Linear model predictive control with lifted bilinear models by Koopman-based approach

Masaki Kanai\textsuperscript{a,b} and Masaki Yamakita\textsuperscript{a}

\textsuperscript{a}Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo, Japan; \textsuperscript{b}Hitachi, Ltd., Omika-cho, Hitachi-shi, Ibaraki, Japan

\section*{ABSTRACT}
This study proposes a linear Model Predictive Control (MPC) method that combines high prediction accuracy with low computational cost, using a lifted bilinear model based on Koopman theory. In recent years, there have been a growing interest in approaches to learning prediction models that determine the performance of MPC through lifting linearization and lifting bilinearization based on Koopman theory. In these methods, a linear or bilinear model reflecting nonlinear characteristics of the target system can be obtained by lifting the states to a higher dimensional space with observable functions (observables). In particular, lifting bilinearization provides more accurate models for the nonlinear input affine system, but when combined with MPC, it requires solving a nonlinear optimization problem, which is computationally expensive. Therefore, in this study, we formulate a linear MPC using a lifted bilinear model that can make accurate predictions for the input affine system, thereby realizing an MPC algorithm with high accuracy and low computational cost. In the proposed method, we first formulate a prediction error correction method for the lifted bilinear model by introducing constraints based on the observables and corrective inputs. Furthermore, we propose a linear MPC that can make accurate predictions using the lifted bilinear model by utilizing prior-predicted states based on the optimal solution at the previous time. We evaluate the effectiveness of the proposed method through numerical simulations using a planar drone.

\section*{1. Introduction}
Model Predictive Control (MPC) is gaining attention as a method that enables the systematic design of optimal control controllers for a variety of systems, taking into account constraints on states and inputs [1, 2]. Since MPC uses a prediction model of the target system to compute a sequence of control inputs that optimizes the states for a finite time horizon, the prediction accuracy of the states is crucial to achieve high-performance control that can accommodate the objective function.

In recent years, techniques using lifting linearization based on Koopman theory have been actively studied to obtain linear prediction models that reflect the characteristics of nonlinear systems [3]. Lifting linearization is a method that allows the state to be lifted to a higher dimensional space using the nonlinear outputs of the system, called observables, and to represent the time evolution of the observables as linear dynamics. The lifted linear model obtained in this way allows various theories for linear systems that have already been established to be applied to nonlinear systems.

In lifting linearization, a linear model is learned in a data-driven manner based on observables. At this time, it is important to select appropriate observables, and a variety of methods have been proposed for this purpose. One popular method is Extended Dynamic Mode Decomposition (EDMD) [4], which uses a dictionary of generic observables such as monomials up to a certain degree. Also, techniques that optimize or learn the observables itself have been proposed [5, 6]. In another direction, there are approaches such as Dual Faceted Linearization (DFL) [7–9] that utilize the physical knowledge of the target system to determine the observables.

Here, the challenge was that model learning by lifting linearization may not be sufficiently accurate, especially for input affine systems that describe dynamics such as aerial and ground vehicles. The main reason for this is that these systems add inputs via state-dependent rotational matrices, and linear models with state-independent input coefficient matrices cannot accurately represent their characteristics.

To address this problem, approaches have been proposed in which lifted bilinear models are learned for the input affine system in a data-driven manner, leading to nonlinear MPC with performance comparable to that of the case with a full knowledge of the target system [10]. This method exploits the fact that the Koopman Canonical Transformation (KCT) allows the input affine system to be globally transformed into a linear system in a high-dimensional space under certain conditions [11]. However, the use of the lifted bilinear
models as prediction models for MPC has faced challenges of computational cost because it requires to solve nonlinear optimization problems.

Therefore, in this study, we establish an accurate and low-computational-cost linear MPC algorithm utilizing lifted bilinear models that can make accurate predictions for an input affine system. In the proposed method, we first construct a prediction error correcting MPC that compensates for the prediction error of the lifted bilinear model by introducing constraints on the observables and corrective inputs. Then, we approximate the lifted bilinear model as a time-varying linear model using the prior-predicted states and linearize the nonlinear constraints. The prior-predicted states can be easily obtained by taking the advantage of the MPC algorithm’s feature of iteratively solving the optimization on the prediction horizon at each time.

The structure of the remainder of this paper is as follows. Section 2 describes preliminary knowledge of the lifting linearization and bilinearization, and how to identify the lifted bilinear model. In Section 3, we propose a prediction error correction method for MPC using the lifted bilinear model. In Section 4, we extend the method of Section 3 and propose an algorithm for linear MPC utilizing the lifted bilinear model. Section 5 presents the results of a numerical simulation that deals with a planar drone as an example of an input affine system to evaluate the effectiveness of the proposed method. Finally, Section 6 provides conclusion.

2. Lifted bilinear model

2.1. Koopman operator

The Koopman operator [12] was originally proposed for autonomous systems as a method to map nonlinear dynamics to an infinite-dimensional linear space. Later, efficient algorithms such as EDMD [4] were proposed to approximate nonlinear dynamics to finite-dimensional linear spaces in a data-driven manner.

