


Article

# State-Constrained Sub-Optimal Tracking Controller for Continuous-Time Linear Time-Invariant (CT-LTI) Systems and Its Application for DC Motor Servo Systems

Jihwan Kim <sup>1,\*</sup> , Ung Jon <sup>2</sup> and Hyeongcheol Lee <sup>1,\*</sup><sup>1</sup> Department of Electrical and Biomedical Engineering, Hanyang University, Seoul 04763, Korea<sup>2</sup> Department of Electrical Engineering, Hanyang University, Seoul 04763, Korea; unginext@hanyang.ac.kr

\* Correspondence: iminai@hanyang.ac.kr (J.K.); hclee@hanyang.ac.kr (H.L.); Tel.: +82-2-2220-1685 (H.L.)

Received: 22 July 2020; Accepted: 13 August 2020; Published: 18 August 2020



**Abstract:** In this paper, we propose an analytic solution of state-constrained optimal tracking control problems for continuous-time linear time-invariant (CT-LTI) systems that are based on model-based prediction, the quadratic penalty function, and the variational approach. Model-based prediction is a concept taken from model-predictive control (MPC) and this is essential to change the direction of calculation for the solution from backward to forward. The quadratic penalty function plays an important role in deriving the analytic solution since it can transform the problem into a form that does not have inequality constraints. For computational convenience, we also propose a sub-optimal controller derived from the steady-state approximation of the analytic solution and show that the proposed controller satisfies the Lyapunov stability. The main advantage of the proposed controller is that it can be implemented in real time with a lower computational load compared to the implicit MPC. Finally, the simulation results for a DC motor servo system are shown and compared with the results of the direct multi-shooting method and the implicit MPC to verify the effectiveness of the proposed controller.

**Keywords:** continuous-time linear time-invariant (CT-LTI); DC motor servo system; linear-quadratic tracking (LQT); model-based prediction; quadratic penalty function; state-constrained optimal control; variational approach

## 1. Introduction

Recently, interest has been increasing in control systems that require limitations on the state of the target system. For example, optimal trajectory control for industrial robots [1,2], which limits the workspace for co-work between humans and machines, and optimal powertrain control for hybrid vehicle systems [3–5], which has limitations on battery capacity, have become more critical to industry. For analytical and computational convenience, the target system for industrial purposes is often linearized, therefore several studies on linear optimal controllers with state constraints were performed. However, these studies imply large computational loads that make it difficult to implement in real time. The following section explains why these computational loads are caused.

### 1.1. Solutions and Their Approximations of the Optimal Control Problems

Assume that the target system is a continuous-time linear time-invariant (CT-LTI) system as follows:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad (1)$$

$$\mathbf{y} = \mathbf{C}_y \mathbf{x}, \tag{2}$$

where  $\mathbf{x}(t)$  is the state of the system,  $\mathbf{u}(t)$  is the input of the system, and  $\mathbf{y}(t)$  is the output of the system. Let the tracking error  $\mathbf{e}(t)$  be

$$\mathbf{e} = \mathbf{r} - \mathbf{y}, \tag{3}$$

where  $\mathbf{r}(t)$  is the reference for the output. Then, the linear-quadratic tracking (LQT) problem can be expressed as follows [6–8]:

$$\begin{aligned} \underset{\mathbf{e}, \mathbf{u}}{\text{minimize}} \quad & J = \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \\ \text{subject to} \quad & \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} - \dot{\mathbf{x}} = 0, \end{aligned} \tag{4}$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are positive definite matrices weighting for the tracking error and the input,  $J$  is the cost function,  $t_0$  is the initial time, and  $t_f$  is the final time. Assume that pair  $(\mathbf{A}, \mathbf{B})$  is controllable and pair  $(\mathbf{A}, \mathbf{C}_y)$  is observable. Then, by the Lagrange multiplier method [6–8] and the variational approach (Theorem A1), the LQT problem (4) can be transformed into a problem of finding solutions of the Riccati equation with boundary conditions and an auxiliary dynamic equation related to the reference. The Riccati equation and the auxiliary dynamic equation can be derived as follows:

$$\dot{\mathbf{P}} = -\mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} - \mathbf{C}_y^T \mathbf{Q} \mathbf{C}_y, \tag{5}$$

$$\dot{\mathbf{g}} = -(\mathbf{A}^T - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T) \mathbf{g} - \mathbf{C}_y^T \mathbf{Q} \mathbf{r}, \tag{6}$$

where  $\mathbf{P}(t)$  and  $\mathbf{g}(t)$  are the solutions to be determined. These solutions are usually calculated in the backward direction in time or by using iterations since  $\mathbf{A} - \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}$ , which is the system matrix of the closed-loop system, must have all the negative real parts of the eigenvalues. However, if there are no constraints on the input and the state, it is well known in control engineering that the solutions of the equations can be approximated to their steady-state values, and the solutions become independent of the time. In the case of input constraints alone, the optimal input can be determined by Pontryagin’s minimum principle [6–9]. In general, the solutions determined by Pontryagin’s minimum principle are closely related with the solutions of the Equations (5) and (6) and have a form simple enough to be implemented on modern microcontrollers. On the other hand, if there are inequality constraints on the state, the problem statement (4) is not valid, therefore it must be redefined as follows:

$$\begin{aligned} \underset{\mathbf{e}, \mathbf{u}}{\text{minimize}} \quad & J = \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \\ \text{subject to} \quad & \begin{cases} \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} - \dot{\mathbf{x}} = 0 \\ \mathbf{C}_h \mathbf{x} + \mathbf{w} \leq 0 \end{cases}, \end{aligned} \tag{7}$$

where  $\mathbf{C}_h$  and  $\mathbf{w}$  are time-invariant parameters of the inequality constraints. The vector inequality  $\mathbf{h} \leq 0$  means that  $h_i \leq 0$  ( $i = 1, \dots, n$ ), where  $\mathbf{h} = [h_1 \ \dots \ h_n]^T$ . In this case, the problem related to the direction of the calculation is hard to avoid.

Since the problem (7) is difficult to solve exactly, many kinds of research were proposed to solve this problem approximately. The first group of research consists of numerical approaches including dynamic programming methods [6,10,11] or direct and indirect methods [11–13]. In general, these methods discretize the target system and apply numerical methods. Since this procedure does not change the direction of the calculation, most of these methods need backward calculations or iterations for all the time steps. The second group of research comprises model-predictive control (MPC) methods including implicit MPC [14–17] and explicit MPC [18–20]. The main difference from the first group is that time-forward calculation is possible since this method predicts the optimal states and inputs of the target system for a short time ahead. However, this method also requires repetitive calculations for the predictions, and the precision of the calculation decreases if the time length of the prediction is not long enough, therefore it still requires many computations in general. To reduce the computational

load, explicit MPC was proposed. For explicit MPC design, methods dividing the state space are essential, but the results of these methods are not easy to analyze in practice [21] since they are based on numerical iterations. In summary, the computational problem of the state-constrained optimal tracking control is that the direction of the exact computation for the solution is a time-backward calculation which leads to increasing computational loads.

### 1.2. Outline and Scope of the Paper

In this paper, we propose an analytic solution of state-constrained optimal tracking control problems for CT-LTI systems that are based on the model-based prediction, the quadratic penalty method, and the variational approach in Section 2. The model-based prediction is a concept taken from MPC, and this is essential to change the direction of the calculation for the solution from backward to forward. The quadratic penalty method plays an important role in deriving the analytic solution since it can transform the problem into a form that does not have inequality constraints. For computational convenience, we also propose a sub-optimal controller derived from the steady-state approximation of the analytic solution, and show that the proposed controller satisfies the Lyapunov stability in Section 3. Finally, the simulation results for a DC motor servo system are shown and compared with the results of the direct multi-shooting method and implicit MPC to verify the effectiveness of the proposed controller in Section 4.

