On the Stabilization of Unstable Internal Dynamics: Model Predictive Control of Feedback Linearized Vehicle Kinematics

Bernd Juris[1] and Pu Li[1]

***Abstract*—In this paper, we propose an approach to stabilize unstable internal dynamics occurring from the feedback linearization of a kinematic vehicle model. Since the input affine model formulated has no full relative degree, a diffeomorphic transformation is performed to reduce the internal dynamics to an integration of the steering speed of the vehicle. We then introduce a Lyapunov candidate to ensure asymptotic stability. Incorporating this Lyapunov function, the resulting input-output linearized system is stabilized with a model predictive control (MPC) scheme. The new internal dynamics is posed as equality constraints in the feedback linearized MPC framework. Numerical results demonstrate that the proposed approach provides a satisfactory control performance and takes a significant lower computation time in comparison to a nonlinear MPC based on the original model.**

***Index Terms*—model predictive control, feedback linearization, Lyapunov stability, vehicle guidance, kinematic model**

I. Introduction

Nonlinear model predictive control (NMPC) offers improved performance for complex nonlinear systems. The importance of MPC in various applications is constantly increasing, for example, it becomes a well-known applicable method in different disciplines in the automotive sector [1]. However, NMPC faces computational challenges due to its requirement of numerical computation for solving nonlinear model equations and calculating gradients. Recent approaches provide numerical solutions through either collocation on finite elements [2] and [3] or direct multiple shooting methods [4]. Such discretizations often result in large-scale optimization problems within the NMPC framework. To address this issue, the method of combined multiple shooting and collocation [5] was developed to reduce the necessary number of time intervals while providing a high numerical accuracy. Although this method reduces the size of the underlying optimization problem, it still retains its nonlinear nature. In this study, we follow another approach, namely perform a feedback linearization to formulate a linear-quadratic and hence convex optimal control problem (OCP), which is used for the direct computation of the optimal control sequence for a linear MPC. The proposed method yields a trivial model structure that allows for a lean and simple implementation.

However, especially in robot kinematics or vehicle guidance applications, an issue remains: there will almost certainly be an unstable internal dynamics, if the system has no full relative degree. The aim of this study is to address this issue and ensure stability of the resulting linear MPC.

A. Feedback Linearization

Feedback linearization is a control strategy used to transform a nonlinear system into an equivalent linear one through a specific feedback mechanism. The primary goal is to design a control law that compensates the system's nonlinear behavior, allowing a standard linear controller design.

In this study, we consider a nonlinear system expressed as an input affine form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \tag{1a}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control vector, and $\boldsymbol{f}$ and $\boldsymbol{G}$ are nonlinear function vector and matrix, respectively. The output mapping is given as

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}). \tag{1b}$$

Feedback linearization aims to transform the system so that the relationship between the input and the output becomes linear. For this purpose, a transformation on the control input

$$\boldsymbol{u} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{\kappa}(\boldsymbol{x})\boldsymbol{v} \tag{2}$$

is to be found such that a new (virtual) input vector $\boldsymbol{v}$ can be expressed as a linear control input [6].

This method can be applied to both single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems. Its applicability mainly depends on the system's (vectorial) relative degree. Systems that provide a full relative degree can be reformulated via an input-state linearization. It transforms the entire state space of the system into one with fully linear dynamics. For non-full relative degree systems, the input-output linearization provides a linear formulation between the outputs $\boldsymbol{y}$ and the virtual inputs $\boldsymbol{v}$ without necessarily transforming all state variables. This casually leads to internal dynamics whose stability properties need to be analyzed.

B. Predictive Control

Predictive controllers can be applied to tackle instable internal dynamics or to take constraints into account. In [7] a cascaded control scheme for a pendubot is used, where the input-output feedback linearization forms the inner loop while a nonlinear predictive control is adopted for the outer loop.

[1] Bernd Juris and Pu Li are with the Faculty of Computer Science and Automation, Process Optimization Group, Ilmenau University of Technology, Ilmenau, Germany {bernd.juris, pu.li}@tu-ilmenau.de

This allows the control of the fast and unstable mode in the inner loop, while the outer loop handles the slower internal dynamics.

Deng et. al. [8] directly apply feedback linearization and model predictive control to a class of constrained problems. They found that the transformation of the input constraints lead to state dependent constraints. To preserve linearity within the MPC they approximate the states using a neural network.

The primary advantage of such approaches lies in the significant simplification of the MPC formulation achieved by feedback linearization to form a linear model. This not only simplifies the design process but also can provide guarantees for stability and convergence. Additionally, reducing the computational burden of NMPC within the underlying OCP presents a critical benefit for real-time applications.