Consider the discrete-time nonlinear autonomous system

\[ x_{k+1} = f_d(x_k), \]

where \( k \) is the time index and \( x_k \in \mathbb{R}^n \). Let \( \psi : \mathbb{R}^n \rightarrow \mathbb{C} \) be observables, which are nonlinear functions of the state. The Koopman operator \( \mathcal{K} \) is an infinite-dimensional linear operator acting on the observables, and the time evolution of the observables according to (1) can be expressed as linear dynamics:

\[ (\mathcal{K}\psi)(x_k) = \psi(f_d(x_k)). \]

Also, for a continuous-time nonlinear autonomous system

\[ \dot{x}(t) = f_c(x(t)) \]

with \( x \in \mathbb{R}^n \) as state, there exists a Koopman operator semigroup \( \mathcal{K}_{\Delta t} \) corresponding to a sample time \( \Delta t > 0 \). The infinitesimal generator \( \hat{\mathcal{K}} \) of this semigroup is

\[ (\hat{\mathcal{K}}\psi)(x(t)) = \nabla \psi \cdot f_c(x(t)), \]

which is referred to as the continuous-time Koopman operator.

In addition, Koopman eigenfunctions \( \phi : \mathbb{R}^n \rightarrow \mathbb{C} \) are functions corresponding to Koopman eigenvalues \( \lambda \in \mathbb{C} \) of \( \hat{\mathcal{K}} \) that evolves linearly along the solution of (3), whose future time values are represented by the following relation:

\[ \phi(x(t + \Delta t)) = (\mathcal{K}_{\Delta t}\phi)(x(t)) = e^{\lambda \Delta t}\phi(x(t)). \]

Although Koopman operator is generally infinite dimensional, it can be approximated as a finite dimensional matrix by data-driven methods such as EDMD using a finite number of selected observables. Such methods have also been extended to be applied to control systems with inputs [13, 14]. This allows us to identify the state evolution of a nonlinear system lifted in higher dimensional space by observables as linear dynamics, which is called lifting linearization. Furthermore, various linear control theories can be applied to the obtained linear model.

2.2. Lifting bilinearization

In this paper, we consider the following input affine systems as target systems:

\[ \dot{x} = f(x) + g(x)u, \]

where \( x \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \). Input affine systems can describe the dynamics of aerial and ground vehicles and mechanical systems. In such systems, the input generally act on state via a state-dependent matrix \( g(x) \). For example, in vehicle dynamics, the driving force affects the state via an attitude angle-dependent rotation matrix.

When the target system (6) is identified as a linear model using the lifting linearization, the coefficient matrix of the input is obtained as a state-independent constant matrix. Therefore, the effect of state-dependent coefficients on the input may not be appropriately captured. Especially for systems with large attitude angle changes, it may not be possible to obtain a sufficiently accurate model for the original system.

On the other hand, it is known that input affine systems can be transformed into bilinear systems under certain conditions using the Koopman Canonical Transformation (KCT) [10, 11].

Let \( (\lambda_i, \phi_i), i = 1, \ldots, n_c \) be eigenvalue-eigenfunction pairs of Koopman operator for the autonomous dynamics \( \dot{x} = f(x) \) of (6). Also, let \( \phi_i : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( z = T(x) = [\phi_1(x) \cdots \phi_N(x)]^T \), then KCT is defined as

\[ x = Cz. \]
\[ \dot{z} = Fz + \sum_{i=1}^{m} L_{gi} T(x)u_i, \]  
where \( F \in \mathbb{R}^{n_z \times n_z} \) is a diagonal matrix with \( \lambda_i \) in \( (i, i) \) elements. Note that in (7a), the state \( x \) of the original system (6) is represented by a linear combination of Koopman eigenfunctions.

At this time, if the space spanned by the Koopman eigenfunctions \( \phi_i, i = 1, \ldots, n_z \) forms an invariant subspace of \( L_{gi}, i = 1, \ldots, m \), then the input affine system (6) is bilinearizable in \( n_z \)-dimensional space. As a result, \( L_{gi} T(x) = G_i z, G_i \in \mathbb{R}^{n_z \times n_z} \), and the input affine system can be expressed in the following Koopman Bilinear Form (KBF) (see [11] for details):

\[ \dot{z} = Fz + \sum_{i=1}^{m} G_i z u_i, z \in \mathbb{R}^{n_z}. \]  

### 2.3. Identification of lifted bilinear model

Now we describe how to identify the lifted bilinear model represented by KBF when the target system (6) is unknown. First, assume that the following state and input sequence are obtained as the observation data of the target system:

\[ \mathcal{D} = \{x_k, u_k\}, k = 0, \ldots, M, \]  

where the sampling interval is \( \Delta t \). In the data acquiring experiment, a controller designed for stabilization or tracking may be applied to obtain the data. The data \( \mathcal{D} \) may be a collection of multiple data sets acquired through separate experiments. In such a case, when constructing matrices \( Z_+, Z_u \) discussed below, care should be taken not to associate data across different data sets.

