonal matrix with main diagonal entries the corresponding arguments. Then, filters at successive time instants are computed by the recursion

$$\pi_{n+1} = \frac{O_{y_{n+1}} P^T \pi_n}{1^T O_{y_{n+1}} P^T \pi_n}, \tag{3}$$

which is initialized by π_0. Here, 1 denotes the $|E_1| \times 1$ all-ones vector.

2 POMDPs

POMDPs are controlled HMMs. An appropriate version of recursion (3) produces the filter sequence $\{\pi_n\}$. The filtered distribution π_n is called **belief state** in the context of POMDPs and it is used by the POMDP controller to choose the next action. Since π_n contains real (nonnegative) valued entries for any n, a POMDP is equivalent to a continuous-state MDP with underlying states the beliefs.

Focusing on the finite horizon scenario and assuming finite state and action spaces, the key ingredients of a POMDP model are:

- x_n is the state of the underlying Markov chain at time n taking values in $E_1 = \{1, 2, \ldots, |E_1|\}$,

[2]Sometimes in the literature, O_{xy} for finite-valued observations are called *symbol probabilities*.

- u_n is the control or action or decision variable at time n taking values in $\mathcal{U} = \{1, 2, \ldots, U\}$,
- y_n is the (noisy) observation of x_n taking values in the observation space E_2, which can be finite with cardinality $|E_2|$ or continuous-valued,
- $P(u)$ is the $|E_1| \times |E_1|$ transition matrix of the underlying Markov chain X, when the control action takes the value u, i.e., $P_{ij}(u) = P(x_{n+1} = j | x_n = i, u_n = u), \forall (i, j) \in E_1 \times E_1$,
- $O(u)$ is the observation likelihood matrix with

$$O_{xy}(u) = P(y_n = y | x_n = x, u_{n-1} = u) \quad \text{or} \quad O_{xy}(u) = p(y_n = y | x_n = x, u_{n-1} = u),$$

- $c(x_n, u_n)$ is the incurred cost for state x_n and action u_n,
- $c_N(x_N)$ is the terminal cost.

Note: The observation likelihood matrix is action-dependent. This corresponds to a **controlled sensing** feature. Other terms for this feature are *sensor scheduling*, *measurement control* or *active sensing*. In signal processing parlance, this feature corresponds to the so-called *sensor-adaptive signal processing* with applications in adaptive radar (allocate more resources to more critical targets), cognitive radio (how to sense the radio spectrum for available channels), etc. In plain words, the controller can adjust the accuracy of the measurements by choosing less noisy sensors (or sensing modes) at particular time instants based on u_n, but with higher measurement cost though. Clearly, this controlled sensing feature may be absent from a POMDP model with the observation likelihoods being action-independent. Finally, although we call the measurement control a "controlled sensing" feature of a POMDP, the reader should be aware of the fact that the term *controlled sensing* in the literature *usually* corresponds to problems where the action only affects the observation distribution and not the dynamics of the underlying Markov chain.

As in the case of finite horizon MDPs, let $\mu = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ denote a policy. Then, the **goal** is to solve the problem:

$$\min_{\mu} J_N^{\mu}(\pi_0), \tag{4}$$

where

$$J_N^{\mu}(\pi_0) = E\left[c_N(x_N) + \sum_{k=0}^{N-1} c(x_k, u_k) \middle| \pi_0 \right]$$

and π_0 is the initial measure of the underlying Markov chain. The optimal policy corresponds to $\mu^* = \arg\min_{\mu} J_N^{\mu}(\pi_0)$ for any initial distribution π_0 and may not be unique. Also, as we describe in the next section, a POMDP is equivalent to a continuous-state MDP and therefore, it suffices to consider only deterministic policies μ.

2.1 Belief State Formulation

We recall that in the MDP framework, $u_n = \mu_n^*(x_n)$. In the POMDP context,

$$u_n = \mu_n^*(\mathcal{F}_n), \qquad \mathcal{F}_n = \{\pi_0, u_0, y_1, \ldots, u_{n-1}, y_n\} \tag{5}$$

and the size of $\mathcal{F}_n$ increases with n. To implement the controller, it is desirable to replace $\mathcal{F}_n$ with a sufficient statistic that does not grow in size with time n. Such a sufficient statistic for $\mathcal{F}_n$ is π_n. Let π_n be the $|E_1| \times 1$ vector with generic element the posterior probability

$$\pi_n(i) = P(x_n = i|\mathcal{F}_n).$$

The vector π_n is called **belief state** or **information state** at time n and it is updated over time via (3) as follows:

$$\pi_{n+1} = \frac{O_{y_{n+1}}(u_n)P^T(u_n)\pi_n}{1^T O_{y_{n+1}}(u_n)P^T(u_n)\pi_n}, \quad O_{y_{n+1}}(u_n) = \text{diag}(O_{1y_{n+1}}(u_n), \ldots, O_{|E_1|y_{n+1}}(u_n)). \tag{6}$$