C. Contributions

Our study focuses on the stabilization of unstable internal dynamics which appears when applying feedback linearization to a kinematic vehicle model. Using a closed-loop formulation, we achieve a simultaneous consideration of both internal and external dynamics. We ensure the system's asymptotic stability using a Lyapunov function, while the resulting strictly convex formulation of the underlying OCP guarantees convergence of the optimization algorithm. Therefore, this approach leads to reducing computational complexity while maintaining the benefits of nonlinear prediction of the system behavior for model predictive control.

Potentially, our closed-loop approach allows consideration of input constraints and its applicability to a broader class of systems with challenging dynamic properties.

II. Feedback Linearization

A kinematic bicycle model in the input affine form (1) reads

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_5 \cos\left(x_3 + \beta(x_4)\right) \\ x_5 \sin(x_3 + \beta(x_4)) \\ x_5 \tan(x_4) \cos(\beta(x_4)) L_2 \\ 0 \\ 0 \end{bmatrix}, \tag{3a}$$

$$\boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \tag{3b}$$

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T, \tag{3c}$$

with

$$\beta(x_4) = \arctan(L_1 \tan(x_4)). \tag{3d}$$

Fig. 1 shows the principle of the kinematic bicycle model. Table I lists the symbols with their physical meaning. In addition, in (3a) $L_1 = \frac{l_r}{l_f + l_r}$ and $L_2 = \frac{1}{l_f + l_r}$, respectively. We linearize this model using the transformation $\boldsymbol{z} = T(\boldsymbol{x})$ where

$$T(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_5 \cos(x_3 + \beta(x_4)) \\ x_2 \\ x_5 \sin(x_3 + \beta(x_4)) \\ x_4 \end{bmatrix}. \tag{4}$$

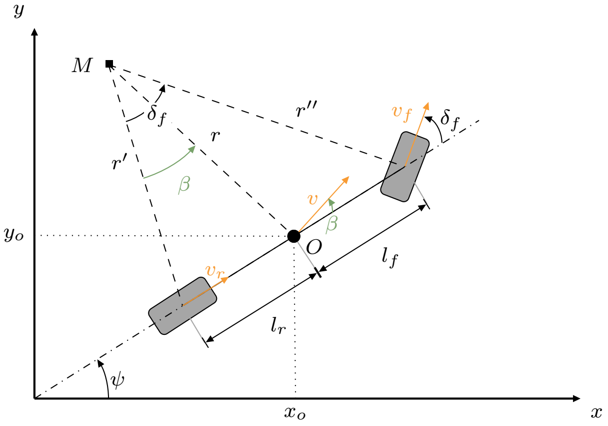

Fig. 1. Principle of the kinematic bicycle model

TABLE I
Physical Meaning of Symbols

Symbol	Physical meaning
x_1	x-position
x_2	y-position
x_3	Yaw angle ψ
x_4	Steering angle δ_f
x_5	Velocity v
O	center of gravity (COG)
l_f	Distance between COG and front axis
l_r	Distance between COG and rear axis
u_1	Acceleration
u_2	Steering speed

The first four entries in $T(\boldsymbol{x})$ in (4) are determined by the model dynamics (3a). The outputs yield a vectorial relative degree $\boldsymbol{\rho} = \begin{bmatrix} 2 & 2 \end{bmatrix}^T$. Since we consider a fifth order system there will appear an internal dynamics of order one. Hence, we define $\boldsymbol{z} = \begin{bmatrix} z_{\xi,1}, & z_{\xi,2}, & z_{\xi,3}, & z_{\xi,4} & z_{\eta,5} \end{bmatrix}^T$, where the index ξ indicates the assignment of the states to the external dynamics and η the assignment to the internal dynamics accordingly. The fifth entry of (4) is selected as x_4 to ensure it being a diffeomorphism.

Proof. A diffeomorphism $T(x)$ must satisfy the following two conditions:

(i) $T(\boldsymbol{x})$ must be bijective (one-to-one and onto);
(ii) $T(\boldsymbol{x})$ and its inverse $T^{-1}(\boldsymbol{z})$ must be continuously differentiable.

