Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Exploit Agile Mobility of Steerable-Wheeled Mobile Robots: A Fast Motion Planning Approach

Wang Xi[1], Jingxuan Tang[1], Zhiyuan Zeng[1], Yushan Li[2] and Jianping He[1,†]
[1]Shanghai Jiao Tong University, China. [2]KTH Royal Institute of Technology, Sweden.
[†]Corresponding author, email: jphe@sjtu.edu.cn

Abstract—**This paper addresses real-time motion planning problem for steerable-wheeled mobile robots (SWMRs). Despite significant progress in SWMR control, most existing approaches design controllers for specific task scenarios and actuator limitations, and are further restricted to four-wheel rectangular layouts, resulting in limited versatility. Motivated by these issues, we formulate SWMR motion planning as an optimization problem that simultaneously maximizes motion performance while enforcing actuator feasibility. The proposed formulation supports flexible combinations of task objectives, wheel placements, and actuator constraints without modifying the underlying framework.**

Formulation alone does not guarantee real-time solvability; we therefore design a tailored fast algorithm that iteratively contracts the non-convex feasible set toward a local minimum. Although contraction-based iterations are uncommon, their anytime feasibility and computational efficiency make them particularly suitable for real-time robotic applications. We then rigorously establish the descent, convergence, and feasibility properties of the proposed algorithm. Simulation results on a trajectory tracking task demonstrate a reduction in computation time and a great improvement in constraint satisfaction compared to baseline approaches.

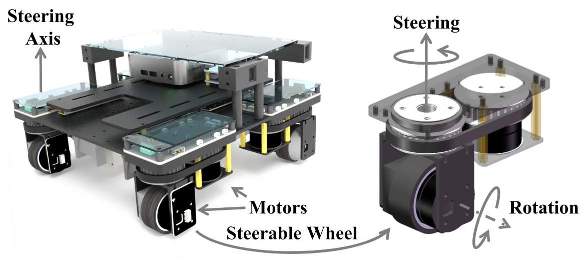


Fig. 1: Illustration of SWMR. Each wheel is independently steerable about its vertical axis.

I. Introduction

Steerable-wheeled mobile robots (SWMRs) have been increasingly employed in various industries, such as warehouse logistics and precision agriculture, to name a few [27]. How to plan their motions by controlling the steerable wheels plays a key role for SWMRs to accomplish complex tasks [14, 15]. Conventional control strategies decompose motion into distinct modes (e.g., translation and rotation), as in [31, 34, 36]. These modal approaches are straightforward and can achieve satisfying performance in many scenarios when combined with reinforcement learning techniques, e.g., see [2, 32]. However, there is still a gap to fully exploit the omnidirectional mobility of SWMRs to achieve agile motion in practice.

The major difficulty of meeting the above goal lies in dealing with the complex dynamics [3] and the kinematic singularities [28] of SWMRs. A variety of dedicated control and planning strategies have been proposed in the literature, which can be broadly categorized into three classes: nonlinear control law designs, Instantaneous Center of Rotation (ICR) planning based, and Model Predictive Control (MPC) based methods. For instance, representative nonlinear trajectory tracking controllers were implemented on *iMoro* [18, 19]. ICR-based planning methods were developed for the *AZIMUT-3* platform [6, 8] and the *ExoMars* prototype [23]. Although the former two types of methods demonstrate agile motion capabilities to some extent, they typically rely on extensive parameter tuning and adopt task-specific formulations. By contrast, the MPC-based methods provide a more general planning framework. For example, Liu et al. [17] achieved trajectory tracking using linearized MPC, while Connette et al. [10] incorporated potential field methods for singularity avoidance. More recently, Xi et al. [33] developed an efficient ICR-based MPC approach; however, it does not support orientation planning and suffers from numerical instability when representing straight-line motion. These methods show the potential of achieving versatile SWMR motion planning by modifying the cost functions and constraints, but their high computational costs still remain an obstacle for practical deployments.