## 2. Analytic Solution of State-Constrained Optimal Tracking Problems

In this section, we describe the analytic solution of state-constrained optimal tracking problems. This solution can be derived by using the model-based prediction, inequality constraints using prediction, the quadratic penalty function, and the variational approach.

### 2.1. Model-Based Prediction

Suppose that the target system is equal to (1), then a model-based prediction with a fixed time interval  $\tau$  can be written as follows [16,22]:

$$\hat{\mathbf{x}}_{\tau}(t) = \mathbf{x}(t + \tau|t) = e^{\mathbf{A}\tau}\mathbf{x}(t) + \int_t^{t+\tau} e^{\mathbf{A}(t+\tau-\eta)} \mathbf{B}\mathbf{u}(\eta)d\eta \tag{8}$$

Assume that the time interval  $\tau$  is short enough to consider the input as a constant. Then, we can approximate (8) as

$$\hat{\mathbf{x}}_{\tau} \approx \mathbf{A}_d\mathbf{x} + \mathbf{B}_d\mathbf{u}, \tag{9}$$

where

$$\mathbf{A}_d(\tau) = e^{\mathbf{A}\tau} \text{ and } \mathbf{B}_d(\tau) = \int_0^{\tau} e^{\mathbf{A}\eta}d\eta \cdot \mathbf{B}.$$

The calculation of the above matrices is described in [22] (pp. 114–117) and this can be performed by using *c2d* MATLAB® command, etc.

### 2.2. Inequality Constraints Using Prediction

Assume that the left sides of the inequality constraints are

$$h_i(\mathbf{x}) = \mathbf{C}_{h_i}\mathbf{x} + w_i, (i = 1, 2, \dots, n), \tag{10}$$

where

$$\mathbf{C}_h = \begin{bmatrix} \mathbf{C}_{h_1} \\ \vdots \\ \mathbf{C}_{h_n} \end{bmatrix} \text{ and } \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}.$$

Then, the inequality constraints have a form of

$$h_i(\mathbf{x}) \leq 0 \tag{11}$$

Since (11) should be valid at all times, the following inequalities also should be valid:

$$h_i(\hat{\mathbf{x}}_\tau) \leq 0 \tag{12}$$

Let

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_n(\mathbf{x}) \\ h_1(\hat{\mathbf{x}}_\tau) \\ \vdots \\ h_n(\hat{\mathbf{x}}_\tau) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_h \mathbf{x} + \mathbf{w} \\ \mathbf{C}_h \mathbf{A}_d \mathbf{x} + \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{w} \end{bmatrix}, \tag{13}$$

where

$$\mathbf{C}_h = \begin{bmatrix} \mathbf{C}_{h_1} \\ \mathbf{C}_{h_2} \\ \vdots \\ \mathbf{C}_{h_n} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}.$$

Then, (11) and (12) can be rewritten by using (13) as follows:

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0. \tag{14}$$

### 2.3. Quadratic Penalty Function

Suppose that the penalty function  $p_i$  of the inequality constraint  $h_i \leq 0$  is

$$p_i(h_i) = \frac{1}{2} \alpha_i(h_i) h_i^2, \quad (i = 1, 2, \dots, 2n), \tag{15}$$

where

$$\alpha_i(h_i) = \begin{cases} 0, & h_i < 0 \\ q_i, & h_i \geq 0 \end{cases}$$

and  $q_i > 0$  is the weight for  $\alpha_i$  [23,24]. As shown in Figure 1, the meaning of the penalty function is the violation costs of the inequality constraint.

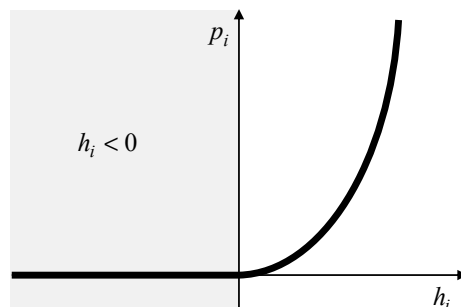


Figure 1. An example of the penalty function (15).

The quadratic penalty function is defined as the sum of the penalty functions:

$$p(\mathbf{h}) = \sum_{i=1}^{2n} p_i(h_i) = \frac{1}{2} \mathbf{h}^T(\mathbf{x}, \mathbf{u}) \text{diag}(\boldsymbol{\alpha}^T) \mathbf{h}(\mathbf{x}, \mathbf{u}), \tag{16}$$

where

$$\alpha = \begin{bmatrix} \alpha_x \\ \alpha_\tau \end{bmatrix}, \alpha_x = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}, \text{ and } \alpha_\tau = \begin{bmatrix} \alpha_{n+1} \\ \vdots \\ \alpha_{2n} \end{bmatrix},$$

and the diag function is defined as

$$\text{diag}(\begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix}) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}.$$

### 2.4. Variational Approach

By using the Lagrange multiplier method and the penalty function method [23,24], Problem (7) can be transformed into the following problem:

$$\underset{x,u}{\text{minimize}} J_a = \int_{t_0}^{t_f} L(x, \dot{x}, u) dt, \tag{17}$$

$$L(x, \dot{x}, u) = \frac{1}{2}(\mathbf{r} - \mathbf{C}_y \mathbf{x})^T \mathbf{Q}(\mathbf{r} - \mathbf{C}_y \mathbf{x}) + \frac{1}{2} \mathbf{u}^T \mathbf{R} \mathbf{u} + \lambda^T (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} - \dot{\mathbf{x}}) + \frac{1}{2} \mathbf{h}^T \text{diag}(\alpha^T) \mathbf{h}, \tag{18}$$

where  $\lambda(t)$  is the Lagrange multiplier. Let the Hamiltonian function of (17) be

$$H(x, u) = \frac{1}{2}(\mathbf{r} - \mathbf{C}_y \mathbf{x})^T \mathbf{Q}(\mathbf{r} - \mathbf{C}_y \mathbf{x}) + \frac{1}{2} \mathbf{u}^T \mathbf{R} \mathbf{u} + \lambda^T (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) + \frac{1}{2} \mathbf{h}^T \text{diag}(\alpha^T) \mathbf{h}, \tag{19}$$

and  $\mathbf{Q}_x = \text{diag}(\alpha_x^T)$ ,  $\mathbf{Q}_\tau = \text{diag}(\alpha_\tau^T)$ . Then, substituting (13) into (19) yields

$$H(x, u) = \frac{1}{2}(\mathbf{r} - \mathbf{C}_y \mathbf{x})^T \mathbf{Q}(\mathbf{r} - \mathbf{C}_y \mathbf{x}) + \frac{1}{2} \mathbf{u}^T \mathbf{R} \mathbf{u} + \lambda^T (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) + \frac{1}{2} (\mathbf{C}_h \mathbf{x} + \mathbf{w})^T \mathbf{Q}_x (\mathbf{C}_h \mathbf{x} + \mathbf{w}) + \frac{1}{2} (\mathbf{C}_h \mathbf{A}_d \mathbf{x} + \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{w})^T \mathbf{Q}_\tau (\mathbf{C}_h \mathbf{A}_d \mathbf{x} + \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{w}). \tag{20}$$

By Corollary A1, the following equations hold:

$$\begin{aligned} \dot{\lambda} &= \mathbf{C}_y^T \mathbf{Q}(\mathbf{r} - \mathbf{C}_y \mathbf{x}) - \mathbf{A}^T \lambda - \mathbf{C}_h^T \mathbf{Q}_x (\mathbf{C}_h \mathbf{x} + \mathbf{w}) - \mathbf{A}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau (\mathbf{C}_h \mathbf{A}_d \mathbf{x} + \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{w}) \\ &= -\mathbf{A}^T \lambda - (\mathbf{Q}_1 + \mathbf{A}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau \mathbf{C}_h \mathbf{A}_d) \mathbf{x} - \mathbf{A}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{C}_y^T \mathbf{Q} \mathbf{r} \\ &\quad - (\mathbf{C}_h^T \mathbf{Q}_x + \mathbf{A}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau) \mathbf{w}, \end{aligned} \tag{21}$$

$$0 = \mathbf{u}^T \mathbf{R} + \lambda^T \mathbf{B} + (\mathbf{C}_h \mathbf{A}_d \mathbf{x} + \mathbf{C}_h \mathbf{B}_d \mathbf{u} + \mathbf{w})^T \mathbf{Q}_\tau \mathbf{C}_h \mathbf{B}_d, \tag{22}$$

$$\mathbf{Q}_1 = \mathbf{C}_y^T \mathbf{Q} \mathbf{C}_y + \mathbf{C}_h^T \mathbf{Q}_x \mathbf{C}_h. \tag{23}$$

