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

Motion planning for 4WS vehicle with autonomous selection of steering modes via an MIQP-MPC controller

Ngoc Thinh Nguyen[1], Pranav Tej Gangavarapu[1], Nicolas Mandel[1], Ralf Bruder[1], Floris Ernst[1]

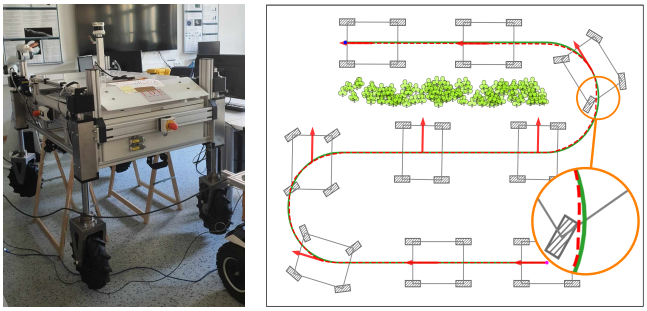
***Abstract*— Navigation in agricultural fields imposes various constraints on manoeuvrability, which can be tackled by using four-wheel steering (4WS) vehicles which are capable of switching between multiple steering mechanisms with distinct kinematic properties. For example, parallel positive steering (PPS) with four wheels in parallel to each other can maintain the vehicle's heading when moving along a curve. Symmetric negative steering (SNS) with two wheels on each side sharing the same steering angle can turn with a small radius. This paper presents a controller capable of selecting and switching between the two aforementioned modes autonomously for better trajectory tracking performance with special heading requirements for agricultural applications. The controller is implemented as a Model Predictive Control (MPC) controller formulated as a mixed-integer quadratic programming (MIQP) problem for the 4WS vehicle. Practical constraints, such as limits on wheel velocities, steering angles and their rate-of-changes are taken into account. A Python implementation confirms the real-time execution capability of the controller and simulation results highlight its effectiveness.**

***Index Terms*— Four-wheel steering, trajectory tracking, Model Predictive Control, mixed-integer programming.**

I. Introduction

Modern agriculture benefits from the advances in automation and enables the deployment of sophisticated tools to agricultural processes for higher efficiency. However, the environments provided by smaller, ecological enterprises do not allow for the application of large industrial machines, and require modular solutions, capable of operating in diverse surroundings [1]. The environmental challenges in a agriculture setting require better maneuverability than what typical Ackermann or differential steering can offer, with 63% of agricultural robots employing 4WS [2]. On top of increased stability on inclines [3], 4WS vehicles provide improved maneuverability through a variety of steering modes: a) negative steering, b) positive steering, c) active front or rear steering only (Ackermann steering), d) zero turn [4]–[6], among which two special settings named *symmetric negative steering* (SNS) and *parallel positive steering* (PPS) are the focus of this work. Each configuration comes with different constraints on wheel alignment and allows distinctive motion properties, e.g. PPS does not provide any rotational motion while translational movement is not applicable for the zero turn. Comprehensive reviews on the operation modes of a 4WS vehicle and studies of their characteristics can be found in [6]–[9].

[1] University of Luebeck, Institute for Robotics and Cognitive Systems (ROB), Luebeck, Germany.
{ngocthinh.nguyen,nicolas.mandel,ralf.bruder,
,floris.erns} @uni-luebeck.de
pranav.gangavarpu@student.uni-luebeck.de



(a) (b)

Fig. 1: (a) Prototype of a 4WS vehicle at ROB[1], (b) Scenario A: tracking results (dashed red line) of the MIQP-MPC controller w.r.t. a reference trajectory (green) with heading angle specifications that require the controller to switch between steering modes to maintain a constant field of view towards the row of plants.

In [4], a 4WS platform is used for weed detection while following a predefined path, which using the PPS mechanism to enable parallel displacement during turns. In [5], a path-tracking controller based on a back-stepping approach is used to compensate the crabwise sliding effect in the SNS mode caused by the field's terrain. In [10], the design of a sliding mode controller for 4WS tracking is presented for the negative steering configuration. The authors of [9] developed a controler for 4WS based on sliding mode control theory for decoupling axles. Arvind [7] decreased the turning radius of a 4WS plattform by studying the kinematics of the steering system geometry. In [8], a controler was designed to explicitly model the rear wheel steering angle in the optimization problem (OP). However, to the best of our knowledge, allowing automated mixing of different configurations has not yet been considered, with existing work related to 4WS vehicles focussing on a single steering mode when acting on the vehicle. For example, Fig. 1 (a) shows the prototype platform the controller is developed for and (b) illustrates a scenario in which the vehicle initially follows the reference trajectory with SNS, then switches to PPS in order to maintain the field of view in direction of the row of plants, and finally returns to the SNS mode.

