2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

K-BMPC: Derivative-based Koopman Bilinear Model Predictive Control For Tractor-trailer Trajectory Tracking With Unknown Parameters

Zehao Wang, Han Zhang, and Jingchuan Wang

***Abstract*—Nonlinear dynamics bring difficulties to controller design for control-affine systems such as tractor-trailer vehicles, especially when the parameters in the dynamics are unknown. To address this constraint, we propose a derivative-based lifting function construction method, show that the corresponding infinite dimensional Koopman bilinear model over the lifting function is equivalent to the original control-affine system. Further, we analyze the propagation and bounds of state prediction errors caused by the truncation in derivative order. The identified finite dimensional Koopman bilinear model would serve as predictive model in the next step. Koopman Bilinear Model Predictive control (K-BMPC) is proposed to solve the trajectory tracking problem. We linearize the bilinear model around the estimation of the lifted state and control input. Then the bilinear Model Predictive Control problem is approximated by a quadratic programming problem. Further, the estimation is updated at each iteration until the convergence is reached. Moreover, we implement our algorithm on a tractor-trailer system, taking into account the longitudinal and side slip effects. The open-loop simulation shows the proposed Koopman bilinear model captures the dynamics with unknown parameters and has good prediction performance. Closed-loop tracking results show the proposed K-BMPC exhibits elevated tracking precision with the commendable computational efficiency. The experimental results demonstrate the feasibility of K-BMPC.**

***Index Terms*—Koopman operator, tractor-trailer trajectory tracking, model predictive control**

I. Introduction

Tractor-trailer vehicles are widely used nowadays, particularly in fields such as agriculture and logistics due to their large cargo capacity and high transport efficiency. Despite their numerous benefits, achieving high-accuracy tractor-trailer tracking control is challenging, particularly for optimization-based trajectory planning methods [1], [2]. These methods discretize the kinematic constraints with a large sampling period to reduce the number of optimization variables. Nevertheless, the outcome of such methods would violate the dynamics of tractor-trailer to great extent. Therefore, a controller is needed for tractor-trailer vehicles to follow the trajectories with low tracking errors and high computational efficiency. Model Predictive Control (MPC) presents an attractive approach to trajectory tracking control, due to its adaptability to performance metrics and constraints [3]. However, MPC problem becomes difficult to solve in real time because of the nonlinear terms in model dynamics and long prediction horizons. Compared to nonlinear models, locally linearized models carry advantages in computational efficiency. However, their accuracy declines when the vehicle states move away from the point of linearization [4].

In addition to the locally linearized models, Koopman operator has been gaining attention for its ability to predict the flow of nonlinear dynamics using a infinite-dimensional linear model [5]. Extended Dynamic Mode Decomposition (EDMD) and is a data-driven tool to identify finite dimensional approximations of the Koopman operator and hence applied to approximate a variety of nonlinear dynamics [6]–[9]. However, the lifting functions used to construct EDMD-based Koopman models generally depend on the expert selections [10]. This means a lot of tuning in practice. To this end, deep neural networks are employed to overcome the difficulties in lifting function construction [11]–[13], however the interpretability of the model is limited.

Prior work on Koopman-based MPC design primarily focuses on combining linear MPC with the linear lifted models to increase the computational speed, however the accuracy of the lifted linear model is not guaranteed when the original states and controls are coupled. Koopman bilinear models are considered to balance the accuracy and computational speed, and the characteristics such as the bilinearizability and reachability are proved in [14]. However, the bilinear term brings difficulties in controller design. To address the difficulties, a few works try to linearize the bilinear models based on the current state of the system [15], but the linearization suffers from the disadvantage of local linearization. Folkestad et al. solve Koopman nonlinear MPC using sequential quadratic programming, however the Hessian of Lagrangian can not be computed directly from the bilinear structure [16], [17].

The challenge in tractor-trailer trajectory tracking using MPC is the nonlinearity in the dynamics. To address the challenge, we generalize derivative-based Koopman operators [9] to Koopman bilinear models, transform the tractor-trailer dynamic into a bilinear model. Then, we propose an iterative strategy to solve the Koopman bilinear MPC problems. Simulation and experimental results demonstrate strengths of our method. Our contributions are twofold.

All authors are with Department of Automation, Institute of Medical Robotics, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, and Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai 200240, China. Han Zhang is the corresponding author (email: zhanghan_tc@sjtu.edu.cn).

This work was supported by the National Key R&D Program of China under Grant 2022YFC3601403, and the ZF (China) Investment Co., Ltd

- We propose a lifting function construction method based on the derivatives of the dynamics, show that the corresponding infinite dimensional Koopman bilinear model is equivalent to the original control-affine system. Moreover, under the assumption that the derivative order is truncated, we analyze the state prediction error propagation and its bounds.
- We propose a Koopman bilinear MPC framework (K-BMPC) to solve the bilinear MPC problem using iterative quadratic programming. In K-BMPC, the bilinear model is linearized around the estimation of the state and control input. Then we transform the original MPC problem to a quadratic programming (QP) problem.