The bijectivity is ensured when $\boldsymbol{x}$ can be recovered uniquely from $\boldsymbol{z}$. In our transformation, we have $x_1 = z_{\xi,1}$, $x_2 = z_{\xi,3}$ and $x_4 = z_{\eta,5}$. The velocity x_5 and the yaw angle x_3 can be recovered as follows

$$x_5 = \sqrt{z_{\xi,2}^2 + z_{\xi,4}^2},$$
$$x_3 = \operatorname{atan2}(z_{\xi,4}, z_{\xi,2}) - \beta(z_{\eta,5}).$$

The differentiability of condition (ii) can be shown by evaluating the Jacobian of (4)

$$\frac{\partial T}{\partial \boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_1 & -a_1 \frac{\partial \beta}{\partial x_4} & a_2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -a_2 & -a_2 \frac{\partial \beta}{\partial x_4} & a_1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5}$$

where $a_1 = x_5 \sin(x_3 + \beta)$ and $a_2 = x_5 \cos(x_3 + \beta)$. The Jacobian has a full rank and is continuous for all $\boldsymbol{x}$, except when the velocity $x_5 = 0$ or the steering angle $x_4 = \pm\frac{\pi}{2}$, which would leave $\tan(x_4)$ undefined. The determinant of the Jacobian

$$\det\left(\frac{\partial T}{\partial \boldsymbol{x}}\right) = x_5 \tag{6}$$

is non-zero for $\forall x_5 \neq 0$. Thus, the transformation is locally invertible for any non-zero velocity. □

As a result, the diffeomorphism conditions are fulfilled if the vehicle is moving and the steering wheels are not at a 90-degree angle, which is given during normal operations of the vehicle. However, these singularities should be taken into account carefully in the practical implementation.

Given the outputs from (3c) we can calculate the entries of the decoupling matrix

$$\mathcal{D}(\boldsymbol{x}) = \begin{bmatrix} L_{g_1} L_f h_1 & L_{g_2} L_f h_1 \\ L_{g_1} L_f h_2 & L_{g_2} L_f h_2 \end{bmatrix}$$

using the Lie-derivatives for y_1:

$$L_f h_1 = x_5 \cos(x_3 + \beta), \tag{7a}$$

$$L_f^2 h_1 = -x_5 \sin(x_3 + \beta)\,\dot{x}_3, \tag{7b}$$

$$L_{g_1} L_f h_1 = \cos(x_3 + \beta), \tag{7c}$$

$$L_{g_2} L_f h_1 = -x_5 \sin(x_3 + \beta)\,\frac{\partial \beta}{\partial x_4} \tag{7d}$$

and for y_2:

$$L_f h_2 = x_5 \sin(x_3 + \beta), \tag{8a}$$

$$L_f^2 h_2 = x_5 \cos(x_3 + \beta)\,\dot{x}_3 \tag{8b}$$

$$L_{g_1} L_f h_2 = \sin(x_3 + \beta), \tag{8c}$$

$$L_{g_2} L_f h_2 = x_5 \cos(x_3 + \beta)\,\frac{\partial \beta}{\partial x_4}, \tag{8d}$$

where

$$\frac{\partial \beta}{\partial x_4} = \frac{L_1\left(1 + \tan^2(x_4)\right)}{1 + L_1^2 \tan^2(x_4)} \tag{8e}$$

Consequently,

$$\mathcal{D}(\boldsymbol{x}) = \begin{bmatrix} \cos(x_3 + \beta) & -x_5 \sin(x_3 + \beta)\,\frac{\partial \beta}{\partial x_4} \\ \sin(x_3 + \beta) & x_5 \cos(x_3 + \beta)\,\frac{\partial \beta}{\partial x_4} \end{bmatrix} \tag{9}$$

can be used to formulate the linear control law

$$\boldsymbol{u} = \mathcal{D}(\boldsymbol{x})^{-1} \begin{bmatrix} v_1 - L_f^2 h_1 \\ v_2 - L_f^2 h_2 \end{bmatrix}. \tag{10}$$

The external dynamics of the feedback linearized system has the following state-space form

$$\begin{bmatrix} \dot{z}_{\xi,1} \\ \dot{z}_{\xi,2} \\ \dot{z}_{\xi,3} \\ \dot{z}_{\xi,4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{\xi,1} \\ z_{\xi,2} \\ z_{\xi,3} \\ z_{\xi,4} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{11a}$$

and the corresponding outputs

$$\boldsymbol{y}_{\boldsymbol{z}} = \boldsymbol{h}(T^{-1}(\boldsymbol{z})) = \begin{bmatrix} z_{\xi,1} & z_{\xi,3} \end{bmatrix}^T, \tag{11b}$$

which are identical with (3c). The internal dynamics is expressed as

$$\dot{z}_{\eta,5} = u_2. \tag{12}$$

The input vector of the original systems $\boldsymbol{u}$ can be retrieved through (10) using the inverse transformation

$$T^{-1}(\boldsymbol{z}) = \begin{bmatrix} z_{\xi,1} \\ z_{\xi,3} \\ \operatorname{atan2}(z_{\xi,4}, z_{\xi,2}) - \beta(z_{\eta,5}) \\ z_{\eta,5} \\ \sqrt{z_{\xi,2}^2 + z_{\xi,4}^2} \end{bmatrix}. \tag{13}$$

To facilitate readability we use from now on $z_\eta := z_{\eta,5}$.

