Structured Hammerstein-Wiener model learning for model predictive control

Ryuta Moriyasu, Taro Ikeda, Sho Kawaguchi, Kenji Kashima

***Abstract*—This paper aims to improve the reliability of optimal control using models constructed by machine learning methods. Optimal control problems based on such models are generally non-convex and difficult to solve online. In this paper, we propose a model that combines the Hammerstein-Wiener model with input convex neural networks, which have recently been proposed in the field of machine learning. An important feature of the proposed model is that resulting optimal control problems are effectively solvable exploiting their convexity and partial linearity while retaining flexible modeling ability. The practical usefulness of the method is examined through its application to the modeling and control of an engine airpath system.**

***Index Terms*—Model predictive control, Machine learning, Convex optimization, Input convex neural network**

I. Introduction

In recent years, there has been an increase in research on control modeling that utilizes machine learning methods such as neural networks and Gaussian processes for model predictive control (MPC); see e.g., [1]–[4]. In the case of complex dynamics, first-principles modeling using physical laws requires advanced knowledge and experience, but there is a possibility that such dynamics can be modeled in a short time without advanced knowledge through data-driven modeling via machine learning methods. Also, factors that are difficult to handle and ignored in physical models can be implicitly learned from the data, which can result in more accurate models than physical models in some cases.

However, machine learning models cannot directly be utilized for numerical optimization-based control methods such as MPC [5], [6]. This is mainly because the resulting optimal control problem (OCP) to be solved at each time is a non-convex optimization problem [7], whose globally optimal solution is difficult to find. Besides, resulting control laws often have discontinuities that can cause hunting of the control input and lead to reliability issues such as instability of the control system and degradation of the actuator.

Recently, in the field of machine learning, Input Convex Neural Network (ICNN) [8], which can guarantee the convexity of input-output relations of a model, has been developed, and a control design method that guarantees the convexity of OCPs has been proposed by using Recurrent ICNN (RICNN), which is an extension of ICNN to a recursive structure, as a model for control [9], [10]. A drawback is that the stage cost is limited to monotonically non-decreasing functions with respect to the state variables. This rules out, for instance, a quadratic cost that is a typical choice for regulation and tracking control. In addition, the closed loop property such as stability is not guaranteed. See also Remark 2 below for a necessity of further reduction of computation burden.

On the other hand, in the area of system identification, various nonlinear models with a special structure that makes OCP convex have been developed. A typical example is the Hammerstein-Wiener (H-W) models, consisting of linear dynamics and static nonlinearity. In the control design using these models, the static nonlinearity can be canceled out [11], [12]. It means that the OCP with cost allowing state and output regulation becomes a convex problem, which can be efficiently solvable. However, when considering constraints on inputs and outputs, especially for the MIMO cases, it is difficult to guarantee the convexity of the feasible set of variables in the obtained model. As a result, the examples of studies that include constraints are mainly for the SISO cases [13]. In addition, for the purpose of control design, it is preferable that the static nonlinear function is bijective. It is, however, difficult to identify complex nonlinear function while guaranteeing invertibility [14], [15].

The purpose of this paper is to propose a novel model, which we refer to as *structured Hammerstein-Wiener models*, and its control design method by combining the techniques developed in both machine learning and control theory. An important feature is that we can take into account constraints on inputs and outputs in a nonlinear multi-input/output system, and can also ensure global optimality and continuity of the control law. This paper is organized as follows: Section II details the proposed model and its identification procedure. Section III reveals the properties, e.g., uniqueness and continuity, of OCP associated to the proposed model. In Section IV, the effectiveness of the proposed method is demonstrated with its application to MPC of an engine airpath system.

Notation: The set of real and non-negative numbers are denoted as $\mathbb{R}$ and $\mathbb{R}_+$. A function is said to be of class C^r if it is r times continuously differentiable. A function f on $\mathbb{R}_+$ is said to be of class $\mathcal{K}$ if it is $f(0) = 0$ and strictly increasing. For vectors x, y, $x_{(i)}$ denotes the i-th element of x and $x \odot y$ is the Hadamard product. Function $\mathrm{softplus}(v) := \log(1 + e^v) > 0$. The vector $1_n := [1, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^n$.

R. Moriyasu and T. Ikeda are with Toyota Central R&D Labs, Aichi, Japan. S. Kawaguchi is with Toyota Industries Coorporation, Aichi, Japan. K. Kashima is with Graduate School of Informatics, Kyoto University, Kyoto, Japan. (e-mail: kk@i.kyoto-u.ac.jp)

II. Modeling and Identification

A. Structured Hammerstein-Wiener Model

We denote the input $u \in \mathbb{R}^{n_u}$, the disturbance $d \in \mathbb{R}^{n_d}$. Tracking to the reference and constraints on internal states are described by using the output $y \in \mathbb{R}^{n_y}$ and $z \in \mathbb{R}^{n_z}$, respectively. We suppose $n_u = n_y$. We assume that the time series data of these signals are available, as well as their time derivatives (true value or difference approximation), if necessary. Our goal is to develop systems from u, d to y, z, and obtain a control law that makes y follow the reference $r \in \mathbb{R}^{n_y}$ while satisfying the box constraints for the input $u = [u_{(1)}, \ldots, u_{(n_u)}]^{\mathrm{T}}$

$$u \in \mathcal{U} := \{u : \underline{u}_{(i)} \leq u_{(i)} \leq \overline{u}_{(i)} \ (i = 1, \ldots, n_u)\} \quad (1)$$

and output $z = [z_{(1)}, \ldots, z_{(n_z)}]^{\mathrm{T}}$

$$z \in \mathcal{Z} := \{z : z_{(i)} \leq \overline{z}_{(i)} \ (i = 1, \ldots, n_z)\}. \quad (2)$$