In order to combine different steering modes autonomously when controlling a 4WS platform, a Model Predictive Control (MPC) approach with mixed-integer programming [11]–[13] is a promising candidate, thanks to the capability of MPC to handle various constraints and the

capability of MIP to distinguish between cases. An intuitive solution would be to index the steering modes, which are represented by their mathematical models and constraints, allowing the OP to select the indices autonomously while minimizing a desired cost function (e.g. tracking errors, energy consumption). The approach sounds simple, however, a direct implementation can easily result in a nonlinear mixed-integer formulation which is unsolvable for most existing solvers [13]. Instead, the model-switching problem requires to be re-formulated into a linear form via the big-M approach and by linearizing the models [11]–[14]. Furthermore, it is essential to ensure that the switch happens within the physical limits of the hardware by imposing smoothing constraints, i.e., limiting the change rates of the steering angles and of the wheel velocities, besides general physical constraints such as maximum speed and the range of the steering angles.

Combining all the preceding elements, we present a novel design of an MPC controller for trajectory tracking of a 4WS vehicle that is capable of switching between the SNS and the PPS configuration. All actuator limits, including the rate of change of the steering angles, are considered in the controller to enable smooth transitions between modes and ensure feasibility on a real 4WS platform. We formulate the controller as a mixed-integer quadratic programming (MIQP) problem in which an integer variable determines the steering mode at every predicted step. The implementation in Python achieves real-time performance and has been successfully validated under extensive simulations.

II. 4WS VEHICLE MODELLING

This section presents the mathematical models and constraints associated with different operating modes of a 4WS vehicle. In general, there are four steering configurations as shown in Fig. 3 [6]–[9]:
(a) Negative steering (also known as *reverse-phase* [9]): front wheels and rear wheels turn in opposite directions in order to reduce the turning radius.
(b) Positive steering (also known as *same-phase* [9]): two front wheels and two rear wheels turn in the same direction.
(c) Ackermann steering: behave as a car-like system, which turns by using only the front wheels.
(d) Zero turn: steering front wheels inward and rear wheels outward, thus allows the vehicle to turn on the spot.

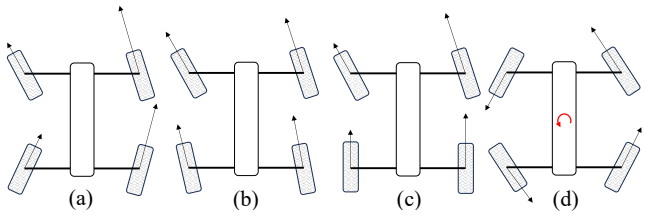


Fig. 2: Four steering configurations of a 4WS vehicle: (a) negative steering, (b) positive steering, (c) Ackermann steering, (d) zero turn.

Comprehensive reviews on the operation modes of a 4WS vehicle and studies of their characteristics can be found in [6]–[9]. This paper focuses on two special configurations with the most distinct properties: the *symmetric negative steering* (SNS) and the *parallel positive steering* (PPS) modes. The first configuration, which is fully named as *symmetrical front and the rear wheels steering* [8], belongs to the category of negative 4WS (c.f. Fig. 3 (a)) and has its two left/right wheels turning for the same absolute values. It provides the shortest-possible turning radius, while still allowing the vehicle to perform a combination of translation and rotation motions. The second configuration is the PPS mode, which is a special case of the positive steering category (c.f. Fig. 3 (b)). It aligns the four wheels in parallel to each other, therefore allowing the vehicle to move translationally only, maintaining the heading direction while navigating. Hence, the two chosen configurations exhibit the least common properties of a 4WS system, i.e. navigation with high dexterity, with a range of turning radii from infinity (i.e. no rotation with the PPS) to smallest-possible values. A brief study of kinematic models and constraints of the two chosen configurations is given in the next sections.

A. Symmetric negative steering (SNS) mode

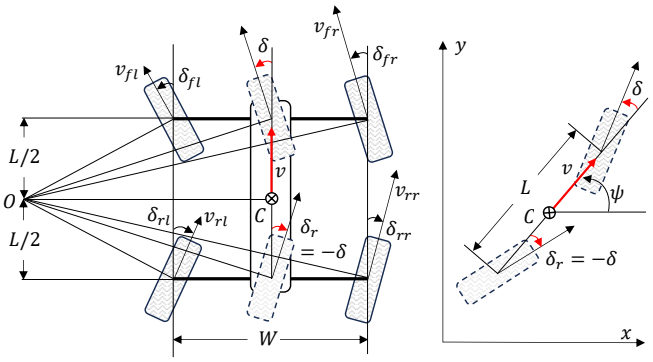


Fig. 3: SNS and the simplified bicycle model.