III. Stabilizing Internal Dynamics

It is easily recognized that the internal dynamics (12) is not inherently stable, since it is an integrator fed with the input u_2. However, stable internal dynamics must be guaranteed since the control sequence generated by the MPC theoretically would only account for external dynamics. Our idea is to ensure a stable behavior within an MPC formulation that considers both internal and external dynamics. For this purpose, we formulate the discrete-time representation of the external dynamics

$$\boldsymbol{z}_\xi^{k+1} = \underbrace{\begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{z}_\xi^k + \underbrace{\begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{v}^k \tag{14a}$$

and the internal dynamics

$$z_\eta^{k+1} = z_\eta^k + T_s u_2(\boldsymbol{z}_\xi^k, \boldsymbol{z}_\eta^k, \boldsymbol{v}^k) \tag{14b}$$

where T_s is the sampling time and $u_2(\cdot)$ the nonlinear steering dynamics expressed as

$$u_2 = \frac{-\left(1 + {L_1}^2 \tan^2(\boldsymbol{z}_\eta^k)\right)\cos^2(\boldsymbol{z}_\eta^k)}{L_1\left((z_{\xi,2}^k)^2 + (z_{\xi,4}^k)^2\right)} \left(v_1^k z_{\xi,4}^k - v_2^k z_{\xi,2}^k\right). \tag{15}$$

For a feedback linearized MPC (FLMPC), we formulate the underlying discretized OCP as follows

$$\min_{\boldsymbol{V}} \quad \sum_{k=0}^{N-1} \left(\left|\left|\boldsymbol{y}_{\boldsymbol{z}}^k - \boldsymbol{y}_r^k\right|\right|_{\boldsymbol{Q}}^2 + \left|\left|\boldsymbol{v}^k\right|\right|_{\boldsymbol{R}}^2 + w_\eta (z_\eta^k)^2 \right) \tag{16a}$$

$$\text{subject to} \quad (14a) - (15), \tag{16b}$$

$$\boldsymbol{y}^k = \boldsymbol{h}\left(z_{\xi,1}^k, z_{\xi,3}^k\right), \tag{16c}$$

$$\left|z_\eta^k\right| \leq \delta_{f,\max}, \ \forall \ k \in [0, 1, \ldots, N-1] \tag{16d}$$

where the weighting matrices $\boldsymbol{Q} \succeq 0$, $\boldsymbol{R} \succ 0$ and $w_\eta \geq 0$ and the steering angle constraint $\delta_{f,\max} < \frac{\pi}{2}$.

To establish asymptotic stability under the MPC framework, we use a Lyapunov-based approach to account for both the linear external and the nonlinear internal dynamics. Following the ideas of [9], the following proof leverages the system's feedback-linearized structure and controllability properties.

Proof. We define the Lyapunov candidate at the i-th time sample of the MPC as

$$\mathcal{V}_N\left(\boldsymbol{y}_z^i, z_\eta^i\right) = \min_{\boldsymbol{V}^i} \left[\sum_{k=0}^{N-1} \ell\left(\boldsymbol{y}_z^{i+k}, z_\eta^{i+k}, \boldsymbol{v}^{i+k}\right) \right] \tag{17}$$

using the stage cost

$$\ell\left(\boldsymbol{y}_z^k, z_\eta^k, \boldsymbol{v}^k\right) = \left|\left|\boldsymbol{y}_{\boldsymbol{z}}^k - \boldsymbol{y}_r^k\right|\right|_{\boldsymbol{Q}}^2 + \left|\left|\boldsymbol{v}^k\right|\right|_{\boldsymbol{R}}^2 + w_\eta (z_\eta^k)^2$$

to penalize the deviations from the reference as well as minimizing the virtual inputs and the internal state.

