

Hamilton-Jacobi-Bellman eq., Riccati eq., Linear Quadratic Regulator

Control Theory, Lecture 9

by Sergei Savin

Spring 2021

■ Hamilton-Jacobi-Bellman equation

- ▶ Definitions
- ▶ Cost, optimal cost
- ▶ Differentiating optimal cost

■ Algebraic Riccati equation

- ▶ HJB for LTI
- ▶ Linear Quadratic Regulator
- ▶ Numerical methods

HAMILTON-JACOBI-BELLMAN EQUATION

Definitions

Let us define dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (1)$$

with initial conditions $\mathbf{x}(0)$.

Additionally we define *control policy* as:

$$\mathbf{u} = \pi(\mathbf{x}, t) \quad (2)$$

To connect with the previous ways we talked about control, we can say that choosing different control gains and different feed-forward control amounts to choosing a different control policy.

HAMILTON-JACOBI-BELLMAN EQUATION

Cost, optimal cost

Let J be an additive cost function:

$$J(\mathbf{x}_0, \pi(\mathbf{x}, t)) = \int_0^\infty g(\mathbf{x}, \mathbf{u}) dt \quad (3)$$

where $g(\mathbf{x}, \mathbf{u})$ is instantaneous cost and $\mathbf{x}_0 = \mathbf{x}(0)$ is the initial conditions. Notice that J depends on \mathbf{x}_0 rather than $\mathbf{x}(t)$, since initial conditions and control policy completely define the trajectory of the system $\mathbf{x}(t)$.

Let J^* be the optimal (lowest possible) cost. In other words:

$$J^*(\mathbf{x}_0) = \inf_{\pi} J(\mathbf{x}_0, \pi(\mathbf{x}, t)) \quad (4)$$

Optimal cost is attained when optimal policy is attained:

$$\pi = \pi^*(\mathbf{x}, t)$$

HAMILTON-JACOBI-BELLMAN EQUATION

Differentiating optimal cost

Since $J^*(\mathbf{x}_0)$ does not depend on t , its full derivative is zero:

$$\frac{dJ^*(\mathbf{x}_0)}{dt} = 0 \quad (5)$$

At the same time, we can expand the full derivative as follows:

$$\frac{dJ^*}{dt} = \frac{\partial J^*}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial J^*}{\partial t} = 0 \quad (6)$$

Observe that $\frac{\partial J^*}{\partial t} = g(\mathbf{x}, \mathbf{u})$, and $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. Therefore:

$$\frac{\partial J^*}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}) + g(\mathbf{x}, \mathbf{u}) = 0 \quad (7)$$

With this, we can formulate *Hamilton-Jacobi-Bellman equation* (HJB):

$$\min_{\mathbf{u}} \left[g(\mathbf{x}, \mathbf{u}) + \frac{\partial J^*}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}) \right] = 0 \quad (8)$$

This can be loosely interpreted as follows: the value in square brackets is $\dot{J}(\mathbf{x}_0, \pi)$, which is equal to 0 when $\pi = \pi^*(\mathbf{x}, t)$, and is positive otherwise (in the small vicinity of π^*), as $J(\mathbf{x}_0, \pi^*)$ is smaller than any $J(\mathbf{x}_0, \pi)$, $\pi^* \neq \pi$.

We can find control that delivers minimum to the function (8):

$$u^* = \arg \min_{\mathbf{u}} \left[g(\mathbf{x}, \mathbf{u}) + \frac{\partial J^*}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}) \right] \quad (9)$$

For LTI, dynamics is:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (10)$$

We can choose quadratic cost:

$$g(\mathbf{x}, \mathbf{u}) = \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u} \quad (11)$$

Then HJB becomes:

$$\min_{\mathbf{u}} [\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u} + \frac{\partial J^*}{\partial \mathbf{x}}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u})] = 0 \quad (12)$$

where $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ and $\mathbf{R} = \mathbf{R}^\top > 0$.

ALGEBRAIC RICCATI

HJB for LTI, part 2

There is a theorem that says that for LTI with quadratic cost, J^* has the form:

$$J^* = \mathbf{x}^\top \mathbf{S} \mathbf{x} \quad (13)$$

where $\mathbf{S} = \mathbf{S}^\top \geq 0$.

Then HJB becomes:

$$\min_{\mathbf{u}} \left[\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u} + \mathbf{x}^\top \mathbf{S} (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) + (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u})^\top \mathbf{S} \mathbf{x} \right] = 0$$

Simplifying, we get:

$$\min_{\mathbf{u}} \left[\mathbf{u}^\top \mathbf{R} \mathbf{u} + \mathbf{x}^\top (\mathbf{Q} + \mathbf{S} \mathbf{A} + \mathbf{A}^\top \mathbf{S}) \mathbf{x} + \mathbf{x}^\top \mathbf{S} \mathbf{B} \mathbf{u} + \mathbf{u}^\top \mathbf{B}^\top \mathbf{S} \mathbf{x} \right] = 0$$

ALGEBRAIC RICCATI

Linear Quadratic Regulator

Finding partial derivative of the HJB with respect to \mathbf{u} and setting it to zero (as it is an extreme point) we get:

$$2\mathbf{u}^\top \mathbf{R} + 2\mathbf{x}^\top \mathbf{S}\mathbf{B} = 0 \quad (14)$$

This expression can be transposed and \mathbf{u} separated:

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S}\mathbf{x} \quad (15)$$

This is the desired control law. We can see that it is *proportional*. We can re-write it as:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (16)$$

where $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S}$ is the controller gain. This control law is called Linear Quadratic Regulator (LQR).

Substituting found control law into the HJB, we find:

$$\min_{\mathbf{u}} [\mathbf{x}^\top (\mathbf{Q} + \mathbf{S}\mathbf{A} + \mathbf{A}^\top \mathbf{S})\mathbf{x} + \mathbf{x}^\top \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S}\mathbf{x} - \mathbf{x}^\top \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S}\mathbf{x} - \mathbf{x}^\top \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S}\mathbf{x}] = 0 \quad (17)$$

Simplifying, we get:

$$\mathbf{x}^\top (\mathbf{Q} + \mathbf{S}\mathbf{A} + \mathbf{A}^\top \mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S})\mathbf{x} = 0 \quad (18)$$

which would hold for all \mathbf{x} iff:

$$\mathbf{Q} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S} + \mathbf{S}\mathbf{A} + \mathbf{A}^\top \mathbf{S} = 0 \quad (19)$$

This is the *Algebraic Riccati equation*.

ALGEBRAIC RICCATI

Numerical methods

There are a number of ways to solve LQR:

- In MATLAB there is a function $[K, S, P] = \text{lqr}(A, B, Q, R)$, where $P = \text{eig}(A - B * K)$
- In Python, there is $S = \text{scipy.linalg.solve_continuous_are}(A, B, Q, R)$
- In Drake (by MIT and Toyota Research) there is a function $(K, S) = \text{LinearQuadraticRegulator}(A, B, Q, R)$

THANK YOU!

Lecture slides are available via Moodle.

You can help improve these slides at:

github.com/SergeiSa/Control-Theory-Slides-Spring-2021

Check Moodle for additional links, videos, textbook suggestions.