The SNS system is illustrated in Fig. 3, in which the two wheels on each side turn to the same absolute angular values but in different directions and rotate at the same speeds [6]–[8]. The steering angles and the velocities of the four wheels need to satisfy the following constraints in order to create a unique center of rotation (CoR) O:

$$\delta_{rl} = -\delta_{fl}, \;\; \delta_{rr} = -\delta_{fr}, \tag{1a}$$
$$v_{rl} = v_{fl}, \quad v_{rr} = v_{fr}, \tag{1b}$$
$$\cot \delta_{fr} - \cot \delta_{fl} = 2W/L, \tag{1c}$$
$$v_{fl} \sin \delta_{fl} = v_{fr} \sin \delta_{fr}, \tag{1d}$$

with $\{\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}\}$ being the steering angles and $\{v_{fl}, v_{fr}, v_{rl}, v_{rr}\}$ the translation velocities in non-slip condition of the front left, front right, rear left and rear right wheels, respectively. Note that (1c)–(1d) are derived from geometrical relations shown in Fig. 3. The 4WS system is simplified into a 2-DOF bicycle model placed at its center point C which is represented by the two dashed wheels

plotted at the center of each axle:

$$\dot{x} = v\cos\psi, \tag{2a}$$
$$\dot{y} = v\sin\psi, \tag{2b}$$
$$\dot{\psi} = 2v\tan\delta/L, \tag{2c}$$

in which the states $\{x, y, \psi\}$ represent the position and heading angle of the vehicle. The inputs are the translation velocity v at C and the front steering angle δ of the bicycle model, while the rear steering angle is constrained by the SNS configuration (c.f. Fig. 3):

$$\delta_r = -\delta. \tag{3}$$

Note that the bicycle system (2) will serve as the motion prediction model in 4WS mode for designing the controller in the next section. The controller will provide the desired inputs $\{v, \delta\}$ of the bicycle model, from which the real inputs applied to the two front wheels of the 4WS vehicle can be calculated with:

$$\cot\delta_{fr} = \cot\delta + W/L, \tag{4a}$$
$$\cot\delta_{fl} = \cot\delta - W/L, \tag{4b}$$
$$v_{fr} = v\tan\delta/\sin\delta_{fr}, \tag{4c}$$
$$v_{fl} = v\tan\delta/\sin\delta_{fl}, \tag{4d}$$

while the inputs for the two rear wheels can be obtained using (1a)–(1b).

In order to be integrated into the MIQP-MPC controller, the model (2) needs to be linearized and discretized. Combining the first-order Taylor expansion and the Euler discretization method with time step Δt yields:

$$\begin{aligned} X_{k+1} =& f_1(X^*, U^*, X_k, U_k) \triangleq A_1(X^*, U^*, \Delta t)X_k \\ &+ B_1(X^*, U^*, \Delta t)U_k + C_1(X^*, U^*, \Delta t), \end{aligned} \tag{5}$$

in which, the state vector $X_k = [x_k, y_k, \psi_k]^\top$ represents the position and heading angle and the input vector $U_k = [v_k, \delta_k]^\top$ gathers the velocity and steering angle of the simplified bicycle model at step k. The matrices $A_1(X^*, U^*, \Delta t)$, $B_1(X^*, U^*, \Delta t)$ and $C_1(X^*, U^*, \Delta t)$ are defined w.r.t. operating point (X^*, U^*)[1] and are explicitly given in Appendix A. The states and inputs as well as the input rates need to respect their lower and upper limits:

$$X_{\min} \le X_k \le X_{\max}, \tag{6}$$
$$U_{\min} \le U_k \le U_{\max}, \tag{7}$$
$$|U_{k+1} - U_k| \le \Delta U_{\max}, \tag{8}$$

in which, $\{X_{\min}, X_{\max}\}$ gather the lower and upper bounds of the states while $\Delta U_{\max} \triangleq [\Delta v_{\max}, \Delta\delta_{\max}]^\top$ gives the maximum rates of change. Note that, $\{U_{\min}, U_{\max}\}$ which give the bounds of the bicycle inputs $\{v, \delta\}$ are specifically designed such that the real inputs of the 4WS vehicle in SNS mode respect their limitations, i.e. $|v_i| \le v_{\max}$, $|\delta_i| \le \delta_{\max}$, $\forall i \in \{fr, fl, rr, rl\}$ with $\{v_{\max}, \delta_{\max}\}$ the maximum allowed wheel's velocity and steering angle. By analyzing the nonlinear input mapping (4), the final formulations of the constraints on $\{v, \delta\}$ are given by:

$$|v| \le \frac{v_{\max}}{\sqrt{\left(1 + \frac{W\sin\delta_{\max}}{W\sin\delta_{\max} + L\cos\delta_{\max}}\right)^2 + \frac{L^2\sin^2\delta_{\max}}{(W\sin\delta_{\max} + L\cos\delta_{\max})^2}}},$$
$$|\delta| \le \operatorname{arccot}\left(\frac{W}{L} + \cot\delta_{\max}\right), \tag{9}$$

with $\delta_{\max} \in (0, \pi/2)$ and W, L the parameters as in Fig. 3.

B. Parallel positive steering (PPS) mode

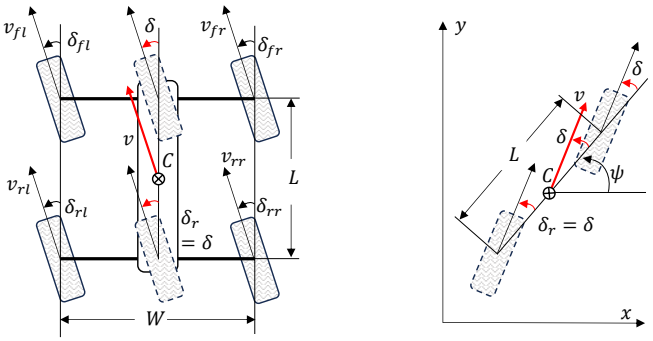


Fig. 4: PPS and the simplified bicycle model.