Next, let the vector of observables be \( h(x) \in \mathbb{R}^{n_c} \). Here, the observables may be selected based on any knowledge of the physical model of the target system. In a more general way, a generic basis function, such as a monomial up to a certain degree, may be chosen. In that case, however, a sufficiently high order may have to be chosen to obtain a highly accurate model.

Here, a simple way to satisfy (7a) is to include the original state in the observables. Let \( \eta(x) \in \mathbb{R}^{n_{\eta}} \) be a vector of observables other than the original state \( x \), and assume the vector of observables in the form

\[ h(x) = \begin{bmatrix} x \\ \eta(x) \end{bmatrix}. \]  

Now we consider the following discrete-time lifted bilinear model with \( z_k \) as the lifted states:

\[ z_{k+1} = Az_k + \sum_{i=1}^{m} B_i z_k u_k(i), \]  

\[ x_k = Cz_k, \]  

where \( u_k(i) \) is the \( i \)-th component of \( u_k \). Given data \( \mathcal{D} \) and calculating \( z_k = h(x_k), k = 0, \ldots, M \), the coefficient matrices \( A, B_1, \ldots, B_m \) of (11a) can be obtained by solving the optimization problem

\[ \min_{A,B_1,\ldots,B_m} \sum_{k=0}^{M-1} \left\| z_{k+1} - \left( Az_k + \sum_{i=1}^{m} B_i z_k u_k(i) \right) \right\|^2. \]  

(12) is the optimization problem of determining the coefficient matrix so as to minimize the prediction error between samplings. The solution can be obtained by the least-squares method. Let \( H = [A B_1 \cdots B_m] \in \mathbb{R}^{n_z \times (m+1)} \) and construct matrices as

\[ Z_+ = [z_2, \ldots, z_M], \]  

\[ Z_u = \begin{bmatrix} z_1 & \cdots & z_{M-1} \\ z_1 u_1(1) & \cdots & z_{M-1} u_{M-1}(1) \\ \vdots & \ddots & \vdots \\ z_1 u_1(m) & \cdots & z_{M-1} u_{M-1}(m) \end{bmatrix}. \]  

Note that each column of \( Z_+ \) and \( Z_u \) is associated with the left and right sides of (11a) at each sampling. Then, the estimation of \( H \) can be calculated by

\[ \tilde{H} = Z_+ Z_u^T (Z_u Z_u^T)^{-1}. \]  

Also, from (10), the matrix \( C \) in (11b) can be obtained as

\[ C = [I_n \quad O_{n_y \times n_y}]. \]  

### 3. Prediction error correction for MPC with lifted bilinear model

#### 3.1. MPC with lifted bilinear model

In this section, we consider MPC using the lifted bilinear model obtained as the prediction model. First, we consider the formulation as a nonlinear MPC and construct the optimization problem to be solved at each time as follows:

\[ \text{minimize} \quad J_1 \]  

subject to

\[ z_0 = h(x_0) \]  

\[ z_{k+1} = Az_k + \sum_{i=1}^{m} B_i z_k u_k(i), \quad k = 0, \ldots, N_p - 1 \]  

\[ x_k = Cz_k, \quad k = 0, \ldots, N_p \]  

\[ A_c \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_c \geq 0, \quad k = 0, \ldots, N_p - 1, \]  

where \( N_p \) is the prediction horizon and \( x_t \) is the observed states at time \( t \). \( A_c, b_c \) are a matrix and a vector constituting linear constraints on the state and the
input. The objective function $J_1$ is given by

$$J_1 = \|x_{N_p} - x_{N_p}^r\|_S^2$$

$$+ \sum_{k=0}^{N_p-1} (\|x_k - x_k^r\|_Q^2 + \|u_k - u_k^r\|_R^2),$$

where $S, Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$ are positive semi-definite weight matrices and $x_k^r, k = 0, \ldots, N_p$ and $u_k^r, k = 0, \ldots, N_p - 1$ are the target state and the target input, respectively. We use the notation of weighted norm $\|x\|_Q = \sqrt{x^TQx}$.

In the optimization problem (16), the observed state $x_t$ at the current time is lifted to the lifted state space by the vector of observables $h(\cdot)$, and the evolution of the lifted states on the prediction horizon is predicted by the lifted bilinear model. Furthermore, by projecting the predicted lifted states at each step onto the original state space and evaluating it in the objective function, the optimal input sequence $u_k, k = 0, \ldots, N_p - 1$ is obtained.

### 3.2. Prediction error correction

The lifted bilinear model is expected to have higher prediction accuracy for the input affine system than the lifted linear model. However, since it is still an approximated model based on the data, some amount of error is inevitable. Furthermore, in this study, the lifted bilinear model is further approximated as a time-varying linear model to formulate a linear MPC, as described in detail in the next section. This may further increase the prediction error.

