Riccati Differential Equation

Tamas Kis | tamas.a.kis@outlook.com | https://tamaskis.github.io

CONTENTS

1 RICCATI DIFFERENTIAL EQUATION

1.1 Definition

The finite-horizon linear quadratic regular (LQR) optimal control problem is defined as

$$\boxed{\begin{aligned} \underset{\mathbf{u}(t)}{\text{minimize}} \quad & \int_{t_0}^{T} \left(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} + 2\mathbf{x}^T\mathbf{S}\mathbf{u}\right)\, dt + \mathbf{x}(T)^T\mathbf{P}_T\mathbf{x}(T) \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ & \mathbf{P}(T) = \mathbf{P}_T \end{aligned}} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$, $\mathbf{R} \in \mathbb{R}^{m\times m}$, $\mathbf{S} \in \mathbb{R}^{n\times m}$, $\mathbf{x}(T) \in \mathbb{R}^n$, and $\mathbf{P}_T \in \mathbb{R}^{n\times n}$. The solution to the finite-horizon LQR problem is

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

where

$$\mathbf{K}(t) = \mathbf{R}^{-1}\left[\mathbf{B}^T\mathbf{P}(t) + \mathbf{S}^T\right] \tag{2}$$

and where $\mathbf{K} \in \mathbb{R}^{m\times n}$ and $\mathbf{P} \in \mathbb{R}^{n\times n}$. The matrix function $\mathbf{P}(t)$ is found by solving the **Riccati differential equation** *backwards* in time (i.e. from $t = T$ to $t = t_0$) using the terminal condition $\mathbf{P}(T) = \mathbf{P}_T$. The Riccati differential equation is given by [5]

$$\boxed{\dot{\mathbf{P}} = -\left[\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - (\mathbf{P}\mathbf{B} + \mathbf{S})\mathbf{R}^{-1}(\mathbf{B}^T\mathbf{P} + \mathbf{S}^T) + \mathbf{Q}\right]} \tag{3}$$

1.2 Solving the IVP

The Riccati differential equation is a matrix-valued ODE of the form

$$\frac{d\mathbf{M}}{dt} = \mathbf{F}(t, \mathbf{M})$$

where $\mathbf{M} \in \mathbb{R}^{p\times q}$ (where p and q are arbitrary scalars). However, MATLAB's ODE solvers are only equipped to solve *vector*-valued ODEs of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y})$$

where $\mathbf{y} \in \mathbb{R}^p$.

We can transform a matrix-valued ODE to a vector-valued ODE using the `mat2vec_ode` and `mat2vec_IC` functions of the *IVP Solver Toolbox*, and then transform the results of the vector-valued IVP into the results of the corresponding matrix-valued IVP using the `vec2mat_sol` function of the *IVP Solver Toolbox* [3, pp. 38–45][2].

1.3 Conditions for Existence and Uniqueness

Let's define the matrix $\mathbf{M}$ as

$$\boxed{\mathbf{M} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{R} \end{bmatrix}} \tag{4}$$

A unique solution to the Riccati differential equation exists if and only if the following conditions are satisfied [1][4, p. 35]:

1. $\mathbf{M}$ is symmetric positive semidefinite ($\mathbf{M} \succeq 0$). If $\mathbf{S} = \mathbf{0}$, this condition reduces to the following two conditions:
   (a) $\mathbf{Q}$ is symmetric positive semidefinite ($\mathbf{Q} \succeq 0$).
   (b) $\mathbf{R}$ is symmetric positive definite ($\mathbf{R} \succ 0$).
2. $\mathbf{P}_T$ is symmetric positive semidefinite ($\mathbf{P}_T \succeq 0$).
3. $(\mathbf{A}, \mathbf{B})$ stabilizable.
4. $(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{S}^T, \mathbf{Q} - \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^T)$ detectable.
   - If $\mathbf{S} = \mathbf{0}$, this condition reduces to $(\mathbf{A}, \mathbf{Q}^{1/2})$ detectable.

REFERENCES

[1] `icare`. MathWorks. Accessed: December 23, 2021. URL: https://www.mathworks.com/help/control/ref/icare.html.

[2] Tamas Kis. *IVP Solver Toolbox*. 2021. URL: https://github.com/tamaskis/IVP_Solver_Toolbox-MATLAB.

[3] Tamas Kis. *Solving Initial Value Problems for ODEs*. 2021. URL: https://tamaskis.github.io/documentation/Solving_Initial_Value_Problems_for_ODEs.pdf.

[4] Eugene Lavretsky and Kevin A. Wise. *Robust and Adaptive Control with Aerospace Applications*. London, UK: Springer-Verlag, 2013.

[5] *Linear-quadratic regulator*. Wikipedia. Accessed: December 22, 2021. URL: https://en.wikipedia.org/wiki/Linear-quadratic_regulator.