

## Article

# Optimization of Weight Matrices for the Linear Quadratic Regulator Problem Using Algebraic Closed-Form Solutions

Daegyun Choi <sup>1</sup>, Donghoon Kim <sup>1,\*</sup> and James D. Turner <sup>2</sup>

<sup>1</sup> Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA; daegyun.choi@uc.edu

<sup>2</sup> Independent Researcher, 9399 Wade Blvd., Frisco, TX 75035, USA; jamesdanielturner75@gmail.com

\* Correspondence: donghoon.kim@uc.edu

**Abstract:** This work proposes an analytical gradient-based optimization approach to determine the optimal weight matrices that make the state and control input at the final time close to zero for the linear quadratic regulator problem. Most existing methodologies focused on regulating the diagonal elements using only bio-inspired approaches or analytical approaches. The method proposed, contrarily, optimizes both diagonal and off-diagonal matrix elements based on the gradient. Moreover, by introducing a new variable composed of the steady-state and time-varying terms for the Riccati matrix and using the coordinate transformation for the state, one develops algebraic equations-based closed-form solutions to generate the required states and numerical partial derivatives for an optimization strategy that does not require the computationally intensive numerical integration process. The authors test the algorithm with one- and two-degrees-of-freedom linear plant models, and it yields the weight matrices that successfully satisfy the pre-defined requirement, which is the norm of the augmented states less than  $10^{-5}$ . The results suggest the broad applicability of the proposed algorithm in science and engineering.

**Keywords:** optimalfeedback control; weight matrices; optimization; algebraic closed-form solutions



**Citation:** Choi, D.; Kim, D.; Turner, J.D. Optimization of Weight Matrices for the Linear Quadratic Regulator Problem Using Algebraic Closed-Form Solutions. *Electronics* **2023**, *12*, 4526. <https://doi.org/10.3390/electronics12214526>

Academic Editor: Nikolay Hinov

Received: 28 August 2023

Revised: 11 October 2023

Accepted: 30 October 2023

Published: 3 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In optimal control theory, one defines optimal control problems to determine control signals that minimize a performance index as well as satisfy the physical constraints [1]. Whereas it is difficult to find explicit expressions for the optimal control of nonlinear systems, linear systems can provide explicit equations for the optimal control input. For the advantage of using explicit equations, many works linearize nonlinear systems at the equilibrium point. One of the widely used state feedback control problems is defined for a linear system, where a quadratic performance index is defined using the state, control, terminal state, and associated unspecified weight matrices, called the linear quadratic regulator (LQR) problem. In general, the weight matrices are adjusted by users to find the optimal control and corresponding states that satisfy the requirement or desired behavior set by users. For instance, if it is important that the intermediate state is to be small, a large state weight matrix can be selected. The other weight matrices, such as the terminal state and control weight matrices concerning the terminal state and control effort, are selected in a similar manner. Although a key of solving the LQR problem is to determine the weight matrices properly, most studies have determined these matrices by trial-and-error, which is time-consuming work. For instance, considering a second-order system with one degree-of-freedom (DOF) requires determining seven symmetric components of the weight matrices (i.e., three for the state weight  $Q \in \mathbb{R}^{2 \times 2}$ , one for the control weight  $R \in \mathbb{R}$ , and three for the terminal state weight  $S_f \in \mathbb{R}^{2 \times 2}$ ). However, if one considers a two-DOFs system, the number of the symmetric components in the weight matrices increases to 23 (i.e., 10 for  $Q \in \mathbb{R}^{4 \times 4}$ , three for  $R \in \mathbb{R}^{2 \times 2}$ , and 10 for  $S_f \in \mathbb{R}^{4 \times 4}$ ). That is, the number of

components to be determined in the weight matrices exponentially increases if the size of the system increases. Moreover, determining the weight matrices generally depends on the engineers' knowledge, and this approach does not always guarantee that the system response will meet the specified performance. For this reason, it is critical to design a proper optimization strategy for determining the weight matrices while satisfying the user-specified performance and reducing the engineers' effort to find the proper weight matrices, especially for high DOF applications.

Many researchers have proposed weight matrices selection approaches to reduce the effort for determining weight matrices using bio-inspired heuristic algorithms. Among a number of bio-inspired optimization algorithms, some researchers employed a genetic algorithm (GA) to optimize the weight matrices for various system models and fitness functions. Marada et al. [2] formulated the LQR problem based on the linearized inverted pendulum system and found the state and control weight matrices using the GA. Their approach minimizes a fitness function that consists of a weighted summation of the settling time and the control input considering the location of system closed-loop poles for system stability. Dhiman et al. [3] also utilized the same system model and optimization algorithm, but the considered fitness function was the quadratic performance index. The GA-based weight optimization approach was also utilized for the vehicle suspension system by Yu et al. [4] and linearized spacecraft attitude dynamics by Kukreti et al. [5]. For the vehicle suspension system, the fitness function was composed of suspension performance indices, such as vertical acceleration of the car, dynamic tire load, and suspension working space. Similarly, Kukreti et al. considered the control performance for the fitness function, which is defined as the weighted summation of the final time and attitude and angular velocity errors. In addition to the GA, to optimize the weight matrices of the LQR problem, scholars applied many different optimization algorithms (e.g., a particle swarm optimization [6], its variations [7,8], ant-lion algorithm [9], ant colony optimization [10], Jaya's algorithm [11], and bat algorithm [12], etc.) into diverse dynamic systems, and more studies including the mentioned research are summarized in Table 1. The aforementioned approaches commonly considered the diagonal components of the state and control weight matrices, as well as the algebraic Riccati matrix equation. Also, these approaches, based on bio-inspired optimization algorithms, take offline optimization which requires a huge computational burden.

Unlike weight selection using bio-inspired heuristic algorithms, some scholars proposed analytical approaches to determine the weight matrices. Elumalai and Raaja [13] proposed a time-domain-based algebraic Riccati and Lagrange optimization technique to determine the weight matrices, where they satisfy a pre-defined performance for a two-dimensional torsion system. Sarkar and Dewan [14] presented a pole-placement approach that satisfies user-defined time domain specifications for generating the state-feedback gain for an inverted pendulum system. Yang [15] presented a pole assignment design of a quaternion-based spacecraft attitude control problem for generating the weight matrices by considering a balance between the performance and fuel consumption.

**Table 1.** Relevant research papers.

Refs.	System Model	Approach	Consideration
[2]	Inverted pendulum	Genetic algorithm	- Weighted summation of settling time and control input
[3]	Inverted pendulum	Genetic algorithm	- Performance index of the LQR problem
[4]	Active suspension system	Genetic algorithm	- Vertical acceleration of the car, dynamic tire load, and suspension working space
[5]	Spacecraft attitude dynamics	Genetic algorithm	- Weighted summation of the final time, attitude, and angular velocity error

Table 1. Cont.

Refs.	System Model	Approach	Consideration
[6]	Inverted double pendulum	Particle swarm optimization	- Performance index of the LQR problem
[7]	Inverted pendulum	Adaptive particle swarm optimization	- Integrated state error
[8]	Semi-active suspension system	Improved particle swarm optimization	- Vertical acceleration, suspension deflection and tire dynamic load of the passive suspension vehicle
[9]	Inverted double pendulum	Ant-lion algorithm	- Crane position, upswing and downswing angle
[10]	Vehicle suspension system	Ant colony optimization	- Suspension travel, suspension velocity, tire deflection, and tire velocity
[11]	Deregulated power system	Jaya's algorithm	- State error
[12]	Vehicle suspension system	Bat algorithm	- Ride comfort and passenger safety
[16]	Vehicle suspension system	Adaptive predator-prey optimization	- Integral square error of the state
[17]	Two-axis CNC system	Artificial bee colony optimization	- Settling time and overshoot
[18]	Inverted double pendulum	Neuro Evolution of Augmenting Topologies	- State error
[13]	Two-dimensional torsion system	Lagrange optimization	- System response (overshoot, setting time, steady state error)
[14]	Inverted pendulum	Pole-placement approach	- Time-domain specifications
[15]	spacecraft attitude dynamics	Pole assignment design	- Control performance and fuel consumption

Most of the existing studies only find diagonal components for the state and control weight matrices using bio-inspired heuristic optimization algorithms along with the infinite-time algebraic Riccati matrix equation, and the limited number of studies deals with analytical methods for finding diagonal components of the state and control weight matrices as shown in Table 1. Considering only diagonal components for the weight matrices may not provide better control performance if the states of the considered system are coupled with each other [19]. On the other hand, this work optimizes all symmetric components of the weight matrices in the LQR problem to minimize the state and control values at the final time by utilizing analytic gradients. Also, the optimization process only contains algebraic equations for the principal equations and the partial derivatives, which are developed by employing the steady-state and time-varying terms for the time-varying differential Riccati matrix equation [20] and utilizing the coordinate transformation for the states [21]. The main contributions of this work include the following:

- We design a gradient-based optimization strategy for determining *diagonal and off-diagonal components of the state, control, and terminal state weight matrices*; to provide more flexibility for optimization.
- We find *only algebraic equations-contained closed-form solutions for the principal equations and the sensitivity partial derivatives*, including the time-varying Riccati matrix equation, which require less computational cost for the optimization.

As a result, no numerical integration is required for solving the finite-time fixed-time optimal feedback control as well as optimizing the weight matrices. In simulation studies, two most widely used numerical examples are presented to demonstrate the effectiveness of the proposed approach, for second-order linear differential systems with one and two DOFs.

## 2. Formulation of Linear Quadratic Regulator Problem

The optimal state feedback control problem for a linear time-invariant and continuous-time model with a finite and fixed-time, called the LQR problem, is to find the control inputs that minimize the following performance index [22]:

$$\mathcal{L} = \frac{1}{2} \mathbf{x}^T(t_f) S_f \mathbf{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [\mathbf{x}^T(t) Q \mathbf{x}(t) + \mathbf{u}^T(t) R \mathbf{u}(t)] dt, \quad (1)$$

subject to

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{2}$$

with the initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$  and terminal condition  $\mathbf{x}(t_f) = \mathbf{x}_f$ . The optimal control is obtained by [22]:

$$\mathbf{u}^*(t) = -R^{-1}B^T S(t)\mathbf{x}(t), \tag{3}$$

where the time-varying Riccati matrix is found by integrating the following backward in time:

$$\dot{S}(t) = -S(t)A - A^T S(t) + S(t)BR^{-1}B^T S(t) - Q. \tag{4}$$

This work assumes that  $(A, B)$  is controllable and  $(A, D)$  is observable. Note that  $D \in \mathbb{R}^{g \times n}$  with  $g \leq n$  is defined as  $Q = D^T D$  and can be obtained by Cholesky decomposition. In addition, the proof of the stability of the given closed-loop control system is discussed in Appendix A. To find the optimal control  $\mathbf{u}^*(t)$ , indeed, it is required to properly select the weight matrices ( $Q, R,$  and  $S_f$ ), which are the user-defined parameters in general. Then, one can compute the time-varying Riccati matrix, state, and control input through numerical integration. If the user-defined requirement, such as the norm of the states less than a certain value at the final time, is satisfied, one can obtain the state with the optimal control input as shown in Figure 1. In general, the weight matrices are determined by the engineers' knowledge and trial-and-error. The engineers need to change the weight matrices iteratively until the requirement is satisfied. However, it is not only time-consuming work but also hard to guarantee that the system response satisfies the specified performance.

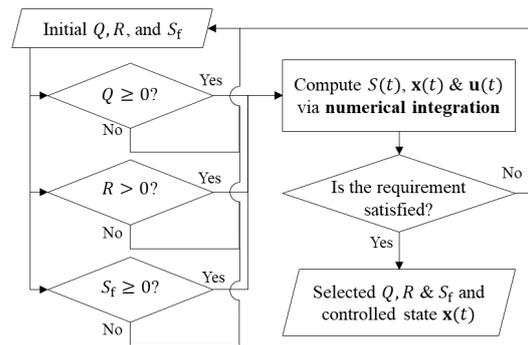


Figure 1. Conventional weight matrices selection procedure.

### 3. Optimization of Weight Matrices

To efficiently determine the weight matrices of the LQR problem, this work proposes an optimization process for the weight matrices. Unlike the aforementioned studies in the introduction, this work considers all symmetric components of the weight matrices ( $Q, R,$  and  $S_f$ ) in the formulation to gain more flexibility in optimization. Here, the unknown parameters for the optimization process are the symmetric elements of  $Q, R,$  and  $S_f$  are as follows:

$$Q = \begin{bmatrix} q_1 & \cdots & q_n \\ \vdots & \ddots & \vdots \\ q_n & \cdots & q_N \end{bmatrix}, \tag{5}$$

$$R = \begin{bmatrix} r_1 & \cdots & r_m \\ \vdots & \ddots & \vdots \\ r_m & \cdots & r_M \end{bmatrix}, \tag{6}$$

$$S_f = \begin{bmatrix} s_1 & \cdots & s_n \\ \vdots & \ddots & \vdots \\ s_n & \cdots & s_N \end{bmatrix}. \tag{7}$$

For the simple notation, the symmetric components of the weight matrices are gathered into one vector as

$$\begin{aligned} \mathbf{w} &\equiv [w_1 \dots w_P]^T \in \mathbb{R}^P \\ &= [q_1 \dots q_N \ r_1 \dots r_M \ s_1 \dots s_N]^T. \end{aligned} \tag{8}$$

For optimization, this work introduces a new variable consisting of the state and control, called an augmented state, as follows:

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \in \mathbb{R}^{n+m}. \tag{9}$$

The goal of the proposed optimization process is to find the optimal weight matrices that minimize the performance index while  $\mathbf{y}(t)$  at the final time is to be zero. Thus, for Equation (9), one applies Taylor expansion at  $t = t_f$  as a function of the weight matrices' symmetric elements defined in Equations (5)–(7), which leads to

$$\mathbf{y}(t_f) = \mathbf{y}(t_f, Q, R, S_f) + \sum_{k=1}^N \mathbf{y}_{,q_k} dq_k + \sum_{l=1}^M \mathbf{y}_{,r_l} dr_l + \sum_{k=1}^N \mathbf{y}_{,s_k} ds_k. \tag{10}$$

It is important to note that  $\beta_{,\alpha}$  indicates the partial derivative of an arbitrary variable  $\beta$  with respect to an arbitrary variable  $\alpha$ . That is, “,” between the variable and the subscript indicates the partial derivative. Targeting  $\mathbf{y}(t_f) = \mathbf{0}$  and collecting the partial derivatives in Equation (10) into a global Jacobian matrix  $J$  leads to

$$\begin{aligned} \mathbf{0} &= \mathbf{y}(t_f, Q, R, S_f) + J\mathbf{dw} \\ &= \mathbf{y}_f + J\mathbf{dw}, \end{aligned} \tag{11}$$

where

$$J = [\mathbf{y}_{,q_1} \dots \mathbf{y}_{,q_N} \ \mathbf{y}_{,r_1} \dots \mathbf{y}_{,r_M} \ \mathbf{y}_{,s_1} \dots \mathbf{y}_{,s_N}] \in \mathbb{R}^{(n+m) \times P}, \tag{12}$$

$$\mathbf{dw} = [dq_1 \dots dq_N \ dr_1 \dots dr_M \ ds_1 \dots ds_N]^T \in \mathbb{R}^P. \tag{13}$$

Here, each element in the correction vector  $\mathbf{dw}$  represents changes in each symmetric element of the weight matrices. Since the number of unknown parameters ( $P = 2N + M$ ) is larger than the number of final conditions ( $n + m$ ), one can obtain the solution for  $\mathbf{dw}$  by minimizing the following:

$$\mathcal{H} = \frac{\mathbf{dw}^T \mathbf{dw}}{2} + \lambda^T (\mathbf{y}_f + J\mathbf{dw}), \tag{14}$$

The necessary conditions for the optimization are derived as

$$\mathcal{H}_{,\mathbf{dw}} = \mathbf{dw} + J^T \lambda = \mathbf{0}, \tag{15}$$

$$\mathcal{H}_{,\lambda} = \mathbf{y}_f + J\mathbf{dw} = \mathbf{0}. \tag{16}$$

Then, manipulating two necessary conditions, the solution for  $\lambda$  is given by

$$\lambda = (JJ^T)^{-1} \mathbf{y}_f, \tag{17}$$

and the minimum norm optimization solution for the parameter correction vector is obtained as

$$\mathbf{dw} = -J^T (JJ^T)^{-1} \mathbf{y}_f. \tag{18}$$

Therefore, the symmetric components of the weight matrices are updated as

$$\mathbf{w}_{\text{update}} = \mathbf{w}_{\text{previous}} + d\mathbf{w}. \tag{19}$$

Figure 2 displays the procedure of the proposed optimization process.

