Optimal Control Theory

Benjamin Dubois-Taine

July 22, 2020

The University of British Columbia
Today we will cover

- Discrete time control and the Bellman equations
Today we will cover

- Discrete time control and the Bellman equations
- Continuous time control and the Hamilton-Jacobi-Bellman equations
Today we will cover

- Discrete time control and the Bellman equations
- Continuous time control and the Hamilton-Jacobi-Bellman equations
- An important special case: Linear-Quadratic Gaussian and Linear-Quadratic Regulator problems

This content is taken from [1, Chapter 12].
Introduction

Today we will cover

- Discrete time control and the Bellman equations
- Continuous time control and the Hamilton-Jacobi-Bellman equations
- An important special case: Linear-Quadratic Gaussian and Linear-Quadratic Regulator problems
- Pontryagin’s maximum principle

This content is taken from [1, Chapter 12].
Today we will cover

- Discrete time control and the Bellman equations
- Continuous time control and the Hamilton-Jacobi-Bellman equations
- An important special case: Linear-Quadratic Gaussian and Linear-Quadratic Regulator problems
- Pontryagin’s maximum principle
- (time allowing) Optimal Estimation and Kalman filter

This content is taken from [1, Chapter 12].
Today we will cover

- Discrete time control and the Bellman equations
- Continuous time control and the Hamilton-Jacobi-Bellman equations
- An important special case: Linear-Quadratic Gaussian and Linear-Quadratic Regulator problems
- Pontryagin’s maximum principle
- (time allowing) Optimal Estimation and Kalman filter

This content is taken from [1, Chapter 12].
Define

- $x \in \mathcal{X}$ the state of the agent’s environment.
- $u \in \mathcal{U}(x)$ the action chosen at state $x$.
- $\text{next}(x, u) \in \mathcal{X}$ the resulting state from applying action $u$ in state $x$.
- $\text{cost}(x, u) \geq 0$ the cost of applying $u$ in state $x$.
Example: plane tickets

- $\mathcal{X} = \text{set of cities}$
- $\mathcal{U}(x) = \text{flights available from city } x$
- $\text{next}(x, u) = \text{the city where the flight lands}$
- $\text{cost}(x, u) = \text{price of the flight}$
Example: plane tickets

- $\mathcal{X} =$ set of cities
- $\mathcal{U}(x) =$ flights available from city $x$
- $next(x, u)$ the city where the flight lands
- $\text{cost}(x, u)$ price of the flight

**Goal:** find cheapest way to get to your destination
Define

- $x \in \mathcal{X}$ the state of the agent's environment.
- $u \in \mathcal{U}(x)$ the action chosen at state $x$.
- $\text{next}(x, u) \in \mathcal{X}$ the resulting state from applying action $u$ in state $x$.
- $\text{cost}(x, u) \geq 0$ the cost of applying $u$ in state $x$.

**Goal:** find action sequence $(u_0, \ldots, u_{n-1})$ minimizing the total cost

$$J(x, u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)$$

where $x_{k+1} = \text{next}(x_k, u_k)$, and $x_0$ and $x_n$ given.
Define

- \( x \in \mathcal{X} \) the state of the agent’s environment.
- \( u \in \mathcal{U}(x) \) the action chosen at state \( x \).
- \( \text{next}(x, u) \in \mathcal{X} \) the resulting state from applying action \( u \) in state \( x \).
- \( \text{cost}(x, u) \geq 0 \) the cost of applying \( u \) in state \( x \).

**Goal:** find action sequence \((u_0, \ldots, u_{n-1})\) minimizing the total cost

\[
J(x., u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)
\]

where \( x_{k+1} = \text{next}(x_k, u_k) \), and \( x_0 \) and \( x_n \) given.

- We can think of this as a graph where nodes are states, and actions are arrows connecting the nodes.
Goal: find action sequence \((u_0, \ldots, u_{n-1})\) minimizing the total cost

\[
J(x., u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)
\]
Discrete Control and the Bellman Equations

**Goal:** find action sequence \((u_0, \ldots, u_{n-1})\) minimizing the total cost

\[
J(x., u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)
\]

We need a **control law**, namely a mapping from states to actions.
Goal: find action sequence \((u_0, \ldots, u_{n-1})\) minimizing the total cost

\[
J(x., u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)
\]

We need a control law, namely a mapping from states to actions. Defining the optimal value function as

\[
v(x) = \min_{u \in \mathcal{U}(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \}
\]  

(1)

the associated optimal control law is

\[
\pi(x) = \arg \min_{u \in \mathcal{U}(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \}
\]

(2)
Goal: find action sequence \((u_0, \ldots, u_{n-1})\) minimizing the total cost

\[
J(x., u.) = \sum_{k=0}^{n-1} \text{cost}(x_k, u_k)
\]

We need a control law, namely a mapping from states to actions. Defining the optimal value function as

\[
v(x) = \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \}
\]

the associated optimal control law is

\[
\pi(x) = \arg \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \}
\]

Those are the Bellman equations.
We want to solve

\[ v(x) = \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \} \]

\[ \pi(x) = \arg \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \} \]

Let's go back to the graph analogy. Assume the graph is acyclic. Suppose we start at \( x_0 \) and want to reach \( x_f \).

- set \( v(x_f) = 0 \)
- once every successor of a state \( x \) has been visited, apply the formula for \( v \) to \( x \).
We want to solve

\[ v(x) = \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \} \]

\[ \pi(x) = \arg \min_{u \in U(x)} \{ \text{cost}(x, u) + v(\text{next}(x, u)) \} \]

Let’s go back to the graph analogy. Assume the graph is acyclic. Suppose we start at \( x_0 \) and want to reach \( x_f \).
Discrete Control and the Bellman Equations

We want to solve

$$v(x) = \min_{u \in \mathcal{U}(x)} \left\{ \text{cost}(x, u) + v(\text{next}(x, u)) \right\}$$

$$\pi(x) = \arg \min_{u \in \mathcal{U}(x)} \left\{ \text{cost}(x, u) + v(\text{next}(x, u)) \right\}$$

Let's go back to the graph analogy. Assume the graph is acyclic. Suppose we start at $x_0$ and want to reach $x_f$.