When the lifted model is used as a prediction model, especially for the nonlinear component $\eta(x)$ of the original states, the prediction error is likely to be large if the choice of the observables is not adequate. And since the prediction of the original states is calculated using the predictions of the nonlinear elements, the prediction errors may affect them in various ways. As a result, the objective function may not be properly evaluated and the performance of the closed-loop system with MPC may deteriorate.

To compensate for the effects of such errors, we consider state constraints based on the relationship of the observables to ensure that the values of the lifted states are properly balanced on the prediction horizon. Also, to avoid inconsistency in the state-input relationship in the state equation, a corrective input term is added to the component of the lifted states that correspond to nonlinear functions. The optimization problem with these considerations is formulated as follows:

$$\text{minimize } J_2$$

subject to

$$z_0 = h(x_t)$$

$$z_{k+1} = Az_k + \sum_{i=1}^{m} B_i z_k u_k(i)$$

$$+ \left[ O_{n \times 1} \begin{array}{c} \Delta u_k \end{array} \right], \quad k \in \{0, \ldots, N_p - 1\}$$

$$x_k = Cz_k, \quad k \in \{0, \ldots, N_p\}$$

$$z_k(n + 1 : n_z) = \eta(z_k(1 : n)), \quad k \in \{1, \ldots, N_p\}$$

$$A_c \begin{bmatrix} x_k \\ u_k \end{bmatrix} + b_c \geq 0, \quad k \in \{0, \ldots, N_p - 1\},$$

In the optimization problem (17), by (17e), we consider that $z_k(1 : n)$ corresponding to the original state and $z_k(n + 1 : n_z)$ corresponding to the portion of the lifted states augmented by the nonlinear function vector $\eta(x)$ satisfy the relationship of the observables at each time step. By solving this optimization problem, the optimal corrective inputs are computed so that the elements of the lifted states at each step satisfy the constraints, by which the prediction error due to the lifted model inaccuracy can be compensated. Since an improvement in the prediction accuracy of $z_t$ leads to an improvement in that of $x_t$, the control performance of the MPC is expected to be improved.

Here, the corrective input $\Delta u_k$ is computed to be as small as possible thanks to the objective function, so the optimization of the state trajectory is realized using the original input as much as possible. In addition, if the lifted model accurately approximates the nonlinear model, the corrective input becomes small enough. In other words, it is expected that the corrective input $\Delta u_k$ compensates for the uncertainty in the dynamics of the prediction model as necessary, but does not interfere with the optimization calculation of the original input.

Note that (17c) is a prediction model based on the lifted bilinear model, which encompasses the lifted linear model. Indeed, if the components of $\eta(x)$ contain a constant and all the corresponding elements of $B_i$ other than the constant are 0, then $B_i z_k \equiv B_i$ and the bilinear terms disappear. In such a case, the model becomes linear with respect to the input. Therefore, the proposed prediction error correction MPC can be equally applied to the case where the prediction model is a lifted linear model.
4. Linear MPC algorithm based on lifted bilinear model

4.1. Linearization with prior-predicted states

The problem (17) is a nonlinear optimization problem because it involves nonlinear constraints (17e) in addition to the bilinear prediction model (17c).

To formulate a linear MPC, we consider linearizing a bilinear prediction model (17c) and nonlinear constraints (17e) with the a previously obtained state sequence prediction for the prediction horizon. We call this prior-predicted states. Fortunately, MPC calculates the optimal state sequence at the same time it finds the optimal input sequence at each time. Therefore, the prior-predicted states can be easily obtained by utilizing the optimal state sequence calculated at the previous time. The details of this procedure are described in the next subsection.

Let $z_k^*, k = 0, \ldots, N_p$ be the prior-predicted states at the $k$th step of the prediction horizon. Also, the prior-predicted value of the original states corresponding to $z_k^*$ can be obtained by $x_k^* = Cz_k^*, k = 0, \ldots, N_p$.

First, the lifted bilinear model for the prediction horizon can be regarded as a time-varying linear system on the prediction horizon, using the prior-predicted states as follows:

$$z_{k+1} = Az_k + \sum_{i=1}^{m} B_i z_k^* u_k(i)$$

$$z_{k+1} = Az_k + \tilde{B}(z_k^*) u_k,$$  \hspace{1cm} (18)

where $\tilde{B}(z_k^*) = [B_1 z_k^* \cdots B_m z_k^*]$.