II. TRACTOR-TRAILER SYSTEM DYNAMICS

In this section, we present the dynamics and constraints of a tractor-trailer vehicle with unknown slip parameters. As shown in Fig.1, the tractor-trailer vehicle system is formed by a tractor The lengths of the tractor and trailer are denoted by l_0 and l_1 respectively, and l_H represents the hitching offset distance.

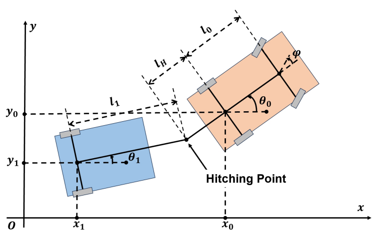


Fig. 1. Schematic of a tractor-trailer vehicle system.

Compared to classical tractor-trailer trajectory tracking control methods [18], [19], we take the velocity as a variable rather than a constant value. The longitudinal and side slips are also considered to achieve high accuracy control. Hence, the dynamic model of a tractor-trailer system with unknown slip parameters takes the form

$$\frac{d}{dt}\begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \\ \theta_1 \\ tan\varphi \\ v \end{bmatrix} = \begin{bmatrix} \mu v \cos\theta_0 \\ \mu v \sin\theta_0 \\ \mu v \tan\kappa\varphi / l_0 \\ \mu v(\sin\delta\theta - \tan\kappa\varphi\cos\delta\theta l_H / l_0)/l_1 \\ \omega \\ a \end{bmatrix}. \quad (1)$$

The state vector is defined as $x = [x_0, y_0, \theta_0, \theta_1, tan\varphi, v]^T$, where we denote (x_0, y_0) as the position of tractor. θ_0 and θ_1 are defined as the orientation angle of tractor and trailer respectively. φ is the front-wheel steering angle and v is the linear velocity. $\delta\theta = \theta_0 - \theta_1$ denotes the jack-knife angle. Further, The control vector is defined as $u = [\omega, a]^T$, where ω is the derivative of $tan\varphi$ and a is acceleration. μ and κ are unknown traction parameters representing the longitudinal and side slip influence respectively [3].

The aim of tractor-trailer tracking control is to keep the position and orientation of the tractor and trailer close to the reference trajectory. To this end, we define the output $y = [x_0, y_0, \theta_0, \theta_1, tan\varphi, v, x_1, y_1]^T$ by adding the position of the trailer (x_1, y_1) to the state vector. (x_1, y_1) are computed through the geometrical relationship.

$$\begin{aligned} x_1 &= x_0 - l_H\cos\theta_0 - l_1\cos\theta_1, \\ y_1 &= y_0 - l_H\sin\theta_0 - l_1\sin\theta_1, \end{aligned} \quad (2)$$

the dynamic of (x_1, y_1) is computed through (1) and (2).

In addition to the dynamics, we consider input and output constraints due to the physical limits. The input constraints take the form

$$-u_{max} \le u \le u_{max}, \quad (3)$$

where $u_{max} = [\omega_{max}, a_{max}]^T$ represent maximum steering angle velocity and linear acceleration respectively. The output constraints are defined as

$$\begin{aligned} -\tan\varphi_{max} \le &\tan\varphi \le \tan\varphi_{max}, \\ -v_{max} \le &v \le v_{max}, \\ -\delta\theta_{max} \le &\theta_0 - \theta_1 \le \delta\theta_{max}, \end{aligned} \quad (4)$$

where $\tan\varphi_{max}$ and v_{max} represent maximum steering angle and linear velocity respectively, $\delta\theta_{max}$ is maximum jack-knife angle for collision avoidance between the tractor and trailer.

III. KOOPMAN THEORY

In this section, we describe the Koopman operator, its identification methods and bilinear realization. Moreover, we elaborate the lifting function construction based on the derivatives of the dynamics, analyze the propagation and bounds of the prediction errors under truncated derivatives.

A. Koopman operator and EDMD for controlled system

For a control-affine continuous-time system of the form

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i = F(x,u), y = h(x), \quad (5)$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^{n_y}$is the output. Denote the observable $\phi \in \mathcal{F} : \mathbb{R}^{n_x \times m} \to \mathbb{R}$, where $\mathcal{F}$ is an infinite-dimensional function space that is composed of all square-integrable real-valued functions with compact domain $X \times U \subset \mathbb{R}^{n_x} \times \mathbb{R}^m$. The continuous time Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ describes the evolution of each observable in (5)

$$\mathcal{K}\phi(x,u) = \frac{\partial\phi}{\partial x}F(x,u).$$

Moreover, the corresponding discrete-time Koopman operator $\mathcal{K}_{T_s}$ with sampling time T_s takes the form

$$\mathcal{K}_{T_s} = e^{T_s\mathcal{K}}.$$

The dimension of Koopman operator $\mathcal{K}$ is typically infinite, so we construct a finite-dimensional approximation using Extended Dynamic Mode Decomposition (EDMD).