In this paper, we employ the continuous-time Hammerstein-Wiener model represented by

$$v(t) = \Psi(u(t); d(t)), \quad (3)$$

$$\dot{x}(t) = A(d(t))x(t) + B(d(t))v(t) + c(d(t)), \quad (4)$$

$$y(t) = \Phi^{-1}(x(t); d(t)), \quad (5)$$

$$z(t) = \Xi(x(t), v(t); d(t)), \quad (6)$$

where $v \in \mathbb{R}^{n_u}$ and $x \in \mathbb{R}^{n_y}$ are converted input and output signals. We impose the following assumption that makes the resulting OCP effectively solvable.

Definition 1: If functions $\Psi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_u}$, $\Phi : \mathbb{R}^{n_y} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_y}$, and $\Xi : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_z}$ satisfy

1) $\Psi(\cdot; d)$ and $\Phi(\cdot; d)$ are bijective, and
2) $\Xi(\cdot, \cdot; d)$ is convex,

for any fixed $d \in \mathbb{R}^{n_d}$, the system in (3) to (6) is called a *structured Hammerstein-Wiener model*.

B. Parameterization of bijective and convex mappings

In this section, we parameterize the bijective and convex mappings in Definition 1. In recent years, there have been many proposals in the area of flow-based generative models (Normalizing Flow) [16]. In this area, for the ease of the evaluation of the determinant of the Jacobian, neural networks with a special structure in which the Jacobian is a block triangular matrix are used. Since such property is unnecessary for the purpose of this paper, we extend the Bijective NN [17], which is a relatively old known model structure with higher degrees of freedom.

Definition 2: Let $\varphi^{(\cdot)} : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\eta} \to \mathbb{R}^{n_\xi}$ be a elementwise nonlinear function. Function $f(\xi; \eta, \theta)$ with $\theta = (\Omega^{(i)}, \beta^{(i)})_{i=1}^{L}$ where $\Omega^{(\cdot)} : \mathbb{R}^{n_\eta} \to \mathbb{R}^{n_\xi \times n_\xi}$, $\beta^{(\cdot)} : \mathbb{R}^{n_\eta} \to \mathbb{R}^{n_\xi}$ is said to be a bijective neural network (abbr. BNN) if

$$f(\xi; \eta, \theta) = \xi^{(L)}, \quad (7)$$

$$\xi^{(i)} = \varphi^{(i)}\left(\Omega^{(i)}(\eta)\xi^{(i-1)} + \beta^{(i)}(\eta), \eta\right) \ (i = 1, ..., L), \quad (8)$$

$$\xi^{(0)} = \xi, \quad (9)$$

$\Omega^{(i)}(\eta)$ is nonsingular and $\varphi^{(i)}(\cdot, \eta)$ is bijective for any i, η. If, in addition, $\Omega^{(i)}(\eta)$ is diagonal for any i, η, f is said to be a diagonal BNN. A BNN f is said to be C^r-diffeomorphic neural network (abbr. C^r-BNN) if $\varphi^{(i)}, \Omega^{(i)}, \beta^{(i)}$ $(i = 1, \ldots, L)$ are C^r with respect to ξ, η.

As a component of neural networks, φ, Ω, β are called as an activation function, weighting matrix, and bias vector, respectively. If f is a C^r-BNN, $f(\cdot; \cdot, \theta)$ is a C^r function.

Proposition 1: Let $f(\xi; \eta, \theta)$ be a BNN. Then, $f(\cdot; \eta, \theta)$ is bijective for any $\eta \in \mathbb{R}^{n_\eta}$.

Proof: The inverse mapping is given by

$$f^{-1}(\xi; \eta, \theta) = \xi^{(0)}, \quad (10)$$

$$\xi^{(i-1)} = \left(\Omega^{(i)}(\eta)\right)^{-1}\left(\varphi^{(i)^{-1}}\left(\xi^{(i)}, \eta\right) - \beta^{(i)}(\eta)\right) \quad (11)$$

$$(i = 1, ..., L),$$

$$\xi^{(L)} = \xi. \quad (12)$$

■

For convex mappings, [8] proposed a specific neural network that guarantees the convexity with respect to selected input variables. We utilize similar architecture with modification to make them differentiable.