Motivated by the prior works, this paper investigates a fast and versatile motion planning approach in a non-convex MPC framework, enabling agile motion with low real-time computational costs. This problem faces two challenges. First, the steering rate of wheels is bounded in practice, which is a nonholonomic constraint and will easily render singularity. These features introduce couplings between the constraints and control inputs on wheels, significantly increasing the problem complexity. Second, the non-convexity is inevitably introduced when various wheel placements and actuator constraints are considered. Although state-of-the-art nonlinear optimization solvers [30, 35] can theoretically work, the feasibility of solutions and low real-time computational costs need to be guaranteed during practical operations. Our contributions are summarized as follows.

- We develop an MPC-based versatile motion planning approach for SWMRs that accommodates arbitrary numbers of wheels with arbitrary placements. Together with dynamically adjustable task objectives and actuator con-

TABLE I: Key notations in description and optimization

Notation	Definition
x, y, ψ	Positions and orientation of robot
$\xi_{\{w,c\}}$	Robot state $(x, y, \psi)^\top$ in coordinate $\{\mathbb{W}, \mathbb{C}\}$
φ_n, ϑ_n	Steering, Rotation angle of n-th wheel
z_n^x, z_n^y	Output velocity of wheel n in $\mathbb{C}$
H_n	characteristic matrix of wheel n
k, K	Step index and total steps
$\bigcirc_{i,n}^k$	i-th variable/function of wheel n in time step k
$\mathcal{P}$, $\mathcal{Q}$	Optimization problem and convex subproblem
$\boldsymbol{u} \in \mathbb{R}^{3\times K}$	Optimization variable of $\mathcal{P}^K$
$\hat{\boldsymbol{u}}_{(m)}$	Estimation for solution in m-th iteration

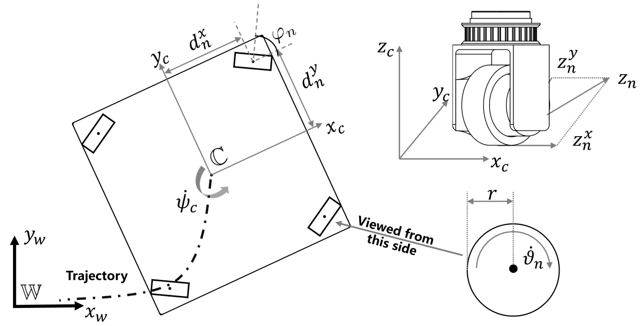


Fig. 2: Schematic model and explanation of notations.

straints, it achieves a high level of versatility across task scenarios and SWMR configurations. To the best of our knowledge, few methods offer this degree of flexibility.

- We propose a fast, structure-aware solution algorithm that makes this intuitive yet challenging non-convex formulation practically solvable in real time. This algorithm resembles sequential quadratic programming, but iteratively tightens the non-convex feasible set for computation. Moreover, we rigorously establish descent, convergence, and anytime feasibility properties for this method.
- We validate the proposed approach in representative trajectory tracking scenarios, and compare it with mainstream solvers. Results confirm the proposed approach's advantages in rigorous solution feasibility and low computational costs.

The remainder of this paper is organized as follows. In Section II, some preliminaries and the problem of interest are presented. The motion planning approach is given in Section III. Simulation results are shown in Section IV. Finally, Section V concludes the paper.

II. Preliminaries and Problem Formulation

In this section, we present the fundamental modeling of SWMRs and the problem of interest. Key notations used in this paper are summarized in Table I.

A. Kinematic Modeling of SWMRs

The kinematic model of SWMRs comprises two components: chassis (macro) and steerable wheels (micro).

1) Chassis: Let $\xi_w = (x_w, y_w, \psi_w)^\top \in \mathbb{R}^3$ denote the state of a SWMR. This state is expressed in the world-fixed coordinate frame $\mathbb{W}$. The variables x_w and y_w represent planar coordinates and ψ_w denotes the orientation. We denote the chassis-fixed frame as $\mathbb{C}$[1], and the chassis velocity $\dot{\xi}_c$ relates to the velocity in $\mathbb{W}$ as below.

$$\dot{\xi}_w = \begin{bmatrix} \mathcal{R}(\psi_w) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \dot{\xi}_c, \text{ with } \mathcal{R}(\psi_w) = \begin{bmatrix} \cos\psi_w & -\sin\psi_w \\ \sin\psi_w & \cos\psi_w \end{bmatrix}, \tag{1}$$

where $\mathcal{R}(\cdot) \in \mathbb{SO}(2)$ is a class of orthogonal rotation matrix.