- set $v(x_f) = 0$
- once every successor of a state $x$ has been visited, apply the formula for $v$ to $x$. 
Discrete Control and the Bellman Equations

- For cyclic graphs, this approach will not work.
For cyclic graphs, this approach will not work.

The Bellman equations are still valid.
Discrete Control and the Bellman Equations

- For cyclic graphs, this approach will not work.
- The Bellman equations are still valid.
- Need to design iterative schemes: **Value Iteration** and **Policy Iteration**
Value Iteration proceeds as follows:

- start with some guess $v^{(0)}$ of the optimal value function.
Value Iteration proceeds as follows:

- start with some guess $v^{(0)}$ of the optimal value function.
- construct a sequence of guesses

$$v^{(i+1)}(x) = \min_{u \in U(x)} \left\{ \text{cost}(x, u) + v^{(i)}(\text{next}(x, u)) \right\}$$
Value Iteration proceeds as follows:

- start with some guess $v^{(0)}$ of the optimal value function.
- construct a sequence of guesses

$$v^{(i+1)}(x) = \min_{u \in U(x)} \{ \text{cost}(x, u) + v^{(i)}(\text{next}(x, u)) \}$$

This algorithm can be shown to converge at a linear rate [2].
Value Iteration proceeds as follows:

- start with some guess $v^{(0)}$ of the optimal value function.
- construct a sequence of guesses

$$v^{(i+1)}(x) = \min_{u \in U(x)} \left\{ \text{cost}(x, u) + v^{(i)}(\text{next}(x, u)) \right\}$$

This algorithm can be shown to converge at a linear rate [2]. Each iteration costs $O(|X||U|)$. 
Policy Iteration proceeds as follows:

- start with some guess $\pi^{(0)}$ of the optimal law.
Policy Iteration proceeds as follows:

- start with some guess $\pi^{(0)}$ of the optimal law.
- construct a sequence of guesses

\[
\nu^{\pi^{(i)}}(x) = \text{cost}(x, \pi^{(i)}(x)) + \nu^{\pi^{(i)}}(\text{next}(x, \pi^{(i)}(x)))
\]

\[
\pi^{(i+1)}(x) = \arg\min_{u \in \mathcal{U}(x)} \left\{ \text{cost}(x, u) + \nu^{\pi^{(i)}}(\text{next}(x, u)) \right\}
\]
Policy Iteration proceeds as follows:

- start with some guess $\pi^{(0)}$ of the optimal law.
- construct a sequence of guesses

\[ v^{\pi^{(i)}}(x) = cost(x, \pi^{(i)}(x)) + v^{\pi^{(i)}}(next(x, \pi^{(i)}(x))) \]

\[ \pi^{(i+1)}(x) = \arg \min_{u \in \mathcal{U}(x)} \{ cost(x, u) + v^{\pi^{(i)}}(next(x, u)) \} \]

Need to relax the first line or solve a system of linear equations.
**Policy Iteration** proceeds as follows:

- start with some guess $\pi^{(0)}$ of the optimal law.
- construct a sequence of guesses

$$
\nu^{\pi^{(i)}}(x) = \text{cost}(x, \pi^{(i)}(x)) + \nu^{\pi^{(i)}}(\text{next}(x, \pi^{(i)}(x)))
$$

$$
\pi^{(i+1)}(x) = \arg \min_{u \in \mathcal{U}(x)} \{ \text{cost}(x, u) + \nu^{\pi^{(i)}}(\text{next}(x, u)) \}
$$

Need to relax the first line or solve a system of linear equations. Under certain assumptions, this is faster than value iteration [3].
**Policy Iteration** proceeds as follows:

- start with some guess $\pi^{(0)}$ of the optimal law.
- construct a sequence of guesses

\[
v^{\pi^{(i)}}(x) = \text{cost}(x, \pi^{(i)}(x)) + v^{\pi^{(i)}}(\text{next}(x, \pi^{(i)}(x)))\]

\[
\pi^{(i+1)}(x) = \arg\min_{u \in \mathcal{U}(x)} \{ \text{cost}(x, u) + v^{\pi^{(i)}}(\text{next}(x, u))\}
\]

Need to relax the first line or solve a system of linear equations. Under certain assumptions, this is faster than value iteration [3]. However each iteration is more costly.
It is also of interest to consider the stochastic setting where we have

\[ p(y \mid x, u) = "\text{probability that } \text{next}(x, u) = y" \]
It is also of interest to consider the stochastic setting where we have
\[ p(y \mid x, u) = \text{"probability that } next(x, u) = y\text{"} \]

The Bellman equation for the optimal control law becomes
\[
\pi(x) = \arg \min_{u \in U(x)} \left\{ \text{cost}(x, u) + \mathbb{E}\left[ v(next(x, u)) \right] \right\}
\]
Discrete Control and the Bellman equations

• It is also of interest to consider the stochastic setting where we have

\[ p(y \mid x, u) = \text{"probability that } \text{next}(x, u) = y \text{"} \]

• The Bellman equation for the optimal control law becomes

\[ \pi(x) = \arg \min_{u \in \mathcal{U}(x)} \left\{ cost(x, u) + \mathbb{E}\left[ v(\text{next}(x, u)) \right] \right\} \]

• everything we have seen so far generalizes to this setting.
• It is also of interest to consider the stochastic setting where we have

\[ p(y \mid x, u) = \text{"probability that } next(x, u) = y \" \]

• The Bellman equation for the optimal control law becomes

\[ \pi(x) = \arg \min_{u \in U(x)} \{ cost(x, u) + \mathbb{E}[v(next(x, u))] \} \]

• everything we have seen so far generalizes to this setting.