Additionally, define the cost vectors

$$c(u) = [c(1,u), c(2,u), \ldots, c(|E_1|,u)]^T \text{ and } c_N = [c_N(1), c_N(2), \ldots, c_N(|E_1|)]^T.$$

The following key result can be proved:

Theorem 1: For the finite state-action finite horizon POMDP considered in the beginning of Section 2, assuming also finite observations:

1. The optimal cost $J_N^{\mu^*}(\pi_0)$ for any given π_0 is achieved by deterministic policies $\mu^* = \{\mu_0^*, \ldots, \mu_{N-1}^*\}$ with $u_n = \mu_n^*(\pi_n)$ for any n.

2. μ^* can be recovered via the following Bellman's backward recursion: For any π_N, set $J_N(\pi_N) = c_N^T \pi_N$. For $k = N-1, \ldots, 0$ compute:

$$J_k(\pi) = \min_u \left\{ c(u)^T\pi + \sum_{y \in E_2} J_{k+1}\left(\frac{O_y(u)P^T(u)\pi}{1^T O_y(u)P^T(u)\pi}\right) 1^T O_y(u)P^T(u)\pi \right\},$$

$$\mu_k^*(\pi) = \operatorname*{argmin}_u \left\{ c(u)^T\pi + \sum_{y \in E_2} J_{k+1}\left(\frac{O_y(u)P^T(u)\pi}{1^T O_y(u)P^T(u)\pi}\right) 1^T O_y(u)P^T(u)\pi \right\}.$$

   Then, for any initial distribution π_0, $J_N^{\mu^*}(\pi_0)$ corresponds to $J_0(\pi_0)$ of the above recursion and $\mu^* = \{\mu_0^*, \ldots, \mu_{N-1}^*\}$ is an optimal policy for the problem.

Remarks:

1. The above theorem shows that there is a DP formulation to solve POMDPs based on belief states.

2. The DP recursion is intractable in practice, since it has to be evaluated for any π_N, π in the probability simplex.

3. **Complexity of MDPs and POMDPs**: MDPs in all their variants (finite horizon, infinite horizon discounted and infinite horizon average cost) are solvable in polynomial time by Dynamic Programming (finite horizon problems), linear programming, or successive approximation techniques (infinite horizon). It has been shown that they are complete for $\mathcal{P}$

($\mathcal{P}$ is the class of all decision problems that can be solved by a deterministic (single-tape or multi-tape) polynomial-time Turing machine), and therefore most likely unsolvable via parallelism. On the other hand, the deterministic versions of these three variants can be solved in parallel. Moreover, POMDPs (in the complexity discussion here, it may be helpful to think of POMDPs as "degraded MDPs", i.e., as MDPs with partially observed states) have been shown to be PSPACE-complete (PSPACE is the class of all decision problems solvable by a Turing machine in polynomial space with respect to the input size), and hence, even less likely to be solved in polynomial time than NP-complete problems. Finally, deciding whether the optimal policy in an unobserved MDP (open-loop control problem) has expected cost (undiscounted, over a finite horizon) equal to zero is an NP-complete problem. In other words, MDPs with no observations are NP-complete.

4. The factors $1^T O_y(u) P^T(u) \pi$ replace the transition probabilities in the DP algorithm for MDPs.

5. In control theory, such setups are often called *problems with imperfect state information.*

Despite, the mentioned difficulties, the following extraordinary result is of interest:

Theorem 2: Consider the finite horizon POMDP discussed so far (finite state, finite action and finite observation spaces). Then:

1. $J_k(\pi)$ is piecewise linear and concave with respect to π, i.e.,

$$J_k(\pi) = \min_{g \in \mathcal{G}_k} g^T \pi,$$

where $\mathcal{G}_k$ is a finite set of vectors at time k and $\mathcal{G}_N = \{c_N\}$. Therefore, $J_k(\pi)$ has a finite-dimensional characterization.

2. At time k, the simplex can be partitioned into at most $|\mathcal{G}_k|$ convex polytopes, each polytope being $\mathcal{P}_r = \{\pi : J_k(\pi) = g_r^T \pi\}$. Then, the optimal policy consists of a single action per polytope, i.e., $\forall \pi \in \mathcal{P}_r$

$$\mu_k^*(\pi) = u_r^*.$$

Therefore, the optimal policy has a finite-dimensional characterization as well.