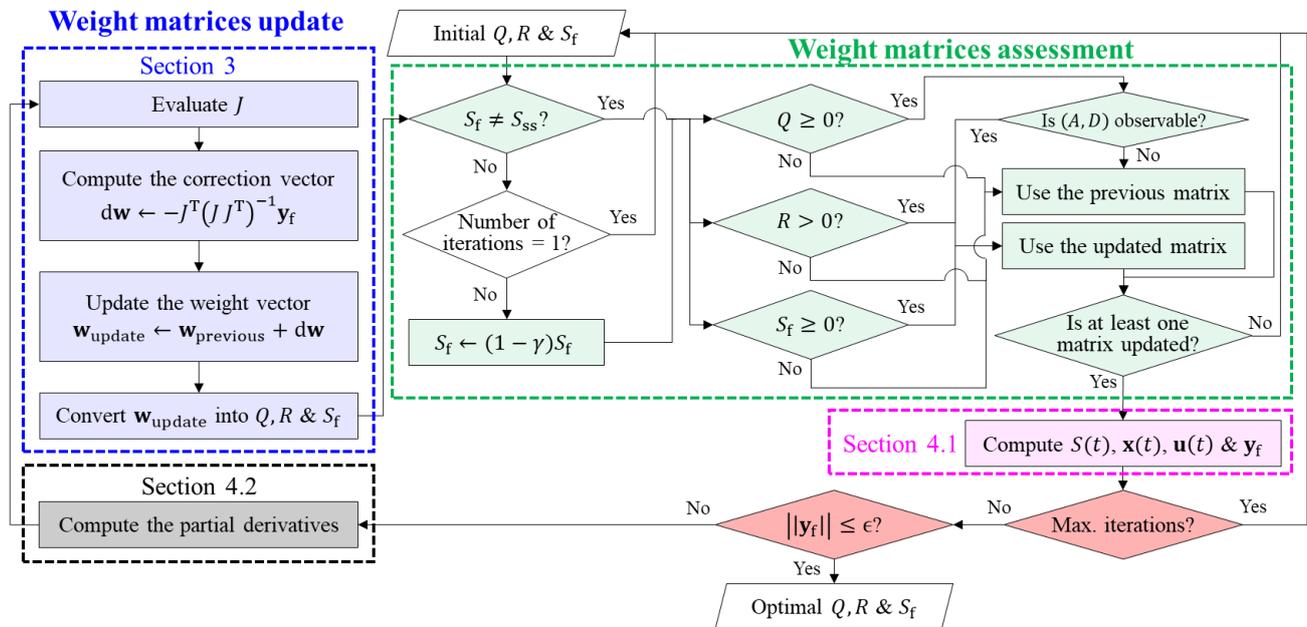


Figure 2. Proposed weight matrices optimization procedure.

Once initial weight matrices are assumed, one needs to confirm whether  $S_f$  is equal to  $S_{ss}$  or not. This condition comes from deriving the closed-form solution for the time-varying Riccati matrix that will be explained in Section 4.1.1. If  $S_f = S_{ss}$  at the first step, users need to redefine the initial  $S_f$ . However, if  $S_f = S_{ss}$  in a second or higher iteration,  $S_f$  obtained is slightly changed by replacing it with  $(1 - \gamma)S_f$ . Note that  $\gamma$  is a user-defined small constant value, which is considered not only to avoid the equality condition but also to use the values as close to the updated values as possible. After that, one requires to verify whether the weight matrices satisfy the definiteness condition or not. That is,  $Q$  and  $S_f$  must be positive semi-definite, and  $R$  must be positive definite. In particular, after finding  $D$  matrix via Cholesky decomposition, the observability for  $(A, D)$  pair is evaluated. After confirming the definiteness condition for each updated weight matrix, one finds  $S(t)$ ,  $\mathbf{x}(t)$ , and  $\mathbf{u}(t)$ , and evaluates  $\mathbf{y}_f$  using the confirmed weight matrices only. Note that it uses previous weight matrices if the updated weight matrices violate the definiteness conditions. For instance, if all weight matrices violate the definiteness condition, one should try different initial weight matrices. However, for example, if only the updated  $Q$  violates the definiteness condition, it uses  $Q$  obtained from the previous step and the updated  $R$  and  $S_f$  for the next step. Next, it evaluates the two-stage stopping conditions. First, it terminates the process when the current iteration number exceeds the maximum iteration number, which is defined by users, and then repeats the entire process with the newly assumed initial weight matrices. This means that the initial guesses for the weight matrices used are not properly selected, and it requires starting from new initial guesses to find the optimal weight matrices. Otherwise, it evaluates the second stopping condition. If the resulting solution meets the requirement defined by users ( $\epsilon$ ), it terminates the process and endorses the weight matrices as optimal. If not, it updates the weight matrices. To proceed with the update process, one evaluates the Jacobian  $J$ . After computing  $d\mathbf{w}$  and updating  $\mathbf{w}$  using  $d\mathbf{w}$ , one converts  $\mathbf{w}_{\text{update}}$  into the corresponding updated weight matrix. This procedure continues until the resulting solution satisfies the requirement.

The proposed optimization process requires finding the state and control using Equations (2)–(4) to evaluate the augmented state at the final time and computing the partial derivatives of Equations (2)–(4) with respect to each symmetric component of the weight matrices to determine the Jacobian matrix in Equation (12). Then, the partial derivatives for the state, control, and time-varying Riccati matrix with respect to each symmetric element of the weight matrices  $w_p$  (for  $p = 1, \dots, P$ ) are derived as

$$\dot{\mathbf{x}}_{rw_p}(t) = A\mathbf{x}_{rw_p}(t) + B\mathbf{u}_{rw_p}(t), \tag{20}$$

$$\mathbf{u}_{rw_p}(t) = R^{-1}R_{rw_p}R^{-1}B^T S(t)\mathbf{x}(t) - R^{-1}B^T S_{rw_p}(t)\mathbf{x}(t) - R^{-1}B^T S(t)\mathbf{x}_{rw_p}(t), \tag{21}$$

$$\begin{aligned} \dot{S}_{rw_p}(t) = & -S_{rw_p}(t)A - A^T S_{rw_p}(t) + S_{rw_p}(t)BR^{-1}B^T S(t) \\ & - S(t)BR^{-1}R_{rw_p}R^{-1}B^T S(t) + S(t)BR^{-1}B^T S_{rw_p}(t) - Q_{rw_p}, \end{aligned} \tag{22}$$

where  $\mathbf{x}_{rw_p}(t_0) = \mathbf{0}$  and  $S_{rw_p}(t_0) = 0$ , because  $\mathbf{x}(t_0)$  and  $S(t_0)$  are constants. So, to evaluate the augmented state, it first numerically integrates the differential Riccati matrix equation in Equation (4) backward in time and then integrates the controlled system response computed by substituting the optimal control defined in Equation (3) into Equation (2) forward in time. In the same manner, to compute the Jacobian matrix, it integrates the partial derivatives in Equations (20) and (22) forward in time. In fact, this process is technically straightforward. However, it requires the exponential computational load as the dimension of the state increases because of multiple numerical integrations. Hence, this work develops closed-form algebraic expressions for the principal equations and partial derivatives. The use of closed-form solutions not only eliminates the need for introducing numerical integration methods but also allows computing analytic sensitivity partial derivatives. Otherwise, it would require extensive numerical integration calculations.

#### 4. Closed-Form Solutions for Principal Equations and Partial Derivatives

This section introduces the closed-form solutions for the principal equations and the sensitivity partial derivatives. The closed-form solutions for the principal equations are mainly used to solve the LQR problem and find the augmented state at the final time, and the closed-form solutions for the partial derivatives are utilized to obtain the Jacobian in the optimization process.

##### 4.1. Derivation of Closed-Form Solutions for Principal Equations

This work considers the time-varying Riccati matrix, the controlled state, control input, and the performance index as the principal equations. It is important to note that the closed-form solutions derived here do not require numerical integration, so there is a minimal computational load to solve the LQR problem.

##### 4.1.1. Time-Varying Riccati Matrix

It eliminates numerical integration by introducing the following closed-form solution for the differential Riccati matrix equation in Equation (4), where the solution consists of a steady-state term and a time-varying term [20,23]:

$$S(t) = S_{ss} + Z^{-1}(t). \tag{23}$$

Note that  $Z(t)$  is invertible, and the detailed condition is described after Equation (27). Also, the final condition is given by  $S_f$ , and the steady-state solution  $S_{ss}$  satisfies the algebraic Riccati matrix equation:

$$-S_{ss}A - A^T S_{ss} + S_{ss}BR^{-1}B^T S_{ss} - Q = 0. \tag{24}$$

Substituting Equation (23) into Equation (4), the differential Lyapunov matrix equation for  $Z(t)$  is derived as

$$\dot{Z}(t) = \bar{A}Z(t) + Z(t)\bar{A}^T - BR^{-1}B^T, \tag{25}$$

where  $\bar{A}$  is defined as [24]

$$\bar{A} = A - BR^{-1}B^T S_{ss}, \tag{26}$$

and from Equation (23), the final condition of  $Z(t)$  is found to be

$$Z(t_f) = (S_f - S_{ss})^{-1}. \tag{27}$$

Note that the condition  $S_f \neq S_{ss}$  must be satisfied during the optimization process so that  $S_f - S_{ss}$  is invertible. This condition aligns with the reversibility of  $Z(t)$  in Equation (23). The closed-form solution for  $Z(t)$  with a steady-state term and a time-varying term is derived as

$$Z(t) = Z_{ss} + e^{\bar{A}(t-t_f)} Z_b e^{\bar{A}(t-t_f)}, \tag{28}$$

where  $e^{(\cdot)}$  is the  $\mathbb{R}^{n \times n}$  exponential matrix, and the boundary condition of  $Z(t)$  is defined by

$$Z_b = (S_f - S_{ss})^{-1} - Z_{ss}. \tag{29}$$

Here, the steady-state solution for  $Z_{ss}$  satisfies the algebraic Lyapunov matrix equation [25]:

$$\bar{A}Z_{ss} + Z_{ss}\bar{A}^T - BR^{-1}B^T = 0. \tag{30}$$

As described above, several equations and solutions for the algebraic Riccati matrix equation and algebraic Lyapunov matrix equation are interconnected with each other to obtain the closed-form solution for the time-varying Riccati matrix. To sum up, the time-varying Riccati matrix can be obtained by substituting the solution of Equations (24) and (28) into Equation (23) as the closed-form solution. See Appendix B.1 for derivations of Equations (25) and (28).

#### 4.1.2. State for the Closed-Loop Control System

From the problem formulation, the governing differential equation for the controlled state is derived by substituting Equations (3) and (23) into Equation (2):

$$\dot{\mathbf{x}}(t) = [\bar{A} - BR^{-1}B^T Z^{-1}(t)] \mathbf{x}(t), \tag{31}$$

with the initial condition  $\mathbf{x}_0$ . The closed-form solution for Equation (31) is expressed as [21,26]

$$\mathbf{x}(t) = \Phi(t, t_0) \mathbf{x}_0. \tag{32}$$

Note that the state transition matrix  $\Phi(t, t_0)$  has the explicit form defined as

$$\Phi(t, t_0) = Z(t) e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0), \tag{33}$$

where  $Z(t)$  is defined by Equation (28), and  $\Phi(t, t_0)$  satisfies the following properties:

$$\Phi(t_2, t_0) = \Phi(t_2, t_1) \Phi(t_1, t_0), \tag{34}$$

$$\Phi(t_0, t_1) = \Phi^{-1}(t_1, t_0), \tag{35}$$

$$\Phi(t_0, t_0) = I. \tag{36}$$

Therefore, the closed-form solution for the state is expressed as

$$\mathbf{x}(t) = \left( Z_{ss} + e^{\bar{A}(t-t_f)} Z_b e^{\bar{A}(t-t_f)} \right) e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0) \mathbf{x}_0. \tag{37}$$

See Appendix B.2 for derivations of Equations (32) and (33).

#### 4.1.3. Control for the Closed-Loop Control System

The control input is defined by introducing Equation (23) into Equation (3):

$$\mathbf{u}^*(t) = -R^{-1}B^T(S_{ss} + Z^{-1}(t))\mathbf{x}(t). \tag{38}$$

Using Equation (32) and expanding Equation (38) lead to

$$\mathbf{u}^*(t) = -R^{-1}B^T S_{ss} \Phi(t, t_0) \mathbf{x}_0 - R^{-1}B^T Z^{-1}(t) \Phi(t, t_0) \mathbf{x}_0, \tag{39}$$

and substituting Equation (33) into Equation (39) yields:

$$\mathbf{u}^*(t) = -R^{-1}B^T S_{ss} Z(t) e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0) \mathbf{x}_0 - R^{-1}B^T e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0) \mathbf{x}_0. \tag{40}$$

Finally, the closed-form solution for the control input is expressed as

$$\mathbf{u}^*(t) = -R^{-1}B^T \left( S_{ss} \left( Z_{ss} + e^{\bar{A}(t-t_f)} Z_b e^{\bar{A}(t-t_f)} \right) + I \right) e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0) \mathbf{x}_0. \tag{41}$$

#### 4.1.4. Performance Index

The performance index on  $[t, t_f]$   $\mathcal{L} = \frac{1}{2} \mathbf{x}^T(t) S(t) \mathbf{x}(t)$  is already a closed-form solution, because the closed-form solution for  $\mathbf{x}(t)$  and  $S(t)$  are given by Equations (37) and (23), respectively.

Algorithm 1 summarizes the process for solving the LQR problem using the closed-form solutions. The process for finding the solution for the LQR problem using the closed-form solutions contains backward computation for obtaining the Riccati matrix from the given  $S_f$  and forward computation for obtaining the state and optimal control. Note that numerical integration processes, like Runge–Kutta methods, are not incorporated. In contrast, the conventional method requires numerical integration processes, such as backward integration for the Riccati matrix and forward integration for the controlled state.