The choice of $\boldsymbol{Q} \succeq 0$ and $w_\eta \geq 0$ ensures $\mathcal{V}_N > 0, \ \forall \boldsymbol{y}_z \neq \boldsymbol{y}_r$ and $z_\eta \neq 0$. Obviously $\mathcal{V}_N\left(\boldsymbol{y}_r, 0\right) = 0$, which guarantees positive definiteness of the chosen function.

To ensure a monotonic decrease of $\mathcal{V}_N$, let

$$\boldsymbol{V}^* = \left[(\boldsymbol{v}^0)^*, \quad (\boldsymbol{v}^1)^*, \quad \ldots, \quad (\boldsymbol{v}^{N-1})^*\right]$$

be the optimal input sequence at sampling time i. At $i+1$, we construct a candidate sequence by appending a feasible but not necessarily optimal control sequence to a by-1-shifted control

$$\begin{aligned} \widetilde{\boldsymbol{V}} &= \left[\widetilde{\boldsymbol{v}}^0, \quad \widetilde{\boldsymbol{v}}^1, \quad \ldots, \quad \widetilde{\boldsymbol{v}}^{N-1}\right] \\ &= \left[(\boldsymbol{v}^1)^*, \quad (\boldsymbol{v}^2)^*, \quad \ldots, \quad (\boldsymbol{v}^{N-1})^*, \quad \boldsymbol{0}\right], \end{aligned}$$

to obtain

$$\widetilde{\mathcal{V}}_N\left(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}\right) = \mathcal{V}_N\left(\boldsymbol{y}_z^i, z_\eta^i\right) - \sum_{k=0}^{N-1} \ell\left(\boldsymbol{y}_z^{i+k}, z_\eta^{i+k}, \widetilde{\boldsymbol{v}}^{i+k}\right).$$

By optimality of $\mathcal{V}_N\left(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}\right)$,

$$\mathcal{V}_N\left(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}\right) \leq \widetilde{\mathcal{V}}_N\left(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}\right)$$

holds. Thus

$$\mathcal{V}_N\left(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}\right) - \mathcal{V}_N\left(\boldsymbol{y}_z^i, z_\eta^i\right) \leq -\sum_{k=0}^{N-1} \ell\left(\boldsymbol{y}_z^{i+k}, z_\eta^{i+k}, \widetilde{\boldsymbol{v}}^{i+k}\right),$$

which satisfies the Lyapunov condition:

$$\Delta\mathcal{V}_N \leq -\lambda_{\min}(\boldsymbol{Q}) \left|\left|\boldsymbol{y}_z^k - \boldsymbol{y}_r\right|\right|^2 - w_\eta \left(z_\eta^k\right)^2,$$

where $\lambda_{\min}(\boldsymbol{Q}) > 0$ is the smallest eigenvalue of $\boldsymbol{Q}$ and $\Delta\mathcal{V}_N = \mathcal{V}_N(\boldsymbol{y}_z^{i+1}, z_\eta^{i+1}) - \mathcal{V}_N(\boldsymbol{y}_z^i, z_\eta^i)$.

Consequently, by the LaSalle Invariance Principle [6], all trajectories will converge to the largest invariant set where $\boldsymbol{y}_z = \boldsymbol{y}_r, \ z_\eta = 0$ and thus $\Delta\mathcal{V}_N = 0$.

□

Remarks:

(i) The external dynamics (14a) is controllable, since $\operatorname{rank}\left(B, AB, A^2B, A^3B\right) = 4$.

(ii) The prediction horizon N needs to be sufficiently long to ensure the desired controller performance. However, a finite horizon is applicable since the control sequence resulting from the MPC leads to a decrease in the Lyapunov function (see Remark (iv)).

(iii) The initial steering angle $u_2(t_0)$ needs to satisfy $|u_2(t_0)| \leq \delta_{f,\max}$, and the MPC ensures recursive feasibility

(iv) In (15) the MPC formulation stabilizes the internal dynamics because the optimal solution of the underlying OCP leads to $(v_1^k z_{\xi,4}^k - v_2^k z_{\xi,2}^k) < L_1((z_{\xi,2}^k)^2 + (z_{\xi,4}^k)^2)$ for $z_\eta \to 0$ and $\boldsymbol{v} \to \boldsymbol{0}$ and we ensure the nonlinear terms remain bounded i.e., $|z_\eta| = |\delta_f| \leq \delta_{f,\max} < \frac{\pi}{2}$ as in (16d) and $z_{\xi,2}^2 + z_{\xi,4}^2 \neq 0$ according to (5), (6) and (13).

(v) The choice of w_η will affect the performance as indicated in the Lyapunov condition.