We finally note that algorithms for solving such POMDPs rely on the provided finite-dimensional characterizations.

2.2 Brief Reference to Algorithms for Solving POMDPs

We will not proceed any further with the theory and algorithms for POMDPs. The interested reader is referred to the relevant literature. We note here that POMDPs are heavily motivated by applications in path planning, human–robot interaction and in controlled sensing where one wishes to penalize uncertainty of the estimates.

For finite horizon POMDPs, optimal algorithms are based on exact value iteration and are computationally intractable in general. Known schemes are the Incremental Pruning algorithm, Monahan's Algorithm and Witness algorithm.

Suboptimal algorithms based on value iteration include Lovejoy's algorithm and point-based value iteration algorithms such as SARSOP. Moreover, the belief compression and the open loop feedback control algorithms are two additional examples.

Finally, there are algorithms exploiting structural results for POMDPs and heuristics based on MDP solutions. Policy gradient approaches for POMDPs exist, as well as discrete stochastic optimization-based search algorithms for estimating the best policy from a finite set of possible policies.

3 Linear Quadratic Regulation

The linear quadratic regulator (LQR) problem is one of the most fundamental control problems. The LQR is an important part of the solution to the Linear Quadratic Gaussian (LQG) problem.

3.1 Deterministic Setup

Focusing on the discrete-time deterministic version of the problem, consider the time-homogeneous discrete-time system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ x_k &\in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}, u_k \in \mathbb{R}^m, B \in \mathbb{R}^{n\times m} \end{aligned}$$

and assume that the system state is fully observable[3]. Let the cost function be:

$$J = c_N(x_N) + \sum_{k=0}^{N-1} c(x_k, u_k),$$

where the one-stage and terminal costs in matrix-vector form are:

$$\begin{aligned} c(x,u) &= \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \\ c_N(x) &= x^T W_N x. \end{aligned}$$

All the involved quadratic forms are assumed positive semi-definite ($\succeq 0$) and $Q \succ 0$, where $\succ 0$ denotes positive definiteness. The above is a model for regulation of (x, u) to the point $(0, 0)$. The assumptions on the involved matrices ensure that the cost function is convex and has a unique minimum.

Value Function: The value function for this problem is

$$J_k(x) = x^T W_k x, \quad k \leq N.$$

Optimal Control: The optimal control for this problem is linear in the system state:

$$u_k^* = K_k^* x_k, \quad k < N$$

[3]Controllability or stabilizability, etc. will not be discussed here.

with

$$K_k^* = -\left(Q + B^T W_{k+1} B\right)^{-1} \left(S + B^T W_{k+1} A\right), \quad k < N.$$

Riccati Recursion: W_k satisfies the Riccati (backward) recursion:

$$W_k = R + A^T W_{k+1} A - (S^T + A^T W_{k+1} B)(Q + B^T W_{k+1} B)^{-1}(S + B^T W_{k+1} A), \quad k < N.$$

Remark: Observe that the optimal control is of the form $u_k^* = \mu_k^*(x_k)$.

Infinite-Horizon LQR: Suppose that $N = \infty$ (clearly, no terminal cost exists in this setup). Then, one may run the Riccati recursion up to convergence, in which case

$$W_\infty = R + A^T W_\infty A - (S^T + A^T W_\infty B)(Q + B^T W_\infty B)^{-1}(S + B^T W_\infty A).$$

This equation is called *algebraic Riccati equation* and W_∞ is the unique positive definite solution to this equation. The optimal control in this case is

$$u_k = K_\infty^* x_k = -\left(Q + B^T W_\infty B\right)^{-1} \left(S + B^T W_\infty A\right) x_k.$$

Remark: Observe that the optimal control is of the form $u_k^* = \mu^*(x_k)$.

3.2 Stochastic Fully Observed Setup

Consider white noise disturbances in the system state:

$$x_{k+1} = Ax_k + Bu_k + \xi_k.$$

$\{\xi_k\}$ is a white noise sequence such that $E[\xi_k] = 0, \forall k$, $E[\xi_k \xi_k^T] = \Xi$ and $E[\xi_s \xi_k^T] = 0, s \neq k$. The system state is fully observable at every time instant.

Optimality equation: The optimality equation for this problem is:

$$J_k(x) = \inf_u \left\{c(x,u) + E_\xi[J_{k+1}(Ax + Bu + \xi)]\right\}, \quad k < N,$$

with $J_N(x) = x^T W_N x$ as before.