Next, the nonlinear constraint (17e) on the lifted state at the $k$th step of the prediction horizon can be reformulated as the following linear constraint by linear approximation around the prior-predicted states:

$$z_k(n+1:n_z) = \eta(x_k^*) + \frac{\partial \eta}{\partial x} \bigg|_{x=x_k^*} (z_k(1:n) - x_k^*).$$

From the above, given the prior-predicted states $z_k^*, k = 0, \ldots, N_p$, the optimization problem for the proposed LMPC is formulated as

$$\min_{u_k} J_2$$

subject to

$$z_0 = h(x_0)$$

$$x_k^* = Pz_k^*, \quad k = 0, \ldots, N_p$$

$$z_{k+1} = Az_k + \tilde{B}(z_k^*) u_k$$

$$+ \begin{bmatrix} O_{n \times 1} \\ \Delta u_k \end{bmatrix}, \quad k = 0, \ldots, N_p - 1$$

$$x_k = Cz_k, \quad k = 0, \ldots, N_p$$

$$z_k(n+1:n_z) = \eta(x_k^*) + \frac{\partial \eta}{\partial x} \bigg|_{x=x_k^*} (z_k(1:n) - x_k^*),$$  \hspace{1cm} (19)

$$k \in \{1, \ldots, N_p\}$$

$$A_z x_k + b_c \geq 0, \quad k \in \{0, \ldots, N_p - 1\},$$  \hspace{1cm} (20f)

$$A_z x_k + b_c = 0, \quad k = N_p$$  \hspace{1cm} (20g)

In the optimization problem (20), the only nonlinear calculation is calculation of the lifted states $z_0$ at the initial step (20b) and the other calculations all consist of a quadratic form objective function and linear constraints. Therefore, the formulation can be converted to the dense form for the quadratic programming problem, and solved with an efficient QP solver.

4.2. Obtaining prior-predicted states

Next, we explain how to obtain the prior-predicted states $z_k^*$. Given the optimal input sequence $u_k$ and $\Delta u_k$ obtained by solving the problem (20) at a certain time, using the bilinear prediction model (20d), it is possible to calculate the optimal value of the lifted states at each step $z_k^*, k = 0, \ldots, N_p$. Provided that the control interval coincides with the sampling interval $\Delta t$ of the discrete-time prediction model (20d), the prior-predicted states at the next time $t + \Delta t$ can be approximately predicted by shifting the optimal state sequence $z_k^*$ at the previous time by one step, except for the $N_p$th step. For the $N_p$th step, the value of $(N_p-1)$th step can be used. That is, the prior predicted state $z_k^*$ can be obtained as follows:

$$z_k^* = z_{k+1}^*, \quad k = 0, \ldots, N_p - 1$$

$$z_{N_p}^* = z_{N_p}^*.$$  \hspace{1cm} (20)

At the initial time of the control loop, the above method cannot be applied because no previous solution is available. Instead, we can use the lifted state of the system in all steps of the prediction horizon. That is, given the initial state $x_{00}$, the prior-predicted states are given by

$$z_k^* = h(x_{00}), \quad k = 0, \ldots, N_p.$$  \hspace{1cm} (22)

The prior-predicted states at the initial time also may be obtained by solving a nonlinear optimization problem with a nonlinear model in advance. This is expected to improve the control performance around the initial time because the approximation of the lifted bilinear model and the nonlinear constraints are more accurate.

For cases where the control interval and the sample time in the discrete-time prediction model do not match, the optimal state sequence at the previous time can be interpolated to virtually shift the optimal state sequence by the control interval.

4.3. Algorithm

From the discussion so far, the proposed linear MPC algorithm based on the lifted bilinear model can be summarized as Algorithm 1.
Algorithm 1 LMPC based on lifted bilinear model

Require: Reference trajectory \((x^r, u^r)\)

while Controller is running do
  if \(t=0\) then
    Set the prior-predicted states \(z_k^*\) by (22)
  end if
  Observe current state \(x_t\) of the system
  Solve the optimization problem (20) with the prior-predicted states \(z_k^*\)
  Update the prior-predicted states \(z_k^*\) by (21)
  Apply first input \(u_0\) to the system
end while

The major advantage of this algorithm is that it can simultaneously compensate for the prediction error and improve the model accuracy of the approximated lifted bilinear model. In other words, even if there is an error in the prior-predicted states at a certain time and the time-varying linear prediction model is not accurate enough, the resulting prediction error can be reduced by the corrective input so that the constraints on the lifted state are satisfied. Then, by utilizing the optimal state sequence with reduced prediction errors, the approximation accuracy of the lifted bilinear model and constraints on the lifted state is improved. This iterative accuracy improvement effect allows the proposed LMPC algorithm to achieve prediction performance equivalent to that of a lifted bilinear model (11), while being in the form of a quadratic programming problem which can be solved efficiently.

5. Numerical simulations

5.1. Target system

In this section, we evaluate the effectiveness of the proposed method through numerical simulations. As a target system, we consider a drone dynamics operating in the two-dimensional plane shown in Figure 1.