In the PPS mode as shown in Fig. 4, the four wheels rotate at the same speed while being aligned in the same direction:

$$\delta_{rl} = \delta_{fl} = \delta_{rr} = \delta_{fr} = \delta, \tag{10a}$$
$$v_{rl} = v_{fl} = v_{rr} = v_{fr} = v, \tag{10b}$$

with $\{\delta_{\_}, v_{\_}\}$ the steering angles and velocities of the four wheels as defined in (1a)–(1d). The mechanism allows the vehicle to only move along that direction and eliminates any rotating motion. Hence, the vehicle maintains a constant heading angle, even when moving on curves (i.e. having only translational motion). The associated 2-DOF bicycle model is given by:

$$\dot{x} = v\cos(\psi + \delta), \tag{11a}$$
$$\dot{y} = v\sin(\psi + \delta), \tag{11b}$$
$$\dot{\psi} = 0, \tag{11c}$$

with the states $\{x, y, \psi\}$ and inputs $\{v, \delta\}$ as defined in (2). Fig. 4 illustrates that the front and rear steering angles of the bicycle model in the PPS mode are equal:

$$\delta_r = \delta. \tag{12}$$

Then the system (11) is linearized with the first-order Taylor expansion and discretized with the Euler method:

$$\begin{aligned} X_{k+1} =& f_2(X^*, U^*, X_k, U_k) \triangleq A_2(X^*, U^*, \Delta t)X_k \\ &+ B_2(X^*, U^*, \Delta t)U_k + C_2(X^*, U^*, \Delta t), \end{aligned} \tag{13}$$

with (X_k, U_k) the state and input vectors at step k as defined in (5) and the matrices $\{A_2(\cdot), B_2(\cdot), C_2(\cdot)\}$ linearized at the operating points as similar to (5). In the PPS mode, the states

[1] In this work, the operating point takes the latest measured values of the system. Other option could be the closest reference points [15].

and inputs of the central bicycle model are also subject to the same constraints as defined in (6)–(8), while the actual inputs of the 4WS vehicle can be obtained from (10a)–(10b).

Remark 1: The switch between the SNS and PPS modes is activated through a change on the calculation of the rear steering angle δ_r of the simplified bicycle model. When $\delta_r = -\delta$ as in (3), the vehicle is in the SNS configuration and model (5) is used while $\delta_r = \delta$ as in (12) indicates the PPS mode with model (13). A smooth transition between those configurations is ensured by applying a rate limiter to δ_r:

$$|\delta_{r_{k+1}} - \delta_{r_k}| \leq \Delta\delta_{\max}, \tag{14}$$

similar to the front steering angle δ in (8). These help limiting the rotation speeds of the servo motors in real experiments.

III. MIQP-MPC FOR TRAJECTORY TRACKING WITH AUTOMATIC SELECTION OF STEERING MODES

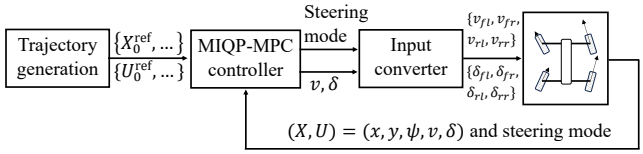


Fig. 5: MPC scheme for trajectory tracking of a 4WS vehicle.

In this section we present the design of an MPC (Model Predictive Control) controller for trajectory tracking of a 4WS vehicle as shown in Fig. 5. At every step, the MPC controller optimizes the system's behavior within N_p steps (so called *prediction horizon*) in the future [15], [16]. It uses the mathematical models for adjusting the inputs so that the predicted motion starting from the current state optimizes a cost function. Then, the first element of the optimal input sequences is applied to the system. New measurements are made and used to update the OP before starting to solve again. The MPC strategy is capable of handling various constraints on the system's states and inputs including their variations as they can be explicitly formulated in the OP [15]–[17]. Furthermore, in this work, integer variables are added to the MPC OP and allow selecting the most appropriate steering mode between the SNS and PPS for better tracking performance:

$$\begin{cases} \bar{X}_{i+1} = f_1(X_k, U_k, \bar{X}_i, \bar{U}_i), \\ \bar{\delta}_{r_i} = -\bar{\delta}_i \end{cases} \tag{15}$$

$$\text{or} \quad \begin{cases} \bar{X}_{i+1} = f_2(X_k, U_k, \bar{X}_i, \bar{U}_i), \\ \bar{\delta}_{r_i} = \bar{\delta}_i \end{cases}$$