• This is called a Markov Decision Process (MDP)
Continuous Control

- State $x \in \mathbb{R}^{n_x}$ and actions $u \in \mathcal{U}(x) \subset \mathbb{R}^{n_u}$ are real-valued vectors.
Continuous Control

- State $x \in \mathbb{R}^{n_x}$ and actions $u \in \mathcal{U}(x) \subset \mathbb{R}^{n_u}$ are real-valued vectors.
- Assume that our trajectory is given by

$$dx = f(x, u)dt + F(x, u)dw$$

where $dw$ is $n_w$-dimensional Brownian motion. We can also write the previous as

$$x(t) = x(0) + \int_0^t f(x(s), u(s))ds + \int_0^t F(x(s), u(s))dw(s)$$
Continuous Control

- State $x \in \mathbb{R}^{n_x}$ and actions $u \in \mathcal{U}(x) \subset \mathbb{R}^{n_u}$ are real-valued vectors.
- Assume that our trajectory is given by

$$dx = f(x, u)dt + F(x, u)dw$$

where $dw$ is $n_w$-dimensional Brownian motion. We can also write the previous as

$$x(t) = x(0) + \int_0^t f(x(s), u(s))ds + \int_0^t F(x(s), u(s))dw(s)$$

- Discretizing this into time steps of size $\Delta$, i.e. $t = k\Delta$, gives

$$x_{k+1} = x_k + \Delta f(x_k, u_k) + \sqrt{\Delta} F(x_k, u_k) \epsilon_k$$

where $\epsilon_k \sim \mathcal{N}(0, I^{n_w})$ and $x_k = x(k\Delta)$. 

Continuous Control

- We also need a cost function.
- For now assume finite horizon-problems, i.e. a final time $t_f$ is specified.
Continuous Control

- We also need a cost function.
- For now assume finite horizon-problems, i.e. a final time $t_f$ is specified.
- Separate the total cost into cost rate $\ell$ and final cost $h$. 

Total cost is then $J(x, u) = h(x(t_f)) + \int_{0}^{t_f} \ell(x(t), u(t), t) \, dt$.

Discretizing this gives $J(x, u) = h(x_n) + \Delta n - 1 \sum_{k=0}^{n-1} \ell(x_k, u_k, k\Delta)$ (4)

where $n = t_f / \Delta$. 

Continuous Control

- We also need a cost function.
- For now assume finite horizon-problems, i.e. a final time $t_f$ is specified.
- Separate the total cost into cost rate $\ell$ and final cost $h$.
- Total cost is then

$$J(x, u) = h(x(t_f)) + \int_0^{t_f} \ell(x(t), u(t), t) \, dt$$
Continuous Control

- We also need a cost function.
- For now assume finite horizon-problems, i.e. a final time $t_f$ is specified.
- Separate the total cost into cost rate $\ell$ and final cost $h$.
- Total cost is then
  \[ J(x, u.) = h(x(t_f)) + \int_0^{t_f} \ell(x(t), u(t), t)dt \]
- Discretizing this gives
  \[ J(x, u.) = h(x_n) + \Delta \sum_{k=0}^{n-1} \ell(x_k, u_k, k\Delta) \] (4)
  where $n = t_f / \Delta$. 
Continuous Control

- To summarize, we have

\[ x_{k+1} = x_k + \Delta f(x_k, u_k) + \sqrt{\Delta} F(x_k, u_k) \epsilon_k \]  \hspace{1cm} (5)

with \( \epsilon_k \sim N(0, I_{n_w}) \), and

\[ J(x., u.) = h(x_n) + \Delta \sum_{k=0}^{n-1} \ell(x_k, u_k, k\Delta) \]  \hspace{1cm} (6)
Continuous Control

- To summarize, we have

\[ x_{k+1} = x_k + \Delta f(x_k, u_k) + \sqrt{\Delta} F(x_k, u_k) \epsilon_k \]  

with \( \epsilon_k \sim \mathcal{N}(0, I_{nw}) \), and

\[ J(x., u.) = h(x_n) + \Delta \sum_{k=0}^{n-1} \ell(x_k, u_k, k\Delta) \]  

- From (5) we can see that

\[ x_{k+1} = x_k + \Delta f(x_k, u_k) + \xi \]

where \( \xi \sim \mathcal{N}(0, \Delta S(x_k, u_k)) \) and \( S(x, u) = F(x, u)F(x, u)^T \).
• To summarize, we have

\[ x_{k+1} = x_k + \Delta f(x_k, u_k) + \sqrt{\Delta F(x_k, u_k)} \epsilon_k \]  

with \( \epsilon_k \sim \mathcal{N}(0, I^{n_w}) \), and

\[ J(x., u.) = h(x_n) + \Delta \sum_{k=0}^{n-1} \ell(x_k, u_k, k\Delta) \]  

• From (5) we can see that

\[ x_{k+1} = x_k + \Delta f(x_k, u_k) + \xi \]

where \( \xi \sim \mathcal{N}(0, \Delta S(x_k, u_k)) \) and \( S(x, u) = F(x, u)F(x, u)^T \).

• With this we can define the **optimal value function** similarly

\[ v(x, k) = \min_u \{ \Delta\ell(x, u, k\Delta) + \mathbb{E}[v(x + \Delta f(x, u) + \xi, k + 1)] \} \]  

(7)
Continuous Control

- We will simplify \( \mathbb{E}[v(x + \Delta f(x, u) + \xi)] \).
- Setting \( \delta = \Delta f(x, u) + \xi \), Taylor expansion gives
  \[
  v(x + \delta) = v(x) + \delta^T v_x(x) + \frac{1}{2} \delta^T v_{xx}(x) \delta + o(\delta^3)
  \]
- Then
  \[
  \mathbb{E}[v(x + \delta)] = v(x) + \Delta f(x, u)^T v_x(x) + \frac{1}{2} \mathbb{E}[\xi^T v_{xx}(x)\xi] + o(\Delta^2)
  \]
- Now,
  \[
  \mathbb{E}[\xi^T v_{xx}\xi] = \mathbb{E}[\text{tr}(\xi^T v_{xx}\xi)]
  = \mathbb{E}[\text{tr}(\xi\xi^T v_{xx})]
  = \text{tr}(\text{Cov}[\xi]v_{xx})
  = \text{tr}(\Delta Sv_{xx})
  \]
Continuous control