IV. Simulation Results

We consider two tracking scenarios to illustrate the properties of the proposed FLMPC approach and compare with those of the NMPC based on the original kinematic model (3). The OCP underlying the NMPC reads

$$\min_{\boldsymbol{u}} \int_{t_0}^{t_f} (\|\boldsymbol{y} - \boldsymbol{y}_r\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}\|_{\boldsymbol{R}}^2) \, dt, \tag{18}$$

$$\text{s.t. } (3a) - (3d)$$

where $\boldsymbol{Q} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$ and $\boldsymbol{R} = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix}$ are chosen the same for both OCP formulations (16) and (18) in order to imitate a practically relevant tuning and $w_\eta = 1$ in (16a). The prediction horizon is set to a length of 10 seconds to ensure a sufficient long time of prediction. The sampling time $T_c = 0.1$ second and the prediction horizon is shifted by T_c after each sample.

The results of a first experiment are shown in Fig. 2 which compares the transient behavior of the FLMPC with that of the NMPC. The vehicle is starting with an initial deviation of $\Delta y = -1$ meter from its target trajectory heading along the x-axis at a speed of 30 kilometer per hour (kph). It is clearly seen that both approaches lead to satisfactory transitions, where the NMPC shows a faster reaction. This is mainly due to the impact of tuning the weighting parameters. Using $\boldsymbol{R}$ in both objectives leads to a weighting of the steering speed and the acceleration in the heading direction of the vehicle for the

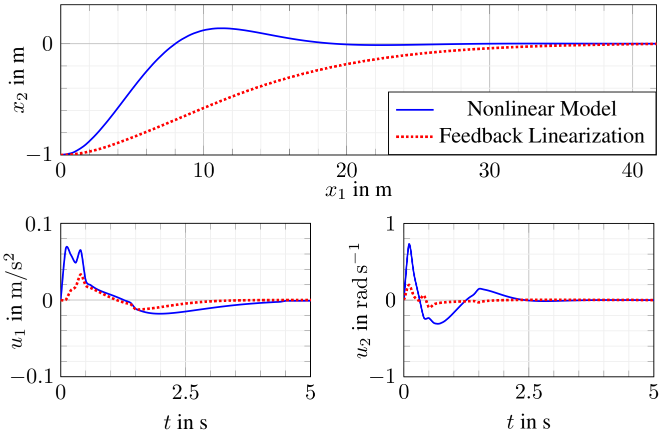

Fig. 2. Transient tracking behavior

NMPC (18), whereas for the FLMPC (16) the accelerations in x- and y-direction are weighted directly. Furthermore, the additional term $w_\eta z_\eta^2$ in (16a) aims at minimizing the yaw rate causing a less aggressive transient behavior. Accordingly, Fig. 3 shows that the steering angle generated by FLMPC is significantly smaller than the one generated by NMPC.

The second experiment considered is a moose-test where the vehicle starts on a given trajectory at the origin of the coordinate system with an initial velocity of 30 kph heading along the x-axis. The target trajectory mimics an evasive maneuver at a velocity of 30 kph as shown in Fig. 4 with the dashed line. According to the underlying OCPs for both approaches, no box constraints or general (in-)equalities were considered except the ones mentioned in section III for the FLMPC. Fig. 4 shows dissatisfactory tracking behavior by the NMPC, whereas the FLMPC is able to track the trajectory using the same configuration. Note that this issue cannot be resolved by adding a term for the steering angle u_2 in the objective function of the NMPC. However, the NMPC can be provided with bounded inputs implemented as hard constraints to work properly. To stabilize the tracking behavior, we constrain the NMPC input variables as follows

$$\begin{bmatrix} -7 \\ -0.16 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} 6 \\ 0.16 \end{bmatrix} \tag{19}$$

which match the range of the steering signals corresponding to the FLMPC. The result shown in Fig. 5 allows us to draw the same conclusion as that from Fig. 2. The NMPC is slightly more aggressive but both approaches yield satisfactory tracking behavior.