More specifically, EDMD approximates the discrete-time Koopman operator over a lifting function $\psi : X \times U \to \mathbb{R}^M$ using experimental data $\{x^{(k)}, u^{(k)}\}_{k=1}^K$. The approximated $\tilde{\mathcal{K}}_{T_s}$ is the solution to the optimization problem

$$\min_{\tilde{\mathcal{K}}} \sum_{k=1}^{K-1} \|\tilde{\mathcal{K}}^T\psi(x^{(k)}, u^{(k)}) - \psi(x^{(k+1)}, u^{(k+1)})\|_2^2. \quad (6)$$

B. Koopman Bilinear Model Realization

We focus on using continuous Koopman bilinear model with the following form to predict the evolution of states in (5)

$$\begin{aligned} \frac{d}{dt}z &= \bar{A}z + \bar{B}u + \sum_{j=1}^{m} u_j \bar{H}_j z, \\ y &= Cz, z(0) = \psi_x(x(0)), \end{aligned} \tag{7}$$

where $z \in \mathbb{R}^N (N >> n)$ is the lifted state, $\bar{A} \in \mathbb{R}^{N\times N}$, $\bar{B} \in \mathbb{R}^{N\times m}$, $\bar{H}_j \in \mathbb{R}^{N\times N}$ and $C \in \mathbb{R}^{n_y\times N}$. The initial condition of lifted state $z(0))$ is given by $z(0) = \psi_x(x(0))$, where

$$\psi_x(x) = [h^T(x), \phi_i(x) | i = 1, \ldots, N - n_y]^T.$$

The first n_y observables are defined as the output vector y for convenience in tracking the trajectory. In practice, we use the following discrete-time model

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k + \sum_{j=1}^{m} u_{j,k} H_j z_k, \\ y_k &= Cz_k, z_0 = \psi_x(x_0). \end{aligned} \tag{8}$$

To identify matrices $A, B, \{H_j\}$ in (8), we stack $\psi_x(x)$ with the product of $[\psi_x(x), 1]^T$ and u to obtain a $(m+1)N + m$ dimensional lifting function

$$\psi(x, u) = [\psi_x^T(x), u \otimes [\psi_x(x), 1]^T]^T.$$

Then, (6) is solved to compute an approximation of the discrete-time Koopman operator $\tilde{\mathcal{K}}_{T_s}$. We are able to obtain the matrices $A, B, \{H_j\}$ in (8) from $\tilde{\mathcal{K}}_{T_s}$

$$\tilde{\mathcal{K}}_{T_s} = \begin{bmatrix} A & H_1 & \cdots & H_m & B \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}^T.$$

$C = [I, 0]$ because the first n_y row of lifting function is the output of the system.

C. Derivative-based lifting function design for control-affine system

For control-affine system (5), we denote $F_i(x, u), f_i(x)$, $g_i(x)$ and $h_i(x)$ as i-th row of $F(x, u), f(x), g(x)$ and $h(x)$ respectively. More specifically, the 0-th order derivative is defined as the dynamic itself

$$f_i(x) + \sum_{j=1}^{m} g_{i,j}(x) u_j \triangleq F_{i,0}^{(0)}(x) + \sum_{j=1}^{m} F_{i,j}^{(0)}(x) u_j. \tag{9}$$

Then, we define $F_i^{(n+1)}(x) = [F_{i,0}^{(n+1)}(x), F_{i,1}^{(n+1)}(x), \cdots]$ from the n-th derivatives $F_i^{(n)}$

$$\begin{aligned} \frac{d}{dt} F_{i,l}^{(n)}(x) &= \frac{\partial F_{i,l}^{(n)}}{\partial x}^T [f(x) + \sum_{j=1}^{m} g_j(x) u_j] \\ &= \frac{\partial F_{i,l}^{(n)}}{\partial x}^T f(x) + \sum_{j=1}^{m} \frac{\partial F_{i,l}^{(n)}}{\partial x}^T g_j(x) u_j \\ &\triangleq F_{i,(m+1)l}^{(n+1)}(x) + \sum_{j=1}^{m} F_{i,(m+1)l+j}^{(n+1)}(x) u_j, \end{aligned} \tag{10}$$

where $F_{i,l}^{(n)}$ is the l-th element in $F_i^{(n)}$, each $F_{i,l}^{(n)}$ will induce $m+1$ elements in $F_i^{(n+1)}$. Then, we define $h_i^{(n)}(x) = [h_{i,0}^{(n)}(x), h_{i,1}^{(n)}(x), \cdots]$ in a similar way to (9), (10).