Going back to

\[ v(x, k) = \min_u \{ \Delta \ell(x, u, k\Delta) + \mathbb{E}[v(x + \Delta f(x, u) + \xi, k + 1)] \} \]

and with

\[ \mathbb{E}[v(x + \delta)] = v(x) + \Delta f(x, u)^T v_x(x) + \frac{1}{2} tr(\Delta S(x, u)v_{xx}(x)) + o(\Delta^2) \]

we get

\[ \frac{v(x, k) - v(x, k + 1)}{\Delta} = \min_u \{ \ell + f^T v_x + \frac{1}{2} tr(Sv_{xx}) \} \]
\[
\frac{v(x, k) - v(x, k+1)}{\Delta} = \min_u \{ \ell + f^T v_x + \frac{1}{2} \text{tr}(S v_{xx}) \}
\]

and recall that \( k \) in \( v(x, k) \) represents time \( k\Delta \), so that the LHS is

\[
\frac{v(x, t) - v(x, t + \Delta)}{\Delta}
\]

As \( \Delta \to 0 \), this is \(-\frac{\partial}{\partial t} v\), which we denote \(-v_t\). So for \( v(x, t_f) = h(x) \) and \( 0 \leq t \leq t_f \), we have

\[
-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u) v_{xx}(x)) \} \tag{8}
\]

and the associated **optimal control law**

\[
\pi(x, t) = \arg \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u) v_{xx}(x)) \} \tag{9}
\]
Continuous Control

\[
\frac{v(x, k) - v(x, k + 1)}{\Delta} = \min_u \{ \ell + f^T v_x + \frac{1}{2} \text{tr}(S v_{xx}) \}
\]

and recall that \( k \) in \( v(x, k) \) represents time \( k\Delta \), so that the LHS is

\[
\frac{v(x, t) - v(x, t + \Delta)}{\Delta}
\]

As \( \Delta \to 0 \), this is \( -\frac{\partial}{\partial t} v \), which we denote \( -v_t \). So for \( v(x, t_f) = h(x) \) and \( 0 \leq t \leq t_f \), we have

\[
-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \} \quad (8)
\]

and the associated **optimal control law**

\[
\pi(x, t) = \arg \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \} \quad (9)
\]

Those are the **Hamilton-Jacobi-Bellman (HJB)** equations.
Continuous Control: solve the HJB Equations

\[-v_t(x, t) = \min_u \left\{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u) v_{xx}(x)) \right\} \]

with \( v(x, t_f) = h(x) \).

- Non-linear second-order PDE.
\[-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \} \]

with \( v(x, t_f) = h(x) \).

- Non-linear second-order PDE.
- May not have a classic solution.
Continuous Control: solve the HJB Equations

\[-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \}\]

with \( v(x, t_f) = h(x) \).

- Non-linear second-order PDE.
- May not have a classic solution
- Numerical methods relying on "viscosity" exist
Continuous Control: solve the HJB Equations

\[-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \}\]

with \( v(x, t_f) = h(x) \).

- Non-linear second-order PDE.
- May not have a classic solution
- numerical methods relying on "viscosity" exist
- suffers from "curse of dimensionality"
Continuous Control: solve the HJB Equations

\[-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \} \]

with \( v(x, t_f) = h(x) \).

- Non-linear second-order PDE.
- May not have a classic solution.
- Numerical methods relying on "viscosity" exist.
- Suffers from "curse of dimensionality".
- Several methods for approximate solutions exist and work well in practice.
Two infinite-horizon costs used in practice:

- **Discounted cost** formulation

  
  \[ J(x, u) = \int_0^\infty \exp(-\alpha t) \ell(x(t), u(t)) dt \]
Two infinite-horizon costs used in practice:

- **Discounted cost** formulation

  \[ J(x\cdot, u\cdot) = \int_0^\infty \exp(-\alpha t)\ell(x(t), u(t))\,dt \]

- **Average cost per stage** formulation

  \[ J(x\cdot, u\cdot) = \lim_{t_f \to \infty} \frac{1}{t_f} \int_0^{t_f} \ell(x(t), u(t))\,dt \]
Two infinite-horizon costs used in practice:

- **Discounted cost** formulation
  \[
  J(x, u.) = \int_{0}^{\infty} \exp(-\alpha t) \ell(x(t), u(t)) dt
  \]

- **Average cost per stage** formulation
  \[
  J(x, u.) = \lim_{t_f \to \infty} \frac{1}{t_f} \int_{0}^{t_f} \ell(x(t), u(t)) dt
  \]

Both those formulations bring similar HJB equations, except that they do not depend on time.
Two infinite-horizon costs used in practice:

- **Discounted cost** formulation
  \[
  J(x., u.) = \int_{0}^{\infty} \exp(-\alpha t) \ell(x(t), u(t)) dt
  \]

- **Average cost per stage** formulation
  \[
  J(x., u.) = \lim_{t_f \to \infty} \frac{1}{t_f} \int_{0}^{t_f} \ell(x(t), u(t)) dt
  \]

Both those formulations bring similar HJB equations, except that they do not depend on time. In that sense they are easier to solve using numerical approximations.
Two infinite-horizon costs used in practice:

- **Discounted cost** formulation

  \[ J(x_{\cdot}, u_{\cdot}) = \int_0^\infty \exp(-\alpha t) \ell(x(t), u(t)) dt \]

- **Average cost per stage** formulation

  \[ J(x_{\cdot}, u_{\cdot}) = \lim_{t_f \to \infty} \frac{1}{t_f} \int_0^{t_f} \ell(x(t), u(t)) dt \]

Both those formulations bring similar HJB equations, except that they do not depend on time.
In that sense they are easier to solve using numerical approximations. However the finite-horizon problem also advantages.
Linear-Quadratic-Gaussian control

- An important class of optimal control problems
- Unlike many other problems, it is possible to find a closed-form formula
- We will derive solutions in both the continuous and discrete cases
LQG: the Continuous Case

We make the following assumptions

- dynamics: \( \text{\dfrac{dx}{dt}} = (Ax + Bu) \) \( dt + Fdw \)
- cost rate: \( \ell(x, u) = \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx \)
- final cost: \( h(x) = \frac{1}{2} x^T Q^f x \)

where \( R, Q \) and \( Q^f \) are symmetric, \( R \) is positive definite, and set \( S = FF^T \).
LQG: the Continuous Case

We make the following assumptions

- dynamics: \( dx = (Ax + Bu)dt + Fdw \)
- cost rate: \( \ell(x, u) = \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx \)
- final cost: \( h(x) = \frac{1}{2} x^T Q^f x \)

where \( R, Q \) and \( Q^f \) are symmetric, \( R \) is positive definite, and set \( S = FF^T \).