Let

$$\mathbf{R}_\tau = \mathbf{R} + \mathbf{B}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau \mathbf{C}_h \mathbf{B}_d. \tag{24}$$

Then, the optimal input is

$$\mathbf{u} = -\mathbf{R}_\tau^{-1} \mathbf{B}^T \lambda - \mathbf{R}_\tau^{-1} \mathbf{B}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau \mathbf{C}_h \mathbf{A}_d \mathbf{x} - \mathbf{R}_\tau^{-1} \mathbf{B}_d^T \mathbf{C}_h^T \mathbf{Q}_\tau \mathbf{w}. \tag{25}$$

### 2.5. Analytical Solution of the Problem

The following procedure is the same method used in the derivation of the Riccati Equation (5) and the auxiliary dynamic Equation (6) [7,8].

**Theorem 1.** Assume that the costate is proportional to the state. Then, the costate can be written as

$$\lambda = \mathbf{P}\mathbf{x} - \mathbf{g}, \tag{26}$$

where  $\mathbf{P}(t)$  and  $\mathbf{g}(t)$  are values to be determined. Then, the following dynamic equations hold:

$$\dot{\mathbf{P}} = -\mathbf{P}\mathbf{A}_z - \mathbf{A}_z^T\mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P} - \mathbf{Q}_2, \tag{27}$$

$$\begin{aligned} \dot{\mathbf{g}} &= -(\mathbf{A}_z^T - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T)\mathbf{g} - \mathbf{C}_y^T\mathbf{Q}\mathbf{r} \\ &+ [\mathbf{C}_h^T\mathbf{Q}_x - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau(\mathbf{I} - \mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)]\mathbf{w}, \end{aligned} \tag{28}$$

where

$$\mathbf{Q}_2 = \mathbf{Q}_1 + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau(\mathbf{I} - \mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)\mathbf{C}_h\mathbf{A}_d, \tag{29}$$

$$\mathbf{A}_z = \mathbf{A} - \mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{C}_h\mathbf{A}_d. \tag{30}$$

**Proof.** By differentiating (26),

$$\dot{\lambda} = \dot{\mathbf{P}}\mathbf{x} + \mathbf{P}\dot{\mathbf{x}} - \dot{\mathbf{g}}. \tag{31}$$

Substituting (1), (25), and (26) into (31) yields

$$\begin{aligned} \dot{\lambda} &= (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A})\mathbf{x} + \mathbf{P}\mathbf{B}\mathbf{u} - \dot{\mathbf{g}} \\ &= (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A}_z)\mathbf{x} - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\lambda - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{w} - \dot{\mathbf{g}} \\ &= (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A}_z - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P})\mathbf{x} - \dot{\mathbf{g}} + \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{g} - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{w}, \end{aligned} \tag{32}$$

Since the left sides of (21) and (32) are equal, the right sides of (21) and (32) are also equal. Therefore,

$$\begin{aligned} 0 &= (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A}_z - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q}_1 + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{C}_h\mathbf{A}_d)\mathbf{x} \\ &- \dot{\mathbf{g}} + \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{g} + (\mathbf{C}_h^T\mathbf{Q}_x + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)\mathbf{w} - \mathbf{C}_y^T\mathbf{Q}\mathbf{r} \\ &+ \mathbf{A}^T\lambda + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{C}_h\mathbf{B}_d\mathbf{u}. \end{aligned} \tag{33}$$

Substituting (25) and (26) into (33) yields

$$\begin{aligned} 0 &= (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A}_z + \mathbf{A}_z^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q}_2)\mathbf{x} - \dot{\mathbf{g}} - (\mathbf{A}_z^T - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T)\mathbf{g} - \mathbf{C}_y^T\mathbf{Q}\mathbf{r} \\ &+ [\mathbf{C}_h^T\mathbf{Q}_x - \mathbf{P}\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau(\mathbf{I} - \mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)]\mathbf{w}. \end{aligned} \tag{34}$$

Since (34) should be valid for all the states, Equations (27) and (28) hold. □

### 3. State-Constrained Sub-Optimal Tracking Controller

Exact solutions of (27) and (28) have to be calculated in the backward direction in time, and this is not proper for real-time implementations. Therefore, in this section, we propose a sub-optimal controller that is stable and proper for real-time implementations.

#### 3.1. State-Constrained Sub-Optimal Tracking Controller

The steady-state values of (27) and (28) are

$$0 = \mathbf{P}_s\mathbf{A}_z + \mathbf{A}_z^T\mathbf{P}_s - \mathbf{P}_s\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P}_s + \mathbf{Q}_2, \tag{35}$$

$$\mathbf{g}_s = -(\mathbf{A}_c^T)^{-1}\mathbf{C}_y^T\mathbf{Q}\mathbf{r} + (\mathbf{A}_c^T)^{-1} \left[ \begin{array}{l} \mathbf{C}_h^T\mathbf{Q}_x - \mathbf{P}_s\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau \\ + \mathbf{A}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau(\mathbf{I} - \mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau) \end{array} \right] \mathbf{w}, \tag{36}$$

where

$$\mathbf{A}_c = \mathbf{A}_z - \mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P}_s. \tag{37}$$

Substituting (26) into (25) yields

$$\mathbf{u} = -(\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P}_s + \mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{C}_h\mathbf{A}_d)\mathbf{x} + \mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{g}_s - \mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{w}. \tag{38}$$

Notably, (38) becomes the steady-state LQT controller [7,8] that uses steady-state values of (5) and (6) if  $\mathbf{Q}_x$  and  $\mathbf{Q}_\tau$  are null matrices.

The sub-optimal controller is implemented by calculating (35)–(38), but  $\alpha_\tau$  is needed in these calculations. Since  $\alpha_x$  indicates current violations of the state constraints and  $\alpha_\tau$  indicates possible violations of the state constraints that occur in a moment,  $\alpha_\tau$  can be identified approximately by the following procedure:

1. Identify  $\alpha_x$  using current state values and calculate  $\mathbf{Q}_x = \text{diag}(\alpha_x^T)$ .
2. Calculate (24) and (35) using an algebraic Riccati equation solver with  $\mathbf{Q}_\tau = 0$ .
3. Calculate (36)–(38) using the result of step 2 and applying  $\mathbf{Q}_\tau = 0$ .
4. Calculate (9) using the result of step 3.
5. Identify  $\alpha_\tau$  using the result of step 4.

Since  $\alpha_x$  has limited values (zero or a fixed value for each element), offline calculation results for lower computational loads can be used in step 2. The sub-optimal controller requires the following procedure in addition to the above procedure.

6. Calculate  $\mathbf{Q}_\tau = \text{diag}(\alpha_\tau^T)$  using the result of step 5.
7. Calculate (24) and (35) using an algebraic Riccati equation solver with the result of step 6.
8. Calculate (36)–(38) using the result of step 7.

Since  $\alpha_\tau$  also has limited values, step 7 can be performed offline. The main advantage of this procedure is that the total computation time of the procedure is limited to a fixed upper boundary. In the case of offline calculations of steps 2 and 7, it is obvious that the total computation load of the proposed controller is lower than that of the implicit MPC.