---

**Algorithm 1** Computation procedure for solving the LQR problem using the closed-form solutions

---

- 1: **Inputs:**  $A, B, t_0, t_f, \mathbf{x}_0, \mathbf{x}_f, Q, R, S_f$
  - 2: **Outputs:**  $\mathbf{x}(t), \mathbf{u}^*(t)$
  - 3:  $S_{ss} \leftarrow 0 = -S_{ss}A - A^T S_{ss} + S_{ss}BR^{-1}B^T S_{ss} - Q$
  - 4:  $\bar{A} \leftarrow A - BR^{-1}B^T S_{ss}$
  - 5:  $Z_{ss} \leftarrow 0 = \bar{A}Z_{ss} + Z_{ss}\bar{A}^T - BR^{-1}B^T$
  - 6:  $Z(t_f) \leftarrow (S_f - S_{ss})^{-1}$
  - 7: **for**  $t \leftarrow t_f$  **to**  $t_0$  **do**
  - 8:      $Z(t) \leftarrow Z_{ss} + e^{\bar{A}(t-t_f)} [(S_f - S_{ss})^{-1} - Z_{ss}] e^{\bar{A}(t-t_f)}$
  - 9:      $S(t) \leftarrow S_{ss} + Z^{-1}(t)$
  - 10: **end for**
  - 11:  $Z^{-1}(t_0) \leftarrow Z(t_0) \neq 0$  from line 8
  - 12: **for**  $t \leftarrow t_0$  **to**  $t_f$  **do**
  - 13:      $\mathbf{x}(t) \leftarrow Z(t) e^{-\bar{A}^T(t-t_0)} Z^{-1}(t_0) \mathbf{x}_0$
  - 14:      $\mathbf{u}^*(t) \leftarrow -R^{-1}B^T S(t) \mathbf{x}(t)$
  - 15: **end for**
- 

#### 4.2. Derivation of Closed-Form Solutions for Sensitivity Partial Derivatives

The partial derivative of a function with respect to the independent variable measures the sensitivity of the function. In this section, the sensitivity partial derivatives for the aforementioned equations and variables with respect to the symmetric elements of the weight matrices are derived to form the Jacobian matrix defined in Equation (12) that is used for the proposed optimization process. Note that no numerical integration is needed to compute the partial derivatives because all of the expressions are purely algebraic.

#### 4.2.1. Time-Varying Riccati Matrix Partial Derivatives

Based on the closed-form expressions, the partial derivatives of the Riccati matrix in Equation (23) with respect to the symmetric components of the weight matrices are given by

$$S_{rw_p}(t) = S_{ss,rw_p} - Z^{-1}(t)Z_{rw_p}(t)Z^{-1}(t), \tag{42}$$

where  $S_{ss,rw_p}$  is found by taking the partial derivatives of the algebraic Riccati matrix equation in Equation (24):

$$\bar{A}^T S_{ss,rw_p} + S_{ss,rw_p} \bar{A} = -S_{ss} B R^{-1} R_{rw_p} R^{-1} B^T S_{ss} - Q_{rw_p}. \tag{43}$$

See Appendix C.1 for this derivation. Moreover, the detailed descriptions for  $Q_{rw_p}$ ,  $R_{rw_p}$ , and  $S_{f,rw_p}$  are explained in the first paragraph in Appendix C. Then,  $Z_{rw_p}(t)$  is found by taking the partial derivatives of Equation (28) as follows:

$$\begin{aligned} Z_{rw_p}(t) = & Z_{ss,rw_p} + \left[ e^{\bar{A}(t-t_f)} \right]_{,rw_p} Z_b e^{\bar{A}(t-t_f)} \\ & + e^{\bar{A}(t-t_f)} Z_{b,rw_p} e^{\bar{A}^T(t-t_f)} + e^{\bar{A}(t-t_f)} Z_b \left[ e^{\bar{A}^T(t-t_f)} \right]_{,rw_p}. \end{aligned} \tag{44}$$

Here,  $[\cdot]_{,rw_p}$  denotes the partial derivative of the matrix exponential, and  $Z_{b,rw_p}$  is given by taking the partial derivatives of Equation (29) as follows:

$$Z_{b,rw_p} = - (S_f - S_{ss})^{-1} (S_{f,rw_p} - S_{ss,rw_p}) (S_f - S_{ss})^{-1} - Z_{ss,rw_p}. \tag{45}$$

Also,  $Z_{ss,rw_p}$  is found by taking the partial derivatives of the steady-state Lyapunov matrix in Equation (30) as follows:

$$\bar{A} Z_{ss,rw_p} + Z_{ss,rw_p} \bar{A}^T = - \bar{A}_{rw_p} Z_{ss} - Z_{ss} \bar{A}_{rw_p}^T - B R^{-1} R_{rw_p} R^{-1} B^T, \tag{46}$$

where  $\bar{A}_{rw_p}$  is found by taking the partial derivatives of the closed-loop system dynamics matrix in Equation (26) as follows:

$$\bar{A}_{rw_p} = B R^{-1} R_{rw_p} R^{-1} B^T S_{ss} - B R^{-1} B^T S_{ss,rw_p}. \tag{47}$$

The details of the derivatives for all variables and matrices with respect to the symmetric elements of each weight matrix are explained throughout Appendices C.1–C.5.

#### 4.2.2. State Partial Derivatives

The state partial derivatives are generated by modeling the state as Equation (32), leading to the state partial derivatives given by

$$\mathbf{x}_{rw_p}(t) = \Phi_{rw_p}(t, t_0) \mathbf{x}_0, \tag{48}$$

where  $\Phi_{rw_p}(t, t_0)$  is expressed as

$$\begin{aligned} \Phi_{rw_p}(t, t_0) = & Z_{rw_p}(t) e^{-\bar{A}(t-t_0)} Z^{-1}(t_0) + Z(t) \left[ e^{-\bar{A}(t-t_0)} \right]_{,rw_p} Z^{-1}(t_0) \\ & - Z(t) e^{-\bar{A}(t-t_0)} Z^{-1}(t_0) Z_{rw_p}(t_0) Z^{-1}(t_0). \end{aligned} \tag{49}$$

The detailed expressions of the partial derivative of the matrix exponential are explained in Appendix C.6.

### 4.2.3. Control Partial Derivatives

The partial derivatives of the optimal control in Equation (40) are expressed as

$$\begin{aligned} \mathbf{u}_{rwp}(t) = & R^{-1}R_{rwp}R^{-1}B^T S_{ss}\Phi(t, t_0)\mathbf{x}_0 - R^{-1}B^T S_{ssrwp}\Phi(t, t_0)\mathbf{x}_0 \\ & - R^{-1}B^T S_{ss}\Phi_{rwp}(t, t_0)\mathbf{x}_0 + R^{-1}R_{rwp}R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T \left[ e^{-\bar{A}^T(t-t_0)} \right]_{rwp} Z^{-1}(t_0)\mathbf{x}_0 \\ & + R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)Z_{rwp}(t_0)Z^{-1}(t_0)\mathbf{x}_0. \end{aligned} \tag{50}$$

The detailed derivation process is described in Appendix C.7. Note that Equations (48) and (50) are primarily utilized to form the Jacobian matrix, which is composed of the augmented state.

## 5. Simulation Study

### 5.1. Problem Descriptions

This work considers two of the most widely used example problems for the second-order differential equation with one and two DOFs to validate the efficacy of the closed-form algebraic expressions and the performance of the proposed optimization process. The system of the example problems is shown in Figure 3. The formulations described in (1) and (2) are used, and the system dynamics and control influence matrices for each problem are defined in Table 2. Also, the simulation parameters for each variable are tabulated in Table 3.

**Table 2.** Model parameters [27].

	1 DOF	2 DOFs
State Variables	$\mathbf{x}(t) = [x_1 \ x_2]^T \in \mathbb{R}^2, u(t) \in \mathbb{R}$	$\mathbf{x}(t) = [x_1 \ x_2 \ x_3 \ x_4]^T \in \mathbb{R}^4, \mathbf{u}(t) = [u_1 \ u_2]^T \in \mathbb{R}^2$
Models	$A = \begin{bmatrix} 0 & 1 \\ -\frac{k_1}{m_1} & -\frac{c_1}{m_1} \end{bmatrix}, B = \begin{bmatrix} 0 \\ -\frac{1}{m_1} \end{bmatrix}$	$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_2+k_3}{m_2} & -\frac{k_3}{m_2} & -\frac{c_2+c_3}{m_2} & \frac{c_3}{m_2} \\ \frac{k_3}{m_3} & -\frac{k_3}{m_3} & \frac{c_3}{m_3} & -\frac{c_3}{m_3} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_2} & 0 \\ 0 & \frac{1}{m_3} \end{bmatrix}$

**Table 3.** Simulation parameters [27].

Parameter	Value	
	1 DOF	2 DOFs
Mass (kg)	$m_1 = 1$	$m_2 = 1$ $m_3 = 1$
Spring coefficient (N/m)	$k_1 = 0.64$	$k_2 = 1$ $k_3 = 0.5$
Damping coefficient (Ns/m)	$c_1 = 0.16$	$c_2 = 0.1$ $c_3 = 0.1$
Initial condition $\mathbf{x}(t_0)$	$[10 \ 10]^T$	$[10 \ 1 \ 0 \ 0]^T$
State weight matrix $Q$	$I_{2 \times 2}$	$I_{4 \times 4}$
Control weight matrix $R$	1	$I_{2 \times 2}$
Terminal state weight matrix $S_f$	$I_{2 \times 2}$	$I_{4 \times 4}$
Final time $t_f$ (s)		10
Time interval $dt$ (s)		0.01

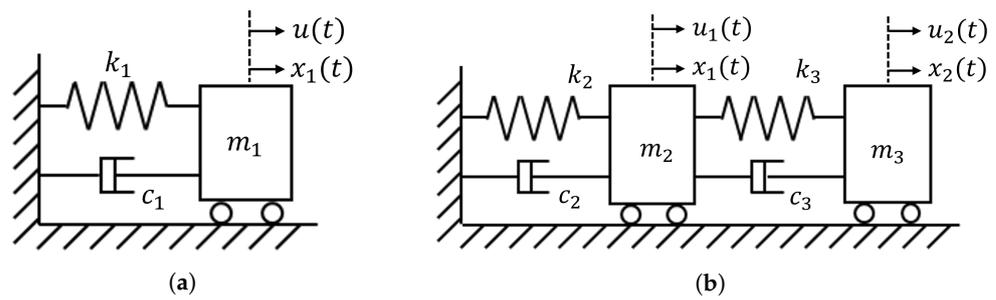


Figure 3. System of the example problems. (a) 1 DOF. (b) 2 DOFs.

5.2. Verification of the Closed-Form Solutions for the LQR Problem

To highlight the efficacy of the closed-form solutions, especially for the principal equations, numerical simulations are conducted using the closed-form solutions that eliminate the numerical integration process. The resulting state and control trajectories computed using the closed-form solutions are compared with the results obtained by a conventional approach, which is to find the Riccati matrix by numerically integrating backward in time and the state and control trajectories by integrating forward in time. Figure 4a,b depicts the comparison results for the state and control trajectories using the parameters listed in Tables 2 and 3. Moreover, the state and optimal control trajectories are displayed in the figures for the changes in state and optimal control trajectories over iterations with the legend of “Initial” in Section 5.3. For Figure 4, it is observed that the state and control differences are less than  $5 \times 10^{-5}$  for both cases. That is, it is confirmed that there is not much difference in terms of the solution trajectories between the conventional and proposed approaches. In addition, it is proved that the proposed approach is computationally more efficient than the conventional approach because, during 1000 simulations, the average computational time using the closed-form solutions is reduced by half compared to the one using the conventional approach, as listed in Table 4. The improvement in the computational speed for solving the LQR problem can contribute to the increase in the control frequency. In fact, the reduction in the computational time is obvious because the proposed approach does not require any numerical integration process. As the dimension of the state and control increases, the computational efficiency also increases exponentially. In addition, the closed-form solution provides a more accurate result compared to the conventional one because the closed-form solution does not contain numerical errors that occurred from numerical integration process.

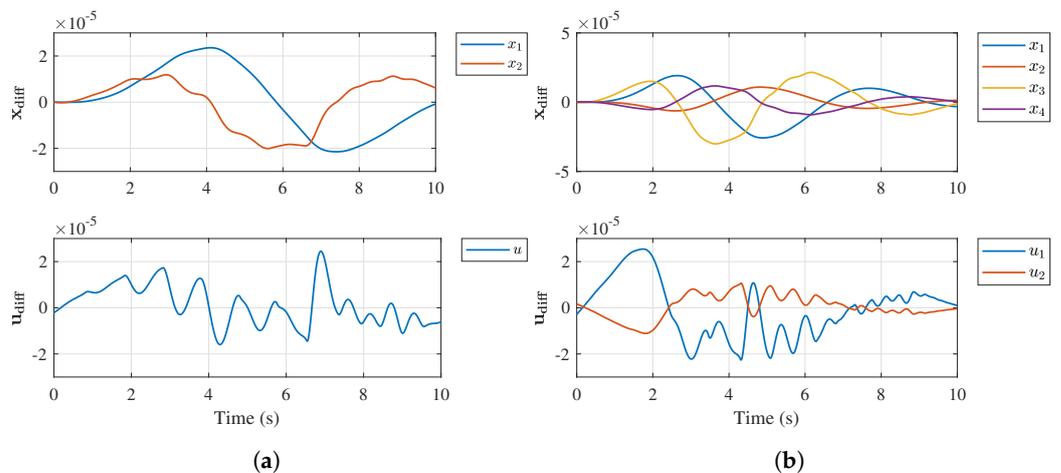


Figure 4. State and control trajectories difference between the closed-form solutions and conventional approach. (a) 1 DOF. (b) 2 DOFs.

**Table 4.** Comparison of average computational time.

	1 DOF	2 DOFs
Conventional: numerical integrations (s)	0.118	0.129
Proposed: closed-form algebraic equations (s)	0.057	0.060

### 5.3. Weight Matrices Optimization

To validate the performance of the proposed optimization approach with the developed closed-form solutions for the partial derivatives and principal equations, the weight matrices given in Table 3 are optimized. The optimization process aims for finding the optimal weight matrices that make the state and the control input at the final time close to zero while minimizing the corrections for the weight matrices. The requirement is set to be the norm of the augmented state at the final time that is less than  $10^{-5}$  (i.e.,  $\epsilon = 10^{-5}$ ), and the maximum number of iterations is 100. With these requirements, two example problems are considered, and the weight matrices listed in Table 3 are used as the initial weight matrices.

Table 5 shows the results obtained using the initial and the optimized weight matrices for the one-DOF example problem. The norm of the augmented state at the final time using initial weight matrices does not meet the requirement, requiring a tuning process for the weight matrices. After applying the optimization process proposed with the closed-form solutions, the norm of the augmented state of  $3.82 \times 10^{-6}$  that satisfies the requirement is obtained. The history of the norm of the augmented state versus the performance index for the one-DOF example is shown in Figure 5, and it is shown that both values evaluated are decreased during optimization from the initial evaluation using the initial weight matrices. Moreover, as shown in Figures 6–8, the proposed approach optimizes the off-diagonal elements of the weight matrices as well as the diagonal elements although the identity matrices are used as the initial conditions. Note that the elements in the green-dotted boxes are the same as the elements on the other side because the weight matrices are symmetric. The proposed optimization approach provides more flexibility compared to existing studies because the proposed one optimizes all components of the weight matrices. In addition, since the proposed optimization process contains the step that checks the definiteness conditions of the obtained matrices at every iteration, one guarantees that all weight matrices updated are positive (semi-)definite. The state and control trajectories for all iterations are shown in Figures 9 and 10. The responses of the state and control input become faster over the iterations to satisfy the requirement. Hence, the trajectories with blue color, which use the optimized weight matrices, are converged to the target states faster than the others.