Recall the HJB equation

\[
-v_t(x, t) = \min_u \left\{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \right\}
\]

with \( v(x, t_f) = h(x) \).
We make the following assumptions

- **dynamics:** \( dx = (Ax + Bu)dt + Fdw \)
- **cost rate:** \( \ell(x, u) = \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx \)
- **final cost:** \( h(x) = \frac{1}{2} x^T Q^f x \)

where \( R, Q \) and \( Q^f \) are symmetric, \( R \) is positive definite, and set \( S = FF^T \).

Recall the HJB equation

\[-v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \}\]

with \( v(x, t_f) = h(x) \).

In our case it reads

\[-v_t(x, t) = \min_u \{ \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx + (Ax + Bu)^T v_x(x) + \frac{1}{2} \text{tr}(Sv_{xx}(x)) \}\]

with \( v(x, t_f) = \frac{1}{2} x^T Q^f x \)
\[ -v_t(x, t) = \min_u \left\{ \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx + (Ax + Bu)^T v_x(x) + \frac{1}{2} \text{tr}(S v_{xx}(x)) \right\} \]

- We make the following guess: \( v(x, t) = \frac{1}{2} x^T V(t)x + a(t) \)
- The derivatives in the HJB equations are
  - \( v_t(x, t) = \frac{1}{2} x^T \dot{V}(t)x + \dot{a}(t) \)
  - \( v_x(x) = V(t)x \)
  - \( v_{xx}(x) = V(t) \)
Plugging back into the HJB equation gives

\[-v_t(x, t) = \min_u \left\{ \frac{1}{2} u^T Ru + \frac{1}{2} x^T Qx + (Ax + Bu)^T V(t)x + \frac{1}{2} \text{tr}(SV(t)) \right\} \]
Plugging back into the HJB equation gives

\[-v_t(x, t) = \min_u \left\{ \frac{1}{2} u^T R u + \frac{1}{2} x^T Q x + (Ax + Bu)^T V(t) x + \frac{1}{2} \text{tr}(SV(t)) \right\} \]

This is simply a quadratic in $u$, whose minimizer is

\[ u^* = -R^{-1}B^T V(t)x \]

and thus

\[-v_t(x, t) = \frac{1}{2} x^T \left( Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t) \right)x + \frac{1}{2} \text{tr}(SV(t)) \]
LQG: the Continuous Case

Plugging back into the HJB equation gives

\[ -v_t(x, t) = \min_u \left\{ \frac{1}{2} u^T R u + \frac{1}{2} x^T Q x + (A x + B u)^T V(t) x + \frac{1}{2} \text{tr}(S V(t)) \right\} \]

This is simply a quadratic in \( u \), whose minimizer is

\[ u^* = -R^{-1} B^T V(t) x \]

and thus

\[ -v_t(x, t) = \frac{1}{2} x^T (Q + A^T V(t) + V(t) A - V(t) B R^{-1} B^T V(t)) x + \frac{1}{2} \text{tr}(S V(t)) \]

Because \( v_t(x, t) = \frac{1}{2} x^T \dot{V}(t) x + \dot{a}(t) \), this gives

\[ -\dot{V}(t) = Q + A^T V(t) + V(t) A - V(t) B R^{-1} B^T V(t) \quad (10) \]

\[ -\dot{a}(t) = \frac{1}{2} \text{tr}(S V(t)) \]
Plugging back into the HJB equation gives

\[-v_t(x, t) = \min_u \left\{ \frac{1}{2} u^T R u + \frac{1}{2} x^T Q x + (A x + B u)^T V(t)x + \frac{1}{2} \text{tr}(S V(t)) \right\}\]

This is simply a quadratic in $u$, whose minimizer is

\[u^* = -R^{-1} B^T V(t)x\]

and thus

\[-v_t(x, t) = \frac{1}{2} x^T (Q + A^T V(t) + V(t)A - V(t)B R^{-1} B^T V(t))x + \frac{1}{2} \text{tr}(S V(t))\]

Because $v_t(x, t) = \frac{1}{2} x^T \dot{V}(t)x + \dot{a}(t)$, this gives

\[-\dot{V}(t) = Q + A^T V(t) + V(t)A - V(t)B R^{-1} B^T V(t)\] \hspace{1cm} (10)

\[-\dot{a}(t) = \frac{1}{2} \text{tr}(S V(t))\]

This is a **continuous-time Riccati** equation.
\[-\dot{V}(t) = Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t)\]

\[-\dot{a}(t) = \frac{1}{2}\text{tr}(SV(t))\]

The boundary conditions \(v(x, t_f) = \frac{1}{2}x^T Q^f x\) imply that \(V(t_f) = Q^f\) and \(a(t_f) = 0\).
LQG: the Continuous Case

\[-\dot{V}(t) = Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t)\]
\[-\dot{a}(t) = \frac{1}{2} \text{tr}(SV(t))\]

The boundary conditions \(v(x, t_f) = \frac{1}{2} x^T Q^f x\) imply that \(V(t_f) = Q^f\) and \(a(t_f) = 0\).

\(\Rightarrow\) This is a simple ODE, which is easy to solve.
LQG: the Continuous Case

$$-\dot{V}(t) = Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t)$$

$$-\dot{a}(t) = \frac{1}{2} \text{tr}(SV(t))$$

The boundary conditions $v(x, t_f) = \frac{1}{2} x^T Q^f x$ imply that $V(t_f) = Q^f$ and $a(t_f) = 0$.

$\Rightarrow$ This is a simple ODE, which is easy to solve.