Definition 3: Let $\varphi_\zeta^{(\cdot)}$ be monotonically non-decreasing convex C^r functions and $\varphi_\eta^{(\cdot)}$ be C^r functions. The parameter set $\theta = \{W_\zeta^{(\cdot)}, W_\xi^{(\cdot)}, W_\eta^{(\cdot)}, W_{\zeta\eta}^{(\cdot)}, W_{\xi\eta}^{(\cdot)}, W_{\eta\eta}^{(\cdot)}, b_\eta^{(\cdot)}, b_{\zeta\eta}^{(\cdot)}, b_{\xi\eta}^{(\cdot)}, b_{\eta\eta}^{(\cdot)}\}$, where all elements of $W_\zeta^{(\cdot)}$ are nonnegative. Then, $\Xi(y; d, \theta)$ is said to be a C^r partially input convex neural network (C^r-PICNN) if

$$\Xi(\xi, \eta; \theta) = \zeta^{(L)}, \quad (13)$$

$$\zeta^{(i)} = \varphi_\zeta^{(i)}\left(W_\zeta^{(i)}\left(\zeta^{(i-1)} \odot \mathrm{softplus}\left(v_\zeta^{(i)}\right)\right) + W_\xi^{(i)}\left(\xi \odot v_\xi^{(i)}\right) + v_\eta^{(i)}\right) \ (i = 1, \ldots, L), \quad (14)$$

$$\eta^{(i)} = \varphi_\eta^{(i)}\left(W_\eta^{(i)}\eta^{(i-1)} + b_\eta^{(i)}\right) \ (i = 1, \ldots, L-1), \quad (15)$$

$$v_\zeta^{(i)} = W_{\zeta\eta}^{(i)}\eta^{(i-1)} + b_{\zeta\eta}^{(i)} \ (i = 1, \ldots, L), \quad (16)$$

$$v_\xi^{(i)} = W_{\xi\eta}^{(i)}\eta^{(i-1)} + b_{\xi\eta}^{(i)} \ (i = 1, \ldots, L), \quad (17)$$

$$v_\eta^{(i)} = W_{\eta\eta}^{(i)}\eta^{(i-1)} + b_{\eta\eta}^{(i)} \ (i = 1, \ldots, L), \quad (18)$$

$$\zeta^{(0)} = \xi, \quad \eta^{(0)} = \eta. \quad (19)$$

By the same argument as in [8], we can show the following:

Proposition 2: Let $\Xi(\xi, \eta; \theta)$ be a C^r-PICNN. Then, $\Xi(\cdot, \eta; \theta)$ is a convex function for any θ, η, and $\Xi(\cdot, \cdot; \theta)$ is a class C^r function for any θ.

C. Identification

In this section, we discuss the identification procedure under the constraints that Ψ, Φ are C^r-BNN and Ξ is C^r-PICNN. Parameterization of such models are given by

$$v(t) = \Psi(u(t), d(t); \theta_\Psi), \quad (20)$$

$$\dot{x}(t) = A(d(t); \theta_A)x + B(d(t); \theta_B)v + c(d(t); \theta_c), \quad (21)$$

$$y(t) = \Phi^{-1}(x(t); d(t), \theta_\Phi), \quad (22)$$

$$z(t) = \Xi(x(t), v(t), d(t); \theta_\Xi). \quad (23)$$

In most identification methods of the H-W models, linear dynamics and nonlinear function (and its inverse) are determined repeatedly [18]. On the contrary, we propose a one-shot learning method based on an analytic inverse of the BNN.

The goal is to determine the parameter $\theta_\bullet$ based on the time series data of u, d, y, z. We begin with elimination of x and v. From (22), we have

$$x = \Phi(y; d, \theta_\Phi), \tag{24}$$

and its time derivative

$$\dot{x} = \frac{\partial \Phi(y;d)}{\partial y}\dot{y} + \frac{\partial \Phi(y;d)}{\partial d}\dot{d}. \tag{25}$$

Consequently, (21),(23),(24),(25) leads to

$$\begin{bmatrix} \dot{y} \\ z \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial \Phi}{\partial y}\right)^{-1} \left(A(d)\Phi + B(d)\Psi + c(d) - \frac{\partial \Phi}{\partial d}\dot{d}\right) \\ \Xi(\Phi, \Psi, d) \end{bmatrix}, \tag{26}$$

where Φ and Ψ denote $\Phi(y; d, \theta_\Phi)$ and $\Psi(u; d, \theta_\Psi)$, respectively. This means the identification reduces to the following minimization:

Problem 1: Suppose that time series data $\left\{u, y, \dot{y}, z, d, \dot{d}\right\}_i (i = 1, \ldots, N)$, where i is the label for the data, and a positive definite matrix K_e are given. Find

$$\theta = \left[\theta_\Psi^{\mathrm{T}}, \theta_\Phi^{\mathrm{T}}, \theta_\Xi^{\mathrm{T}}, \theta_A^{\mathrm{T}}, \theta_B^{\mathrm{T}}, \theta_c^{\mathrm{T}}\right]^{\mathrm{T}} \tag{27}$$

that minimizes

$$\frac{1}{N}\sum_{i=1}^{N} e_i^{\mathrm{T}} K_e e_i, \tag{28}$$

where e_i is the prediction error given by

$$\begin{bmatrix} \dot{y} - \left(\frac{\partial \Phi}{\partial y}\right)^{-1} \left(A(d)\Phi + B(d)\Psi + c(d) - \frac{\partial \Phi}{\partial d}\dot{d}\right) \\ z - \Xi(\Phi, \Psi, d) \end{bmatrix}. \tag{29}$$

This problem can be effectively solvable via Stochastic Gradient Decent methods. This procedure can be implemented by standard machine learning libraries such as Tensorflow as far as the Jacobian $\partial\Phi/\partial y$ is nonsingular.