[1]The definition of an inertial reference frame attached to a moving object can be referred to the "base frame" in [25].

2) Wheels: The N wheels of chassis are indexed by $n \in \{1, 2, \cdots, N\}$, with the n-th wheel placed at $d_n = (d_n^x, d_n^y)^\top \in \mathbb{R}^2$ relative to $\mathbb{C}$. Let φ_n and ϑ_n denote the steering angle and rotation angle of the n-th wheel, respectively. The radius of each wheel is r. The output velocity of the n-th wheel in $\mathbb{C}$ is represented by $z_n = (z_n^x, z_n^y)^\top \in \mathbb{R}^2$, which is governed by the following kinematic relations:

$$z_n^x = -r\dot{\vartheta}_n \sin\varphi_n, \quad z_n^y = r\dot{\vartheta}_n \cos\varphi_n. \tag{2}$$

3) Alignment: Note that for SWMR with N wheels, there are $2N$ degrees of freedom of the wheels, exceeding the 3 degrees of freedom of the chassis. Therefore, the control designs on SWMRs typically restrict attention to a subset of wheel states, which is called *alignment* [26]. The alignment offers several advantages. First, it removes conflicting forces among the wheels. Consequently, chassis motion resistance is reduced and overall efficiency improves. Second, under this condition, the kinematic model is sufficiently representative, eliminating the need for complex wheel–ground friction models [20]. We then introduce the characteristic matrix of the n-th wheel as

$$H_n \triangleq \begin{bmatrix} 1 & 0 & -d_n^y \\ 0 & 1 & d_n^x \end{bmatrix}, \tag{3}$$

based on which the formal definition of alignment follows.

Definition 1 (Alignment) *Given the velocity $\dot{\xi}_c(t) = (\dot{x}_c, \dot{y}_c, \dot{\psi}_c)^\top$, if the output velocities of all wheels satisfy*

$$z_n(t) = H_n \dot{\xi}_c(t), \quad \forall n \in \{1, 2, \cdots, N\}, \tag{4}$$

then the SWMR is called to exhibit alignment at time t.

Notice that (4) unifies the macro and micro states: the projected velocity of the rigid chassis at wheel mounting points matches the output velocity of wheel in $\mathbb{C}$. This consistency requires two conditions: appropriate rotational velocities ($\dot{\vartheta}_n$) and appropriate steering angles (φ_n). When the latter are properly set, the ground-wheel friction naturally synchronizes wheel rotational speeds and satisfies the former condition. Thus, *proper steering angles* are critical for maintaining alignment.

B. Singularity and Nonholonomy

Due to the actuator limits of SWMRs in practice, the steering rate of each wheel should be bounded by a constant:

$$|\dot{\varphi}_n(t)| \leq \omega_{\max},\ \forall t > 0,\ n \in \{1, 2, \cdots, N\}. \tag{5}$$

These actuator limitations may lead to singularities and nonholonomy. A detailed analysis is provided below.

The singularity refers to cases where a desired trajectory drives outputs toward infinity, which commonly occurs in nonlinear robotic systems. Based on the no-lateral-skidding property of alignment, each wheel must satisfy zero lateral velocity. Rigid-body kinematics further implies

$$l_n \left(\varphi_n(\dot{\xi}_c)\right)^\top \dot{\xi}_c = 0, \tag{6}$$

where $l_n(\varphi_n) = [\cos\varphi_n, \sin\varphi_n, d_n^x \sin\varphi_n - d_n^y \cos\varphi_n]^\top$. Supposing the second-order derivative $\ddot{\xi}_c$ exists, the steering rate relates to the desired trajectory as [25]:

$$\dot{\varphi}_n = -\frac{l_n(\varphi_n)\ddot{\xi}_c}{\frac{\mathrm{d}l_n(\varphi_n)}{\mathrm{d}\varphi_n}\dot{\xi}_c}. \tag{7}$$

The structure of l_n makes it easy to identify trajectory velocities $\dot{\xi}_c$ that render the denominator small. Even as the denominator in (7) approaches zero, the required steering speed can approach infinity. Such impractical control demands will break the alignment condition (4), thus producing conflicting forces among wheels and causing unpredictable chassis motion. To avoid singularities, it is necessary to impose additional constraints on the steering rate.