**Table 5.** Optimization results for the 1-DOF problem.

	Initial	Optimized
Norm of the augmented state $\ y_f\ $	$2.68 \times 10^{-2}$	$3.82 \times 10^{-6}$
States at the final time $x(t_f)$	$[1.20 \ -1.70]^T \times 10^{-2}$	$[3.28 \ -1.94]^T \times 10^{-6}$
Control at the final time $u(t_f)$	$1.70 \times 10^{-2}$	$-2.71 \times 10^{-7}$
State weight matrix $Q$	$I_{2 \times 2}$	$\begin{bmatrix} 1.63 & 0.08 \\ 0.08 & 1.02 \end{bmatrix}$
Control weight matrix $R$	1	0.19
Terminal state weight matrix $S_f$	$I_{2 \times 2}$	$\begin{bmatrix} 1.08 & 0.31 \\ 0.31 & 0.50 \end{bmatrix}$
Performance index $\mathcal{L}$	$2.01 \times 10^2$	$1.58 \times 10^2$

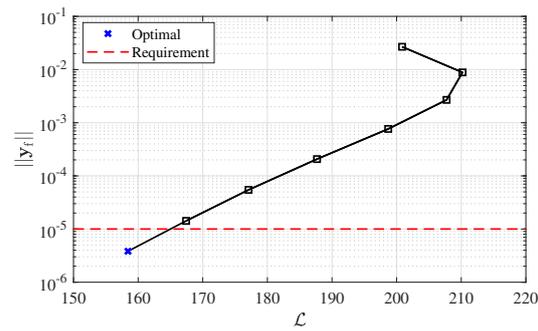


Figure 5. History of the norm of the augmented state and the performance index (1 DOF).

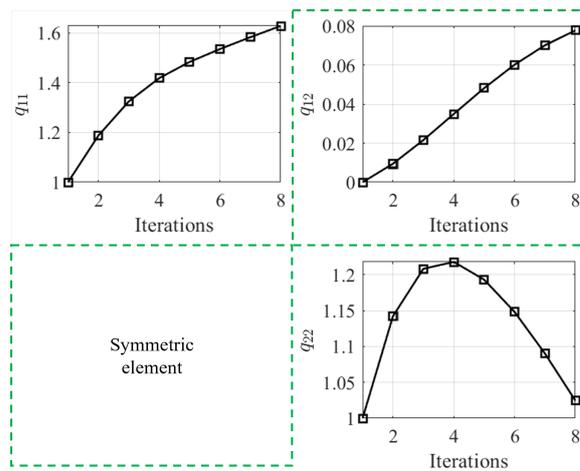


Figure 6. History of symmetric components of  $Q$  (1 DOF).

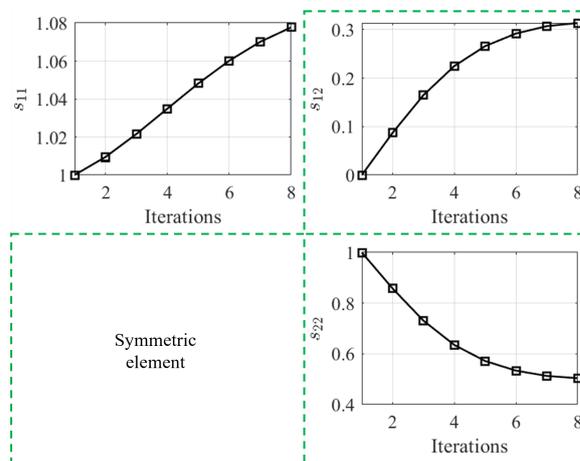


Figure 7. History of symmetric components of  $S_f$  (1 DOF).

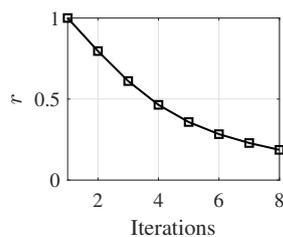


Figure 8. History of symmetric components of  $R$  (1 DOF).

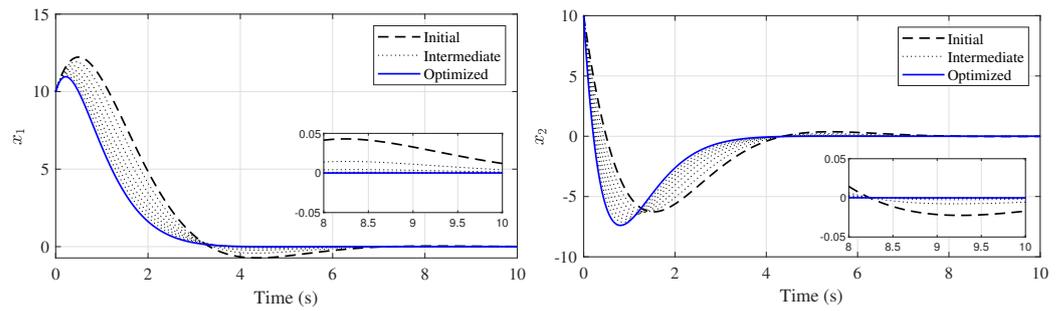


Figure 9. Changes in state trajectories over iterations (1 DOF).

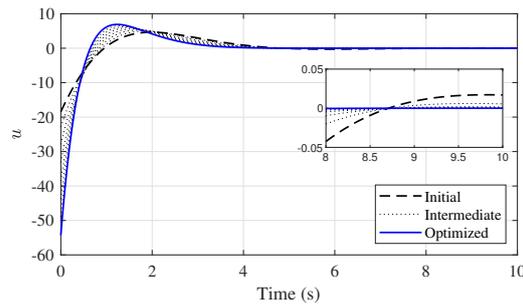


Figure 10. Changes of optimal control trajectories over iterations (1 DOF).

The optimization results for the two-DOFs example problem are tabulated in Table 6 and displayed in Figures 11–16. The proposed approach successfully finds the weight matrices with  $\|y_f\|$  of  $7.71 \times 10^{-6}$  that satisfy the predefined requirement as shown in Table 6. The history of the norm of the augmented state and the performance index over iterations is depicted in Figure 11, and the result shows that both values continuously decrease until the requirement is satisfied. The optimization results for the weight matrices are shown in Table 6, and their histories during optimization are shown in Figures 12–14. It is shown that all symmetric components including the off-diagonal terms are optimized to find the optimal solution. In addition, all weight matrices over iterations are surely positive (semi-)definite because the proposed approach updates the matrix that meets the definiteness conditions. As shown in Figures 15 and 16, the controlled states obtained using the optimized weight matrices are converged into zero values faster compared to the states obtained using the initial weight matrices in order to satisfy the requirements.

For the two cases, even though the identity matrices are used as the initial conditions, the optimal weight matrices that satisfy the predefined requirement are obtained within 10 iterations. Over the iterations, the updated matrices do not violate the definiteness conditions while optimizing all symmetric components of the weight matrices using the developed closed-form solutions.

Moreover, additional simulations using different initial conditions (weight matrices) for better understanding are displayed, and the optimization results are discussed in Appendix D.

Table 6. Optimization results for the 2-DOFs problem.

	Initial	Optimized
Norm of the augmented state $\ y_f\ $	$2.04 \times 10^{-2}$	$7.71 \times 10^{-6}$
States at the final time $x(t_f)$	$\begin{bmatrix} 1.94 \times 10^{-2} \\ -5.26 \times 10^{-3} \\ 2.48 \times 10^{-3} \\ -6.43 \times 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 4.29 \\ -2.36 \\ 1.58 \\ 1.32 \end{bmatrix} \times 10^{-6}$

Table 6. Cont.

	Initial	Optimized
Control at the final time $\mathbf{u}(t_f)$	$\begin{bmatrix} -2.48 \\ 0.64 \end{bmatrix} \times 10^{-3}$	$\begin{bmatrix} -4.84 \\ -2.80 \end{bmatrix} \times 10^{-6}$
State weight matrix $Q$	$I_{4 \times 4}$	$\begin{bmatrix} 1.54 & -0.07 & 0.17 & -0.04 \\ -0.07 & 1.56 & 0.14 & -0.03 \\ 0.17 & 0.14 & 0.75 & 0.26 \\ -0.04 & -0.03 & 0.26 & 0.82 \end{bmatrix}$
Control weight matrix $R$	$I_{2 \times 2}$	$\begin{bmatrix} 0.14 & 0.03 \\ 0.03 & 0.26 \end{bmatrix}$
Terminal state weight matrix $S_f$	$I_{4 \times 4}$	$\begin{bmatrix} 1.17 & 0.10 & -0.15 & -0.07 \\ 0.10 & 0.97 & 0.06 & 0.02 \\ -0.15 & 0.06 & 0.97 & -0.03 \\ -0.07 & 0.02 & -0.03 & 0.99 \end{bmatrix}$
Performance index $\mathcal{L}$	$2.01 \times 10^2$	$1.58 \times 10^2$

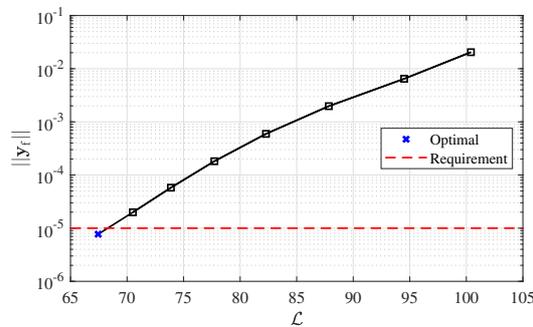


Figure 11. History of the norm of the augmented state and the performance index (2 DOFs).

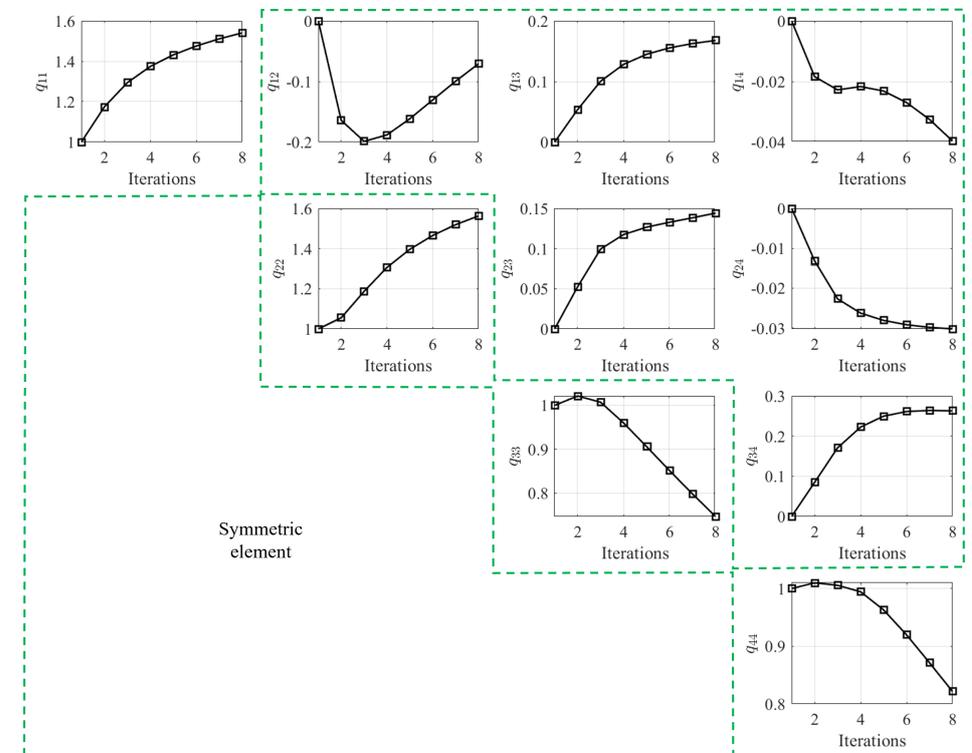


Figure 12. History of symmetric components of  $Q$  (2 DOFs).

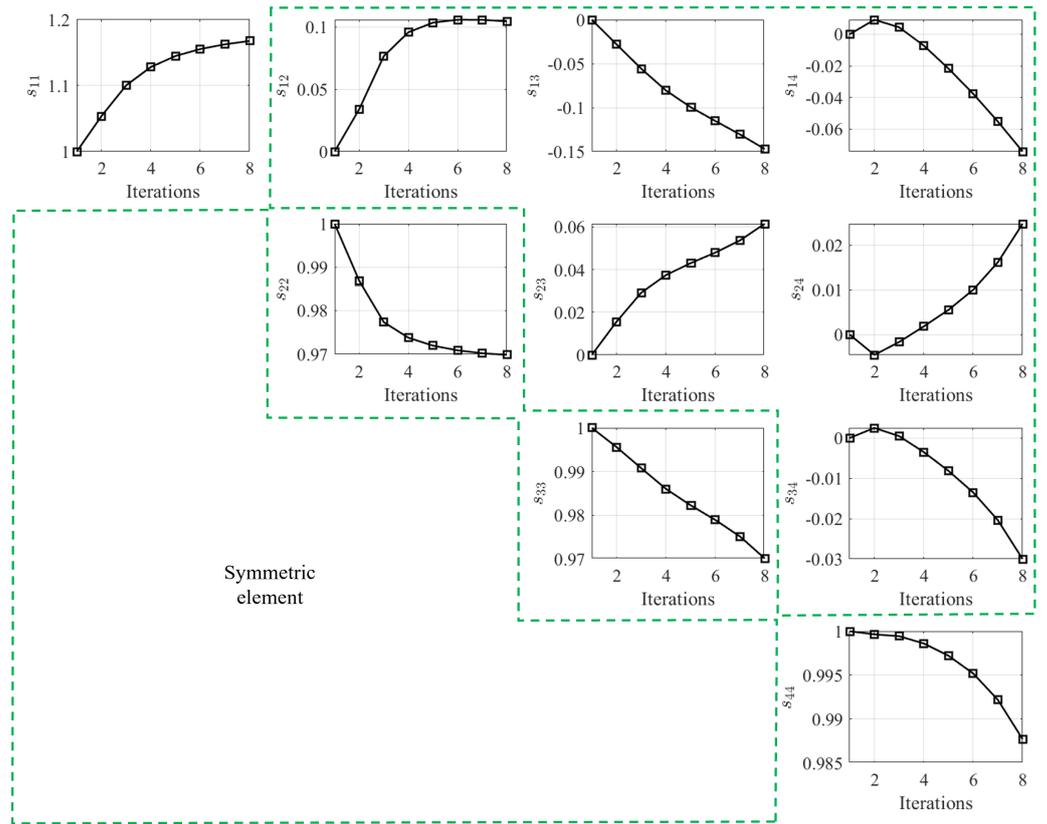


Figure 13. History of symmetric components of  $S_f$  (2 DOFs).

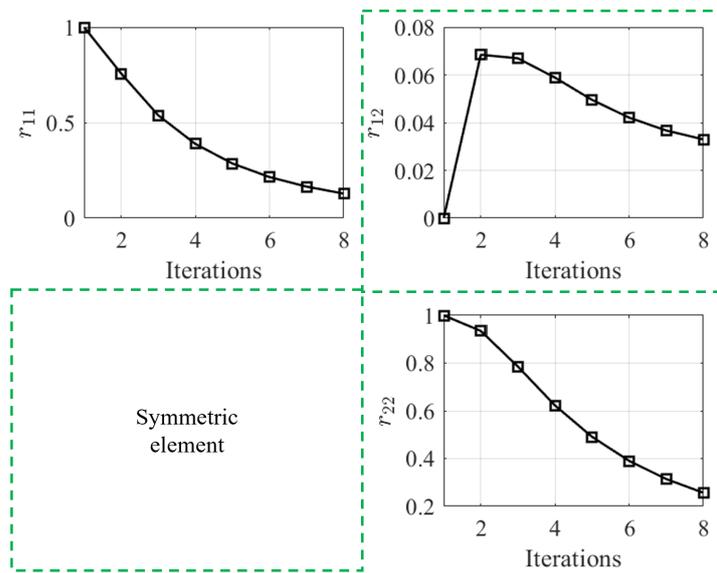


Figure 14. History of symmetric components of  $R$  (2 DOFs).