Remark 1: The inverse function theorem tells us that, for any C^r-BNN Φ, $\partial\Phi/\partial y$ is nonsingular; see also the proof of Proposition 1. Since the set of nonsingular matrices is dense in the matrix field, $\Omega^{(i)}$ included in $\Phi(y; d, \theta_\Phi)$ remains nonsingular during learning almost surely when each element of $\Omega^{(i)}$ is parametrized independently. This can be proven rigorously by introducing a suitable probability space, which will be omitted due to page limitations.

III. Optimization-based Control

A. Uniqueness

In this section, we study how to construct an OCP that can guarantee the uniqueness of the solution, aiming at the implementation of online MPC via numerical optimization. Let us discretize the continuous-time system (3)-(6) with the sampling period Δ as

$$v_k = \Psi(u_k; d_k), \tag{30}$$
$$x_{k+1} = A_\Delta(d_k)x_k + B_\Delta(d_k)v_k + c_\Delta(d_k), \tag{31}$$
$$y_k = \Phi^{-1}(x_k; d_k), \tag{32}$$
$$z_k = \Xi(x_k, v_k; d_k), \tag{33}$$

with $A_\Delta = e^{A\Delta}, B_\Delta = (\int_0^\Delta e^{A(\Delta-\tau)}d\tau)B, c_\Delta = (\int_0^\Delta e^{A(\Delta-\tau)}d\tau)c$. In what follows, finite-time optimization problem with $k = 0, \ldots, n$ is considered, where n represents the prediction horizon length. Denote $U := \left[u_0^{\mathrm{T}}, \ldots, u_{n-1}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{nn_u}$, and V, X, Z, R similarly. $\Psi(V; \bar{d}) := [\Psi(v_0; \bar{d}), \ldots, \Psi(v_n; \bar{d})]^{\mathrm{T}}$, and $\Xi(X, V; \bar{d})$ similarly. We hereafter assume that $d_i = \bar{d}, r_i = \bar{r},\ i = 0, \ldots, n$.

Problem 2: Let $x_0, n, \bar{d}, \bar{r}$, and $\underline{u} < \overline{u}$ be given. Suppose that $Q \succ 0$, f_v is a convex C^2-function, f_z is a convex and monotonically non-decreasing C^2-function. Find U that minimizes

$$f := E^{\mathrm{T}}QE + f_u(U) + f_z(Z) \tag{34}$$
$$E := X - 1_n \otimes \Phi(\bar{r}; \bar{d}) \tag{35}$$

subject to $u_i \in \mathcal{U},\ i = 0, \ldots, n-1$.
The tracking error E is represented in terms of the internal state x. For example, take

$$\nabla y_0 := \frac{\partial \Phi^{-1}(x_0; \bar{d})}{\partial x}, Q_0 := (\nabla y_0)^{\mathrm{T}}(\nabla y_0), \tag{36}$$
$$Q := I_{n\times n} \otimes Q_0. \tag{37}$$

This choice of Q corresponds to $\|Y - R\|^2$ in (34) where Φ^{-1} is linearly approximated around the initial state $(x_0, \bar{d})$. Function f_u characterizes the input cost. We use f_z to describe a soft constraint for (2) in the form of a penalty term, e.g.,

$$f_z(Z) = \sum_{i=0}^{n-1}\sum_{j=1}^{n_z} \max(0, w_j(z_{ji} - \bar{z}_j)^3), \tag{38}$$

with weight w_j. This is just to avoid that the feasible set is empty and the hard constraint (2) can be dealt with similarly.

Theorem 1: Suppose that Ψ is a diagonal BNN, Φ is a BNN, Ξ is PICNN, and $B_\Delta(\bar{d_i})$ is nonsingular. Then, Problem 2 has a unique stationary solution.

Proof: By virtue of the invertibility of Ψ, we can regard V as decision variables, instead of U. Then, X is affine with respect to V:

$$X = \bar{A}(\bar{d})x_0 + \bar{B}(\bar{d})V + \bar{c}(\bar{d}), \tag{39}$$

$$\bar{A} := \begin{bmatrix} A_\Delta \\ A_\Delta^2 \\ \vdots \\ A_\Delta^n \end{bmatrix}, \bar{c} := \begin{bmatrix} c_\Delta \\ (A_\Delta + I)c_\Delta \\ \vdots \\ \left(\sum_{i=0}^{n-1} A_\Delta^i\right)c_\Delta \end{bmatrix}, \tag{40}$$

$$\bar{B} := \begin{bmatrix} B_\Delta & O & \cdots & O \\ A_\Delta B_\Delta & B_\Delta & \cdots & O \\ \vdots & & \ddots & \vdots \\ A_\Delta^{n-1}B_\Delta & A_\Delta^{n-2}B_\Delta & \cdots & B_\Delta \end{bmatrix}. \tag{41}$$