The system is an input affine nonlinear system whose dynamics is given by

\[
\frac{d}{dt} \begin{bmatrix} p_y \\ p_z \\ \theta \\ \omega \end{bmatrix} = \begin{bmatrix} v_y \\ v_z \\ -\frac{1}{m} \sin \theta - \frac{1}{m} \sin \theta \\ -\frac{1}{m} \cos \theta + \frac{1}{m} \cos \theta \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix},
\]

where \(p_y, p_z\) are coordinates in the \(y, z\) axis [m], \(\theta\) is the angle [rad], \(v_y, v_z\) are the velocity in the \(y, z\) axes [m/s], \(\omega\) is the angular velocity [rad/s] and \(T_1, T_2\) are the thrust forces [N] of the left and right propellers. Let \(x = [p_y, p_z, \theta, v_y, v_z, \omega]^T\) as the state and \(u = [T_1, T_2]^T\) as the input. Also, the physical parameters are shown in Table 1.

In the following sections, we first identify a lifted bilinear model for the target system (23). Then, using the obtained model, we apply the LMPC algorithm of the proposed method and evaluate its performance. Two control methods are considered for comparison: nonlinear MPC which uses the true system as the prediction model (NMPC), and linear MPC which uses a linear approximation model around the equilibrium point as the prediction model (Nominal LMPC).

Here, the linear approximation model for the Nominal LMPC is obtained from the first-order Taylor expansion around the equilibrium state \(\bar{x} = [0 0 0 0 0 0]^T\) and the equilibrium input \(\bar{u} = [\frac{mg}{l_{xx}}, \frac{mg}{l_{xx}}]^T\), as

\[
\frac{d}{dt} \bar{x} = A \bar{x} + B \bar{u} + W,
\]

where

\[
A = \begin{bmatrix} 0_{3 \times 3} & I_3 \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{3 \times 2} \\ \frac{1}{l_{xx}} 0_{3 \times 2} \end{bmatrix}, \quad W = \begin{bmatrix} O_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}.
\]

5.2. Model identification

To identify the lifted bilinear model, since the target system (23) is unstable, a stabilizing controller to the origin was applied to obtain the observation data. As a stabilizing controller, LQR was designed based on the nominal model (24). The sampling time for data acquisition was
0.05 [s], and the stabilization operation was performed from 100 random initial states to the origin. The initial states were generated by \( x_0 \sim \mathcal{N}(O_{6 \times 1}, \sigma_0) \), \( \sigma_0 = \text{diag}(1, 1, 0.1, 0, 0, 0) \). To keep exciting the system, a noise signal \( u_n \) was added to the stabilizing feedback input \( u_{st} = -Kx \) and applied to the target system. Here, \( K \) is the LQR gain and \( u_n \sim \mathcal{N}(O_{2 \times 1}, \sigma_u) \), \( \sigma_u = \text{diag}(1, 1) \).

For the observables \( h(x) = [x^T \eta^T(x)]^T \), we chose \( \eta(x) = [\cos \theta \sin \theta 1]^T \), based on the nonlinear components included in the dynamics of the target system (23).

To avoid uncontrollability around \( \theta = 0 \), the input was decomposed as \( u = \bar{u} + \Delta u \), and the difference \( \Delta u \) from the equilibrium input \( \bar{u} \) was considered as a new input to identify the lifted bilinear model. See Appendix 1 for a discussion of the effect on controllability due to the difference in the input choice.

With the identified lifted bilinear model, the prediction accuracy was evaluated using test data, and the results are shown in Figure 2. For the test data, 30 random initial states that were different from the training data were used and 2 [s] of simulations were performed for each trajectory. The predictions are close to the true values for the most of trajectories, showing that the nonlinearity of the target system is well captured by the identified model.

5.3. Control performance

5.3.1. Without state constraints

We evaluated the proposed method using the lifted bilinear model obtained by the identification. First, we show the simulation results for the case without state constraints.

In the simulation, the initial state was set to \([-15, 0, 0, 0, 0, 0]^T\) while the target state was set to the origin to control the drone to move horizontally. The simulation time was 10 [s], the sampling time was 0.05 [s], the prediction horizon length was 30 steps, and the weight matrices in the objective function \( J_2 \) were \( S = 100 \cdot I_6, Q = I_6, R = 0.1 \cdot I_2, \Delta R = 0.001 \cdot I_3 \). The input constraints were set to \( T_1, T_2 \geq 0 \). The effect of noise was not taken into account this time.

We used IPOPT with CasADi [15] as the interface for the NMPC solver, and the MATLAB quadprog function for the LMPC solver for proposed method and the Nominal LMPC. We conducted the simulations with MATLAB 2020b running on a Core i5 CPU@1.20GHz, 24.0 GB memory, Windows 10 PC.

Figures 3 and 4 show the results of the comparison of state and input with each method, respectively. The dashed lines in the figures indicate constrains. First, for the state shown in Figure 3, all methods successfully stabilized the target system to the origin. However, it can be seen that the behaviour of the Nominal LMPC using the linear approximation model differs significantly from that of the NMPC using the true model, while the proposed method yields behaviour that are almost similar to that of the NMPC. It can also be seen that in the proposed method, the behaviour is more similar to NMPC when there is correction for prediction error than when there is no correction. In particular, after around 2 [s], the proposed method almost matches the NMPC. There is a similar trend for the input shown in Figure 4. However, it can be seen that some differences appear for the proposed method compared to NMPC at the time period just after the beginning of the simulation.