in which, $f_1(\cdot), f_2(\cdot)$ are the discrete linear models of the SNS in (5) and the PPS in (13), respectively, with their linearization point defined at (X_k, U_k), the current measured states and inputs. Variables with bar on top such as $\bar{X}_i, \bar{U}_i, \bar{\delta}_i$ represent the predicted values at step i within the prediction horizon ($i \in \{0, 1, \ldots, N_p\}$). The mixed-integer linear formulation of the logical OR operator [12] for the model switching (15) is achieved by applying the big-M approach:

$$z_i L_1 \leq \bar{X}_{i+1} - f_1(X_k, U_k, \bar{X}_i, \bar{U}_i) \leq z_i U_1, \tag{16a}$$
$$-2z_i\delta_{\max} \leq \bar{\delta}_{r_i} + \bar{\delta}_i \leq 2z_i\delta_{\max}, \tag{16b}$$
$$(1 - z_i)L_2 \leq \bar{X}_{i+1} - f_2(\cdot) \leq (1 - z_i)U_2, \tag{16c}$$
$$-2(1 - z_i)\delta_{\max} \leq \bar{\delta}_{r_i} - \bar{\delta}_i \leq 2(1 - z_i)\delta_{\max}, \tag{16d}$$

in which, $f_2(\cdot)$ is the shortened form of (15) and $z_i \in \{0, 1\}$ is the integer variable at the predicted step i, which activates the SNS at 0 and the PPS mode at 1. The coefficients L_1, U_1 have large positive values in order to de-activate the SNS model when $z_i = 1$:

$$L_1 = \min\Big(f_2(X_k, U_k, X, U) - f_1(X_k, U_k, X, U)\Big) - M$$
$$U_1 = \max\Big(f_2(X_k, U_k, X, U) - f_1(X_k, U_k, X, U)\Big) + M$$

both subject to

$$X_{\min} \leq X \leq X_{\max},\ U_{\min} \leq U \leq U_{\max}, \tag{17}$$

with $M > 0$ a large positive value and the limits of states and inputs $\{X_{\min}, X_{\max}, U_{\min}, U_{\max}\}$ from (6)–(7). Then, the coefficients L_2, U_2 can be taken as:

$$L_2 = -U_1, \tag{18}$$
$$U_2 = -L_1. \tag{19}$$

Note that, the constraints (16b) and (16d) are constructed in a similar manner such that $-2\delta_{\max}$ and $2\delta_{\max}$ are the minimum and maximum values of the two middle terms. The complete MIQP optimization problem of the tracking MPC controller at step k is formulated as follows:

$$\min_{X_k, U_k} \sum_{i=0}^{N_p-1} \Big(E_{X,i}^\top Q E_{X,i} + E_{U,i}^\top R E_{U,i} + \Delta_{U,i}^\top \Gamma \Delta_{U,i} + \gamma\Delta_{z,i}^2\Big) + E_{X,N_p}^\top P E_{X,N_p} \tag{20a}$$

subject to

$$E_{X,i} = \bar{X}_i - X_{k+i}^{\text{ref}},\ E_{U,i} = \bar{U}_i - U_{k+i}^{\text{ref}} \tag{20b}$$
$$\Delta_{U,i} = \bar{U}_{i+1} - \bar{U}_i,\ \Delta_{z,i} = z_{i+1} - z_i, \tag{20c}$$
$$\text{model-switching constraints from (16)}, \tag{20d}$$
$$z_i \in \{0, 1\}, \tag{20e}$$
$$X_{\min} \leq \bar{X}_i \leq X_{\max}, \tag{20f}$$
$$U_{\min} \leq \bar{U}_i \leq U_{\max}, \tag{20g}$$
$$|\bar{U}_{i+1} - \bar{U}_i| \leq \Delta U_{\max}, \tag{20h}$$
$$|\bar{\delta}_{r_{i+1}} - \bar{\delta}_{r_i}| \leq \Delta\delta_{\max}, \tag{20i}$$
$$|\bar{U}_0 - U_k| \leq \Delta U_{\max}, \tag{20j}$$
$$|\bar{\delta}_{r_0} - \delta_{r_k}| \leq \Delta\delta_{\max}, \tag{20k}$$
$$\bar{X}_0 = X_k, \tag{20l}$$

for all $i \in \{0, 1, \ldots, N_p\}$,

in which, $Q \in \mathbb{R}^{3\times3}$, $R \in \mathbb{R}^{2\times2}$, $\Gamma \in \mathbb{R}^{2\times2}$ and $P \in \mathbb{R}^{3\times3}$ are the weighting matrices for the state, input, input variation and terminal state, all required to be positive semi-definite.