Also, the convexity of Ξ guarantees that Z is a convex function of V. The assumptions on B_Δ and Q imply that the Hessian of $E^{\mathrm{T}}QE$ is positive-definite. This shows that

$$f(V; x_0, \bar{d}, \bar{r}) := E^{\mathrm{T}}QE + f_u(\Psi(V; \bar{d})) + f_z(\Xi(X, V; \bar{d})) \tag{42}$$

is a strictly convex function of V. By the assumption that Ψ is a diagonal BNN, the constraint (1) can be represented as a box constraint on V, for which the feasibility set is not empty. Finally, the desired result follows from the theory of nonsmooth optimization [19]. ■

Remark 2: The fact that the optimal control problem reduces to a convex problem is attractive because the optimal solution can be obtained via gradient methods. However, some industrial applications require a very fast sampling period (e.g., less than 1 msec to update control input for application to vehicle engine control discussed in the next section) and low-performance computer implementations. In such cases, convexity alone is not sufficient for online optimization. For example, the evaluation of the gradient of the control cost (via backpropagation) for the model in [9] requires the same number of substitutions to the neural network as the horizon length n, which can be a bottleneck of the online implementation. On the other hand, concerning the proposed model, $\partial f/\partial V$ requires the neural network substitution only once since the state transition can be described by a matrix multiplication; see (39). This property reduces the computation burden significantly.

B. Continuity

The stationary solution is a function of $(x_0, \bar{d}, \bar{r})$, which can be viewed as a control law. We next study the continuity of this control law with respect to $(x_0, \bar{d}, \bar{r})$. Let us introduce Lagrange multiplier $\lambda \in \mathbb{R}^{2nn_u}$ such as

$$L(V, \lambda; x_0, \bar{d}, \bar{r}) := f(V; x_0, \bar{d}, \bar{r}) + \lambda^{\mathrm{T}} g(V; x_0, \bar{d}, \bar{r}). \quad (43)$$

Here, $g \leq 0$ represents $u_i \in \mathcal{U}, i = 0, ..., n-1$. Because of the convexity of Problem 2, a V is a globally optimal solution if and only if the KKT condition

$$\frac{\partial L}{\partial V} = 0,\ \lambda \odot g = 0,\ \lambda \geq 0,\ g \leq 0, \quad (44)$$

is satisfied. This inequality condition can be rewritten as an equality condition by using Fisher-Burmeister (FB) function [20] $\phi(a, b) := a + b - \sqrt{a^2 + b^2}$. That is, the KKT condition (44) is equivalent to

$$F(V, \lambda; x_0, \bar{d}, \bar{r}) := \begin{bmatrix} \left(\frac{\partial L}{\partial V}\right)^{\mathrm{T}} \\ \phi(V, -g) \end{bmatrix} = 0. \quad (45)$$

Consider the generalized Jacobian[1] ∂F of F [19] with respect to $\tilde{V} := \left[V^{\mathrm{T}}, \lambda^{\mathrm{T}}\right]^{\mathrm{T}}$. Then, we can show the continuity of the optimal control law.

Theorem 2: Denote $\tilde{V}^*$ the $(x_0, \bar{d}, \bar{r})$-dependent unique solution to (45). Then, the control law defined by

$$u(x_0, \bar{d}, \bar{r}) := \Psi^{-1}(v^*(x, \bar{d}, \bar{r}), \bar{d}), \quad (46)$$
$$v^*(x, \bar{d}, \bar{r}) := \tilde{V}^*_{1:n_u}, \quad (47)$$

is locally Lipschitz continuous, where $\tilde{V}^*_{1:n_u}$ represents the first n_u elements of $\tilde{V}^*$, i.e., the optimal input at the first time period.

Proof: From the specific structure of Problem 2, the unique existence of $\tilde{V}^*$ follows from

P1) linear independent constraint qualification is satisfied,

P2) strong second-order sufficient condition[2] is satisfied.

This also means any $H \in \partial F$ is nonsingular [21]. Then, the local Lipschitz continuity of $\tilde{V}^*$ follows from the implicit function theorem for locally Lipschitz continuous functions [19]. ■

As explained in Sections I and IV, the continuity of the control law is significant for implementation. In addition, this continuity guarantees the nonsingularity of $\partial F/\partial \tilde{V}$, which is an important property when we apply continuation or homotopy type methods [22]. For example, instead of solving the optimization at each time step, let us update the $\tilde{V}(t)$ according to $d\tilde{V}(t)/dt = -(\partial F/\partial \tilde{V})^{-1} F(\tilde{V}(t))$. This $\tilde{V}(t)$ realizes $dF(\tilde{V}(t))/dt = -F(\tilde{V}(t))$, and consequently $F(\tilde{V}(t)) \to 0$.