The nonholonomy denotes velocity-level constraints that depend on prior motions, and it renders even simple constraints non-convex, thereby complicating optimization-based methods. A common misconception is that agile SWMR motion can be achieved by closed-loop control of each motor based on the projection relationship in (4). However, constraints of the form (5) are nonholonomic [29, 1.3.2], and the existence of smooth time-invariant controllers (such as PID) for SWMRs is proved to be precluded by Brockett et al. [5]. Furthermore, related studies in other vehicles [22, III] suggest that finding optimal SWMR trajectories satisfying these constraints is likely NP-hard.

C. Problem of Interest

In this paper, we aim to develop a fast MPC-based motion planning approach, which can avoid the singularity issue and accommodate the nonholonomy of steering-rate constraints.

First, the problem is posed as a K-step constrained discrete optimization, which naturally aligns with digital control systems. Let t_0 denote the planning start time. The system is discretized with a uniform sampling interval $\tau > 0$, and indexed by

$$t_k = t_0 + k\tau,\ k = 0, \ldots, K.$$

Note that the alignment condition (4) implies that planning the chassis velocity in frame $\mathbb{C}$

$$\nu^k \triangleq \dot{\xi}_c(t_k) \tag{8}$$

uniquely determines the states of all N wheels. We define the incremental control input u^k such that

$$\nu^{k+1} = \nu^k + u^{k+1},\ k < K. \tag{9}$$

Accordingly, the control sequence $\boldsymbol{u} = \{u^1, \ldots, u^K\}$ constitutes the optimization variables, which corresponds to the chassis acceleration sequence in frame $\mathbb{C}$.

Concerning the constraints of the considered problem, practical scenarios may impose further actuation constraints apart from the intrinsic steering-rate constraint (5). Targeting at a versatile framework, the proposed approach is designed to accommodate such commonly encountered constraints. We denote a feasible set for $\boldsymbol{u}$ specified by j-th constraint as $\mathcal{U}_j$. Commonly considered choices for $\mathcal{U}_j$ include the following wheel output speed constraint [18], the chassis velocity constraint [7], and the velocity output direction constraint [23]:

$$\begin{cases} \text{wheel output speed}: \ \left\{\boldsymbol{u} \,\middle|\, \|H_n\nu^k\| \leq z_{\max}, z_{\max} \in \mathbb{R}^+\right\} \\ \text{chassis velocity}: \ \left\{\boldsymbol{u} \,\middle|\, \|\nu^k\| \leq v_{\max}, v_{\max} \in \mathbb{R}^+\right\} \\ \text{velocity output direction}: \ \left\{\boldsymbol{u} \,\middle|\, \beta^\top H_n\nu^k \geq 0,\ \beta \in \mathbb{R}^2\right\} \end{cases}.$$

Additional constraints can be incorporated in the same manner.

Let $g(\boldsymbol{u}) : \mathbb{R}^{3\times K} \to \mathbb{R}$ be a task-dependent motion planning cost. Different cost functions for $g(\boldsymbol{u})$ can be chosen to address various tasks (such as speed regulation or trajectory tracking). Then, our objective is to solve the following optimization problem:

$$\begin{aligned} \mathcal{P}_{\text{org}}: \ & \min_{\boldsymbol{u}\in\mathbb{R}^{3\times K}} \quad g(\boldsymbol{u}) \\ \text{s.t.} \quad & |\dot{\varphi}_n(t_k)| \leq \omega_{\max},\ k < K,\ n \in \{1, \cdots, N\}, && \text{(10a)} \\ & \boldsymbol{u} \in \bigcap\nolimits_j \mathcal{U}_j. && \text{(10b)} \end{aligned}$$

Let $\mathcal{D}(\cdot)$ denotes the feasible solution set of an optimization problem, and the following assumption is made throughout this paper.