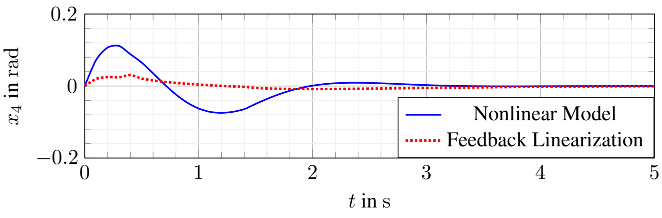

Fig. 3. Transient behavior of steering angle x_4

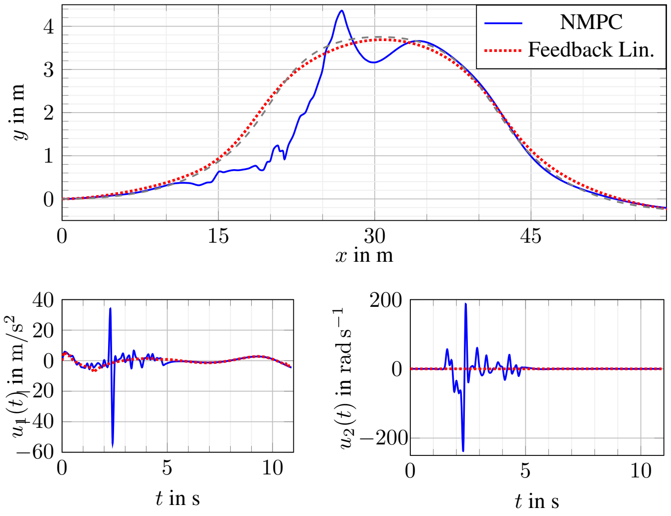

Fig. 4. Tracking behavior of a moose-test

In terms of computational requirements, the FLMPC allows a significant reduction of nonlinearities within the underlying OCP, since there is only one nonlinear state z_η left. The following benchmark considers 10 samples of both the linear and nonlinear MPC implementations. The experiment runs both MPC frameworks for about 11 seconds (i.e. physical time). To gain insight into the computational times of the underlying OCPs, its measured computing times are presented in an analogous manner. The measurement refers to (18) as N-OCP and to (16) as FL-OCP. Both implementations employ IPOPT [10] for solving the underlying OCP using the interfaces of InfiniteOpt [11] and JuMP [12].

Table II shows the minimum and maximum time taken to run both MPC for the 11 seconds of the system and the computation time of the underlying OCPs, respectively. It

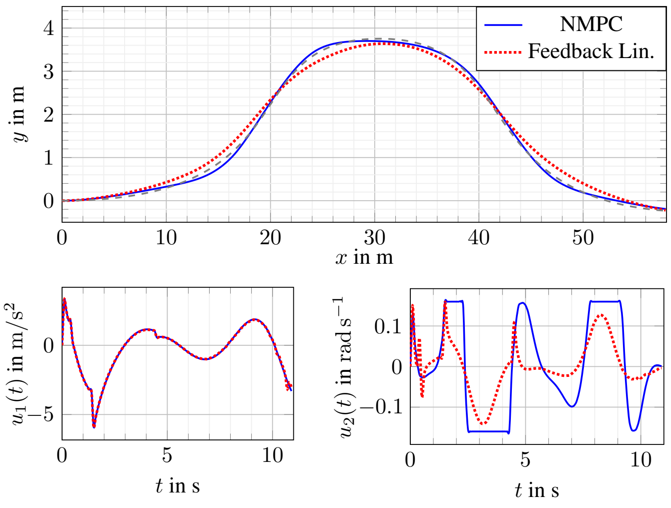

Fig. 5. Tracking behavior with bounded inputs for NMPC

TABLE II
CPU BENCHMARK

	Computation Time[a]		
Scen.	***Candidate***	***min ... max***	***mean*** $\pm\,\sigma$
Lane-keeping	NMPC	10.7 s ... 11.1 s	10.9 s ± 93.1 ms
	FLMPC	2.0 s ... 2.3 s	2.1 s ± 85.3 ms
	N-OCP	96.7 ms ... 102.1 ms	99.9 ms ± 977.4 μs
	FL-OCP	19.7 ms ... 24.4 ms	21.7 ms ± 0.7 ms
Moose-test	NMPC	14.9 s ... 15.5 s	15.1 s ± 184.1 ms
	FLMPC	2.2 s ... 2.4 s	2.3 s ± 67.5 ms
	N-OCP	140.9 ms ... 148.6 ms	143.9 ms ± 1.6 ms
	FL-OCP	20.9 ms ... 26.5 ms	23.5 ms ± 1.1 ms

[a] Computer setup: AMD Ryzen 7 7735U, Windows 11.

shows that, FLMPC takes significantly less CPU time than NMPC. And the CPU time used for each update in the FL-OPC is less than the sampling time (0.1 second), making it applicable for real-time implementation. In the moose-test, it can be seen that each update in the N-OCP needs about 150 milliseconds. Thus, it cannot be used for the online implementation.