C. Control Barrier Function

In what follows, the reduction of decision variables is discussed for implementation in further computationally severe environments. Suppose that $\bar{d}$ and $\bar{r}$ are stationarily realizable in that there exists $\bar{v}$ such that $A(\bar{d})\bar{x} + B(\bar{d})\bar{v} + c(\bar{d}) = 0$ with $\bar{x} := \Phi(\bar{r}; \bar{d})$ and $\bar{u} := \Psi^{-1}(\bar{v}; \bar{d})$ satisfies (1). Then, the $v(t)$ that minimizes

$$J = \int_0^\infty (x - \bar{x})^{\mathrm{T}} Q (x - \bar{x}) + \|v - \bar{v}\|^2 dt \quad (48)$$

subject to (4) with $d(t) = \bar{d}$ is given by

$$v = \bar{v} + K(x - \bar{x}),\ \ K := -B^{\mathrm{T}} P \quad (49)$$

where P is the unique positive definite solution to matrix Riccati equation

$$PA + A^{\mathrm{T}} P - PB^{\mathrm{T}} BP + Q = 0. \quad (50)$$

However, $u = \Psi^{-1}(v(t), \bar{d})$ does not necessarily satisfy the constraints (1) and (2). To guarantee these constraints with lightweight calculation, we propose to use $\Xi(x, v, d)$ as a control barrier function.

Theorem 3: Suppose that $\Xi(x, v, d)$ does not depend on v. Let $\kappa_i(z), i = 1, \ldots, n_z$, be class $\mathcal{K}$ functions. Define

$$\begin{aligned} \mathcal{V}_i(x) := \Big\{ & v : \Psi^{-1}(v, \bar{d}) \in \mathcal{U}, \\ & \frac{\partial \Xi_{(i)}(x; \bar{d})}{\partial x}(A(\bar{d})(x - \bar{x}) + B(\bar{d})(v - \bar{v})) \\ & \leq \kappa_i(\bar{z}_{(i)} - \Xi_{(i)}(x; \bar{d}))\Big\},\ i = 1, \ldots, n_z, \end{aligned} \quad (51)$$

$\mathcal{V}(x) := \bigcap_{i=0}^{n_z} \mathcal{V}_i(x)$ and the control law given by

$$\mathcal{K}(x) := \Psi^{-1}(v^*(x); \bar{d}) \quad (52)$$
$$v^*(x) := \arg\min_{v \in \mathcal{V}} \|v - (\bar{v} + K(x - \bar{x}))\|^2 \quad (53)$$

If $\Xi(x(0); \bar{d}) \in \mathcal{Z}$, $\mathcal{V}(x(t))$ is nonempty and $u(t) = \mathcal{K}(x(t))$ is applied, then the constraints (1) and (2) are satisfied for all $t \geq 0$.

[1] The generalized Jacobian is needed since $\phi(a, b)$ is not differentiable at $a = b = 0$.

[2] $\gamma^{\mathrm{T}} \frac{\partial^2 L}{\partial V^2} \gamma > 0$ for any $\gamma \in \mathbb{R}^{2nn_u}$ satisfying $\gamma \neq 0$, $\frac{\partial g_i}{\partial V} \gamma = 0$ $(i \in I_+)$, where I_+ is the index set of non-active inequalities.

Proof: When $v(t) \in \mathcal{V}_i$,

$$\frac{d\Xi_{(i)}(x(t);\bar{d})}{dt} \le \kappa_i(\bar{z}_{(i)} - \Xi_{(i)}(x(t);\bar{d})). \tag{54}$$

This implies that the time derivative of $\Xi_{(i)}(x(t);\bar{d})$ is non-positive whenever $\Xi_{(i)}(x(t);\bar{d}) = \bar{z}_{(i)}$. Therefore, the forward invariance of $\mathcal{Z}$ follows from the standard theory for control barrier functions [23]. ■

In order to implement this control law, only (53) should be solved online. Note that this is a standard quadratic programming since $\Psi^{-1}(v,\bar{d}) \in \mathcal{U}$ is equivalent to a box constraint on v and the inequality constraints included in (51) are linear inequalities.

IV. Numerical Experiment

A. Engine Airpath System

We verify the effectiveness of the proposed method through a numerical experiment on control of engine airpath system depicted in Fig. 1. This is a heavily nonlinear MIMO system having input- and output-constraints. Control input $u = [u_{(1)}, u_{(2)}, u_{(3)}]^{\mathrm{T}}$ represents opening positions of variable nozzle, throttle, and Exhaust Gas Recirculation (EGR) valve. Exogenous input $d = [d_{(1)}, d_{(2)}]^{\mathrm{T}}$ is engine speed and fuel injection amount. A tracking reference is given for output $y = [y_{(1)}, y_{(2)}, y_{(3)}]^{\mathrm{T}}$, which represents boost pressure (intake manifold gas pressure), EGR ratio, and Pumping Mean Effective Pressure (PMEP). A ceiling value is given for the turbine speed z.