Based on the derivatives of dynamics, we construct an infinite-dimensional lifting function

$$\psi_x(x) = \left[h^{(n)}(x), F^{(n)}(x), n \geq 0\right]^T, \tag{11}$$

where the $F^{(n)}(x) \triangleq [F_1^{(n)}(x), \ldots, F_{n_x}^{(n)}(x)]$, $h^{(n)}(x) \triangleq [h_1^{(n)}(x), \ldots, h_{n_y}^{(n)}(x)]$. According to the necessary and sufficient conditions for bilinear realizations [15], the infinite dimensional Koopman realization (7) over (11) is bilinear.

In practice, we truncate the lifting function to obtain a finite dimensional bilinear model by limiting the order of derivatives less than or equal to a pre-defined degree ρ

$$\psi_x(x) = \left[h^{(n)}(x), F^{(n)}(x), 0 \leq n \leq \rho\right]^T. \tag{12}$$

The truncation in the derivative order would bring errors to the state and output estimation. The corresponding truncated bilinear model is of the form

$$\frac{d}{dt}\begin{bmatrix} \tilde{x}_i \\ F_{i,0}^{(0)}(\tilde{x}) \\ \vdots \\ F_{i,j}^{(0)}(\tilde{x}) \\ \vdots \\ F_{i,0}^{(\rho)}(\tilde{x}) \\ \vdots \end{bmatrix} = \begin{bmatrix} F_{i,0}^{(0)}(\tilde{x}) + \sum_{j=1}^{m} F_{i,j}^{(0)}(\tilde{x}) u_j \\ F_{i,0}^{(1)}(\tilde{x}) + \sum_{j=1}^{m} F_{i,j}^{(1)}(\tilde{x}) u_j \\ \vdots \\ F_{i,m+1}^{(1)}(\tilde{x}) + \sum_{j=1}^{m} F_{i,m+1+j}^{(1)}(\tilde{x}) u_j \\ \vdots \\ 0 \\ \vdots \end{bmatrix}, \tag{13}$$

where $\tilde{x}$ is the estimation of x. Note that $F_{i,0}^{(n)}(\tilde{x})(n > \rho)$ is set to 0. To evaluate the prediction error between the truncated bilinear model and the original control-affine model, we denote

$$\begin{aligned} F_i^{(0)}(x, u) &= F_{i,0}^{(0)}(x) + \sum_{j=1}^{m} F_{i,j}^{(0)}(x) u_j, \\ F_i^{(n+1)}(x, u) &= \frac{d}{dt} F_i^{(n)}(x, u). \end{aligned}$$

The propagation from $\tilde{x}_{i,k}$ to $\tilde{x}_{i,k+1}$ under truncated bilinear model (13) takes the form

$$\tilde{x}_{i,k+1} = \tilde{x}_{i,k} + F_i^{(0)}(\tilde{x}_k, u_k) T_s + \cdots + F_i^{(\rho)}(\tilde{x}_k, u_k) \frac{T_s^\rho}{(\rho)!}.$$

The propagation of state estimation error $e_{i,k} = x_{i,k} - \tilde{x}_{i,k}$ is

$$e_{i,k+1} = e_{i,k} + e_{i,k}^{(1)} T_s + \cdots + e_{i,k}^{(\rho+1)} \frac{T_s^\rho}{\rho!} + F_i^{(\rho+1)}(x_t, u_t) \frac{T_s^{\rho+1}}{(\rho+1)!},$$

where $e_{i,k}^{(j+1)} = F_i^{(j)}(x_k, u_k) - F_i^{(j)}(\tilde{x}_k, u_k)$ and $t \in [t_k, t_{k+1}]$. The bounds and propagation of errors are written in a similar way to [9]

$$e_{i,k} = \sum_{j=1}^{k-1} \sum_{p=1}^{\rho} e_{i,j}^{(p)} \frac{T_s^p}{p!} + \sum_{j=0}^{k-1} F_{i,j,j+1}^{(\rho+1)}(x, u) \frac{T_s^{\rho+1}}{(\rho+1)!},$$

$$|e_{i,k}| \le \frac{(kT_s)^{\rho+1}}{(\rho+1)!}F^{(\rho)}_{i,max}(x,u),$$

where $F^{(\rho+1)}_{i,max}(x,u)$ is the maximum of $F^{(\rho+1)}_i(x,u)$.