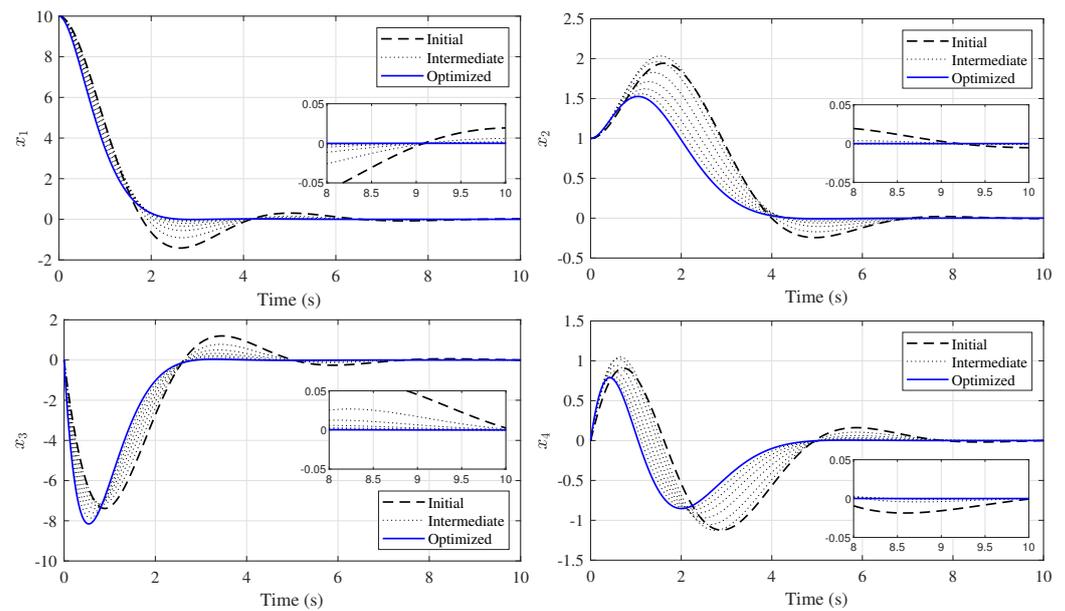


Figure 15. Changes in state trajectories over iterations (2 DOFs).

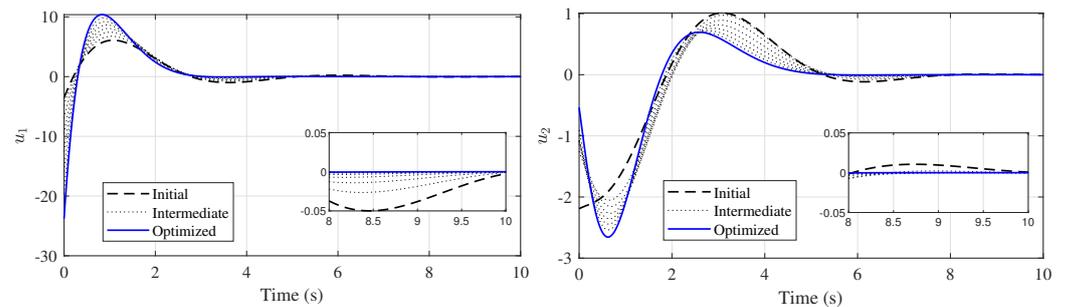


Figure 16. Changes in optimal control trajectories over iterations (2 DOFs).

## 6. Conclusions

This study proposes a gradient-based optimization approach to determine the symmetric components of the weight matrices for the linear quadratic regulator problem by applying Taylor's expansion to the state and control at the final time and finding the minimum norm optimization solution. To prevent the increase in the computational burden that arises from the increase in state dimensions and multiple numerical integration steps in the optimization process, this work develops the algebraic equations that exploit the closed-form solutions for the principal equations and their partial derivatives. Through numerical simulations for the one- and two-degrees-of-freedom second-order dynamic systems, it is validated that the proposed optimization approach finds all symmetric elements of the state, control, and terminal state weight matrices that satisfy the requirement (the norm of the augmented state of less than  $10^{-5}$ ) without violating the definiteness condition. Moreover, it is confirmed that the use of closed-form solutions reduces the computation time by half compared to the use of numerical integrations. In the future, a linear time-varying system model will be considered to extend the applicable systems.

**Author Contributions:** Conceptualization, D.K. and J.D.T.; methodology, D.C., D.K. and J.D.T.; software, D.C. and D.K.; validation, D.C., D.K. and J.D.T.; formal analysis, D.C., D.K. and J.D.T.; investigation, D.C. and D.K.; resources, D.K.; data curation, D.C. and D.K.; writing—original draft preparation, D.C.; writing—review and editing, D.K. and J.D.T.; visualization, D.C.; supervision, D.K.; project administration, D.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

$A \in \mathbb{R}^{n \times n}$	System dynamics matrix
$B \in \mathbb{R}^{n \times m}$	Control influence matrix
$Q \in \mathbb{R}^{n \times n}$	$= Q^T \geq 0$ . State weight matrix
$R \in \mathbb{R}^{m \times m}$	$= R^T > 0$ . Control weight matrix
$S(t_f) \in \mathbb{R}^{n \times n}$	$= S_f = S_f^T \geq 0$ . Terminal state weight matrix
$S(t) \in \mathbb{R}^{n \times n}$	Time-varying Riccati matrix
$J \in \mathbb{R}^{(n+m) \times P}$	Global Jacobian matrix
$S_{ss} \in \mathbb{R}^{n \times n}$	Steady-state Riccati matrix
$Z(t) \in \mathbb{R}^{n \times n}$	Time-varying term of Riccati matrix
$\bar{A} \in \mathbb{R}^{n \times n}$	System stability matrix
$\Phi(t, t_0) \in \mathbb{R}^{n \times n}$	State transition matrix
$\mathbf{x}(t) \in \mathbb{R}^n$	State vector
$\mathbf{u}(t) \in \mathbb{R}^m$	Control input vector
$\mathbf{y}(t) \in \mathbb{R}^{n+m}$	Augmented state vector
$d\mathbf{w} \in \mathbb{R}^P$	Correction vector
$\lambda \in \mathbb{R}^{n+m}$	Lagrange multiplier
$\mathbf{w} \in \mathbb{R}^P$	Vector composed of symmetric elements of weight matrices
$t_0 \in \mathbb{R}$	Initial time
$t_f \in \mathbb{R}$	Terminal time
$q_k \in \mathbb{R}$	Symmetric element of $Q$
$s_k \in \mathbb{R}$	Symmetric element of $S_f$
$r_l \in \mathbb{R}$	Symmetric element of $R$
$\beta_{,\alpha}$	$= \frac{\partial \beta}{\partial \alpha}$ . Partial derivative of an arbitrary variable $\beta$ with respect to an arbitrary variable $\alpha$
$N \in \mathbb{R}$	$= n(n+1)/2$ . Number of symmetric elements of $Q$ and $S_f$
$M \in \mathbb{R}$	$= m(m+1)/2$ . Number of symmetric elements of $R$
$P \in \mathbb{R}$	$= 2N + M$ . Total number of symmetric elements of weight matrices
$\mathcal{L} \in \mathbb{R}$	Performance index
$n \in \mathbb{R}$	Dimension of states
$m \in \mathbb{R}$	Dimension of control input
$Z_{ss} \in \mathbb{R}^{n \times n}$	Steady-state term of $Z(t)$
$Z_b \in \mathbb{R}^{n \times n}$	Boundary condition of $Z(t)$

## Appendix A. Proof of Closed-Loop Control System Stability

Using the obtained optimal control in Equation (3), the closed-loop system is described as

$$\dot{\mathbf{x}}(t) = (A - BR^{-1}B^T S(t))\mathbf{x}(t), \quad (\text{A1})$$

where

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

To prove stability, the positive Lyapunov function is chosen as

$$V = \mathbf{x}(t)^T S(t)\mathbf{x}(t) \geq 0. \quad (\text{A2})$$

Note that  $S(t)$  is a positive semi-definite matrix. The time derivative of the Lyapunov function is derived as

$$\dot{V} = \dot{\mathbf{x}}(t)^T S(t)\mathbf{x}(t) + \mathbf{x}(t)^T \dot{S}(t)\mathbf{x}(t) + \mathbf{x}(t)^T S(t)\dot{\mathbf{x}}(t). \quad (\text{A3})$$

Substituting Equations (A1) and (4) into Equation (A3) yields

$$\begin{aligned} \dot{V} = & \mathbf{x}(t)^T(A - BR^{-1}B^T S(t))^T S(t)\mathbf{x}(t) \\ & + \mathbf{x}(t)^T(-S(t)A - A^T S(t) + S(t)BR^{-1}B^T S(t) - Q)\mathbf{x}(t) \\ & + \mathbf{x}(t)^T S(t)(A - BR^{-1}B^T S(t))\mathbf{x}(t). \end{aligned} \tag{A4}$$

Rearranging Equation (A4) and applying the fact of  $S(t) = S(t)^T$  and  $R = R^T$  results in

$$\dot{V} = -\mathbf{x}(t)^T(Q + S(t)BR^{-1}S(t))\mathbf{x}(t) \leq 0. \tag{A5}$$

It is worth noting that the weight matrices  $Q$  and  $S(t)$  are positive semi-definite and  $R$  is positive definite. Therefore, the given closed-loop control system is stable. In addition, the definiteness conditions for each weight matrix are maintained during the proposed optimization process as discussed in Section 3 because the weight matrices are only updated when satisfying the definiteness conditions. For this reason, one can confirm that the closed-loop control system with the optimized weight matrices is stable.

### Appendix B. Closed-Form Solutions for the Principal Equations

This section describes the derivation process for obtaining the closed-form solutions for the time-varying Riccati matrix and the controlled state.

#### Appendix B.1. Time-Varying Riccati Matrix

Substituting (23) and its derivative into (4) and omitting  $t$  for notation simplification yields

$$\begin{aligned} \dot{S}_{ss} + \dot{Z}^{-1} = & -(S_{ss} + Z^{-1})A - A^T(S_{ss} + Z^{-1}) \\ & + (S_{ss} + Z^{-1})BR^{-1}B^T(S_{ss} + Z^{-1}) - Q. \end{aligned} \tag{A6}$$

Rearranging (A6) by applying the fact of  $\dot{S}_{ss} = 0$  leads to

$$\begin{aligned} \dot{Z}^{-1} = & -S_{ss}A - Z^{-1}A - A^T S_{ss} - A^T Z^{-1} + S_{ss}BR^{-1}B^T S_{ss} + Z^{-1}BR^{-1}B^T S_{ss} \\ & + S_{ss}BR^{-1}B^T Z^{-1} + Z^{-1}BR^{-1}B^T Z^{-1} - Q. \end{aligned} \tag{A7}$$

Using (24), (A7) reduces to

$$\dot{Z}^{-1} = -Z^{-1}A - A^T Z^{-1} + Z^{-1}BR^{-1}B^T S_{ss} + S_{ss}BR^{-1}B^T Z^{-1} + Z^{-1}BR^{-1}B^T Z^{-1}, \tag{A8}$$

and rearranging (A8) leads to

$$\dot{Z}^{-1} = -Z^{-1}(A - BR^{-1}B^T S_{ss}) - (A^T - S_{ss}BR^{-1}B^T)Z^{-1} + Z^{-1}BR^{-1}B^T Z^{-1}. \tag{A9}$$

Using  $\dot{Z}^{-1} = -Z^{-1}\dot{Z}Z^{-1}$  and (26), (A9) is rewritten as

$$-Z^{-1}\dot{Z}Z^{-1} = -Z^{-1}\bar{A} - \bar{A}^T Z^{-1} + Z^{-1}BR^{-1}B^T Z^{-1}. \tag{A10}$$

Multiplying  $Z$  by both sides, the differential Lyapunov matrix equation is derived as

$$\dot{Z} = \bar{A}Z + Z\bar{A} - BR^{-1}B^T. \tag{A11}$$

Hence, the closed-form solution for  $Z(t)$  is given by [24]

$$Z(t) = Z_{ss} + e^{\bar{A}(t-t_f)}(Z(t_f) - Z_{ss})e^{\bar{A}^T(t-t_f)}, \tag{A12}$$

with  $Z(t_f) = (S_f - S_{ss})^{-1}$ .

Appendix B.2. State for the Closed-Loop Control System

One assumes that  $\mathbf{x}(t)$  is expressed as [21,26]

$$\mathbf{x}(t) = Z(t)\mathbf{r}(t), \tag{A13}$$

where  $Z(t)$  is given by (A12) and  $\mathbf{r}(t)$  is a vector function to be determined. Introducing (A13) and its derivative into the left-hand side of (31) and dropping  $t$  for notation simplification yield

$$\begin{aligned} \dot{\mathbf{x}} &= \dot{Z}\mathbf{r} + Z\dot{\mathbf{r}}, \\ &= (\bar{A}Z + Z\bar{A}^T - BR^{-1}B^T)\mathbf{r} + Z\dot{\mathbf{r}}, \end{aligned} \tag{A14}$$

where  $\dot{Z}$  is replaced by (A11). Manipulating (31) and (A14) leads to

$$(\bar{A}Z + Z\bar{A}^T - BR^{-1}B^T)\mathbf{r} + Z\dot{\mathbf{r}} = (\bar{A} - BR^{-1}B^TZ^{-1})Z\mathbf{r}, \tag{A15}$$

and it is further simplified as

$$Z(\dot{\mathbf{r}} + \bar{A}^T\mathbf{r}) = \mathbf{0}. \tag{A16}$$

To satisfy (A16) regardless of  $Z$ ,  $\dot{\mathbf{r}} + \bar{A}^T\mathbf{r} = \mathbf{0}$ . This leads to the following differential equation for  $\mathbf{r}$ :

$$\dot{\mathbf{r}}(t) = -\bar{A}^T\mathbf{r}(t), \tag{A17}$$

where the initial condition is  $\mathbf{r}(t_0) = Z^{-1}(t_0)\mathbf{x}_0$ . Then, the solution for  $\mathbf{r}(t)$  follows as

$$\mathbf{r}(t) = e^{-\bar{A}^T(t-t_0)}\mathbf{r}(t_0). \tag{A18}$$

Substituting (A18) into (A13) yields the closed-form solution of the state as follows:

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x}_0,$$

where  $\Phi(t, t_0)$  denotes the state transition matrix, which is defined as

$$\Phi(t, t_0) = Z(t)e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0).$$

Appendix C. Sensitivity Partial Derivatives with Respect to Each Symmetric Component

Partial derivative models are required for  $S_{ss}$ ,  $\bar{A}$ ,  $Z_{ss}$ ,  $Z_b$ ,  $Z(t)$ ,  $\Phi(t, t_0)$ , and  $\mathbf{u}(t)$  to perform the optimization process proposed. The free variables in the calculation consist of the symmetric elements of  $Q$ ,  $R$ , and  $S_f$ . Closed-form algebraic equations are developed for all partial derivatives with respect to the symmetric elements of each weight matrix. To define the partial derivatives, this work defines a single-entry matrix  $(\cdot)^{ij}$  that all entries are equal to zero except for the entry of  $i$ -th row and  $j$ -th column, which is 1. For example, when each element of  $Q$  is defined as  $q_{ij}$ , that is, the element in  $i$ -th row and  $j$ -th column, the partial derivative of  $Q$  with respect to  $q_{ij}$  is expressed as  $Q^{ij}$ . This is applied to the other weight matrices.