At every step, the OP is free to select between the two steering modes via the integer variable z_i in (20d)–(20e) but is enticed to keep the number of switches as low as possible with the term of $\gamma\Delta_{z,i}^2$. The $\gamma \in \mathbb{R}_+$ in the cost function ensures a smooth motion within the prediction horizon. Additional constraints on states (20f), inputs (20g) and input rates (20h) are imposed which correspond to the simplified bicycle system's constraints given in (6)–(8). The current states and inputs (i.e. the velocities and steering angles of the two wheels of the bicycle system) are used to ensure the continuity and rate limits of the motion via (20j)–(20l). Note that, the reference trajectory $\{X_0^{\text{ref}}, \dots\}$ is created by sampling a 2D geometrical path according to the desired forward velocity and the time step. Then, the 2D poses are merged with the user-defined heading angles, if applicable, otherwise the path direction is used instead, c.f. Fig. 1. Therefore, the reference trajectory does not respect the system model and is a challenge to follow within the vehicle's limits and under consideration of two possible steering modes. Minimizing the term $\Delta_{U,i}^\top \Gamma \Delta_{U,i}$ as in (20a) helps to reduce the chattering problem by smoothing the input commands in such difficult scenarios. Finally, the information which is sent to the input converter block shown in Fig. 5 consists of the optimal values of the first elements $\{\bar{U}_0, z_0\}$ (i.e. the central bicycle model's inputs and the steering mode) which are then converted into the reference values of four steering angles and four velocities, one for each of the four wheels by using the equations given in (4) for the SNS and (10) for the PPS.

Remark 2: The terminal cost $E_{X,N_p}^\top P E_{X,N_p}$ in (20a) facilitates the stability of the closed-loop system, as the MPC controller is formulated without stabilizing terminal constraints [18]. Two major factors, which mainly affect the stability are the terminal weighting matrix P, which needs to provide more weight than the matrix Q in (20a) [18] and the prediction horizon N_p, whose value is determined heuristically [14]–[16] in this work.

Remark 3: The values of L_1, U_1 in (17) require solving a linear OP at every step according to the measured values of the current states and inputs (X_k, U_k). In order to reduce the computing time, they can be calculated once offline beforehand by solving the following nonlinear OP with a nonlinear solver, e.g. IPOPT [19]:

$$L_1' = \min \Big(f_2(X,U,X,U) - f_1(X,U,X,U) \Big) - M,$$
$$U_1' = \max \Big(f_2(X,U,X,U) - f_1(X,U,X,U) \Big) + M,$$

both subject to:

$$X_{\min} \le X \le X_{\max},\ U_{\min} \le U \le U_{\max}, \qquad (21)$$

with $f_1(\cdot), f_2(\cdot)$ the SNS model (5) and the PPS model (13), respectively. In our implementation, we use $M = 10000$, while the absolute values of L_1', U_1' do not exceed 10.

IV. Simulation and Experiment Results

In this section the validation results of the proposed MIQP-MPC controller are presented. A custom built prototype of a 4WS vehicle is under development at University of Lübeck as shown in Fig. 1a. Based on the technical details of the platform, a digital twin of the vehicle is created in Gazebo to enable ROS compatibility, which is used to validate the MIQP-MPC controller. The controller itself is implemented in Python, utilizing DOcplex - an optimization library and the CPLEX solver which are included in IBM ILOG CPLEX Optimization Studio [13]. The controller is targeted to run at 10 Hz and the discretization step in (5), (13) is $\Delta t = 0.1$ seconds. Two tracking scenarios are considered:

- Scenario A (Fig. 1b): A reference path that resembles row navigation in an agricultural field in which the vehicle keeps its heading constant towards the row while moving alongside it.
- Scenario B (Fig. 7): A trajectory with sharp turns and sudden change of heading angles. The vehicle needs to steer to and maintain a constant heading angle at 0° for the second half of the trajectory.

The 4WS vehicle admits the following parameters: dimension of $L \times W = 1.3m \times 0.9m$, maximum steering angle of $\delta_{\max} = 160^\circ/s^2$, maximum steering rate of $\Delta\delta_{\max} = 30^\circ/s$, maximum allowed speed of $v_{\max} = 5^m/s$ and maximum acceleration $a_{\max} = 1^m/s^2$. For both scenarios, the prediction horizon is chosen at $N_p = 10$ steps and the following weighting matrices are used: $Q = \text{diag}(1,1,1)$, $P = 10Q$, $R = \text{diag}(1,0)$, $\Gamma = \text{diag}(0.05, 0.05)$ as in (20a). The second element of R is equal to zero as the reference of the steering angle δ_k^{ref} is unknown. Table I provides a comprehensive analysis of the tracking errors $\{e_x, e_y, e_\psi\}$ of the three states and the computation time at each iteration.

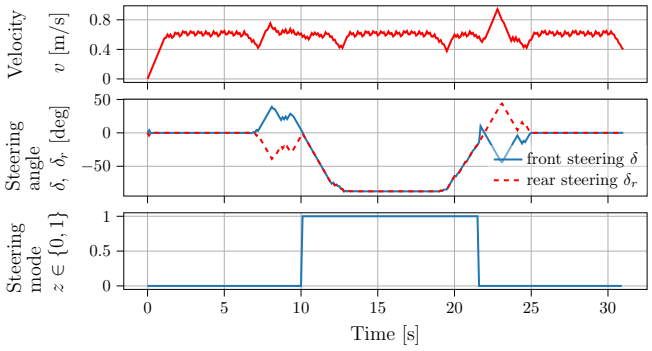


Fig. 6: Control inputs and model index under Scenario A

Fig. 1b shows the tracking result under Sce. A in which the reference trajectory is plotted as a solid green line while the vehicle motion is plotted with a dotted red line. The rectangles represent the 4WS platform with its heading indicated with green arrows. It can be observed that the controller tracks the reference well and the only noticeable mismatch happens at the second turn before switching back to the SNS mode. This is deemed reasonable, as the controller is compromising current tracking performance (with PPS) in order to smoothly transition to SNS in future timesteps and achieves a better tracking performance in the following steps, which is a typical MPC behavior. The command inputs as well as the discrete model switch is