The last column in II gives an idea about how strong the computation time differs between samples, given as standard deviation σ. This shows the influence of the scheduler of the operating system on the simulation setup. Based on the mean computation time, we can conclude that the time reduction achieved by the feedback linearized MPC lies within the range of 80.8 % to 84.8 %. Furthermore, the results indicate that the NMPC is sensitive to the type of maneuver, as the moose-test results in a longer computation time than the first tracking scenario. On the contrary, the FLMPC demonstrates a more consistent computation time across both experiments, i.e. it remains always below the MPC sampling time, thereby ensuring the real-time capability.

It should be noted that the computation time shown in Table II is resulted by using the same solver (i.e. IPOPT) to solve both the N-OPC and the FL-OPC problems. It means that the advantages of the FLMPC in numerical computing are not fully utilized. Our next step will be implementing it with a dedicated QP (quadratic programming) solver, which will further reduce the computation time.

V. CONCLUSION

This paper presents a feedback input-output linearization approach while addressing the challenge of unstable internal dynamics. By employing a Lyapunov function, we demonstrate that the formulated FLMPC stabilizes the whole system including the internal dynamics which is an integrator of an original input. This is achieved by integrating the internal state as an additional term into the objective function. The linearization and the stability are ensured when the vehicle is continuously moving and the steering angle must not reach the theoretical limit of ± 90 degrees. This, in practice, is acceptable as long as we consider trajectory tracking tasks such as lane keeping or evasive maneuvers, while physical limits, i.e. the maximum steering angle of the vehicle, are taken into account. The simulation results demonstrate that the proposed scheme leads to significant improvements in terms of the computational performance, i.e. real-time implementation is possible by employing the FLMPC framework.

The derived internal state of the kinematic model allows to approximate the original state-space by using the proposed inverse transformation and hence the prediction of the input constraints inside the FLMPC framework, which would usually be state-dependent. Thus, further work should investigate the consideration of input constraints to the FLMPC. Since the implementation of constraints is not trivial as the transformation usually depending on the progression of the states of the original state space within the prediction horizon. Implementations of different approaches to resolve this issue can be found in [8], [13], [14]. Our ultimate goal is to incorporate constraints and implement FLMPC using a dedicated solver, with the aim of improving its applicability and computation time.

REFERENCES

[1] D. Hrovat, S. Di Cairano, H. Tseng, and I. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *2012 IEEE International Conference on Control Applications*, 2012, pp. 295–302.

[2] W. Hong, S. Wang, P. Li, G. Wozny, and L. T. Biegler, "A quasi–sequential approach to large–scale dynamic optimization problems," *AIChE Journal*, vol. 52, no. 1, pp. 255–268, 2006.

[3] M. Bartl, P. Li, and L. T. Biegler, "Improvement of state profile accuracy in nonlinear dynamic optimization with the quasi–sequential approach," *AIChE Journal*, vol. 57, no. 8, pp. 2185–2197, 2011.

[4] P. D. Dai and B. T. H. Linh, "Nonlinear model predictive control of quadrotor using direct multiple shooting," in *2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2024, pp. 1–5.

[5] J. Tamimi and P. Li, "A combined approach to nonlinear model predictive control of fast systems," *Journal of Process Control*, vol. 20, no. 9, pp. 1092–1102, 2010.

[6] H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002, pp. 505–550.

[7] K. Guemghar, B. Srinivasan, and D. Bonvin, "Control of pendubot using input-output feedback linearization and predictive control," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 907–912, 2005.

[8] J. Deng, V. Becerra, and R. Stobart, "Input constraints handling in an mpc/feedback linearization scheme," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 219–232, 2009.

[9] L. Grüne and J. Pannek, Eds., *Nonlinear model predictive control: Theory and algorithms*, second edition ed., ser. Communications and control engineering. Cham: Springer, 2017.

[10] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[11] J. L. Pulsipher, W. Zhang, T. J. Hongisto, and V. M. Zavala, "A unifying modeling abstraction for infinite-dimensional optimization," *Computers & Chemical Engineering*, vol. 156, p. 107567, 2022.

[12] M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat, and J. P. Vielma, "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization," *Mathematical Programming Computation*, 2023.

[13] S. L. de Oliveira, V. Nevistic, and M. Morari, "Control of nonlinear systems subject to input constraints," *IFAC Proceedings Volumes*, vol. 28, pp. 13–18, 1995.

[14] M. J. Kurtz and M. A. Henson, "Input-output linearizing control of constrained nonlinear processes," *Journal of Process Control*, vol. 7, no. 1, pp. 3–17, 1997.