Appendix C.1. Partial Derivatives for the Steady-State Riccati Matrix  $S_{ss} = S_{ss}(Q, R)$

Differentiating (24) with respect to  $w_p$  yields

$$\begin{aligned} 0 &= -S_{ss/w_p}A - A^T S_{ss/w_p} + S_{ss/w_p}BR^{-1}B^T S_{ss} \\ &\quad - S_{ss}BR^{-1}R/w_p R^{-1}B^T S_{ss} + S_{ss}BR^{-1}B^T S_{ss/w_p} - Q/w_p, \end{aligned} \tag{A19}$$

and it is rewritten as

$$0 = -S_{SS,rw_p} (A - BR^{-1}B^T S_{SS}) - (A^T - S_{SS}BR^{-1}B^T)S_{SS,rw_p} - S_{SS}BR^{-1}R_{,rw_p} R^{-1}B^T S_{SS} - Q_{,rw_p}. \tag{A20}$$

Using (26), the partial derivatives for the steady-state Riccati matrix equation in (43) are derived as

$$\bar{A}^T S_{SS,rw_p} + S_{SS,rw_p} \bar{A} = -S_{SS}BR^{-1}R_{,rw_p} R^{-1}B^T S_{SS} - Q_{,rw_p}.$$

Appendix C.1.1. Partial Derivatives with Respect to  $q_{ij}$

The partial derivatives of  $S_{SS}$  with respect to  $q_{ij}$  are defined by

$$\bar{A}^T S_{SS,q_{ij}} + S_{SS,q_{ij}} \bar{A} = -Q^{ij}, \tag{A21}$$

where  $Q^{ij}$  is the single-entry matrix that is the partial derivative of  $Q$  with respect to the symmetric element  $q_{ij}$ , and the partial derivatives are computed over  $N$  number of components.

Appendix C.1.2. Partial Derivatives with Respect to  $r_{ij}$

The partial derivatives of  $S_{SS}$  with respect to  $r_{ij}$  are defined by

$$\bar{A}^T S_{SS,r_{ij}} + S_{SS,r_{ij}} \bar{A} = -S_{SS}BR^{-1}R^{ij}R^{-1}BS_{SS}, \tag{A22}$$

where  $R^{ij}$  is the single-entry matrix that is the partial derivative of  $R$  with respect to the symmetric element  $r_{ij}$ , and the partial derivatives are computed over  $M$  number of components.

Appendix C.1.3. Partial Derivatives with Respect to  $s_{ij}$

The partial derivatives of  $S_{SS}$  with respect to  $s_{ij}$  are defined by

$$S_{SS,s_{ij}} = 0. \tag{A23}$$

No other partial derivatives exist for the steady-state Riccati matrix because  $S_{SS}$  is the function of  $Q$  and  $R$ .

Appendix C.2. Partial Derivatives for the Closed-Loop System Dynamics Matrix  $\bar{A} = \bar{A}(Q, R)$

The partial derivatives for the closed-loop system dynamics matrix are given in (47) as follows:

$$\bar{A}_{,rw_p} = BR^{-1}R_{,rw_p} R^{-1}B^T S_{SS} - BR^{-1}B^T S_{SS,rw_p}.$$

Appendix C.2.1. Partial Derivatives with Respect to  $q_{ij}$

The partial derivatives of  $\bar{A}$  with respect to  $q_{ij}$  are defined by

$$\bar{A}_{,q_{ij}} = -BR^{-1}B^T S_{SS,q_{ij}}, \tag{A24}$$

yielding  $N$  matrix partial derivative calculations.

Appendix C.2.2. Partial Derivatives with Respect to  $r_{ij}$

The partial derivatives of  $\bar{A}$  with respect to  $r_{ij}$  are defined by

$$\bar{A}_{,r_{ij}} = BR^{-1}R^{ij}R^{-1}B^T S_{SS} - BR^{-1}B^T S_{SS,r_{ij}}. \tag{A25}$$

The partial derivatives are computed over  $M$  number of components.

#### Appendix C.2.3. Partials with Respect to $s_{ij}$

The partial derivatives of  $\bar{A}$  with respect to  $s_{ij}$  are defined by

$$\bar{A}_{,s_{ij}} = 0. \quad (\text{A26})$$

No additional  $\bar{A}$  partials exist because  $\bar{A}$  is not the function of  $S_f$ .

#### Appendix C.3. Partial Derivatives for the Steady-State Lyapunov Dynamics Matrix

$Z_{ss} = Z_{ss}(Q, R)$

The partial derivatives for the steady-state Lyapunov dynamics matrix are given in (46) as follows:

$$\bar{A}Z_{ss,rw_p} + Z_{ss,rw_p} \bar{A}^T = -\bar{A}_{,rw_p} Z_{ss} - Z_{ss} \bar{A}_{,rw_p}^T - BR^{-1}R_{,rw_p} R^{-1}B^T.$$

#### Appendix C.3.1. Partials with Respect to $q_{ij}$

The partial derivatives of  $Z_{ss}$  with respect to  $q_{ij}$  are defined by

$$\bar{A}Z_{ss,q_{ij}} + Z_{ss,q_{ij}} \bar{A}^T = -\bar{A}_{,q_{ij}} Z_{ss} - Z_{ss} \bar{A}_{,q_{ij}}^T, \quad (\text{A27})$$

where the partial derivatives are computed over  $N$  number of components.

#### Appendix C.3.2. Partials with Respect to $r_{ij}$

The partial derivatives of  $Z_{ss}$  with respect to  $r_{ij}$  are defined by the Lyapunov matrix equation:

$$\bar{A}Z_{ss,r_{ij}} + Z_{ss,r_{ij}} \bar{A}^T = -\bar{A}_{,r_{ij}} Z_{ss} - Z_{ss} \bar{A}_{,r_{ij}}^T - BR^{-1}R^{ij}R^{-1}B^T, \quad (\text{A28})$$

The partial derivative computations are performed over  $M$  number of components.

#### Appendix C.3.3. Partials with Respect to $s_{ij}$

The partial derivatives of  $Z_{ss}$  with respect to  $s_{ij}$  are defined by

$$Z_{ss,s_{ij}} = 0. \quad (\text{A29})$$

No additional  $Z_{ss}$  partials exist.

#### Appendix C.4. Partial Derivatives for the Boundary Condition $Z_b = Z_b(Q, R, S_f)$

The partial derivatives for the boundary condition are given in (45) as follows:

$$Z_{b,rw_p} = -(S_f - S_{ss})^{-1}(S_{f,rw_p} - S_{ss,rw_p})(S_f - S_{ss})^{-1} - Z_{ss,rw_p}.$$

#### Appendix C.4.1. Partials with Respect to $q_{ij}$

The partial derivatives of  $Z_b$  with respect to  $q_{ij}$  are expressed as

$$Z_{b,q_{ij}} = (S_f - S_{ss})^{-1}S_{ss,q_{ij}}(S_f - S_{ss})^{-1} - Z_{ss,q_{ij}}, \quad (\text{A30})$$

yielding  $N$  matrix partial derivative calculations.

#### Appendix C.4.2. Partials with Respect to $r_{ij}$

The partial derivatives of  $Z_b$  with respect to  $r_{ij}$  are given by

$$Z_{b,r_{ij}} = (S_f - S_{ss})^{-1}S_{ss,r_{ij}}(S_f - S_{ss})^{-1} - Z_{ss,r_{ij}}, \quad (\text{A31})$$

yielding  $M$  matrix partial derivative calculations.

#### Appendix C.4.3. Partials with Respect to $s_{ij}$

The partial derivatives of  $Z_b$  with respect to  $s_{ij}$  are given by

$$Z_{b,s_{ij}} = -(S_f - S_{ss})^{-1} S_f^{ij} (S_f - S_{ss})^{-1}. \quad (\text{A32})$$

The partial derivatives are computed over  $N$  number of components.

#### Appendix C.5. Partial Derivatives for the Time-Varying Part of the Riccati Matrix Equation

$$Z(t) = Z(t; Q, R, S_f)$$

The partial derivatives for the time-varying part of the Riccati matrix are given in (44) as follows:

$$\begin{aligned} Z_{,r_{wp}}(t) = & Z_{ss,r_{wp}} + \left[ e^{\bar{A}(t-t_f)} \right]_{,r_{wp}} Z_b e^{\bar{A}(t-t_f)} \\ & + e^{\bar{A}(t-t_f)} Z_{b,r_{wp}} e^{\bar{A}^T(t-t_f)} + e^{\bar{A}(t-t_f)} Z_b \left[ e^{\bar{A}^T(t-t_f)} \right]_{,r_{wp}}. \end{aligned}$$

#### Appendix C.5.1. Partials with Respect to $q_{ij}$

The partial derivatives of  $Z(t)$  with respect to  $q_{ij}$  are given by

$$\begin{aligned} Z_{,q_{ij}}(t) = & Z_{ss,q_{ij}} + \left[ e^{\bar{A}(t-t_f)} \right]_{,q_{ij}} Z_b e^{\bar{A}^T(t-t_f)} \\ & + e^{\bar{A}(t-t_f)} Z_{b,q_{ij}} e^{\bar{A}^T(t-t_f)} + e^{\bar{A}(t-t_f)} Z_b \left[ e^{\bar{A}^T(t-t_f)} \right]_{,q_{ij}}, \end{aligned} \quad (\text{A33})$$

where the matrix exponential partial derivative with respect to  $q_{ij}$  is computed as follows:

$$\left[ e^{\bar{A}^T(t-t_f)} \right]_{,q_{ij}} = \begin{bmatrix} I & 0 \end{bmatrix} \exp \left( \begin{bmatrix} \bar{A} & \bar{A}_{,q_{ij}} \\ 0 & \bar{A} \end{bmatrix} (t - t_f) \right) \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (\text{A34})$$

where  $\exp(\cdot)$  is the matrix exponential. The partial derivative is extracted as the upper right-hand side block of the  $2n \times 2n$  matrix exponential solution, yielding  $N$  matrix partial derivative calculations.

#### Appendix C.5.2. Partials with Respect to $r_{ij}$

The partial derivatives of  $Z(t)$  with respect to  $r_{ij}$  are given by

$$\begin{aligned} Z_{,r_{ij}}(t) = & Z_{ss,r_{ij}} + \left[ e^{\bar{A}(t-t_f)} \right]_{,r_{ij}} Z_b e^{\bar{A}^T(t-t_f)} \\ & + e^{\bar{A}(t-t_f)} Z_{b,r_{ij}} e^{\bar{A}^T(t-t_f)} + e^{\bar{A}(t-t_f)} Z_b \left[ e^{\bar{A}^T(t-t_f)} \right]_{,r_{ij}}, \end{aligned} \quad (\text{A35})$$

where the matrix exponential partial derivative with respect to  $r_{ij}$  is computed as follows:

$$\left[ e^{\bar{A}^T(t-t_f)} \right]_{,r_{ij}} = \begin{bmatrix} I & 0 \end{bmatrix} \exp \left( \begin{bmatrix} \bar{A} & \bar{A}_{,r_{ij}} \\ 0 & \bar{A} \end{bmatrix} (t - t_f) \right) \begin{bmatrix} 0 \\ I \end{bmatrix}. \quad (\text{A36})$$

This yields  $M$  matrix partial derivative calculations.

#### Appendix C.5.3. Partials with Respect to $s_{ij}$

The partial derivatives of  $Z(t)$  with respect to  $s_{ij}$  are given by

$$Z_{,s_{ij}}(t) = e^{\bar{A}(t-t_f)} Z_{b,s_{ij}} e^{\bar{A}^T(t-t_f)}, \quad (\text{A37})$$

yielding  $N$  matrix partial derivative calculations.

Appendix C.6. Partial Derivatives for the State Transition Matrix  $\Phi(t, t_0) = \Phi(t, t_0; Q, R, S_f)$

The partial derivatives for the state transition matrix are given in (49) as follows:

$$\begin{aligned} \Phi_{,r_{w_p}}(t, t_0) = & Z_{,r_{w_p}}(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0) + Z(t)\left[e^{-\bar{A}(t-t_0)}\right]_{,r_{w_p}}Z^{-1}(t_0) \\ & - Z(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0)Z_{,r_{w_p}}(t_0)Z^{-1}(t_0). \end{aligned}$$

Appendix C.6.1. Partials with Respect to  $q_{ij}$

The partial derivatives of  $\Phi(t, t_0)$  with respect to  $q_{ij}$  are expressed as

$$\begin{aligned} \Phi_{,q_{ij}}(t, t_0) = & Z_{,q_{ij}}(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0) + Z(t)\left[e^{-\bar{A}(t-t_0)}\right]_{,q_{ij}}Z^{-1}(t_0) \\ & - Z(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0)Z_{,q_{ij}}(t_0)Z^{-1}(t_0), \end{aligned} \tag{A38}$$

yielding  $N$  matrix partial derivative calculations.

Appendix C.6.2. Partials with Respect to  $r_{ij}$

The partial derivatives of  $\Phi(t, t_0)$  with respect to  $r_{ij}$  are given by

$$\begin{aligned} \Phi_{,r_{ij}}(t, t_0) = & Z_{,r_{ij}}(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0) + Z(t)\left[e^{-\bar{A}(t-t_0)}\right]_{,r_{ij}}Z^{-1}(t_0) \\ & - Z(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0)Z_{,r_{ij}}(t_0)Z^{-1}(t_0), \end{aligned} \tag{A39}$$

yielding  $M$  matrix partial derivative calculations.

Appendix C.6.3. Partials with Respect to  $s_{ij}$

The partial derivatives of  $\Phi(t, t_0)$  with respect to  $s_{ij}$  are expressed as

$$\Phi_{,s_{ij}}(t, t_0) = Z_{,s_{ij}}(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0) - Z(t)e^{-\bar{A}(t-t_0)}Z^{-1}(t_0)Z_{,s_{ij}}(t_0)Z^{-1}(t_0), \tag{A40}$$

yielding  $N$  matrix partial derivative calculations.

Appendix C.7. Partial Derivatives for the Control Trajectory  $\mathbf{u}(t) = \mathbf{u}(t; Q, R, S_f)$

The partial derivatives for the optimal control trajectory are given in (50) as follows:

$$\begin{aligned} \mathbf{u}_{,r_{w_p}}(t) = & R^{-1}R_{,r_{w_p}}R^{-1}B^T S_{ss}\Phi(t, t_0)\mathbf{x}_0 - R^{-1}B^T S_{ss,r_{w_p}}\Phi(t, t_0)\mathbf{x}_0 \\ & - R^{-1}B^T S_{ss}\Phi_{,r_{w_p}}(t, t_0)\mathbf{x}_0 + R^{-1}R_{,r_{w_p}}R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T \left[ e^{-\bar{A}^T(t-t_0)} \right]_{,r_{w_p}} Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)Z_{,r_{w_p}}(t_0)Z^{-1}(t_0)\mathbf{x}_0. \end{aligned}$$

Appendix C.7.1. Partials with Respect to  $q_{ij}$

The partial derivatives of  $\mathbf{u}(t)$  with respect to  $q_{ij}$  are defined by

$$\begin{aligned} \mathbf{u}_{,q_{ij}}(t) = & -R^{-1}B^T S_{ss,q_{ij}}\Phi(t, t_0)\mathbf{x}_0 - R^{-1}B^T S_{ss}\Phi_{,q_{ij}}(t, t_0)\mathbf{x}_0 \\ & - R^{-1}B^T \left[ e^{-\bar{A}^T(t-t_0)} \right]_{,q_{ij}} Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)Z_{,q_{ij}}(t_0)Z^{-1}(t_0)\mathbf{x}_0. \end{aligned} \tag{A41}$$

The partial derivatives are computed over  $N$  number of components.