TABLE I: Tracking errors and computing time

		State errors (absolute values)			Time [ms]
		e_x[m]	e_y[m]	e_ψ[rad]	
Sce. A	Max	0.109	0.090	0.275	72.69
	Mean	0.025	0.010	0.034	49.05
	Std. Dev.	0.017	0.016	0.047	9.85
Sce. B	Max	0.510	0.379	0.753	70.84
	Mean	0.055	0.043	0.039	53.12
	Std. Dev.	0.106	0.076	0.062	5.47

shown in Fig. 6, in which both the velocity and steering angle satisfy all requirements. The binary variable indicates the steering mode switch from 0 (i.e. SNS) to 1 (i.e. PPS mode) and its return to 0 (SNS) at the end of the trajectory. As given in Table I, the maximum computing time for this scenario is 73 ms, never exceeding the sampling time of 100 ms, while the average value is 49 ms per iteration. With the maximum value during calculation lower than the discretization sampling time, the controller can be considered real-time capable. Note that, the calculation of L_1, U_1 as in (21) are done offline as mentioned in Remark 3 and thus, are not reflected in Table I.

For Scenario B, a simple cubic polynomial trajectory shown in Fig. 7 is used as a reference for the controller while the inputs, i.e. vehicle velocity, steering angles and binary mode variables are presented in Fig. 8. The shaded yellow region, limited by two dotted lines, depicts the region requiring PPS mode where the reference heading angle is suddenly switched to and maintained at 0°. For the first half of the task, the SNS mode is used to align the vehicle with the trajectory. The trajectory is intentionally created to include turns with small radii, so that only the SNS mode is able to follow, owing to its smaller turning radius. The tracking results shown in Fig. 7 and Table I prove the effectiveness of the controller by switching the steering modes according to the heading requirements. There is always a trade-off between maintaining the intended heading angle and the tracking on x and y coordinates of the trajectory as can be seen clearly from Fig. 7 between $x = 7$ and $x = 8$. It requires careful design of the cost function (20a) and fine-tuning of the constraints and the weighting matrices in order to achieve a good performance. Even though this scenario imposes more challenges with the tight turns and the sudden heading change, the computing time is not increased w.r.t. those from Sce. A as given in Table I. The maximum value (71 ms) still allows the controller to run at 10 Hz, therefore achieving real-time performance.

V. Conclusions and Future Works

In this paper we introduced an MIQP-MPC controller for autonomous trajectory tracking of a 4WS mobile platform, which is intended to control the custom platform shown in Fig. 1a. The controller is capable of tracking the reference trajectory with special heading angle requirements by adapting its steering modes under consideration of states and inputs constraints. It also considers constraints on the angular rates of steering motors according to a real platform. The controller is mathematically formulated as an MIQP problem and is implemented in Python with *cplex* solver, achieving real-time performance. The results are verified in simulations in a Gazebo environment developed based on a real custom-built autonomous 4WS platform and demonstrate the increased maneuverability of the proposed controller, therefore enabling the platform to operate in small agricultural enterprises. Future work focuses on experimental validation on the real platform.

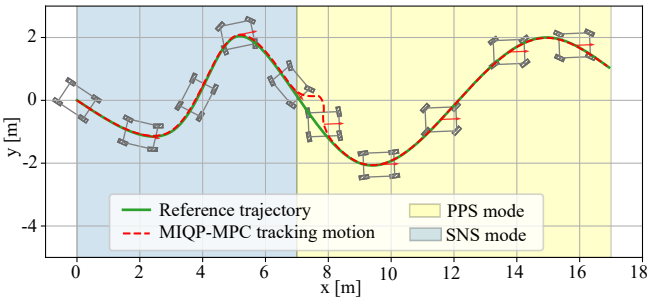


Fig. 7: Trajectory tracking results under Scenario B

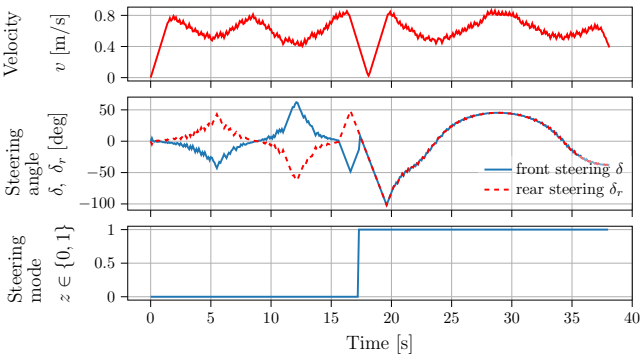


Fig. 8: Control inputs and model index under Scenario B

Appendix

A. Formulations of the model matrices

Matrices used in the discrete linear SNS model:

$$A_1(X^*, U^*, \Delta t) = \begin{bmatrix} 1 & 0 & -\Delta t v^* \sin\psi^* \\ 0 & 1 & \Delta t v^* \cos\psi^* \\ 0 & 0 & 1 \end{bmatrix}, \tag{22a}$$

$$B_1(X^*, U^*, \Delta t) = \Delta t \begin{bmatrix} \cos\psi^* & 0 \\ \sin\psi^* & 0 \\ \frac{2}{L} v^* \tan\delta^* & \frac{2v^*}{L\cos^2\delta^*} \end{bmatrix}, \tag{22b}$$

$$C_1(X^*, U^*, \Delta t) = \Delta t \begin{bmatrix} v^*\psi^* \sin\psi^* \\ -v^*\psi^* \cos\psi^* \\ \frac{2v^*\delta^*}{L\cos^2\delta^*} \end{bmatrix} \tag{22c}$$

Acknowledgment

This work is funded by the German Ministry of Food and Agriculture (BMEL) Project No. 28DK133A20. We thank our project partners and collaborators, especially IDE Automation for manufacturing the 4WS platform.

REFERENCES

[1] A. Botta, P. Cavallone, L. Baglieri, G. Colucci, L. Tagliavini, and G. Quaglia, "A review of robots, perception, and tasks in precision agriculture," *Applied Mechanics*, vol. 3, no. 3, pp. 830–854, 2022.

[2] L. F. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in agriculture robotics: A state-of-the-art review and challenges ahead," *Robotics*, vol. 10, no. 2, p. 52, 2021.

[3] R. Vidoni, M. Bietresato, A. Gasparetto, and F. Mazzetto, "Evaluation and stability comparison of different vehicle configurations for robotic agricultural operations on side-slopes," *Biosystems engineering*, vol. 129, pp. 197–211, 2015.

[4] T. Bak and H. Jakobsen, "Agricultural robotic platform with four wheel steering for weed detection," *Biosystems Engineering*, vol. 87, no. 2, pp. 125–136, 2004.

[5] C. Cariou, R. Lenain, B. Thuilot, and M. Berducat, "Automatic guidance of a four-wheel-steering mobile robot for accurate field operations," *Journal of Field Robotics*, vol. 26, no. 6-7, pp. 504–518, 2009.

[6] M. Harrer and P. Pfeffer, *Steering handbook*. Springer, 2017, vol. 163.

[7] V. Arvind, "Optimizing the turning radius of a vehicle using symmetric four wheel steering system," *International Journal of Scientific & Engineering Research*, vol. 4, no. 12, pp. 2177–2184, 2013.

[8] X. Tan, D. Liu, and H. Xiong, "Optimal control method of path tracking for four-wheel steering vehicles," in *Actuators*, vol. 11, no. 2. MDPI, 2022, p. 61.

[9] T. Hiraoka, O. Nishihara, and H. Kumamoto, "Automatic path-tracking controller of a four-wheel steering vehicle," *Vehicle System Dynamics*, vol. 47, no. 10, pp. 1205–1227, 2009.

[10] L. Tan, S. Yu, Y. Guo, and H. Chen, "Sliding-mode control of four wheel steering systems," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2017, pp. 1250–1255.

[11] J. P. Vielma, "Mixed integer linear programming formulation techniques," *Siam Review*, vol. 57, no. 1, pp. 3–57, 2015.

[12] I. Prodan, F. Stoican, S. Olaru, and S.-I. Niculescu, *Mixed-integer representations in control design: Mathematical foundations and applications*. Springer, 2016.

[13] C. Bliek, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report," in *Proceedings of the twenty-sixth RAMP symposium*, 2014, pp. 16–17.

[14] F. Stoican, I. Prodan, E. I. Grøtli, and N. T. Nguyen, "Inspection trajectory planning for 3d structures under a mixed-integer framework," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 1349–1354.

[15] N. T. Nguyen, I. Prodan, and L. Lefevre, "A stabilizing nmpc design for thrust-propelled vehicles dynamics via feedback linearization," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2909–2914.

[16] H. T. Nguyen, N. T. Nguyen, and I. Prodan, "Notes on the terminal region enlargement of a stabilizing nmpc design for a multicopter," *Automatica*, vol. 159, p. 111375, 2024.

[17] N. T. Nguyen, P. T. Gangavarapu, A. Sahrhage, G. Schildbach, and F. Ernst, "Navigation with polytopes and b-spline path planner," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5695–5701.

[18] A. Boccia, L. Grüne, and K. Worthmann, "Stability and feasibility of state constrained mpc without stabilizing terminal constraints," *Systems & control letters*, vol. 72, pp. 14–21, 2014.

[19] 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.