The optimal control law is given by

$$u^* = -R^{-1} B^T V(t)x$$
\[ -\dot{V}(t) = Q + A^T V(t) + V(t)A - V(t)BR^{-1}B^T V(t) \]
\[ -\dot{a}(t) = \frac{1}{2} \text{tr}(SV(t)) \]

The boundary conditions \( v(x, t_f) = \frac{1}{2}x^T Q^f x \) imply that \( V(t_f) = Q^f \) and \( a(t_f) = 0 \).

\[ \Rightarrow \text{This is a simple ODE, which is easy to solve.} \]

The optimal control law is given by
\[ u^* = -R^{-1}B^T V(t)x \]

- It does not depend on the noise.
- It remains the same in the deterministic case, called the \textbf{linear-quadratic} regulator.
LQR: the Discrete Case

We make the following assumptions

- **dynamics:** \( x_{k+1} = Ax_k + Bu_k \)
- **cost:** \( \ell(x_k, u_k) = \frac{1}{2} u_k^T Ru_k + \frac{1}{2} x_k^T Q x_k \)
- **final cost:** \( h(x_n) = \frac{1}{2} x_n^T Q^f x_n \)

where \( R, Q \) and \( Q^f \) are symmetric, \( R \) is positive definite, and set \( S = FF^T \).
We make the following assumptions

- dynamics: \( x_{k+1} = Ax_k + Bu_k \)
- cost: \( \ell(x_k, u_k) = \frac{1}{2} u_k^T R u_k + \frac{1}{2} x_k^T Q x_k \)
- final cost: \( h(x_n) = \frac{1}{2} x_n^T Q^f x_n \)

where \( R, Q \) and \( Q^f \) are symmetric, \( R \) is positive definite, and set \( S = FF^T \).

Recall the Bellman equation

\[
\nu(x, k) = \min_u \left\{ \ell(x, u, k) + \nu(\text{next}(x, u, k)) \right\}
\]

with \( \nu(x_n) = h(x_n) \).
LQR: the Discrete Case

We make the following assumptions

- dynamics: $x_{k+1} = Ax_k + Bu_k$
- cost: $\ell(x_k, u_k) = \frac{1}{2} u_k^T Ru_k + \frac{1}{2} x_k^T Q x_k$
- final cost: $h(x_n) = \frac{1}{2} x_n^T Q^f x_n$

where $R$, $Q$ and $Q^f$ are symmetric, $R$ is positive definite, and set $S = FF^T$.

Recall the Bellman equation

$$v(x, k) = \min_u \{ \ell(x, u, k) + v(\text{next}(x, u, k)) \}$$

with $v(x_n) = h(x_n)$.

Again we make the assumption that

$$v(x, k) = \frac{1}{2} x^T V_k x$$
The boundary constraint gives $V_n = Q^f$.

Plugging everything gives

$$\frac{1}{2}x^TV_kx = \min_u \left\{ \frac{1}{2}u^TRu + \frac{1}{2}x^TQx + \frac{1}{2}(Ax + Bu)^T V_{k+1} (Ax + Bu) \right\}$$
The boundary constraint gives $V_n = Q^f$.

Plugging everything gives

$$\frac{1}{2} x^T V_k x = \min_u \left\{ \frac{1}{2} u^T R u + \frac{1}{2} x^T Q x + \frac{1}{2} (A x + B u)^T V_{k+1} (A x + B u) \right\}$$

This is simply a quadratic in $u$, and we get

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

which is a **discrete-time Ricatti** equation and the associated optimal control law

$$u_k = -L_k x_k$$

where $L_k = (R + B^T V_{k+1} B)^{-1} V_{k+1} A$
\[ V_k = Q + A^T V_{k+1} A - A^T V_{k+1} B (R + B^T V_{k+1} B)^{-1} B^T V_{k+1} A \]

- Start with \( V_n = Q^f \) and iterate backwards
- Can be computed offline
Deterministic Control: Pontryagin’s Maximum Principle

- Another approach to optimal control theory
- Developed in the Soviet Union by Pontryagin
- Only applies for deterministic problems.
- Avoids the curse of dimensionality.
- Applies for both continuous and discrete time.
Setting:

- dynamics: \( dx = f(x(t), u(t))dt \)
- cost rate: \( \ell(x(t), u(t), t) \)
- final cost: \( h(x(t_f)) \)

with fixed \( x_0 \) and final time \( t_f \).
Setting:

- dynamics: \( \frac{dx}{dt} = f(x(t), u(t)) \) dt
- cost rate: \( \ell(x(t), u(t), t) \)
- final cost: \( h(x(t_f)) \)

with fixed \( x_0 \) and final time \( t_f \).

Recall the HJB equation

\[
-v_t(x, t) = \min_u \left\{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \right\}
\]
Pontryagin’s Maximum Principle: The Continuous Case

Setting:

- dynamics: \( dx = f(x(t), u(t))dt \)
- cost rate: \( \ell(x(t), u(t), t) \)
- final cost: \( h(x(t_f)) \)

with fixed \( x_0 \) and final time \( t_f \).

Recall the HJB equation

\[
- v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \}
\]

Because we are in the deterministic case we have

\[
- v_t(x, t) = \min_u \{ \ell(x, u, t) + f(x, u)^T v_x(x) \}
\]
Pontryagin’s Maximum Principle: The Continuous Case

Setting:

- dynamics: \( dx = f(x(t), u(t))dt \)
- cost rate: \( \ell(x(t), u(t), t) \)
- final cost: \( h(x(t_f)) \)

with fixed \( x_0 \) and final time \( t_f \).