Here, the graph of corrective input in the proposed method is shown in Figure 5. From the graph, we can see that the corrective input is relatively large just after the beginning of the simulation, especially until about 1 [s], but after that it decreases to small value and

**Figure 2.** Prediction by the identified model with test data.

**Figure 3.** Comparison of states.
eventually converges to zero. This is considered to be because there were large error in the prior-predicted states at the start of the simulation and also the large prediction error of the time-varying linear model. Then, as the accuracy of the prior-predicted states improve through iterative calculations, the prediction accuracy of the time-varying linear model improves and the corrective input approaches zero because they are no longer needed.

The performance of each method was quantitatively evaluated by the objective function

\[ J_e = \sum_{k=0}^{N} (\|x_k - \bar{x}\|_Q^2 + \|u_k - \bar{u}\|_R^2), \]  

(25)

where \( N \) is the number of data points obtained in the simulation period. The evaluation results, normalized with the NMPC value as 1, were 1.90 for the Nominal LMPC and 1.01 for the proposed method. These results indicate that the performance of the Nominal LMPC was deteriorated by 90% against the NMPC, whereas the proposed method can achieve control performance comparable to that of the NMPC.

5.3.2. With state constraints

Next, we show the simulation results for the case where \( |p_z| < 0.1 \) is set as the state constraints. In this case, the initial state was set to \([-10, 0, 0, 0, 0, 0, 0]^T\), and other simulation conditions were the same as in the previous case.

Figures 6 and 7 show the comparison results of the state and input by each method, respectively. Even with the state constraints, the behaviour obtained by the proposed method are closer to that of NMPC than Nominal LMPC. Particularly, a major difference is that the constraints on \( p_z \) are often violated in the Nominal LMPC, while the proposed method satisfies them. This is considered to be achieved by the better prediction accuracy of the proposed method.

The objective function values normalized by the NMPC value were 1.06 for the Nominal LMPC and 1.00 for the proposed method. These results show that the proposed method outperforms the Nominal LMPC even in the presence of state constraints, and can
achieve control performance comparable to that of the NMPC.

5.4. CPU time

Finally, we discuss the results of the computation time evaluation. Figure 8 shows the execution time of the optimization solver for each method for the prediction horizons $N_p = 10, 20, 30$. Figures 8(a) and 8(b) show the evaluation results without and with the state constraints, respectively.

The results show that the proposed method significantly reduces the computation time compared to NMPC, regardless of the presence of the state constraints. On the other hand, the proposed method takes longer computation time than the Nominal LMPC. Main reasons for this are considered to be that the proposed method requires a higher order model by the lifting and the constraints on the lifted states.

From the above results, it was shown through verification that the proposed method has the comparable control performance to NMPC, but at a lower computational cost than NMPC.

6. Conclusion

In this study, we proposed an LMPC algorithm that can achieve control performance comparable to NMPC with low computational cost, using the identified lifted bilinear model for the input affine system. In the proposed method, the identified lifted bilinear model is used as the prediction model in MPC, and prediction error correction is performed by introducing constraints based on observables and corrective inputs, and then linearizing the nonlinear prediction model and constraints to achieve accurate prediction in LMPC. Comparison with NMPC and Nominal LMPC in numerical simulation shows the superiority of the proposed method in terms of control performance and computational cost.

The proposed method can accelerate time-consuming optimization calculations in MPC calculations, which is expected to improve the real-time control performance of the actual plants. Although the effectiveness of the proposed method was verified by simulation using a simple model, designing a controller for a more practical plant and verifying it on the real environment remains a future task. In addition, it is also a future challenge to apply the parallel processing method [16] to reduce the computation time of NMPC.

Furthermore, although in this paper the lifted bilinear model was identified assuming the target plant as unknown, in many practical cases a simplified nominal model can be available. So, it is also planned to consider more actively utilizing the nominal model knowledge to extend the proposed method.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Masaki Kanai received the B.E. and M.E. degrees from Tokyo Institute of Technology in 2009 and 2011, respectively. He is currently a researcher in the Autonomous Driving Systems Research Department, Controls and Robotics Innovation Center, R&D Group of Hitachi, Ltd., as well as a doctoral student of the Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology. His research interests include nonlinear control, system identification and robotics.

Masaki Yamakita received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology in 1984, 1986, and 1989, respectively. From 1989 he was a research associate in the department of control engineering of the institute, and from 1993 he was a lecturer at Toyohashi University of Technology. He is currently an associate professor in the Department of Systems and Control Engineering of Tokyo Institute of Technology. His research interests include robotics, learning control,
and robust and nonlinear control. He is a member of IEEE and RSJ.

References

[1] Mayne DQ, Rawlings JB, Rao CV, et al. Constrained model predictive control: stability and optimality. Automatica. 2000;36(6):789–814.