### 3.2. Stability of the Proposed Controller

By the matrix inversion lemma [25,26],

$$\begin{aligned} \mathbf{I} - \mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau &= \mathbf{I} - \mathbf{C}_h\mathbf{B}_d(\mathbf{R} + \mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau\mathbf{C}_h\mathbf{B}_d)^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau \\ &= (\mathbf{I} + \mathbf{C}_h\mathbf{B}_d\mathbf{R}^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)^{-1}, \end{aligned} \tag{39}$$

$$\mathbf{I} - \mathbf{Q}_\tau\mathbf{C}_h\mathbf{B}_d\mathbf{R}_\tau^{-1}\mathbf{B}_d^T\mathbf{C}_h^T = (\mathbf{I} + \mathbf{Q}_\tau\mathbf{C}_h\mathbf{B}_d\mathbf{R}^{-1}\mathbf{B}_d^T\mathbf{C}_h^T)^{-1}. \tag{40}$$

Therefore, the following equations hold:

$$\mathbf{Q}_2 = \begin{bmatrix} \mathbf{C}_y \\ \mathbf{C}_h \\ \mathbf{C}_h\mathbf{A}_d \end{bmatrix}^T \begin{bmatrix} \mathbf{Q} & 0 & 0 \\ 0 & \mathbf{Q}_x & 0 \\ 0 & 0 & \mathbf{Q}_\tau(\mathbf{I} + \mathbf{C}_h\mathbf{B}_d\mathbf{R}^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{C}_y \\ \mathbf{C}_h \\ \mathbf{C}_h\mathbf{A}_d \end{bmatrix}, \tag{41}$$

$$\mathbf{Q}_\tau(\mathbf{I} + \mathbf{C}_h\mathbf{B}_d\mathbf{R}^{-1}\mathbf{B}_d^T\mathbf{C}_h^T\mathbf{Q}_\tau)^{-1} = (\mathbf{I} + \mathbf{Q}_\tau\mathbf{C}_h\mathbf{B}_d\mathbf{R}^{-1}\mathbf{B}_d^T\mathbf{C}_h^T)^{-1}\mathbf{Q}_\tau. \tag{42}$$

These equations show that  $\mathbf{Q}_2$  is positive semi-definite since  $\mathbf{Q}$  is positive definite and  $\mathbf{Q}_x, \mathbf{Q}_\tau$  are positive semi-definite or null. Assume that pair  $\left(\mathbf{A}_c, \begin{bmatrix} \mathbf{C}_y \\ \mathbf{C}_h \\ \mathbf{C}_h\mathbf{A}_d \end{bmatrix}\right)$  is observable since  $\mathbf{C}_y$  is different from  $\mathbf{C}_h$  in general and suitable  $\mathbf{A}_d$  may be selected by changing  $\tau$ . Then, Equation (35) can be rewritten as

$$\mathbf{P}_s\mathbf{A}_c + \mathbf{A}_c^T\mathbf{P}_s = -(\mathbf{P}_s\mathbf{B}\mathbf{R}_\tau^{-1}\mathbf{B}^T\mathbf{P}_s + \mathbf{Q}_2). \tag{43}$$

Substituting (36)–(38) into (1) and applying  $\mathbf{r} = 0$  and  $\mathbf{w} = 0$  yields

$$\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x}. \tag{44}$$

Therefore, it is concluded that the closed-loop system is stable by the Lyapunov stability theorem [27,28] if the observability condition is satisfied.

### 3.3. Model Modification for Input Smoothing

The input generated by the proposed controller may have severe vibration that is not found in numerical solutions. To mitigate this, we propose a modification of the plant model including a low pass filter before the input as shown in Figure 2.

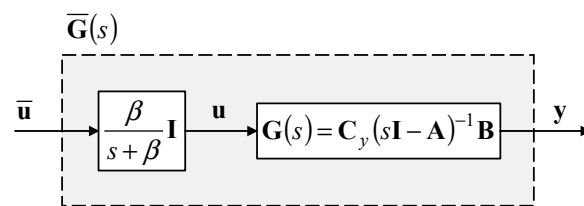


Figure 2. Modified plant model with a low-pass filter.

The state equation of the modified plant model is

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & -\beta \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \mathbf{I} \end{bmatrix} \bar{\mathbf{u}}, \tag{45}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C}_y & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}. \tag{46}$$

Then, the problem (7) should be changed as follows:

$$\begin{aligned} & \underset{\mathbf{e}, \mathbf{u}, \bar{\mathbf{u}}}{\text{minimize}} J = \frac{1}{2} \int_{t_0}^{t_f} \left( \begin{bmatrix} \mathbf{e} \\ \mathbf{u} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{u} \end{bmatrix} + \bar{\mathbf{u}}^T \bar{\mathbf{R}} \bar{\mathbf{u}} \right) dt \\ & \text{subject to} \begin{cases} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & -\beta \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \mathbf{I} \end{bmatrix} \bar{\mathbf{u}} - \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{u}} \end{bmatrix} = 0 \\ \begin{bmatrix} \mathbf{C}_h & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} + \mathbf{w} \leq 0 \end{cases}, \end{aligned} \tag{47}$$

where  $\bar{\mathbf{R}}$  is the weight for  $\bar{\mathbf{u}}$ . Since this problem can be solved by using the same methods described in Sections 2 and 3, we omit a detailed description of the solution.

## 4. Case Study: Application for DC Motor Servo Systems

In this section, to help readers understand how to apply the proposed controller, we show an application of the proposed controller for DC motor servo systems. For precision control of DC motor servo systems, studies including MPC [29], data-driven [30], fuzzy [31], neural network [32], cascade control [33], and the digital twin-based optimization [34] were introduced recently. However, except for MPC, these studies did not consider the state constraints; therefore, we compare MPC and the proposed controller for performance verification. The parameters of the target motor are shown in Table 1. The target motor is a 24 V DC brushed gear motor and its rated torque is 2.94 Nm.



**Table 1.** The parameters of the target motor.

| Name                           | Unit      | Value     |
|--------------------------------|-----------|-----------|
| Rotor inductance ( $L_m$ )     | H         | 0.0065    |
| Armature resistance ( $R_m$ )  | $\Omega$  | 2.3       |
| Back EMF constant ( $K_b$ )    | V·sec/rad | 0.09      |
| Torque constant ( $K_m$ )      | Nm/A      | 0.09      |
| Friction coefficient ( $B_m$ ) | Nm·sec    | 0.00175   |
| Rotor inertia ( $J_m$ )        | Nm·rad    | 0.0000525 |
| Gear ratio ( $\kappa_m$ )      |           | 0.25      |

Then, the state equation of the target system is

$$\dot{\mathbf{x}} = \begin{bmatrix} -\frac{R_m}{L_m} & -\frac{K_b}{L_m} & 0 \\ \frac{K_m}{J_m} & -\frac{B_m}{J_m} & 0 \\ 0 & \kappa_m & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \frac{1}{L_m} \\ 0 \\ 0 \end{bmatrix} u$$

$$\approx \begin{bmatrix} -384.62 & -13.85 & 0 \\ 1714.29 & -33.33 & 0 \\ 0 & 0.25 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 153.85 \\ 0 \\ 0 \end{bmatrix} u, \tag{48}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \tag{49}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

$x_1$  is the motor current,  $x_2$  is the angular speed of the motor, and  $x_3$  is the angular position of the motor. Let  $\beta = 1000$ , then the modified plant model is

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \mathbf{B}\bar{u}, \tag{50}$$

$$\mathbf{y} = \mathbf{C}_y \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \tag{51}$$

where

$$\mathbf{A} = \begin{bmatrix} -384.62 & -13.85 & 0 & 153.85 \\ 1714.29 & -33.33 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & -1000 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1000 \end{bmatrix}, \mathbf{C}_y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

Let  $\tau = 0.001$ , then the parameters for the prediction can be calculated as

$$\mathbf{A}_d = \begin{bmatrix} 0.67 & -0.01 & 0 & 0.09 \\ 1.41 & 0.96 & 0 & 0.07 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.33 \end{bmatrix}, \mathbf{B}_d = \begin{bmatrix} 0.04 \\ 0.04 \\ 0 \\ 0.67 \end{bmatrix}$$