To solve the optimality equation, the solution $J_k(x) = x^T W_k x + \alpha_k$ is tried, where W_k follows the Riccati recursion provided earlier. It turns out that $\alpha_k = \sum_{r=k+1}^N \mathrm{tr}(\Xi W_r)$.

Optimal Control: The optimal control for this problem corresponds to linear state feedback:

$$u_k^* = K_k^* x_k, \quad k < N$$

with K_k^* as in the noiseless scenario.

Remark: Observe that the optimal control is of the form $u_k^* = \mu_k^*(x_k)$.

3.3 Stochastic Setup with Imperfect State Information

Consider now the state-space model:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \xi_k, \\ y_k &= Cx_{k-1} + \epsilon_{k-1}, \\ y_k &\in \mathbb{R}^p, C \in \mathbb{R}^{p\times n}, \epsilon_k \in \mathbb{R}^p, p < n \text{ (typically)} \end{aligned}$$

with the remaining model parameters as before. The system noise is again considered white with

$$E\left(\begin{bmatrix} \xi_k \\ \epsilon_k \end{bmatrix}\right) = 0, \quad \forall k, \qquad \text{cov}\left(\begin{bmatrix} \xi_k \\ \epsilon_k \end{bmatrix}\right) = \begin{bmatrix} \Xi & L \\ L^T & M \end{bmatrix}, \quad \forall k, \qquad E\left(\begin{bmatrix} \xi_k \\ \epsilon_k \end{bmatrix}\begin{bmatrix} \xi_s \\ \epsilon_s \end{bmatrix}^T\right) = 0, \quad k \neq s.$$

Also, $x_0 \sim \pi_0$ for a given initial distribution π_0.

LQG model: $x_0 \sim \mathcal{N}(\mu_0, V_0)$, where μ_0, V_0 are known and the system noise is Gaussian.

Value Function: Let $\mathcal{F}_n$ be as in (5). The optimal value function for this problem is:

$$J_k(\mathcal{F}_k) = E[x_k^T W_k x_k|\mathcal{F}_k] + \sum_{r=k}^{N-1} E[\Delta_r \tilde{W}_r \Delta_r|\mathcal{F}_k] + \alpha_k, \quad J_N(\mathcal{F}_N) = E[x_N^T W_N x_N|\mathcal{F}_N],$$

with all the involved variables as in the previous scenarios, $\Delta_k = x_k - E[x_k|\mathcal{F}_k]$ and $\tilde{W}_k = R + A^T W_{k+1} A - W_k$ with $\tilde{W}_k \succeq 0$.

Optimal Control: In this case,

$$u_k^* = K_k^* \hat{x}_k,$$

where K_k^* is as before and $\hat{x}_k = E[x_k|\mathcal{F}_k]$.

Remark: Observe that the optimal control is of the form $u_k^* = \mu_k^*(\pi_k)$.

Certainty Equivalence: The optimal control u_k^* is exactly the same as it would be if all unknowns were known with values equal to their conditional means based on the observations up to time k. This fact is named "certainty equivalence".

Separation Principle: It turns out that the distribution of the estimation error Δ_k does not depend on $\{u_0, u_1, \ldots, u_{k-1}\}$. Therefore, the problems of optimal (state) estimation and optimal control can be decoupled. This decoupling is called "separation principle" of linear quadratic control design.

LQG problem: Under Gaussianity, optimal state estimates $\hat{x}_k$ are obtained iteratively via Kalman filtering.

Kalman filtering: The Kalman filter computes recursively the *filter*, which is Gaussian, i.e., the posterior state distribution. The optimal system state estimate at every time instant corresponds to the mean value of this distribution (conditional or posterior mean).

Double Separation Principle: Suppose that except state control u_k, we also have observation control $\tilde{u}_k$. Then, it can be shown that the determination of the optimal state control policy can be separated from the determination of the optimal measurement control policy.

Sample Complexity of LQR with unknown dynamics: Recently, there have been approaches to address the sample complexity of the LQR problem in the case of unknown dynamics. A model is first estimated based on few experimental trials and the corresponding error of the estimated model from the true model is also inferred. Then, a controller is designed using both the model and uncertainty estimates. The system identification setup relies on n experiments of the following form: Starting at some given initial state x_0, the dynamics are evolved for a time horizon N using any control sequence $\{u_0, u_1, \ldots, u_{N-1}\}$. The resulting states $\{x_1, x_2, \ldots, x_N\}$ are assumed perfectly observable. Finite sample uncertainty bounds for a pre-specified level of confidence exist as well as estimates on the achievable LQR cost for a pre-specified confidence level as functions of the model dimensionality and the number of independent trials n (samples) used to infer the model and the corresponding uncertainty.