[2] Grüne L, Pannek J. Nonlinear model predictive control. In: Nonlinear model predictive control. Springer; 2017. p. 45–69.

[3] Korda M, Mezić I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica. 2015;25(6):1307–1346.

[4] Williams MO, Kevrekidis IG, Rowley CW. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. J Nonlinear Sci. 2015;25(6):1307–1346.

[5] Williams MO, Hemati MS, Dawson ST, et al. Extend- ing dynamic mode decomposition. J Nonlinear Sci. 2019;29(1):149–160.

[6] Yeung E, Kundu S, Hodas N. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. Proc Am Control Conf. 2019;2019:4832–4839.

[7] Grüne L, Pannek J. Nonlinear model predictive control. In: Nonlinear model predictive control. Springer; 2017. p. 45–69.

[8] Igarashi Y, Yamakita M, Ng J et al. A Robust Method for Dual Faced Linearization (DFL). IFAC Conference on Decision and Control, CDC; 2017. p. 6107–6112.

[9] Yeung E, Kundu S, Hodas N. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. Proc Am Control Conf. 2019;2019:4832–4839.

[10] Korda M, Mezić I. Optimal construction of Koopman eigenfunctions for prediction and control. IEEE Trans Automat Control. 2020;65(12):5114–5129.

[11] Grüne L, Pannek J. Nonlinear model predictive control. In: Nonlinear model predictive control. Springer; 2017. p. 45–69.

[12] Williams MO, Hemati MS, Dawson ST, et al. Extend- ing dynamic mode decomposition. J Nonlinear Sci. 2019;29(1):149–160.

[13] Yeung E, Kundu S, Hodas N. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. Proc Am Control Conf. 2019;2019:4832–4839.

[14] Korda M, Mezić I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica. 2015;25(6):1307–1346.

[15] Williams MO, Hemati MS, Dawson ST, et al. Extend- ing dynamic mode decomposition. J Nonlinear Sci. 2019;29(1):149–160.

[16] Yeung E, Kundu S, Hodas N. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. Proc Am Control Conf. 2019;2019:4832–4839.

Appendix 1. Controllability depending on the input choice

We describe the difference in controllability depending on the choice of input in the target system (23) when approximating a linear system around $\theta = 0$. First, if the inputs is $u = [T_1 T_2]_T$, and having $\theta = 0$ in the coefficient matrices of the input, the approximated linear system is given as

$$
\frac{d}{dt} \begin{bmatrix} p_T \\ p_z \\ v_T \\ v_z \\ \theta \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta & -\frac{1}{I_x} \sin \theta & \frac{1}{I_x} \sin \theta \\ 0 & 0 & \frac{1}{I_z} \cos \theta & \frac{1}{I_z} \cos \theta \\ 0 & 0 & \frac{1}{I_y} \cos \theta & \frac{1}{I_y} \cos \theta \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.
$$

(A1)

In this case, using the coefficient matrices $A_1, B_1$ to compute the controllability matrix $M_{d1} = [B_1 A_1 B_1 \cdots A_2^{-1} B_1]$, the rank of $M_{d1}$ is 4 and the system (A1) is uncontrollable.

On the other hand, consider the case where the input is $\Delta u = [\Delta T_1 \Delta T_2]^T$. First, substituting $T_i = \frac{1}{2} + \Delta T_i, i = 1, 2$ into (23), we obtain

$$
\frac{d}{dt} \begin{bmatrix} p_T \\ p_z \\ v_T \\ v_z \\ \theta \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta & -\frac{1}{I_x} \sin \theta & \frac{1}{I_x} \sin \theta \\ 0 & 0 & \frac{1}{I_z} \cos \theta & \frac{1}{I_z} \cos \theta \\ 0 & 0 & \frac{1}{I_y} \cos \theta & \frac{1}{I_y} \cos \theta \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} = A_2 x + B_2 u.
$$

(A2)

For this system, with a linear approximation around $\theta = 0$ for the first term and $\theta = 0$ in the coefficient matrix of the input of the second term, the approximated linear system becomes

$$
\frac{d}{dt} \begin{bmatrix} p_T \\ p_z \\ v_T \\ v_z \\ \theta \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} \sin \theta & -\frac{1}{m} \sin \theta & -\frac{1}{I_x} \sin \theta & \frac{1}{I_x} \sin \theta \\ 0 & 0 & \frac{1}{I_z} \cos \theta & \frac{1}{I_z} \cos \theta \\ 0 & 0 & \frac{1}{I_y} \cos \theta & \frac{1}{I_y} \cos \theta \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} = A_2 x + B_2 u.
$$

(A3)

In this case, using the coefficient matrices $A_2, B_2$ to compute the controllability matrix $M_{d2} = [B_2 A_2 B_2 \cdots A_2^{-1} B_2]$, the rank of $M_{d2}$ is 6 and the system (A3) is controllable.