IV. BILINEAR MODEL PREDICTIVE CONTROL

A. Formulation of BMPC

Given a reference trajectory $\{y^r_k\}$. We formulate BMPC problem with bilinear models and linear constraints as

$$\begin{aligned}\min_{\substack{\{u_k\}\\\{z_k\}\\\{y_k\}}} &\sum_{k=0}^{N_p}(y_k - y^r_k)^T Q_k (y_k - y^r_k) + \sum_{k=0}^{N_p-1} u_k^T R u_k \\ s.t. z_{k+1} &= Az_k + Bu_k + \sum_{j=1}^{m} u_{j,k}H_j z_k, k = 0,\dots,N_p-1 \\ y_k &= Cz_k, z_0 = \psi_x(x_0), \qquad k=0,\dots,N_p \\ E_k y_k &+ F_k u_k \le l_k, E_{N_p}y_{N_p} \le l_{N_p}, \ k = 0,\dots,N_p-1\end{aligned} \tag{14}$$

where N_p is prediction horizon, $Q_k \in \mathbb{R}^{N_y \times N_y}$ and $R_k \in \mathbb{R}^{m\times m}$ are positive semidefinite, $E_k \in \mathbb{R}^{n_c\times N_y}$, $F_k \in \mathbb{R}^{n_c\times m}$ and $l_k \in \mathbb{R}^{n_c}$ define the input and output constraints in (3) and (4) for brevity, where n_c is the number of constraints. Due to the bilinear term in predictive model, there is no explicit solution to the problem, hence numerical scheme is needed. We linearize the bilinear term around the estimation of z_k and u_k

$$z_{k+1} = \hat{A}_k z_k + \hat{B}_k u_k - \sum_{j=1}^{m}\hat{u}_{j,k}H_j\hat{z}_k, \tag{15}$$

where $\hat{z}_k$ and $\hat{u}_k$ are estimations of z_k and u_k, and

$$\hat{A}_k = (A + \sum_{j=1}^{m}\hat{u}_{j,k}H_j), \hat{B}_k = (B + [H_1\hat{z}_k,\dots,H_m\hat{z}_k]). \tag{16}$$

B. Iterative quadratic programming

We linearize the bilinear term through (15), however the accuracy of the linearization is determined by the accuracy of $\{\hat{z}_k\}$. Thus, we design an iterative quadratic programming method to make $\{\hat{z}_k\}$ approach $\{z_k\}$ iteratively. At i-th iteration time, by replacing $\{z_k\}$, $\{u_k\}$ and $\{y_k\}$ in (14) with $\hat{z}_{i,k} + \Delta z_k$, $\hat{u}_{i,k} + \Delta u_k$ and $\hat{y}_{i,k} + \Delta y_k$ then applying (15), the BMPC problem is converted to a linearized MPC problem

$$\begin{aligned}\min_{\substack{\{\Delta u_k\}\\\{\Delta z_k\}\\\{\Delta y_k\}}} &\sum_{k=0}^{N_p}(\hat{y}_{i,k} + \Delta y_k - y^r_k)^T Q_k (\hat{y}_{i,k} + \Delta y_k - y^r_k) \\ &+ \sum_{k=0}^{N_p-1}(\hat{u}_{i,k} + \Delta u_k)^T R (\hat{u}_{i,k} + \Delta u_k) \\ s.t. \Delta z_{k+1} &= \hat{A}_{i,k}\Delta z_k + \hat{B}_{i,k}\Delta u_k + A_{i,k}\hat{z}_{i,k} \\ &\quad + \hat{B}_{i,k}\hat{u}_{i,k} - \hat{z}_{i,k+1}, \qquad k=0,\dots,N_p-1 \\ \hat{y}_{i,k} &= C\hat{z}_{i,k}, \Delta y_k = C\Delta z_k, \qquad k=0,\dots,N_p \\ E_k(\hat{y}_k &+ \Delta y_k) + F_k(\hat{u}_k + \Delta u_k) \le l_k, \ k=0,\dots,N_p-1 \\ E_{N_P}(\hat{y}_{N_P} &+ \Delta y_{N_P}) \le l_{N_P}, \hat{z}_{i,0} = \psi_x(x_0).\end{aligned} \tag{17}$$

Then, the linearized MPC problem is transformed into a dense QP problem and solved efficiently. After the solution is obtained, by adding the increment, we update $\{\hat{z}_k\}, \{\hat{u}_k\}$ at each iteration until the solution is converged

$$\begin{aligned}\hat{u}_{i+1,k} &= \hat{u}_{i,k} + \Delta u^*_{i,k}, \\ \hat{z}_{i+1,k} &= \hat{z}_{i,k} + \Delta z^*_{i,k},\end{aligned} \tag{18}$$

where $\{\Delta z^*_{i,k}\}$ and $\{\Delta u^*_{i,k}\}$ are the optimal solutions of (17). Besides the estimation update, a initial guess for the first iteration is needed. The initial guess of $\{\hat{z}_k\}$ and $\{\hat{u}_k\}$ for next time-step is derived from the solution of the current time-step

$$\begin{aligned}\hat{u}_k &= u^*_{k+1}, k = 0,\dots,N_p-2, \\ \hat{z}_k &= z^*_{k+1}, k = 0,\dots,N_p-1, \\ \hat{z}_{N_p} &= z^*_{N_p}, \hat{u}_{N_p-1} = u^*_{N_p-1}.\end{aligned} \tag{19}$$

C. K-BMPC algorithm

We summarize the K-BMPC algorithm as follows. At each time-step, the reference $\{y^r_k\}$ is updated. Then we linearize the bilinear MPC problem around the initial guess, solve the linearized problem by applying the iterative QP method. The first step of optimal control is deployed to the system and the initial guess in the next time-step is determined from (19).