The optimal tracking problem is

$$\underset{\mathbf{x}, u, \bar{u}}{\text{minimize}} J = \frac{1}{2} \int_{t_0}^{t_f} \left( \begin{bmatrix} r - x_3 \\ u \end{bmatrix}^T \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r - x_3 \\ u \end{bmatrix} + \bar{u}^2 \right) dt$$

$$\text{subject to} \begin{cases} \mathbf{A} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \mathbf{B}\bar{u} - \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{u} \end{bmatrix} = 0 \\ \mathbf{C}_h \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \mathbf{w} \leq 0, \end{cases} \tag{52}$$

where

$$C_h = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} -3 \\ -3 \\ -50 \\ -50 \end{bmatrix}$$

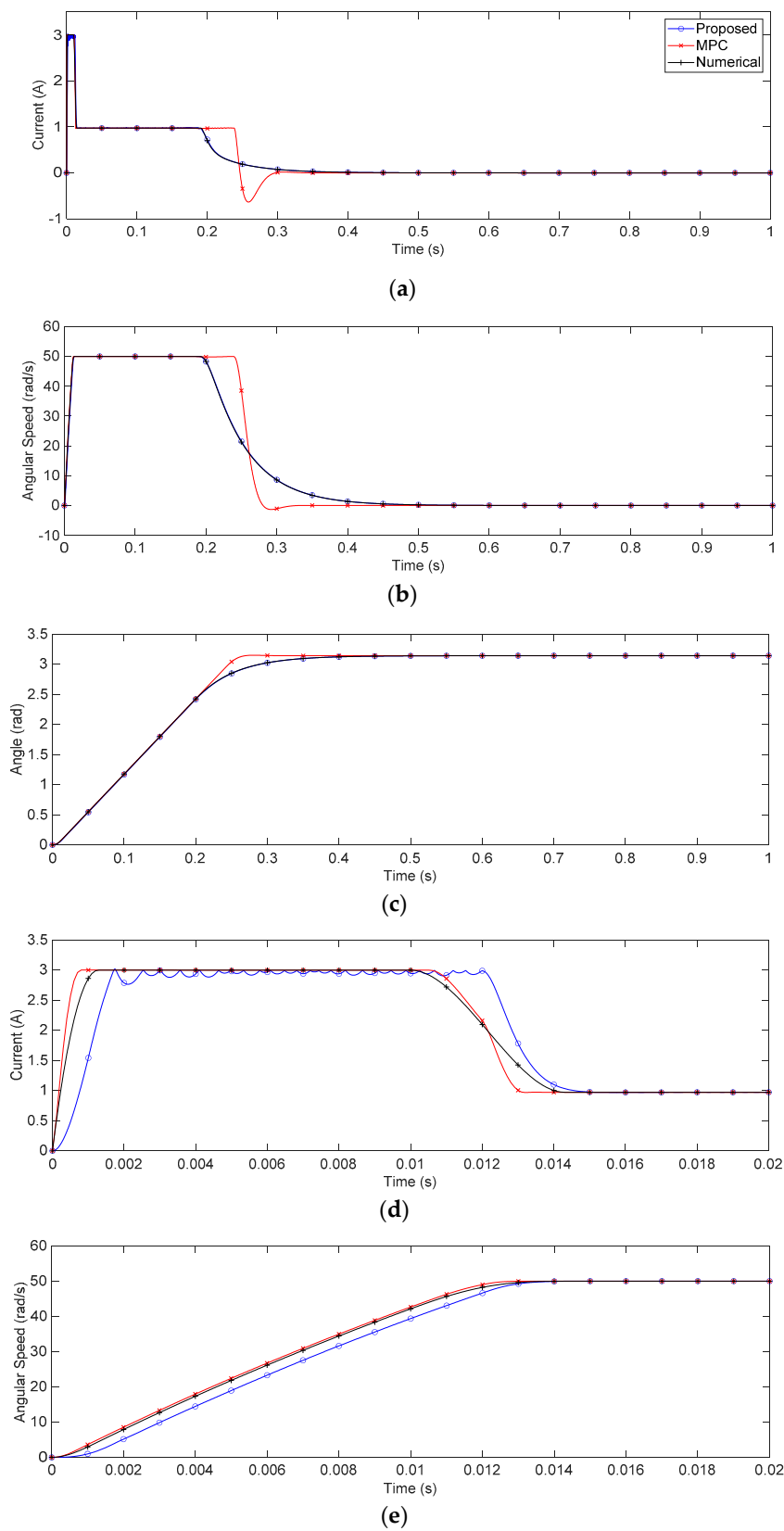
In this case,  $\begin{bmatrix} C_y \\ C_h \\ C_h A_d \end{bmatrix}$  has full rank, therefore the closed-loop system is stable. Available values of  $P_s$

in the cases of  $Q_\tau = 0$  are shown in Table 2. The weight for  $\alpha_x$  is  $[10000 \ 10000 \ 0.001 \ 0.001]^T$  and  $P_s$  was calculated by the MATLAB<sup>®</sup> care function. Since the weight for  $\alpha_\tau$  is  $[100 \ 100 \ 10 \ 10]^T$  and similar calculations can be performed in the cases of  $Q_\tau \neq 0$ , the proposed controller can be implemented by these results. Figure 3 shows the simulation results with  $r = \pi$  and  $x(0) = 0$  without changing  $w$ . The proposed controller was implemented by using Simulink<sup>®</sup> blocks and the ode2 (Heun) fixed-step solver was used for the simulation. The step size was set at 10 $\mu$ s for smooth results. To verify the performance of the proposed controller, we also implemented an implicit MPC by using MATLAB<sup>®</sup> Model Predictive Control Toolbox<sup>™</sup> software. A numerical method based on the direct multi-shooting method implemented by using CasADi software [35] was chosen for comparison since it is close to the optimal solution. The sampling time used in both the numerical method and the implicit MPC was set to 100  $\mu$ s since lower sampling time causes larger computational loads. The prediction and control horizon of the MPC is set to (100, 10).

**Table 2.** Examples of the solutions of (34).

| $\alpha_x$   | $P_s$  | $\alpha_x$   | $P_s$  |
|--|--|--|--|
| $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$         | $\begin{bmatrix} 0.08 & 0.02 & 6.50 & 0.01 \\ * & 0 & 1.46 & 0 \\ * & * & 555.69 & 1 \\ * & * & * & 0 \end{bmatrix}$             | $\begin{bmatrix} 10000 \\ 0 \\ 0 \\ 0 \end{bmatrix}$     | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ |
| $\begin{bmatrix} 0 \\ 10000 \\ 0 \\ 0 \end{bmatrix}$     | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 0 \\ 0.001 \\ 0 \end{bmatrix}$     | $\begin{bmatrix} 0.08 & 0.02 & 6.50 & 0.01 \\ * & 0 & 1.46 & 0 \\ * & * & 555.69 & 1 \\ * & * & * & 0 \end{bmatrix}$             |
| $\begin{bmatrix} 10000 \\ 0 \\ 0.001 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 10000 \\ 0.001 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ |
| $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.001 \end{bmatrix}$     | $\begin{bmatrix} 0.08 & 0.02 & 6.50 & 0.01 \\ * & 0 & 1.46 & 0 \\ * & * & 555.69 & 1 \\ * & * & * & 0 \end{bmatrix}$             | $\begin{bmatrix} 10000 \\ 0 \\ 0 \\ 0.001 \end{bmatrix}$ | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ |
| $\begin{bmatrix} 0 \\ 10000 \\ 0 \\ 0.001 \end{bmatrix}$ | $\begin{bmatrix} 12.21 & -0.13 & 6.63 & 1.29 \\ * & 0.06 & 1.56 & -0.04 \\ * & * & 561.18 & 1 \\ * & * & * & 0.20 \end{bmatrix}$ |  |  |

As shown in Figure 3a–c, the trajectory of the proposed controller and that of the numerical method are very similar, therefore it can be said that the proposed controller is well-approximated. On the other hand, the implicit MPC has different trajectories between 0.2 and 0.4 s though the constraints are maintained. In Figure 3d–e, it seems that the trajectory of the proposed controller has a delay compared to that of the numerical method, but the delay is small enough to be neglected. Since the steady-state LQT controller has similar delay properties [36], it is assumed that the cause of this phenomenon is the steady-state approximation.