Recall the HJB equation

\[
- v_t(x, t) = \min_u \left\{ \ell(x, u, t) + f(x, u)^T v_x(x) + \frac{1}{2} \text{tr}(S(x, u)v_{xx}(x)) \right\}
\]

Because we are in the deterministic case we have

\[
- v_t(x, t) = \min_u \left\{ \ell(x, u, t) + f(x, u)^T v_x(x) \right\}
\]

Suppose optimal control law is given by \( u = \pi(x, t) \)
\[ -v_t(x, t) = \ell(x, \pi(x, t), t) + f(x, \pi(x, t))^T v_x(x, t) \]
Pontryagin’s Maximum Principle: The Continuous Case

\[-v_t(x, t) = \ell(x, \pi(x, t), t) + f(x, \pi(x, t))^T v_x(x, t)\]

Taking derivatives w.r.t. \(x\)

\[0 = v_{tx} + \ell_x + \pi_x^T \ell_u + f_x^T v_x + \pi_x^T f_u^T v_x + v_{xx} f\]
Pontryagin’s Maximum Principle: The Continuous Case

\[-v_t(x, t) = \ell(x, \pi(x, t), t) + f(x, \pi(x, t))^T v_x(x, t)\]

Taking derivatives w.r.t. \(x\)

\[0 = v_{tx} + \ell_x + \pi_x^T \ell_u + f_x^T v_x + \pi_x^T f_u^T v_x + v_{xx} f\]

Observe that \(\dot{v}_x = v_{xx} \dot{x} + v_{tx} = v_{xx} f + v_{tx},\)

\[0 = \dot{v}_x + \ell_x + f_x^T v_x + \pi_x^T (\ell_u + f_u^T v_x)\]
Pontryagin’s Maximum Principle: The Continuous Case

\[-v_t(x, t) = \ell(x, \pi(x, t), t) + f(x, \pi(x, t))^T v_x(x, t)\]

Taking derivatives w.r.t. \(x\)

\[0 = v_{tx} + \ell_x + \pi_x^T \ell_u + f_x^T v_x + \pi_x^T f_u^T v_x + v_{xx} f\]

Observe that \(\dot{v}_x = v_{xx} \dot{x} + v_{tx} = v_{xx} f + v_{tx},\)

\[0 = \dot{v}_x + \ell_x + f_x^T v_x + \pi_x^T (\ell_u + f_u^T v_x)\]

Observe that \(\ell_u + f_u^T v_x = \ell_u(x, \pi(x, t), t) + f_u(x, \pi(x, t))^T v_x(x, t)\)
Pontryagin’s Maximum Principle: The Continuous Case

$$-v_t(x, t) = \ell(x, \pi(x, t), t) + f(x, \pi(x, t))^T v_x(x, t)$$

Taking derivatives w.r.t. $x$

$$0 = v_{tx} + \ell_x + \pi_x^T \ell_u + f_x^T v_x + \pi_x^T f_u^T v_x + v_{xx} f$$

Observe that

$$\dot{v}_x = v_{xx} \dot{x} + v_{tx} = v_{xx} f + v_{tx},$$

$$0 = \dot{v}_x + \ell_x + f_x^T v_x + \pi_x^T (\ell_u + f_u^T v_x)$$

Observe that

$$\ell_u + f_u^T v_x = \ell_u(x, \pi(x, t), t) + f_u(x, \pi(x, t))^T v_x(x, t) = 0$$
We then get

\[-\dot{v}_x(x, t) = f_x(x, \pi(x, t))^T v_x(x, t) + \ell_x(x, \pi(x, t), t)\]
Pontryagin’s Maximum Principle: The Continuous Case

We then get

$$-\dot{v}_x(x, t) = f_x(x, \pi(x, t))^T v_x(x, t) + \ell_x(x, \pi(x, t), t)$$

Setting $p = v_x$, this gives

$$-\dot{p}(t) = f_x(x, \pi(x, t))^T p(t) + \ell_x(x, \pi(x, t), t)$$
Pontryagin’s Maximum Principle: The Continuous Case

We then get

\[-\dot{v}_x(x, t) = f_x(x, \pi(x, t))^T v_x(x, t) + \ell_x(x, \pi(x, t), t)\]

Setting \( p = v_x \), this gives

\[-\dot{p}(t) = f_x(x, \pi(x, t))^T p(t) + \ell_x(x, \pi(x, t), t)\]

The **maximum principle** thus reads

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
-\dot{p}(t) &= f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) &= \arg\min_u \{ \ell(x(t), u, t) + f(x(t), u)^T p(t) \}
\end{align*}
\]

with boundary conditions \( p(t_f) = v_x(x(t_f), t_f) = h_x(x(t_f)) \), and \( x_0, t_f \) given.
Pontryagin’s Maximum Principle: The Continuous Case

Setting the Hamiltonian $H(x, u, p, t) := \ell(x, u, t) + f(x, u)^T p$, the maximum principle reads

\[
\dot{x}(t) = f(x(t), u(t)) \\
-\dot{p}(t) = f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) = \arg\min_u H(x(t), u, p(t), t)
\]

with $p(t_f) = h_x(x(t_f))$
Setting the **Hamiltonian** $H(x, u, p, t) := \ell(x, u, t) + f(x, u)^T p$, the maximum principle reads

\[
\dot{x}(t) = f(x(t), u(t)) \\
-\dot{p}(t) = f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) = \arg\min_u H(x(t), u, p(t), t)
\]

with $p(t_f) = h_x(x(t_f))$

- Simple ODE, cost grows linearly with $n_x$
Setting the **Hamiltonian** \( H(x, u, p, t) := \ell(x, u, t) + f(x, u)^T p \), the maximum principle reads

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
-\dot{p}(t) &= f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) &= \arg\min_u H(x(t), u, p(t), t)
\end{align*}
\]

with \( p(t_f) = h_x(x(t_f)) \)

- Simple ODE, cost grows linearly with \( n_x \)
- existing software packages to solve
Pontryagin’s Maximum Principle: The Continuous Case

Setting the **Hamiltonian** $H(x, u, p, t) := \ell(x, u, t) + f(x, u)^T p$, the maximum principle reads

\[
\dot{x}(t) = f(x(t), u(t)) \\
-\dot{p}(t) = f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) = \arg\min_{u} H(x(t), u, p(t), t)
\]

with $p(t_f) = h_x(x(t_f))$

- Simple ODE, cost grows linearly with $n_x$
- existing software packages to solve
- Only issue is to solve for the Hamiltonian
Pontryagin’s Maximum Principle: The Continuous Case