Appendix C.7.2. Partial with Respect to  $r_{ij}$

The partial derivatives of  $\mathbf{u}(t)$  with respect to  $r_{ij}$  are defined by

$$\begin{aligned} \mathbf{u}_{,r_{ij}}(t) = & R^{-1}R^{ij}R^{-1}B^T S_{ss}\Phi(t, t_0)\mathbf{x}_0 - R^{-1}B^T S_{ss,r_{ij}}\Phi(t, t_0)\mathbf{x}_0 \\ & - R^{-1}B^T S_{ss}\Phi_{,r_{ij}}(t, t_0)\mathbf{x}_0 + R^{-1}R^{ij}R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T \left[ e^{-\bar{A}^T(t-t_0)} \right]_{,r_{ij}} Z^{-1}(t_0)\mathbf{x}_0 \\ & - R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)Z_{,r_{ij}}(t_0)Z^{-1}(t_0)\mathbf{x}_0, \end{aligned} \tag{A42}$$

yielding  $M$  matrix partial derivative calculations.

Appendix C.7.3. Partial with Respect to  $s_{ij}$

The partial derivatives of  $\mathbf{u}(t)$  with respect to  $s_{ij}$  are defined by

$$\begin{aligned} \mathbf{u}_{,s_{ij}}(t) = & -R^{-1}B^T S_{ss}\Phi_{,s_{ij}}(t, t_0)\mathbf{x}_0 \\ & - R^{-1}B^T e^{-\bar{A}^T(t-t_0)}Z^{-1}(t_0)Z_{,s_{ij}}(t_0)Z^{-1}(t_0)\mathbf{x}_0. \end{aligned} \tag{A43}$$

The partial derivatives are computed over  $N$  number of components.

Appendix D. Additional Simulation Study

On top of the simulation study described in Section 5, additional simulations were performed using different initial weight matrices for each example problem. That is, all simulation parameters remain the same, except for the state, control, and terminal state weight matrices listed in Table 3. New initial conditions considered are tabulated in Table A1.

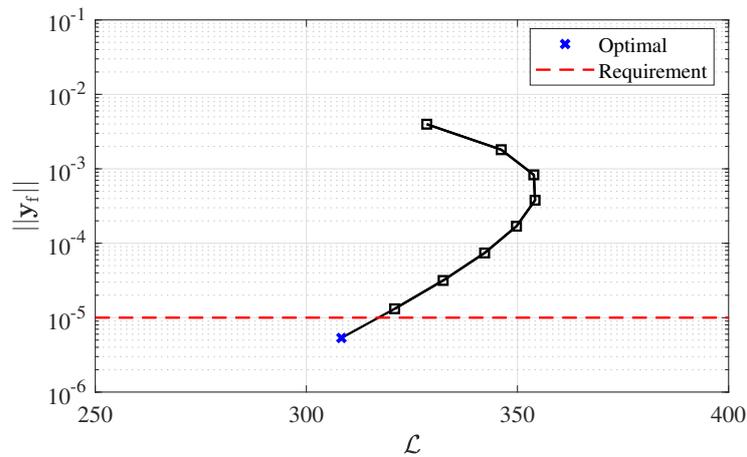
Table A1. Newly considered initial weight matrices.

Parameter	Value	
	1 DOF	2 DOFs
State weight matrix $Q$	$2I_{2 \times 2}$	$2I_{4 \times 4}$
Control weight matrix $R$	1	$5I_{2 \times 2}$
Terminal state weight matrix $S_f$	$4I_{2 \times 2}$	$2I_{4 \times 4}$

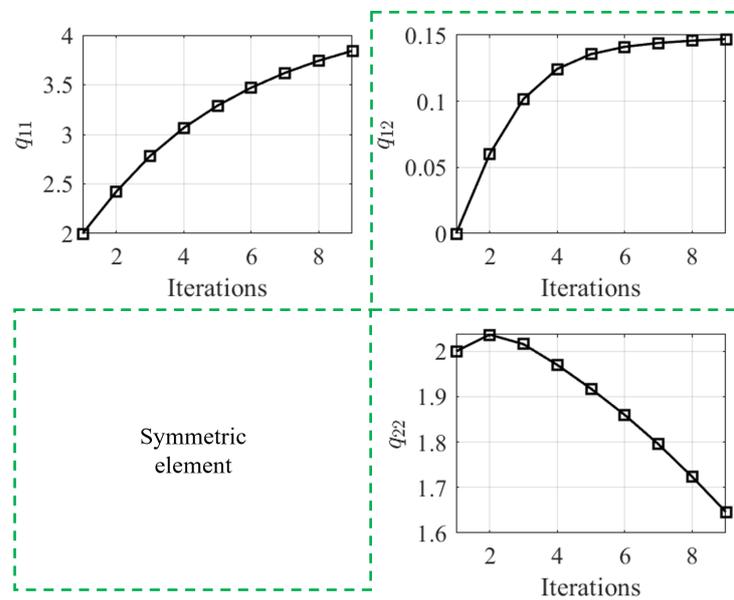
Table A2 displays the initial and optimized results for the one-DOF example problem. Although the norm of the augmented state at the final time using initial weight matrices does not meet the requirement, the norm of the augmented state of  $5.34 \times 10^{-6}$  that satisfies the requirement is obtained after the optimization with the closed-form solutions. Figure A1 depicts the history of the norm of the augmented state and the performance index for the one-DOF example, and it shows that both values are decreased during optimization from the initially evaluated values. Furthermore, from Figures A2–A4, although the history of the updated elements for the weight matrices is similar to the ones discussed in Section 5 and the norm of the augmented state satisfies the requirement, the performance index value is a larger value because this example starts from the weight matrices composed of larger values. From these results, one can say that the optimized weight matrices are highly dependent on the initial weight matrices, and there are multiple solutions that satisfy the requirement. The state and control trajectories for all iterations are shown in Figures A5 and A6. When the optimized weight matrices are utilized, the rate of convergence to a value close to zero becomes faster to meet the requirement.

**Table A2.** Optimization results for the 1-DOF problem with different weight matrices.

	Initial	Optimized
Norm of the augmented state $\ y_f\ $	$3.97 \times 10^{-3}$	$5.34 \times 10^{-6}$
States at the final time $x(t_f)$	$[1.15 \ -0.92]^T \times 10^{-3}$	$[0.55 \ -1.09]^T \times 10^{-6}$
Control at the final time $u(t_f)$	$3.69 \times 10^{-3}$	$5.20 \times 10^{-6}$
State weight matrix $Q$	$2I_{2 \times 2}$	$\begin{bmatrix} 3.84 & 0.15 \\ 0.15 & 1.65 \end{bmatrix}$
Control weight matrix $R$	1	0.28
Terminal state weight matrix $S_f$	$4I_{2 \times 2}$	$\begin{bmatrix} 4.15 & 1.35 \\ 1.35 & 2.01 \end{bmatrix}$
Performance index $\mathcal{L}$	$3.29 \times 10^2$	$3.08 \times 10^2$



**Figure A1.** History of the norm of the augmented state and the performance index (1 DOF).



**Figure A2.** History of symmetric components of  $Q$  (1 DOF).

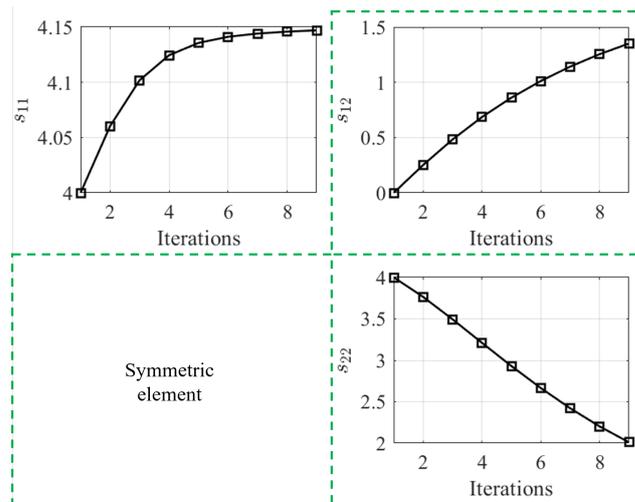


Figure A3. History of symmetric components of  $S_f$  (1 DOF).

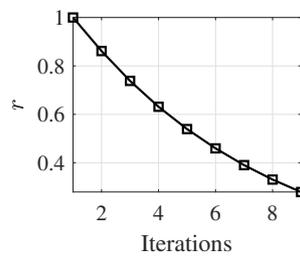


Figure A4. History of symmetric components of  $R$  (1 DOF).

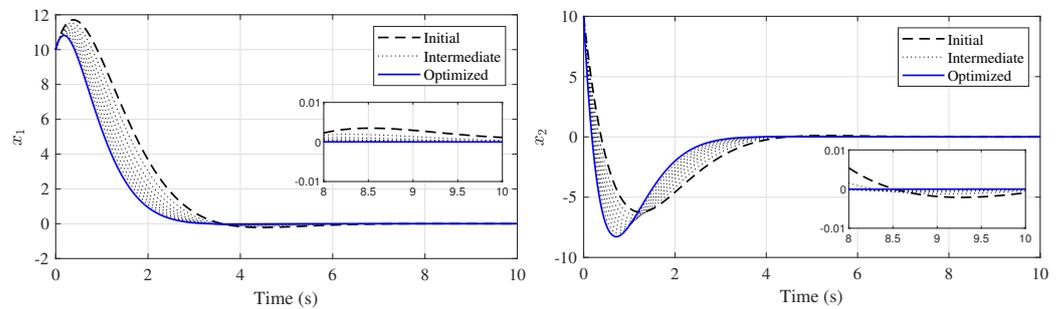


Figure A5. Changes in state trajectories over iterations (1 DOF).

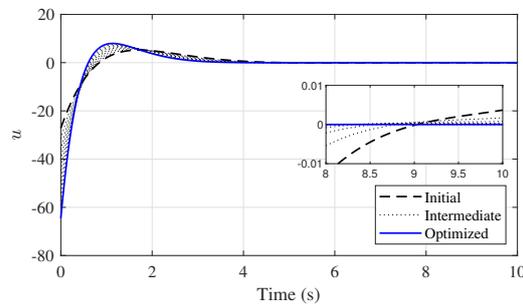


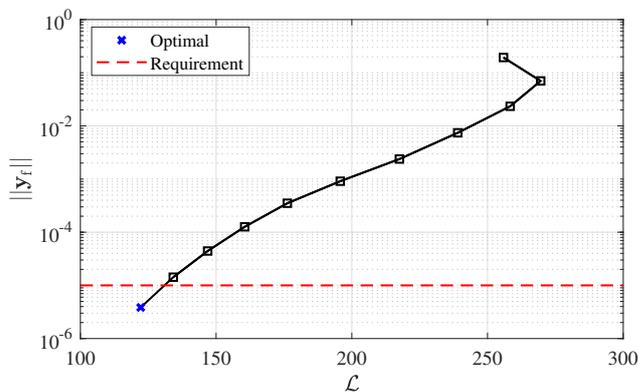
Figure A6. Changes in optimal control trajectory over iterations (1 DOF).

The optimization results for the two-DOFs example problem using different initial weight matrices are listed in Table A3 and displayed in Figures A7–A12. The proposed optimization process successfully finds the weight matrices that satisfy the requirement ( $\|y_f\| = 7.71 \times 10^{-6} < 10^{-5}$ ) as shown in Table A3. Similar to the other results, the history of the norm of the augmented state and the performance index over iterations has a decreasing trend as depicted in Figure A7. Figures A8–A10 illustrate the history of the updated

elements of the weight matrices. It is shown that the updated history for each component is different compared to the ones discussed in Section 5. However, the proposed optimization process finds the solution within 11 iterations without violating the definiteness condition. As shown in Figures A11 and A12, the controlled states obtained using the optimized weight matrices converge into near zero values faster compared to the ones using the initial weight matrices. Also, it is displayed that the states and control trajectories are different from the results displayed in Section 5 because the different initial guesses are used in the optimization process.

**Table A3.** Optimization results for the 2-DOFs problem with different weight matrices.

	Initial	Optimized
Norm of the augmented state $\ y_f\ $	$1.93 \times 10^{-1}$	$3.85 \times 10^{-6}$
States at the final time $x(t_f)$	$\begin{bmatrix} 1.82 \times 10^{-1} \\ -4.50 \times 10^{-2} \\ -3.61 \times 10^{-2} \\ 2.52 \times 10^{-2} \end{bmatrix}$	$\begin{bmatrix} 1.79 \times 10^{-6} \\ -9.53 \times 10^{-7} \\ -1.79 \times 10^{-6} \\ -6.19 \times 10^{-8} \end{bmatrix}$
Control at the final time $u(t_f)$	$\begin{bmatrix} 1.45 \\ -1.01 \end{bmatrix} \times 10^{-2}$	$\begin{bmatrix} 2.72 \times 10^{-6} \\ -2.82 \times 10^{-7} \end{bmatrix}$
State weight matrix $Q$	$2I_{4 \times 4}$	$\begin{bmatrix} 5.20 & 0.18 & 1.80 & 0.04 \\ 0.18 & 6.09 & 1.24 & -0.02 \\ 1.80 & 1.24 & 1.95 & 1.67 \\ 0.04 & -0.02 & 1.67 & 3.17 \end{bmatrix}$
Control weight matrix $R$	$5I_{2 \times 2}$	$\begin{bmatrix} 0.23 & 0.23 \\ 0.23 & 1.17 \end{bmatrix}$
Terminal state weight matrix $S_f$	$2I_{4 \times 4}$	$\begin{bmatrix} 3.80 & 1.28 & 0.83 & -0.05 \\ 1.28 & 1.98 & -0.27 & 0.22 \\ 0.83 & -0.27 & 1.30 & -0.07 \\ -0.05 & 0.22 & -0.07 & 1.90 \end{bmatrix}$
Performance index $\mathcal{L}$	$2.56 \times 10^2$	$1.22 \times 10^2$



**Figure A7.** History of the norm of the augmented state and the performance index (2 DOFs).

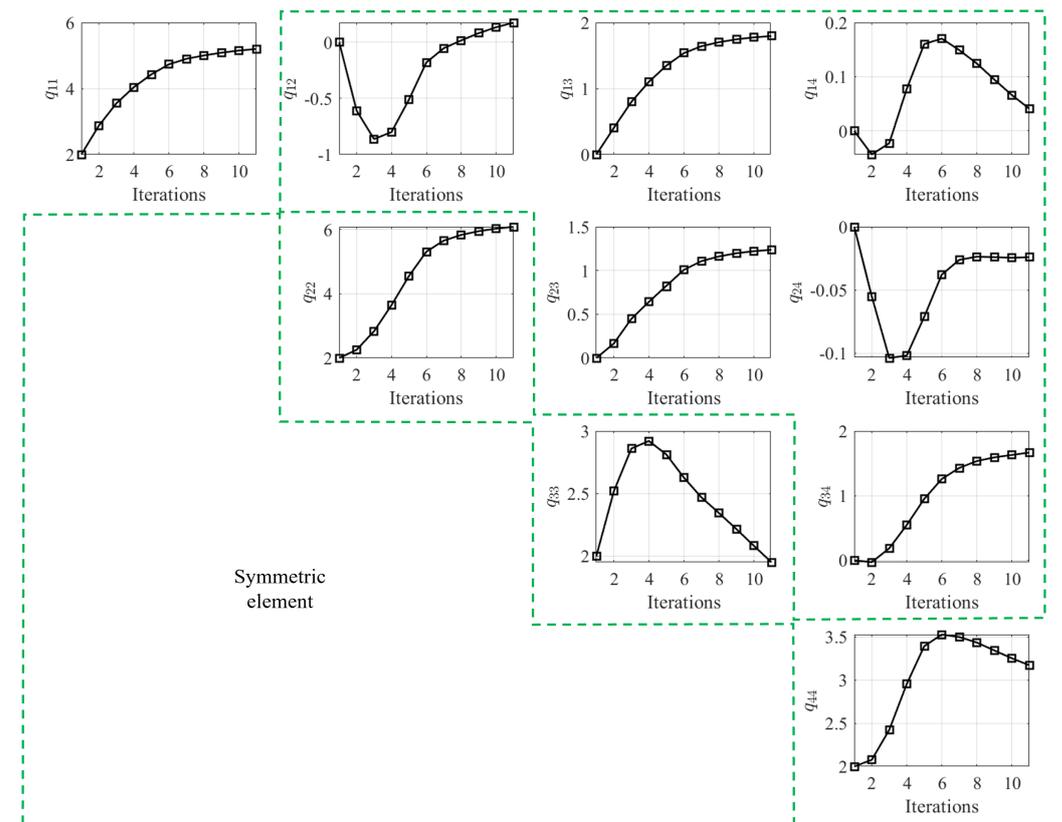


Figure A8. History of symmetric components of  $Q$  (2 DOFs).