**Figure 3.** Simulation results of the proposed controller, the numerical method, and the implicit MPC with the first set of  $r$  and  $w$ : (a) motor currents from 0 to 1 s, (b) angular speeds of the motor rotor from 0 to 1 s, (c) angle of the motor rotor from 0 to 1 s, (d) motor currents from 0 to 0.02 s, (e) angular speeds of the motor rotor from 0 to 0.02 s.

Figure 4 shows simulation results with  $r = 10$ ,  $x(0) = 0$ , and  $\mathbf{w} = [-10 \quad -10 \quad -100 \quad -100]^T$ . All other control parameters used in Figure 4 are the same as those used in Figure 3. These results show that the inequality constraints are maintained, and this leads to the conclusion that the proposed controller does not seem sensitive to  $r$  or  $\mathbf{w}$ . The results of the implicit MPC in Figure 4a–c also differ from these of other methods therefore it is concluded that the implicit MPC differs from the optimal solution.

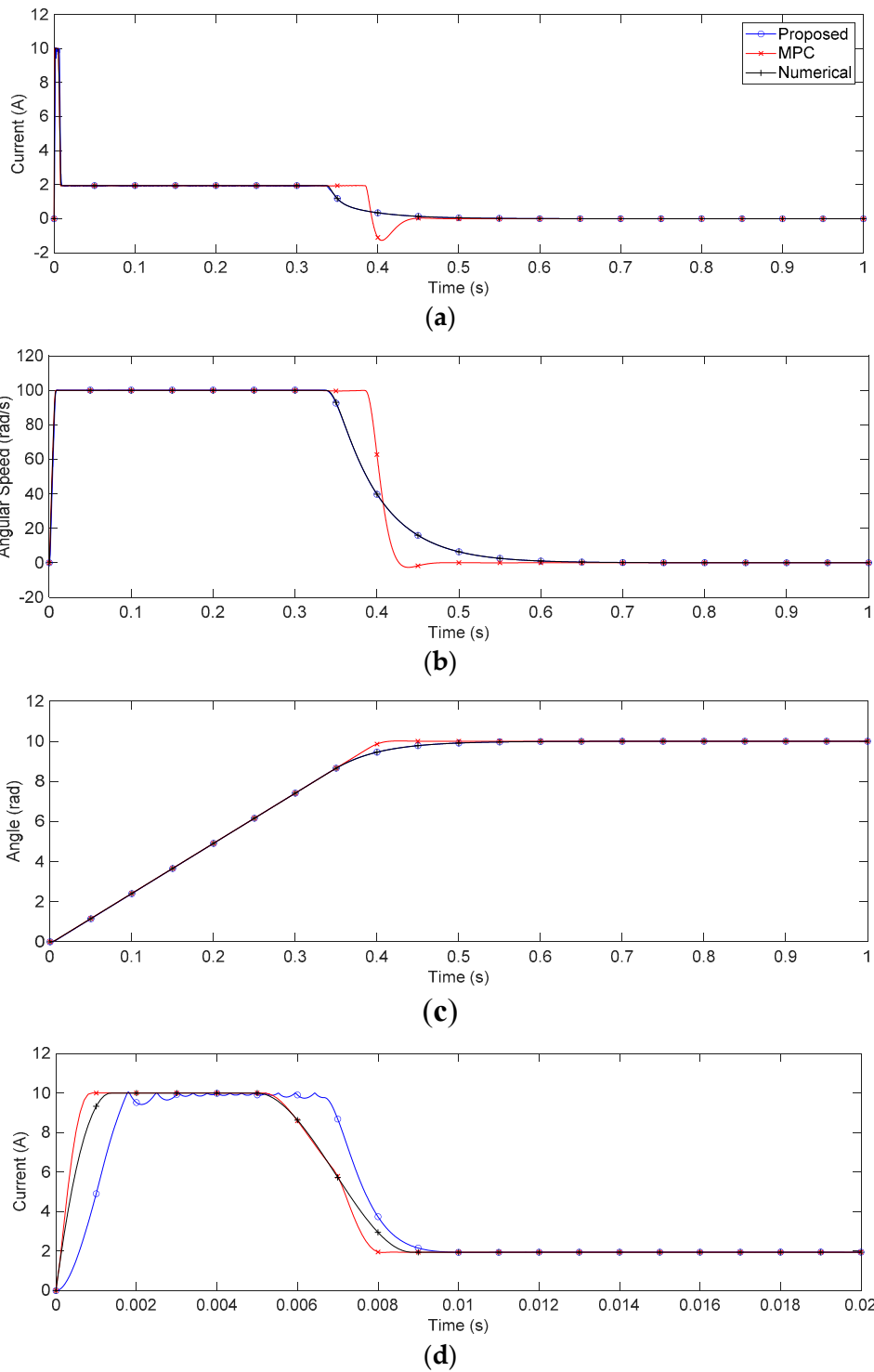
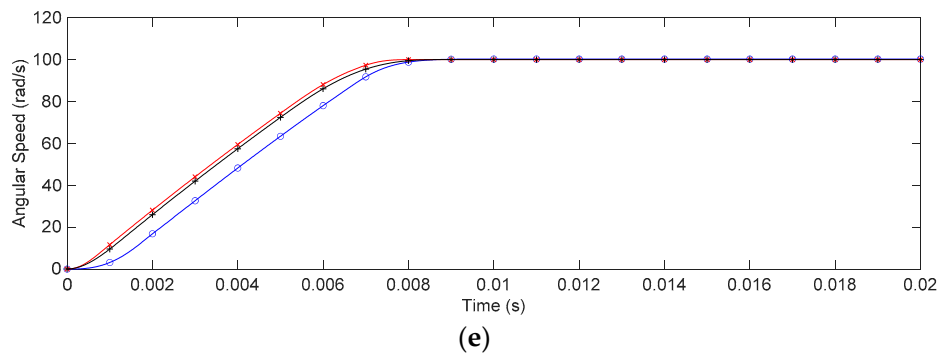
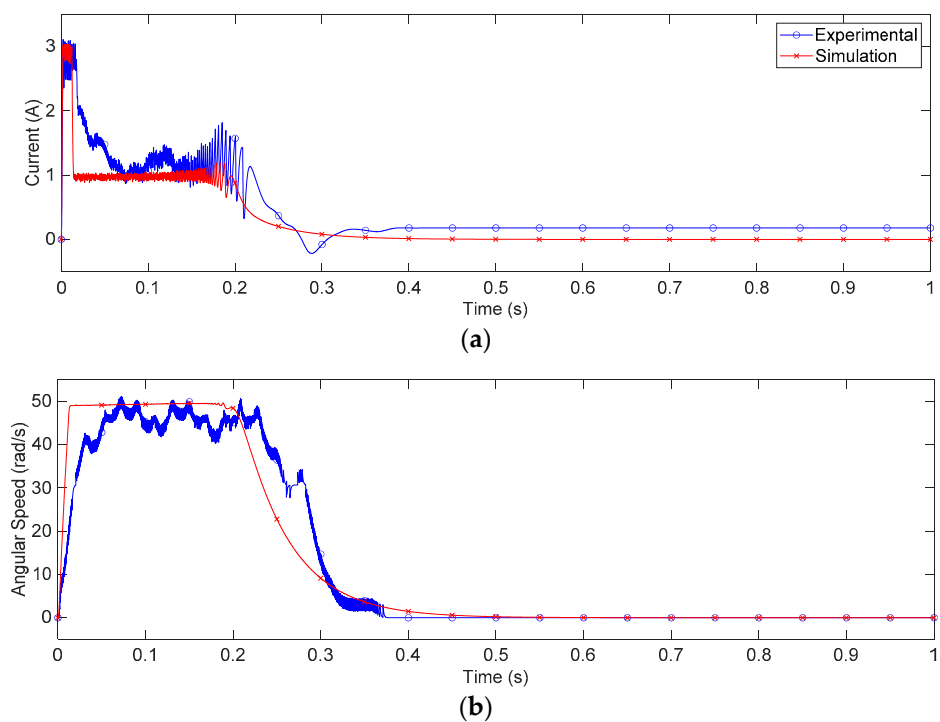


Figure 4. Cont.