By using the data generated by a high-precision simulator equivalent to a real machine, we developed two nonlinear models. One is our proposed model (20)-(23). Concerning the activation function, we employ $\varphi^{(\cdot)}(\xi,\eta) = \sinh^{-1}\left(\alpha^{(\cdot)}(\eta) + \sinh(\xi)\right)$,where $\alpha^{(\cdot)}, \beta^{(\cdot)}$ are 3-layered neural networks whose activation function is $\tanh$. For Ξ, $\varphi_{\zeta}^{(\cdot)}, \varphi_{\eta}^{(\cdot)}$ is the softplus function, which is a convex and monotonically increasing C^{∞} function. The other, for comparison purpose, is a standard 3-layered neural network in the form of

$$\begin{aligned}&[y_{k+1}^{\mathrm{T}}, z_k^{\mathrm{T}}]^{\mathrm{T}}\\ &= W^{(2)}\tanh\left(W^{(1)}[y_k^{\mathrm{T}}, u_k^{\mathrm{T}}, d_k^{\mathrm{T}}]^{\mathrm{T}} + b^{(1)}\right) + b^{(2)},\end{aligned} \tag{55}$$

where $W^{(\cdot)}, b^{(\cdot)}$ is weighting matrices and bias vectors. Detailed architecture such as the number of nodes is tuned so that these models have the almost same degree of representation ability. The resulting model accuracy, which is not shown due to page limitation, is comparable.

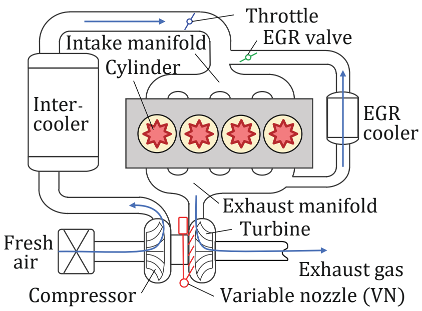


Fig. 1. Engine airpath system

B. Control System Design

The optimization criteria are described for both models. For the proposed model, we solve Problem 2 with $f_u = 0$, f_z in (38), and Q in (37). Then, the control law is given by (46))[3]. For the 3-layered neural network model, we solve the non-convex optimization problem

$$u(y,d,r) = U^{*}_{1:n_u}(y,d,r), \tag{56}$$

$$\begin{aligned} U^{*}(y,d,r) &= \arg\min_{U}\ \|E_y\|^2 + f_z(Z)\\ &\text{s.t.}\ \ \underline{u} \le u_k \le \overline{u}\ \ (k=0,\ldots,n-1), \end{aligned} \tag{57}$$

with $E_y := Y - R$ and f_z in (38). As a reliable nonlinear optimization solver, Sequential Quadratic Programming (SQP) method is applied to both problems with several initial conditions. Note that the convergence of the SQP method to the global optimizer is not guaranteed for non-convex problems.

C. Result

Fig. 2 shows the results of the SQP method for obtaining u for the two control laws described above, with only initial $y_{(1)}$ (for SQP) changed and the other values fixed at specific values. Fig. 2(a) and (b) show the results using the 3-layered neural network and the proposed method, respectively. The vertical axis is normalized to the upper and lower limits for each element of u. Thick lines (A) represent the result obtained by SQP with initial input u taken as the middle of upper and lower limits of u, while the initial input for thin lines (B) is given as the lower limits of u. The vertical axis is normalized to the upper and lower limits for each element of u.

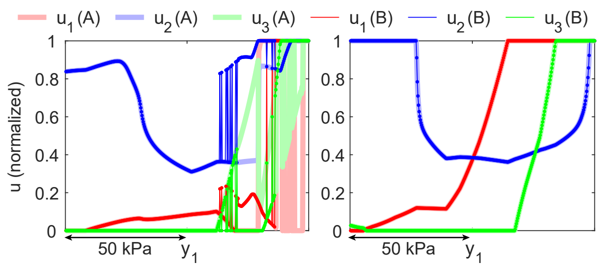
First, since the OCP for the control law for the 3-layered neural network is a non-convex problem, it can be confirmed that different solutions are obtained in (A) and (B) of Fig. 2(a). In addition, there are discontinuities where the solution bifurcates to different solutions. On the other hand, in Fig. 2(b), we can confirm that the solutions are unique and continuous, indicating the effectiveness of the method.

Finally, the results of the MPC simulation using each control law are shown in Fig. 3. The dashed and dotted lines in the figure show the results of MPC simulations using each control law. Observe that the trajectories largely depend on the choice of the initial input. In particular, the result of (A) shows the fluttering of the input and deviation from the target value, which may be caused by the discontinuity of the control law. On the other hand, the thin solid line is the result of the proposed method, and only one case is shown because the result does not depend on the initial value of the search. In this case, good target tracking and constraint satisfaction are achieved.

V. Conclusion

In this paper, we proposed a learning model structure that combines the Hammerstein-Wiener model with Bijective NN and Input Convex NN, which are extended to represent disturbance dependency and differentiability. Using this model, we

[3] The proposed control law $u(x,d,r)$ represents $u(\Phi(y,d),d,r)$.