Algorithm 1 Koopman BMPC

```
Input: Initial state x_0 and input u_0.
û ← u_0 ⊗ 1_{N_p}, ẑ ← ψ_x(x_0), ⊗1_{N_p+1}
while MPC is running do
    Update the reference {y^r_k}, iter ← 0
    while Not converged and iter ≤ iter_max do
        Form Â_{i,k}, B̂_{i,k} from (16)
        Solve (17) to get the increment Δu_i
        Update {ẑ_k} and {û_k} from (18)
        iter ← iter + 1
    end while
    Deploy the first control input of û to (1)
    Determine the initial guess of {ẑ_k} and {û_k} from (19)
end while
```

V. SIMULATION AND EXPERIMENTS RESULTS

A. Open-loop prediction

We test the performance of open-loop prediction in simulation. Basic parametric settings of the tractor-trailer system are listed in Table I. To collect data for identification, we generate a set of 50000 trajectories, each trajectory is

TABLE I
PARAMETRIC SETTINGS FOR TRACTOR-TRAILER IN SIMULATION

Parameter	Settings	Parameter	Settings
l_0	$3.6m$	φ_{max}	$0.6rad$
l_H	$1m$	v_{max}	$1m/s$
l_1	$6m$	a_{max}	$2m/s^2$
$\delta\theta_{max}$	$\pi/3$	ω_{max}	$2rad/s$
μ	$\sim U[0.97, 0.99]$	κ	0.94

simulated forward 40 steps using 4-th Runge-Kutta method with sampling time $T_s = 0.05s$.

A discrete-time Koopman bilinear model is approximated as introduced in Section III-B, the max derivative order is set to $\rho = 2$. More specifically, we compute the derivatives (10) under the assumption that $\mu, \kappa = 1$. The derivatives of v and $\tan\varphi$ in (1) are linear to control inputs. Thus we evaluated the errors of position orientation angle e_{x_0,y_0}, e_{x_1,y_1}, e_{θ_0}, e_{θ_1} respectively. As comparison, we consider the following commonly used models:

- KBM: Koopman bilinear model.
- LKBM: locally linearized Koopman bilinear model at z_0, which is used in [15].
- NM: the nominal model of (1), considering $\mu, \kappa = 1$.
- LLNM: locally linearized nominal model at x_0 and u_0.

We test the prediction performance of the above models under 1000 randomized initial states and control sequences, each initial state is simulated forward 20 steps. The errors (denoted as $e_{x_0,y_0}(10^{-4}m), e_{x_1,y_1}(10^{-4}m), e_{\theta_0}(10^{-4}rad)$ and $e_{\theta_1}(10^{-4}rad)$) are averaged to measure the errors between the truth and the prediction. Table II presents the mean prediction errors and Fig. 2 depicts propagation of state prediction errors for each model.

TABLE II
MEAN PREDICTION ERRORS UNDER RANDOMIZED CONTROL INPUTS

	e_{x_0,y_0}	e_{x_1,y_1}	e_{θ_0}	e_{θ_1}
KBM	**6.12**	**6.94**	**7.18**	**1.24**
LKBM	9.48	12.18	16.35	2.78
NM	50.44	42.45	21.31	4.90
LLNM	53.90	47.63	29.57	6.33

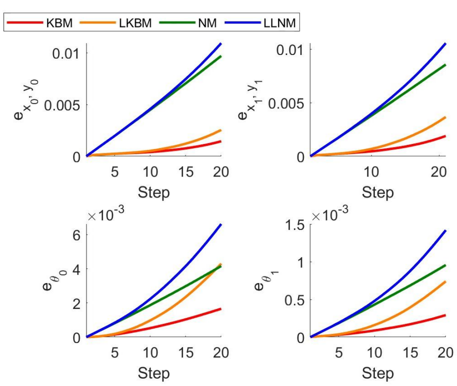


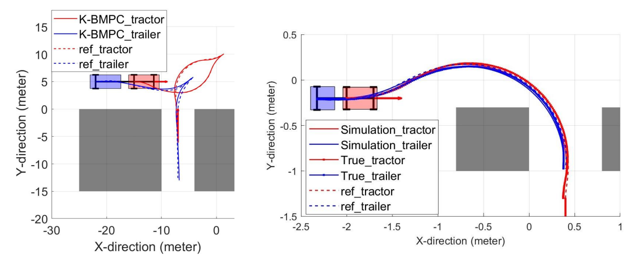
Fig. 2. Propagation of state prediction errors for different models.