Assumption 1 *For the problem $\mathcal{P}_{\text{org}}$:*

1) *The cost function $g(\boldsymbol{u})$ is convex, continuous, and lowered bounded, i.e., $\exists g_{\min} > -\infty$, s.t. $g(\boldsymbol{u}) \geq g_{\min}, \forall \boldsymbol{u}$.*
2) *The feasible set $\mathcal{D}(\mathcal{P}_{\text{org}})$ is nonempty.*
3) *Each $\mathcal{U}_j$ is a closed convex set.*

The above assumption is standard in vehicle trajectory planning to ensure that the problem is well-posed and amenable to algorithmic design.

III. The Fast Motion Planning Approach

In this section, we first show how to address the complex constraint regarding $\dot{\varphi}_n(t_k)$ and reformulate the problem. Then, an iterative convexification based motion planning algorithm is proposed, along with the performance analysis.

A. Equivalent Transformation of The Steering Constraints

As analyzed before, the steering constraints (5) are indispensable and special for SWMR planning. However, it is hard to directly apply the full form (7) due to its complex structure.

This part shows how to transform (5) into a feasible form and reformulate the problem.

First, note that from a micro perspective, the change in steering angle between two successive time steps must be bounded by δ_ω, i.e.,

$$|\varphi_n(t_{k+1}) - \varphi_n(t_k)| \le \delta_\omega \triangleq \omega_{\max}\tau. \tag{11}$$

We assume that τ is sufficiently small such that $\delta_\omega < \pi/2$. This constraint results in a cone-shaped feasible region for $z_n(t_{k+1})$, described by a pair of bilinear inequalities:

$$\left(\mathcal{R}_i z_n(t_k)\right)^\top z_n(t_{k+1}) \ge 0,\ i = 1, 2, \tag{12}$$

where $\mathcal{R}_1 = \mathcal{R}(\pi/2 - \delta_\omega)$ and $\mathcal{R}_2 = \mathcal{R}_1^\top$. Although the aperture of the above cone is fixed, its center direction depends on historical decisions. By substituting (4) and $\nu^k = \dot{\xi}_c(t_k)$ into (12), we have

$$\nu^{k-1\top} H_n^\top \mathcal{R}_i^\top H_n(\nu^{k-1} + u^k) \ge 0,\ i = 1, 2, \tag{13}$$

which eliminates explicit dependence on z_n.

As (13) represents the steering rate constraint for a wheel n at step k, the intersection of these $2 \times K \times N$ constraints constitutes the discrete-time feasible region of (5). Consequently, the original problem $\mathcal{P}_{\mathrm{org}}$ is transformed into:

$$\begin{aligned}
\mathcal{P}^K :\ & \min_{\boldsymbol{u}\in\mathbb{R}^{3\times K}} \quad g(\boldsymbol{u}) \\
\text{s.t.}\ & \nu^{k-1\top} H_n^\top \mathcal{R}_1^\top H_n(\nu^{k-1} + u^k) \ge 0,\ \forall n, k \ge 1, && \text{(14a)}\\
& \nu^{k-1\top} H_n^\top \mathcal{R}_2^\top H_n(\nu^{k-1} + u^k) \ge 0,\ \forall n, k \ge 1, && \text{(14b)}\\
& \boldsymbol{u} \in \textstyle\bigcap_j \mathcal{U}_j. && \text{(14c)}
\end{aligned}$$

In a brief summary, the reformulated $\mathcal{P}^K$ has the following features:

1) ICR-free: solving the problem does not rely on any ICR concepts, avoiding complex processing and numerical issues as in [9, 23, 26, 33].
2) Flexible configuration: more than four wheels with arbitrary mounting positions can be accommodated.
3) Versatile: the formulation considers the critical steering rate constraints and many other convex constraints, covering many task-specific planning scenarios as in [6, 10, 18, 19, 23, 25, 33].

Notice that the bilinear constraints in $\mathcal{P}^K$ are non-convex due to the nonholonomy in SWMRs. Although some mainstream nonlinear programming (NLP) solvers can provide feasible solutions, steering-rate constraints significantly increase computational burden and ultimately limit real-time deployability.

B. Convex Approximation

Inspired by techniques for handling nonconvex matrix inequalities in robust control [11], this subsection shows how to contract the nonconvex feasible set for $\mathcal{P}^K$ into convex subsets, without harming the sustainability in iterations. This treatment is the cornerstone for the proposed algorithm.