Setting the **Hamiltonian** \( H(x, u, p, t) := \ell(x, u, t) + f(x, u)^T p \), the maximum principle reads

\[
\begin{align*}
\dot{x}(t) & = f(x(t), u(t)) \\
-\dot{p}(t) & = f_x(x(t), u(t))^T p(t) + \ell_x(x(t), u(t), t) \\
u(t) & = \arg \min_{u} H(x(t), u, p(t), t)
\end{align*}
\]

with \( p(t_f) = h_x(x(t_f)) \)

- Simple ODE, cost grows linearly with \( n_x \)
- existing software packages to solve
- Only issue is to solve for the Hamiltonian
- For problems where the dynamic is linear and the cost is quadratic w.r.t. the control \( u \), a nice closed form formula exists.
• Derivation in the continuous and discrete case is also possible using Lagrange multipliers
• Optimization using gradient descent is possible in the discrete case
**Goal:** From a sequence of noisy measurements, estimate the true dynamics.

\[
\begin{align*}
x_{k+1} &= Ax_k + w_k \\
y_k &= Hx_k + v_k
\end{align*}
\]

where \( w_k \sim N(0, S) \) and \( v_k \sim N(0, P) \), \( x_0 \sim N(\hat{x}_0, \Sigma_0) \), and \( A, H, S, P, \hat{x}_0, \Sigma_0 \) are known.

\[
\hat{p}_k = p(x_k | y_0, \ldots, y_{k-1})
\]

\[
\hat{p}_0 = N(\hat{x}_0, \Sigma_0)
\]
Optimal Estimation and the Kalman Filter

• **Goal:** From a sequence of noisy measurements, estimate the true dynamics.

• Intimately tied to the problem of optimal control
• **Goal:** From a sequence of noisy measurements, estimate the true dynamics.

• Intimately tied to the problem of optimal control

\[
\begin{align*}
\text{dynamics: } x_{k+1} &= Ax_k + w_k \\
\text{observation: } y_k &= Hx_k + v_k
\end{align*}
\]

where \( w_k \sim \mathcal{N}(0, S) \) and \( v_k \sim \mathcal{N}(0, P) \), \( x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0) \), and \( A, H, S, P, \hat{x}_0, \Sigma_0 \) are known.
**Goal:** From a sequence of noisy measurements, estimate the true dynamics.

Intimately tied to the problem of optimal control

\[
\begin{align*}
\text{dynamics: } & \quad x_{k+1} = Ax_k + w_k \\
\text{observation: } & \quad y_k = Hx_k + v_k
\end{align*}
\]

where \( w_k \sim \mathcal{N}(0, S) \) and \( v_k \sim \mathcal{N}(0, P) \), \( x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0) \), and \( A, H, S, P, \hat{x}_0, \Sigma_0 \) are known.

\[\Rightarrow\] Goal: estimate the probability distribution of \( x_k \) given \( y_0, \ldots, y_{k-1} \):

\[
\begin{align*}
\hat{p}_k &= p(x_k \mid y_0, \ldots, y_{k-1}) \\
\hat{p}_0 &= \mathcal{N}(\hat{x}_0, \Sigma_0)
\end{align*}
\]
Using properties of multivariate Gaussian, it can be shown that

\[
\hat{p}_{k+1} = p(x_{k+1} \mid y_0, \ldots, y_k) \sim \mathcal{N}(\hat{x}_{k+1}, \Sigma_{k+1})
\]

where

\[
\hat{x}_{k+1} = A\hat{x}_k + A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} (y_k - H\hat{x}_k)
\]  

(11)

and

\[
\Sigma_{k+1} = S + A\Sigma_k A^T - A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} H\Sigma_k A^T
\]  

(12)
Using properties of multivariate Gaussian, it can be shown that

\[ \hat{p}_{k+1} = p(x_{k+1} | y_0, \ldots, y_k) \sim \mathcal{N}(\hat{x}_{k+1}, \Sigma_{k+1}) \]

where

\[ \hat{x}_{k+1} = A\hat{x}_k + A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} (y_k - H\hat{x}_k) \] (11)

and

\[ \Sigma_{k+1} = S + A\Sigma_k A^T - A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} H\Sigma_k A^T \] (12)

This is the **Kalman filter**.
Using properties of multivariate Gaussian, it can be shown that

\[ \hat{p}_{k+1} = p(x_{k+1} \mid y_0, \ldots, y_k) \sim \mathcal{N}(\hat{x}_{k+1}, \Sigma_{k+1}) \]

where

\[ \hat{x}_{k+1} = A\hat{x}_k + A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} (y_k - H\hat{x}_k) \quad (11) \]

and

\[ \Sigma_{k+1} = S + A\Sigma_k A^T - A\Sigma_k H^T (P + H\Sigma_k H^T)^{-1} H\Sigma_k A^T \quad (12) \]

This is the Kalman filter.

Recall the Riccati equation for LQR

\[ V_k = Q + A^T V_{k+1} A - A^T V_{k+1} B (R + B^T V_{k+1} B)^{-1} B^T V_{k+1} A \]
Conclusion

What we covered today

- Bellman equations and dynamic programming
What we covered today

- Bellman equations and dynamic programming
- Hamilton-Jacobi-Bellman equations
What we covered today

- Bellman equations and dynamic programming
- Hamilton-Jacobi-Bellman equations
- Linear-Quadratic-Gaussian, Linear-Quadratic regulator and the Riccati equations
What we covered today

- Bellman equations and dynamic programming
- Hamilton-Jacobi-Bellman equations
- Linear-Quadratic-Gaussian, Linear-Quadratic regulator and the Riccati equations
- Pontryagin’s maximum principle
Conclusion

What we covered today

- Bellman equations and dynamic programming
- Hamilton-Jacobi-Bellman equations
- Linear-Quadratic-Gaussian, Linear-Quadratic regulator and the Riccati equations
- Pontryagin’s maximum principle
- Kalman Filter
Conclusion

What we covered today

- Bellman equations and dynamic programming
- Hamilton-Jacobi-Bellman equations
- Linear-Quadratic-Gaussian, Linear-Quadratic regulator and the Riccati equations
- Pontryagin’s maximum principle
- Kalman Filter

What we didn’t cover

- solving non-linear optimal problem using linear relaxation
- duality between optimal control and optimal estimation
Any questions?
Thank you!