As can be seen from Fig.2, Koopman bilinear model exhibits superior prediction performance over NM and LLNM. The performance improvements can be attributed to the identification of slip effects through the Koopman bilinear model. On the other hand, LKBM does not create noticeable discrepancies from the KBM and aligns closely with the ground truth. Nevertheless, as we shall see the linearization at z_0 brings extra prediction errors, we linearize KBM around $\hat{z}$ and $\hat{u}$ to reduce prediction errors.

B. Closed-loop tracking control

1) Simulation results: We implement our algorithm in Matlab 2022b. We select the predictive horizon as $N_p = 20$, relevant controller parameters are assigned as follows: $Q = diag[10, 10, 1, 1, 0, 0, 10, 10]$, $R = diag[0.01, 1]$, $Q_{N_p} = 10Q$, $iter_{max} = 3$. Nominal nonlinear MPC (NMPC) and iteratively locally linearized nominal MPC (LMPC) are chosen as benchmarks. NMPC uses the nominal model as predictive model, LMPC solves the linearized NMPC problem iteratively in a similay way to section IV-C. All simulations are run with the same settings.

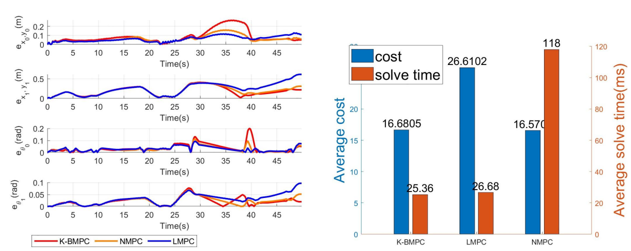
In simulation, the tractor-trailer needs to follow a reference trajectory and park vertically. To this end, the reference trajectory is given by the optimization-based method [2] with the discretization time $T \approx 0.5s$, thereby failing to satisfy the dynamics in a small sampling time $T = 0.05s$. The result trajectory is depicted in Fig. 3(a), and the trajectory tracking errors $e_{x_0,y_0}, e_{x_1,y_1}, e_{\theta_0}, e_{\theta_1}$ are shown in Fig. 4(a). The average MPC cost and solving time for all time-steps are presented in Figure 4(b), mean tracking errors of the controllers are displayed in Table III.



(a) Tractor-trailer tracking trajectory results in simulation

(b) Comparison of tracking trajectory results in simulations and experiments.

Fig. 3. Tractor-trailer trajectory when tracking a given path



(a) Tracking errors for all time-steps.

(b) Average cost and solving time for all time-steps.

Fig. 4. Tracking results comparison between different methods in simulation.

TABLE III
MEAN TRACKING ERRORS COMPARISON BETWEEN DIFFERENT METHODS

	$e_{x_0,y_0}(m)$	$e_{x_1,y_1}(m)$	$e_{\theta_0}(rad)$	$e_{\theta_1}(rad)$
K-BMPC	0.0861	**0.1783**	0.0347	**0.0213**
NMPC	0.0627	0.1903	**0.0281**	0.0223
LMPC	**0.614**	0.2389	0.0283	0.0303

From Table III, we find that the mean trailer position and orientation tracking errors of K-BMPC are lower than that of NMPC and LMPC, while NMPC has lower errors in tractor orientation tracking, LMPC has lower tractor position error. Overall, K-BMPC demonstrates superior tracking performance when compared to LMPC. The improvement in tracking accuracy is due to the fact that Koopman bilinear model identifies slip parameters from data and predicts the evolution of high order derivatives in tractor-trailer dynamics. Finally, the data from Figure 4(b) indicates that LMPC has a cost 59.53% higher than K-BMPC, while the cost of NMPC is sightly lower. Moreover, the average solving time for K-BMPC is significantly less than that of NMPC.

2) Experimental results: Furthermore, we test our algorithm on a miniature tractor-trailer model, as shown in Fig. 5. We use Vicon motion capture system to measure the global positions of the tractor and trailer and a virtual machine with 8GB RAM to compute control inputs. The tractor-trailer model receives and executives the control inputs from the virtual machine. Communication between the Vicon, virtual machine and the tractor-trailer miniature model is achieved by Robot Operating System (ROS). Fig. 3(b) and Table IV detail the tracking performance of the K-BMPC algorithm on a 20-second trajectory in simulation and real-world experiment.

Fig. 5. Miniature tractor-trailer system.