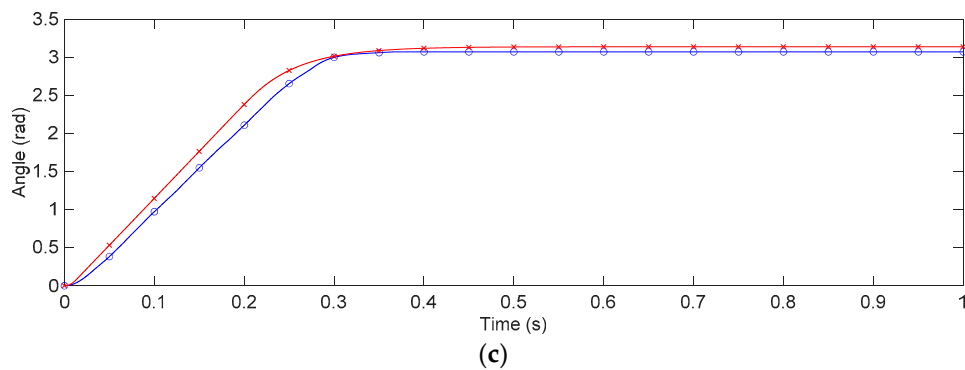


**Figure 4.** Simulation results of the proposed controller, the numerical method, and the implicit MPC with the second set of  $r$  and  $w$ : (a) motor currents from 0 to 1 s, (b) angular speeds of the motor rotor from 0 to 1 s, (c) angle of the motor rotor from 0 to 1 s, (d) motor currents from 0 to 0.02 s, (e) angular speeds of the motor rotor from 0 to 0.02 s.

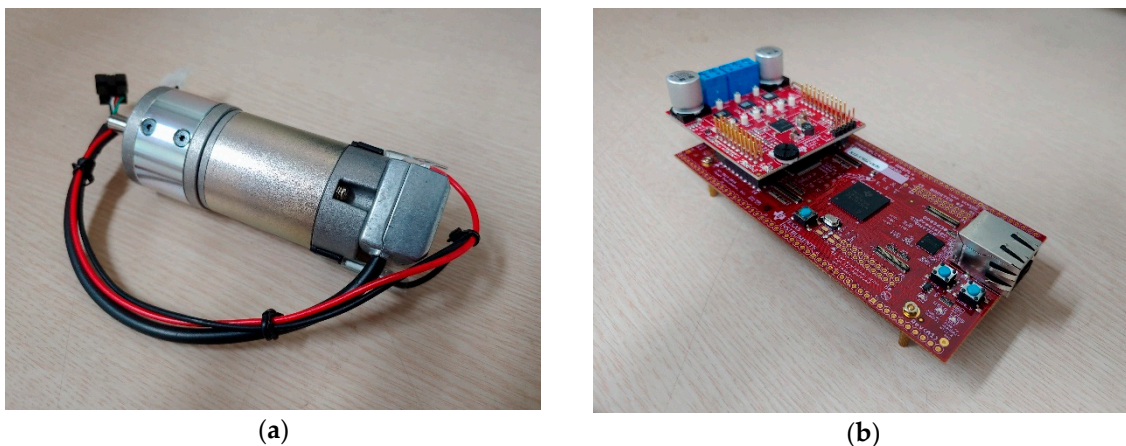
Figure 5 shows experimental and simulation results with  $r = \pi$ ,  $x(0) = 0$ , and  $w = [-3 - 3 - 50 - 50]^T$ . All other control parameters used in Figure 5 are the same as those used in Figure 3. Texas Instruments LAUNCHXL2-570LC43 and BOOSTXL-DRV8323RS (Dallas, TX, USA) are used to control the target motor, and they are shown in Figure 6b. The target motor is equipped with an incremental encoder as shown in Figure 6a, and the encoder has resolutions of 0.0879 degrees. The controller was designed to have sampling time of 100  $\mu$ s, implemented by using Simulink<sup>®</sup> blocks, and converted to C code by using Embedded Coder<sup>®</sup> software and Embedded Coder<sup>®</sup> support package for ARM<sup>®</sup> Cortex<sup>®</sup>-R processors. For comparison, simulations were also performed at the same sampling time in Figure 5. The experimental and simulation results have similar trends in Figure 5, but the experimental results have severe vibrations and performance degradations. Since the real motor has nonlinearities and frictions, it seems that these uncertainties are related to the performance degradations.



**Figure 5.** Cont.



**Figure 5.** Experimental and simulation results of the proposed controller with  $100 \mu\text{s}$  sampling time and the first set of  $r$  and  $w$ : (a) motor currents from 0 to 1 s, (b) angular speeds of the motor rotor from 0 to 1 s, (c) angle of the motor rotor from 0 to 1 s.



**Figure 6.** Target motor and controller: (a) motor, (b) controller.

## 5. Discussion

In this paper, we proposed a sub-optimal tracking controller that does not need numerical iterations or backward calculations for state-constrained optimal tracking problems. The main advantage of the proposed controller is that it can be implemented in real time with a lower computational load compared to the implicit MPC. Though there is a delay compared to the results of the numerical method, the simulation results show that the proposed controller has acceptable performance. However, the proposed controller needs to be verified through more application cases including industrial robots, hybrid vehicles, or other control systems. For future work, the proposed controller may be extended to discrete-time systems. Therefore, studies related to the discrete-time optimal tracking controller based on the proposed method are worth researching. In particular, the controller may be applied to path tracking control for autonomous vehicles [37,38], which have recently been the subject of much research.

**Author Contributions:** Conceptualization, J.K. and U.J.; methodology, J.K.; validation, J.K. and U.J.; writing—original draft, J.K.; writing—review and editing, U.J. and H.L.; supervision, H.L. All authors have read and agreed to the published version of the manuscript.

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

**Acknowledgments:** This work was supported by the research fund of Hanyang University (HY-2017) and “The Technology Innovation Program” (10052501, development of design technology for a device visualizing the virtual driving environment and synchronizing with the actual vehicle driving conditions to test and evaluate ADAS), funded by the Ministry of Trade, Industry, and Energy (MI, Korea).