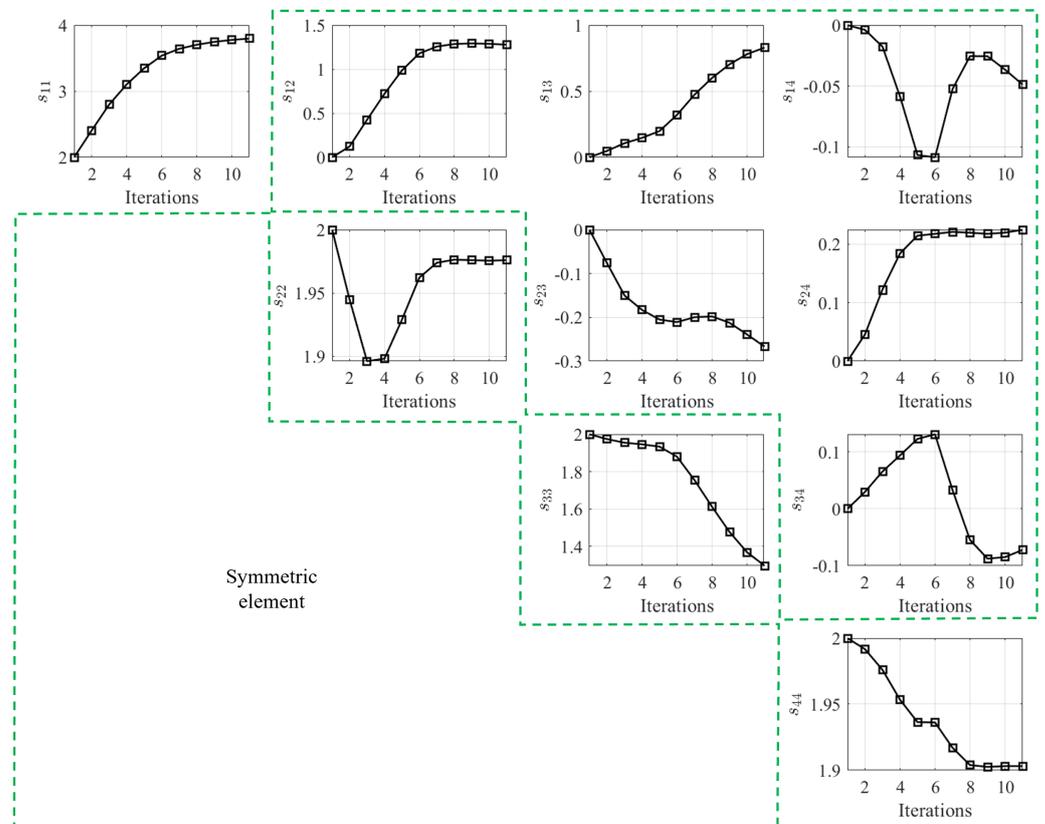


Figure A9. History of symmetric components of  $S_f$  (2 DOFs).

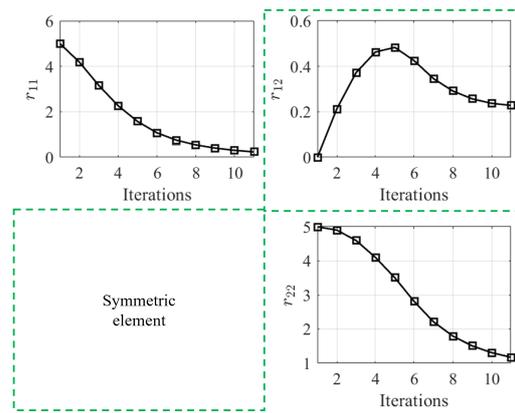


Figure A10. History of symmetric components of  $R$  (2 DOFs).

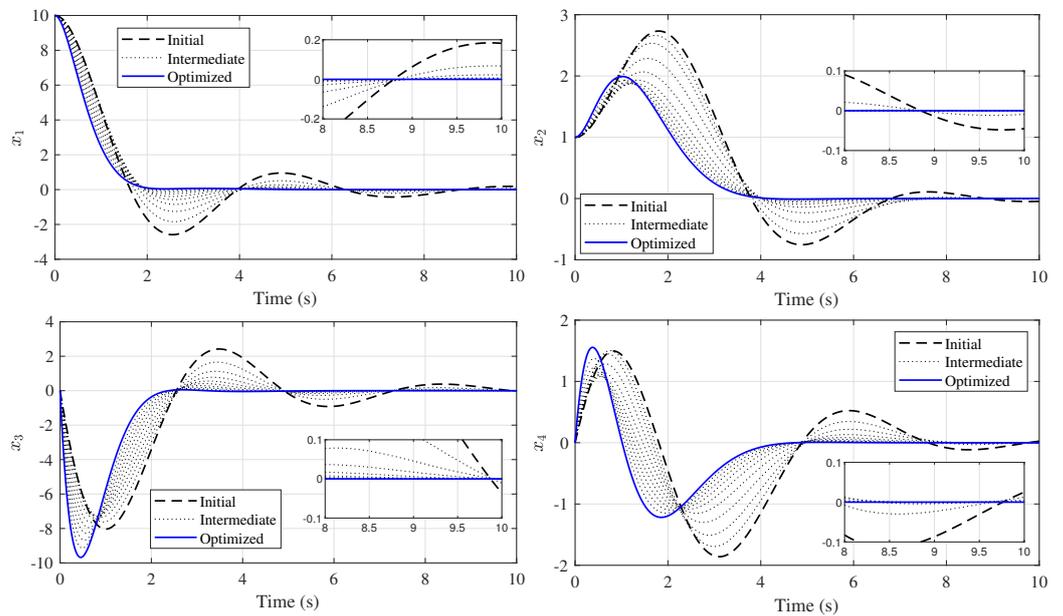


Figure A11. Changes of state trajectories over iterations (2 DOFs).

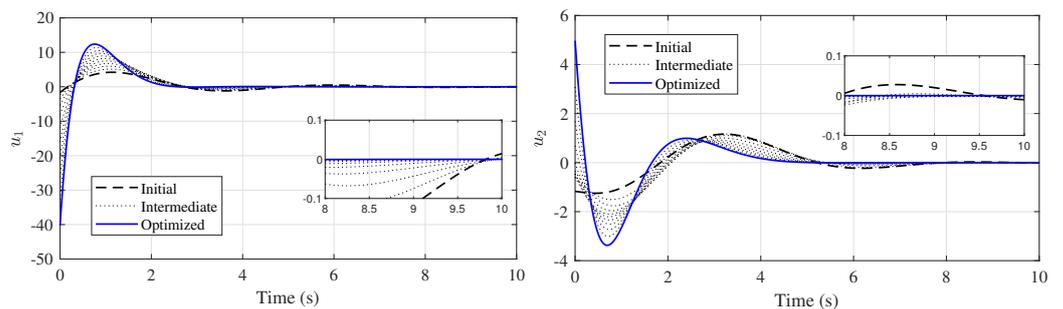


Figure A12. Changes of optimal control trajectories over iterations (2 DOFs).

References

1. Bryson, A.E.; Ho, Y.C. *Applied Optimal Control*; Blaisdell: New York, NY, USA, 1975.
2. Marada, T.; Matousek, R.; Zuth, D. Design of Linear Quadratic Regulator (LQR) Based on Genetic Algorithm for Inverted Pendulum. *MENDEL* **2017**, *23*, 149–156. [CrossRef]
3. Dhiman, V.; Singh, G.; Kumar, M. Modeling and Control of Underactuated System Using LQR Controller Based on GA. In *Proceedings of the Advances in Interdisciplinary Engineering*; Springer: Singapore, 2019; pp. 595–603. [CrossRef]
4. Yu, W.; Li, J.; Yuan, J.; Ji, X. LQR controller design of active suspension based on genetic algorithm. In *Proceedings of the 2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Xi’an, China, 15–17 October 2021; Volume 5, pp. 1056–1060. [CrossRef]

5. Kukreti, S.; Walker, A.; Putman, P.; Cohen, K. Genetic Algorithm Based LQR for Attitude Control of a Magnetically Actuated CubeSat. In Proceedings of the AIAA Infotech @ Aerospace, Kissimmee, FL, USA, 5–9 January 2015. [\[CrossRef\]](#)
6. Habib, M.K.; Ayankoso, S.A. Modeling and Control of a Double Inverted Pendulum using LQR with Parameter Optimization through GA and PSO. In Proceedings of the 2020 21st International Conference on Research and Education in Mechatronics (REM), Cracow, Poland, 9–11 December 2020; pp. 1–6. [\[CrossRef\]](#)
7. Karthick, S.; Jerome, J.; Vinodh Kumar, E.; Raaja, G. APSO Based Weighting Matrices Selection of LQR Applied to Tracking Control of SIMO System. In *Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics*; Springer: New Delhi, India, 2016; pp. 11–20.
8. Yuan, C.Y.; Li, K.T.; Zang, G.R.; Wang, X.C. Optimization of semi-active suspension LQR parameters based on local optimization with a skipping out particle swarm algorithm. In Proceedings of the International Conference on Mechanical Design and Simulation (MDS 2022), Wuhan, China, 18–20 March 2022; International Society for Optics and Photonics; SPIE: Bellingham, WA, USA, 2022; Volume 12261, p. 122614B. [\[CrossRef\]](#)
9. Sun, Z.; Wen, Z.; Xu, L.; Gong, G.; Xie, X.; Sun, Z. LQR Control Method based on Improved Antlion Algorithm. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 663–668. [\[CrossRef\]](#)
10. Manna, S.; Mani, G.; Ghildiyal, S.; Stonier, A.A.; Peter, G.; Ganji, V.; Murugesan, S. Ant Colony Optimization Tuned Closed-Loop Optimal Control Intended for Vehicle Active Suspension System. *IEEE Access* **2022**, *10*, 53735–53745. [\[CrossRef\]](#)
11. Muthukumari, S.; Kanagalakshmi, S.; Sunil Kumar, T.K. Optimal Tuning of LQR for Load Frequency Control in Deregulated Power System for Given Time Domain Specifications. In Proceedings of the 2019 29th Australasian Universities Power Engineering Conference (AUPEC), Nadi, Fiji, 26–29 November 2019; pp. 1–6. [\[CrossRef\]](#)
12. Yuvapriya, T.; Lakshmi, P.; Elumalai, V.K. Experimental Validation of LQR Weight Optimization Using Bat Algorithm Applied to Vibration Control of Vehicle Suspension System. *IETE J. Res.* **2022**, 1–11. [\[CrossRef\]](#)
13. Elumalai, V.K.; Raaja, G.S. A new algebraic LQR weight selection algorithm for tracking control of 2 DoF torsion system. *Arch. Electr. Eng.* **2017**, *66*, 55–75. [\[CrossRef\]](#)
14. Sarkar, T.T.; Dewan, L. Pole-placement, PID and genetic algorithm based stabilization of inverted pendulum. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–6. [\[CrossRef\]](#)
15. Yang, Y. Quaternion-Based LQR Spacecraft Control Design Is a Robust Pole Assignment Design. *J. Aerosp. Eng.* **2014**, *27*, 168–176. [\[CrossRef\]](#)
16. Das, R.R.; Elumalai, V.K.; Ganapathy Subramanian, R.; Ashok Kumar, K.V. Adaptive predator–prey optimization for tuning of infinite horizon LQR applied to vehicle suspension system. *Appl. Soft Comput.* **2018**, *72*, 518–526. [\[CrossRef\]](#)
17. Morar, D.; Dobra, P. Optimal LQR weight matrices selection for a CNC machine controller. In Proceedings of the 2021 23rd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 26–28 May 2021; pp. 21–26. [\[CrossRef\]](#)
18. Fahmizal; Nugroho, H.A.; Cahyadi, A.I.; Ardiyanto, I. Tuning LQR Parameters using Neuro Evolution of Augmenting Topologies (NEAT) on a Double Pendulum Cart. In Proceedings of the 2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), Malang, Indonesia, 23–25 August 2022; pp. 270–275. [\[CrossRef\]](#)
19. Xin, G.; Xin, S.; Cebe, O.; Pollayil, M.J.; Angelini, F.; Garabini, M.; Vijayakumar, S.; Mistry, M. Robust Footstep Planning and LQR Control for Dynamic Quadrupedal Locomotion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4488–4495. [\[CrossRef\]](#)
20. Potter, J.E.; Velde, W.E.V. Optimum mixing of gyroscope and star tracker data. *J. Spacecr. Rocket.* **1968**, *5*, 536–540. [\[CrossRef\]](#)
21. Turner, J.D.; Chun, H.M.; Juang, J.N. An analytic solution for the state trajectories of a feedback control system. *J. Guid. Control. Dyn.* **1985**, *8*, 147–148. [\[CrossRef\]](#)
22. Lewis, F.; Syrmos, V.; Syrmos, V. *Optimal Control*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012. [\[CrossRef\]](#)
23. Potter, J. *A Matrix Equation Arising in Statistical Filter Theory*; Technical Report; NASA: Washington, DC, USA, 1965.
24. Davison, E. The numerical solution of  $\dot{X} = A_1X + XA_2 + D$ ,  $X(0) = C$ . *IEEE Trans. Autom. Control* **1975**, *20*, 566–567. [\[CrossRef\]](#)
25. Golub, G.; Nash, S.; Van Loan, C. A Hessenberg-Schur method for the problem  $AX + XB = C$ . *IEEE Trans. Autom. Control* **1979**, *24*, 909–913. [\[CrossRef\]](#)
26. Juang, J.N.; Turner, J.D.; Chun, H.M. Closed-form solutions for feedback control with terminal constraints. *J. Guid. Control. Dyn.* **1985**, *8*, 39–43. [\[CrossRef\]](#)
27. Ogata, K. *System Dynamics*, 4th ed.; Pearson Education: London, UK, 2013.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.