TABLE IV
MEAN TRACKING ERRORS IN SIMULATIONS AND EXPERIMENTS

	$e_{x_0,y_0}(m)$	$e_{x_1,y_1}(m)$	$e_{\theta_0}(rad)$	$e_{\theta_1}(rad)$
Real model	0.0378	0.0420	0.0258	0.0521
Simulation	0.0284	0.0297	0.0094	0.0089

The results show that while the average tracking errors in the real-model application are higher than that in the simulation, K-BMPC is still able to control the tractor-trailer to track the reference trajectory. The increased errors are likely attributed to factors such as the initial condition errors, topic transmission delay and the uncertainty and dead zone of the actuator.

VI. CONCLUSION

This paper presents K-BMPC, a Koopman-based bilinear MPC for tractor-trailer tracking control. We propose a derivative-based lifting function construction methods to approximate tractor-trailer dynamics with unknown parameters using a Koopman bilinear model. Moreover, when the derivative order is truncated, we analyze the state prediction error propagation and its bounds. K-BMPC linearizes the bilinear term around the estimation of lifted state and control input then solves Koopman bilinear MPC problems iteratively. Closed-loop tracking results show the proposed K-BMPC exhibits elevated tracking precision along with commendable computational efficiency. Compared to LMPC, K-BMPC keeps better tracking accuracy with a similar solving time. The real-world experiments help to verify the feasibility of the method. Future work will be devoted to identify a Koopman bilinear model for a tractor-trailer vehicle using experimental data, along with analyzing the robustness of Koopman bilinear MPC under the localization and modeling uncertainties.

REFERENCES

[1] Li, Bai, et al. "Tractor-trailer vehicle trajectory planning in narrow environments with a progressively constrained optimal control approach." IEEE Transactions on Intelligent Vehicles 5.3 (2019): 414-425.
[2] Li, Bai, et al. "Optimization-based maneuver planning for a tractor-trailer vehicle in a curvy tunnel: A weak reliance on sampling and search." IEEE Robotics and Automation Letters 7.2 (2021): 706-713.
[3] Kayacan, Erkan, et al. "Experimental validation of linear and nonlinear MPC on an articulated unmanned ground vehicle." IEEE/ASME Transactions on Mechatronics 23.5 (2018): 2023-2030.
[4] Gros, Sébastien, et al. "From linear to nonlinear MPC: bridging the gap via the real-time iteration." International Journal of Control 93.1 (2020): 62-80.
[5] Koopman, Bernard O. "Hamiltonian systems and transformation in Hilbert space." Proceedings of the National Academy of Sciences 17.5 (1931): 315-318.
[6] Korda, Milan, and Igor Mezić. "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control." Automatica 93 (2018): 149-160.
[7] Bruder, Daniel, et al. "Data-driven control of soft robots using Koopman operator theory." IEEE Transactions on Robotics 37.3 (2020): 948-961.
[8] Gupta, Shobhit, et al. "Koopman model predictive control for eco-driving of automated vehicles." 2022 American Control Conference (ACC). IEEE, 2022.
[9] Mamakoukas, Giorgos, et al. "Derivative-based koopman operators for real-time control of robotic systems." IEEE Transactions on Robotics 37.6 (2021): 2173-2192.
[10] Cibulka, Vít, Tomáš Haniš, and Martin Hromčík. "Data-driven identification of vehicle dynamics using Koopman operator." 2019 22nd International Conference on Process Control (PC19). IEEE, 2019.
[11] Han, Yiqiang, Wenjian Hao, and Umesh Vaidya. "Deep learning of Koopman representation for control." 2020 59th IEEE Conference on Decision and Control (CDC). IEEE, 2020.
[12] Shi, Haojie, and Max Q-H. Meng. "Deep Koopman operator with control for nonlinear systems." IEEE Robotics and Automation Letters 7.3 (2022): 7700-7707.
[13] Xiao, Yongqian, et al. "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles." IEEE Transactions on Intelligent Vehicles 8.1 (2022): 135-146.
[14] Goswami, Debdipta, and Derek A. Paley. "Bilinearization, reachability, and optimal control of control-affine nonlinear systems: A Koopman spectral approach." IEEE Transactions on Automatic Control 67.6 (2021): 2715-2728.
[15] Bruder, Daniel, Xun Fu, and Ram Vasudevan. "Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics." IEEE Robotics and Automation Letters 6.3 (2021): 4369-4376.
[16] Folkestad, Carl, and Joel W. Burdick. "Koopman NMPC: Koopman-based learning and nonlinear model predictive control of control-affine systems." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.
[17] Folkestad, Carl, Skylar X. Wei, and Joel W. Burdick. "Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking." 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022.
[18] Wu, Tong, and John Y. Hung. "Path following for a tractor-trailer system using model predictive control." SoutheastCon 2017. IEEE, 2017.
[19] Ljungqvist, Oskar, Daniel Axehill, and Anders Helmersson. "Path following control for a reversing general 2-trailer system." 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016.