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

## Appendix A

**Theorem A1** (Variational approach) [6]. *Suppose that the cost function is*

$$J = \int_{t_0}^{t_f} L(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, \lambda) dt. \tag{A1}$$

*Assume that the initial value and the final value of the state are fixed. Then, the necessary conditions for minimizing (A1) are*

$$\frac{\partial L}{\partial \mathbf{x}} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \right) = 0, \tag{A2}$$

$$\frac{\partial L}{\partial \mathbf{u}} = 0, \tag{A3}$$

$$\frac{\partial L}{\partial \lambda} = 0. \tag{A4}$$

**Proof.** See Section 2.5 of [6] (pp. 169–171). □

**Corollary A1:** Suppose that the cost function is

$$J = \int_{t_0}^{t_f} L(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) dt. \tag{A5}$$

Let the Hamiltonian function be

$$H(\mathbf{x}, \mathbf{u}) = L(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) + \lambda^T \dot{\mathbf{x}}. \tag{A6}$$

Then, the necessary conditions for minimizing (A5) are

$$\dot{\lambda} = - \left( \frac{\partial H}{\partial \mathbf{x}} \right)^T, \tag{A7}$$

$$0 = \frac{\partial H}{\partial \mathbf{u}}, \tag{A8}$$

$$\dot{\mathbf{x}} = \left( \frac{\partial H}{\partial \lambda} \right)^T. \tag{A9}$$

**Proof.** By Theorem A1,

$$\frac{\partial L}{\partial \mathbf{x}} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \right) = \frac{\partial H}{\partial \mathbf{x}} + \frac{d}{dt} (\lambda^T) = \frac{\partial H}{\partial \mathbf{x}} + \dot{\lambda}^T = 0, \tag{A10}$$

$$\frac{\partial L}{\partial \mathbf{u}} = \frac{\partial H}{\partial \mathbf{u}} = 0, \tag{A11}$$

$$\frac{\partial L}{\partial \lambda} = \frac{\partial H}{\partial \lambda} - \dot{\mathbf{x}}^T = 0. \tag{A12}$$

Therefore, (A6) and (A7) holds. □

## References

1. Rubio, F.; Llopis-Albert, C.; Valero, F.; Suñer, J.L. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. *Robot. Auton. Syst.* **2016**, *86*, 106–112. [[CrossRef](#)]
2. Ragaglia, M.; Zanchettin, A.M.; Rocco, P. Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. *Mechatronics* **2018**, *55*, 267–281. [[CrossRef](#)]
3. Hung, C.W.; Vu, T.V.; Chen, C.K. The development of an optimal control strategy for a series hydraulic hybrid vehicle. *Appl. Sci.* **2016**, *6*, 93. [[CrossRef](#)]
4. Guo, L.; Gao, B.; Gao, Y.; Chen, H. Optimal energy management for HEVs in eco-driving applications using bi-level MPC. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 2153–2162. [[CrossRef](#)]
5. Chen, Z.; Hu, H.; Wu, Y.; Xiao, R.; Shen, J.; Liu, Y. Energy management for a power-split plug-in hybrid electric vehicle based on reinforcement learning. *Appl. Sci.* **2018**, *8*, 2494. [[CrossRef](#)]
6. Kirk, D.E.; Donald, E. *Optimal Control Theory: An Introduction*; Dover Publications: Mineola, NY, USA, 2004; pp. 53–239.
7. Athans, M.; Falb, P.L. *Optimal Control: An Introduction to the Theory and Its Applications*; Dover Publications: Mineola, NY, USA, 2007; pp. 221–812.
8. Lewis, F.L.; Vrabie, D.; Syrmos, V.L. *Optimal Control*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2012; pp. 110–212.
9. Pontryagin, L.S. *Mathematical Theory of Optimal Processes*; John Wiley & Sons: Hoboken, NJ, USA, 1962; pp. 9–114.
10. Elbert, P.; Ebbesen, S.; Guzzella, L. Implementation of dynamic programming for n-dimensional optimal control problems with final state constraints. *IEEE Trans. Control Syst. Technol.* **2012**, *21*, 924–931. [[CrossRef](#)]
11. Böhme, T.J.; Frank, B.J.C. *Hybrid Systems, Optimal Control and Hybrid Vehicles*; Springer International: Cham, Switzerland, 2017; pp. 167–270.
12. Betts, J.T. Survey of numerical methods for trajectory optimization. *J. Guidance Control. Dyn.* **1998**, *21*, 193–207. [[CrossRef](#)]
13. Betts, J.T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2010; pp. 96–218.
14. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Sokaert, P.O. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [[CrossRef](#)]
15. Qin, S.J.; Badgwell, T.A. A survey of industrial model predictive control technology. *Control Eng. Pract.* **2003**, *11*, 733–764. [[CrossRef](#)]
16. Wang, L. *Model Predictive Control System Design and Implementation Using MATLAB®*; Springer-Verlag: London, UK, 2009; pp. 1–224.
17. Aldaouab, I.; Daniels, M.; Ordóñez, R. MPC for optimized energy exchange between two renewable-energy prosumers. *Appl. Sci.* **2019**, *9*, 3709. [[CrossRef](#)]
18. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. *Automatica* **2002**, *38*, 3–20. [[CrossRef](#)]
19. Bemporad, A.; Borrelli, F.; Morari, M. Model predictive control based on linear programming—The explicit solution. *IEEE Trans. Autom. Control.* **2002**, *47*, 1974–1985. [[CrossRef](#)]
20. Grancharova, A.; Johansen, T.A. *Explicit Nonlinear Model Predictive Control: Theory and Applications*; Springer-Verlag: London, UK, 2012; pp. 1–108.
21. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. Corrigendum to: “The explicit linear quadratic regulator for constrained systems” [*Automatica* 38 (1)(2002) 3–20]. *Automatica* **2003**, *39*, 1845–1846. [[CrossRef](#)]
22. Franklin, G.F.; Powell, J.D.; Workman, M. *Digital Control of Dynamic Systems*, 3rd ed.; Addison-Wesley: Menlo Park, CA, USA, 1998; pp. 96–118.
23. Venkataraman, P. *Applied Optimization with MATLAB Programming*; John Wiley & Sons: Hoboken, NJ, USA, 2002; pp. 265–316.
24. Bryson, A.E. *Applied Optimal Control: Optimization, Estimation and Control*; Hemisphere Publishing Corporation: Washington, DC, USA, 1975; pp. 212–245.
25. Zhang, X.D. *Matrix Analysis and Applications*; Cambridge University Press: Cambridge, UK, 2017; pp. 60–61.
26. Bernstein, D.S. *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed.; Princeton university press: Princeton, NJ, USA, 2009; pp. 77–164.



27. Khalil, H.K.; Grizzle, J.W. *Nonlinear Systems*, 3rd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2002; pp. 111–194.
28. Chen, C.T. *Linear System Theory and Design*, 3rd ed.; Oxford University Press: Oxford, UK, 1998; pp. 121–140.
29. Lin, C.Y.; Liu, Y.C. Precision tracking control and constraint handling of mechatronic servo systems using model predictive control. *IEEE/ASME Trans. Mechatron.* **2011**, *17*, 593–605.
30. Rădac, M.B.; Precup, R.E.; Petriu, E.M.; Preitl, S.; Dragoș, C.A. Data-driven reference trajectory tracking algorithm and experimental validation. *IEEE Trans. Ind. Inf.* **2012**, *9*, 2327–2336. [[CrossRef](#)]
31. Premkumar, K.; Manikandan, B.V.; Kumar, C.A. Antlion algorithm optimized fuzzy PID supervised on-line recurrent fuzzy neural network based controller for brushless DC motor. *Electr. Power Compon. Syst.* **2017**, *45*, 2304–2317. [[CrossRef](#)]
32. Hassan, A.K.; Saraya, M.S.; Elksasy, M.S.; Areed, F.F. Brushless DC motor speed control using PID controller, fuzzy controller, and neuro fuzzy controller. *Int. J. Comput. Appl.* **2018**, *180*, 47–52.
33. Sun, Z.; Pritschow, G.; Zahn, P.; Lechler, A. A novel cascade control principle for feed drives of machine tools. *CIRP Ann.* **2018**, *67*, 389–392. [[CrossRef](#)]
34. Guerra, R.H.; Quiza, R.; Villalonga, A.; Arenas, J.; Castaño, F. Digital twin-based optimization for ultraprecision motion systems with backlash and friction. *IEEE Access* **2019**, *7*, 93462–93472. [[CrossRef](#)]
35. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [[CrossRef](#)]
36. Anderson, B.D.; Moore, J.B. *Optimal Control: Linear Quadratic Methods*; Dover Publications: Mineola, NY, USA, 1990; pp. 68–100.
37. Yao, Q.; Tian, Y. A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking. *Appl. Sci.* **2019**, *9*, 4739. [[CrossRef](#)]
38. Wu, X.; Qiao, B.; Su, C. Trajectory planning with time-variant safety margin for autonomous vehicle lane change. *Appl. Sci.* **2020**, *10*, 1626. [[CrossRef](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).