(a) 3-layered NN based (b) Proposed

Fig. 2. Control law

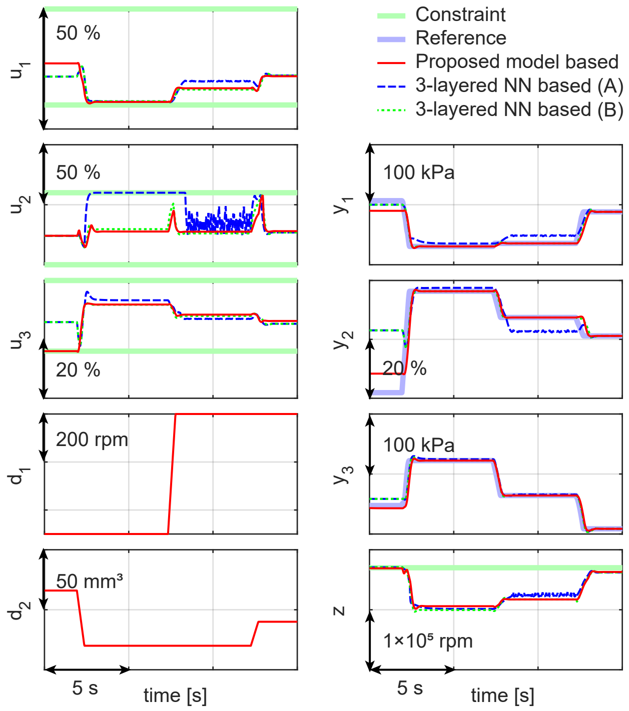


Fig. 3. Results of model predictive control

formulated the optimal tracking control problem with input-output constraints as a convex problem and guarantee the continuity of the control law.

The effectiveness of the method is demonstrated by numerical examples for an engine airpath system, which is a multi-input/output system with input/output constraints. This method is expected to have a wide range of industrial applications, including safety-critical applications because it provides a kind of reliability guarantee for machine learning-based control design.

The usefulness of the MPC methods described after Theorem 10 has already been confirmed by theory and experiment. The proposed modeling framework can also be extended to differentially flat systems. These results will be presented in a future publication. We are currently working on the relaxation of the diagonality of the input operator Ψ and the application of differentiable MPC.

References

[1] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *2004 American Control Conference*, 2004, p. ThA08.3.

[2] R. Hedjar, "Adaptive neural network model predictive control," *International Journal of Innovative Computing, Information and Control*, vol. 9, pp. 1245–1257, 2013.

[3] I. Lenz, R. A. Knepper, and A. Saxena, "DeepMPC: Learning deep latent features for model predictive control," in *Robotics: Science and Systems*, 2015.

[4] R. Moriyasu, S. Nojiri, A. Matsunaga, T. Nakamura, and T. Jimbo, "Diesel engine air path control based on neural approximation of nonlinear MPC," *Control Engineering Practice*, vol. 91, p. 104114, 2019.

[5] M. Ławryńczuk, "Suboptimal nonlinear predictive control based on neural Wiener models," in *Artificial Intelligence: Methodology, Systems, and Applications*. Springer Berlin Heidelberg, 2008, pp. 410–414.

[6] T. X. Nghiem, "Linearized Gaussian processes for fast data-driven model predictive control," in *2019 American Control Conference*, 2019, pp. 1629–1634.

[7] S. Gros, "Implicit non-convex model predictive control," in *Handbook of Model Predictive Control*. Birkhäuser Basel, 2019, pp. 305–333.

[8] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 146–155.

[9] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," in *arXiv*, 2019, p. 1805.11835.

[10] F. Bünning, A. Schalbetter, A. Aboudonia, M. H. de Badyn, P. Heer, and J. Lygeros, "Input convex neural networks for building MPC," *arXiv: 2011.13227*, 2020.

[11] K. P. Fruzzetti, A. Palazoğlu, and K. A. McDonald, "Nolinear model predictive control using Hammerstein models," *Journal of process control*, vol. 7, no. 1, pp. 31–41, 1997.

[12] A. L. Cervantes, O. E. Agamennoni, and J. L. Figueroa, "A nonlinear model predictive control system based on Wiener piecewise linear models," *Journal of process control*, vol. 13, no. 7, pp. 655–666, 2003.

[13] K. H. Chan and J. Bao, "Model predictive control of hammerstein systems with multivariable nonlinearities," *Industrial & engineering chemistry research*, vol. 46, no. 1, pp. 168–180, 2007.

[14] T. Patikirikorala, L. Wang, A. Colman, and J. Han, "Hammerstein-Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems," *Control Engineering Practice*, vol. 20, no. 1, pp. 49 – 61, 2012.

[15] M. Ławryńczuk, *Computationally efficient model predictive control algorithms*. Springer, 2014.

[16] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," *Proceedings of the 32nd International Conference on Machine Learning*, pp. PMLR 37:1530–1538, 2015.

[17] L. Baird, D. Smalenberger, and S. Ingkiriwang, "One-step neural network inversion with PDF learning and emulation," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 966–971.

[18] F. Giri and E. Bai, *Block-oriented nonlinear system identification*. Springer, 2010, vol. 1.

[19] F. Clarke, *Optimization and Nonsmooth Analysis*. John Wiley and Sons, 1983.

[20] A. Fischer, "An NCP-function and its use for the solution of complementarity problems," in *Recent Advances In Nonsmooth Optimization*. World Scientific, 1995, pp. 88–105.

[21] F. Facchinei, A. Fischer, and C. Kanzow, "Regularity properties of a semismooth reformulation of variational inequalities," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 850–869, 1998.

[22] T. Ohtsuka, "A continuation/GMRES method for fast computaion of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

[23] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," *2019 18th European Control Conference (ECC)*, pp. 3420–3431, jun 2019.