Transforming nonconvex problems into convex ones is a common strategy in the literature. Unlike standard relaxation algorithms based on local linearization (such as those used in sequential quadratic programming), we perform the expansion after a convex-concave decomposition in order to achieve a strict contraction of the feasible region. Let $\hat{\boldsymbol{u}} \in \mathbb{R}^{3\times K}$ be a fixed vector with the same dimension as $\boldsymbol{u}$, and we then present the following result.

Proposition 1 (Constraint convexification) *The bilinear constraint* (13) *can be locally contracted around* $\hat{\mathbf{u}}$ *as*

$$C_{i,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}}) = A_{i,n}^k(\boldsymbol{u}) - B_{i,n}^k(\hat{\boldsymbol{u}}) - L_{i,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}}) \le 0, \tag{15}$$

where $A_{i,n}^k(\boldsymbol{u})$*,* $B_{i,n}^k(\hat{\boldsymbol{u}})$*, and* $L_{i,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}})$ *are given by*

$$\begin{cases}
A_{i,n}^k(\boldsymbol{u}) = \dfrac{1}{2}\|H_n\nu^{k-1}\|^2 + \dfrac{1}{2}\|H_n\left(\nu^{k-1} + u^k\right)\|^2 \\
B_{i,n}^k(\hat{\boldsymbol{u}}) = \dfrac{1}{2}\|(I + R_i)H_n\hat{\nu}^{k-1} + R_i H_n \hat{u}^k\|^2 \\
L_{i,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}}) = \mathrm{tr}\left((\nabla_{\boldsymbol{u}} B_{i,n}^k|\boldsymbol{u} = \hat{\boldsymbol{u}})^\top(\boldsymbol{u} - \hat{\boldsymbol{u}})\right)
\end{cases}. \tag{16}$$

Proof: See Appendix A.

Based on Proposition 1, we further obtain a parametric convex subproblem $\mathcal{Q}^K$ for $\mathcal{P}^K$:

$$\begin{aligned}
\mathcal{Q}^K(\hat{\boldsymbol{u}}) :\ & \min_{\boldsymbol{u}\in\mathbb{R}^{3\times K}} f(\boldsymbol{u}, \hat{\boldsymbol{u}}) = g(\boldsymbol{u}) + \rho\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2 \\
& \text{s.t.} \begin{cases} C_{1,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}}) & \le & 0 \\ C_{2,n}^k(\boldsymbol{u}, \hat{\boldsymbol{u}}) & \le & 0 \\ \boldsymbol{u} \in \bigcap\limits_j \mathcal{U}_j \end{cases}.
\end{aligned} \tag{17}$$

Let $\mathrm{ri}(\cdot)$ denote the relative interior of a set, then the parametric convex subproblem $\mathcal{Q}^K$ has the following properties.

Proposition 2 (Convex subproblem) $\forall \hat{\boldsymbol{u}} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$*, the convex subproblem* $\mathcal{Q}^K(\hat{\boldsymbol{u}})$ *of the form* (17) *possesses the following properties:*

1) $\mathcal{D}(\mathcal{Q}^K(\hat{\boldsymbol{u}}))$ *is nonempty and* $\mathcal{D}(\mathcal{Q}^K(\hat{\boldsymbol{u}})) \subseteq \mathcal{D}(\mathcal{P}^K)$*;*
2) $\mathcal{Q}^K(\hat{\boldsymbol{u}})$ *is computationally tractable.*

Proof: This result is straightforward by combining the proof of Proposition 1 and convex analysis. First, given the inequality (23), the solution to $\mathcal{Q}^K$ remains feasible for $\mathcal{P}^K$. This proves the first part of the proposition. Next, since $\hat{\boldsymbol{u}}$ lies in the relative interior of the feasible set of $\mathcal{P}^K$, the convexified constraints admit a strictly feasible point. Therefore, $\mathcal{Q}^K(\hat{\boldsymbol{u}})$ satisfies Slater's condition [4], which implies strong duality and $\mathcal{Q}^K(\hat{\boldsymbol{u}})$ is computationally tractable. ∎

Note that the regularization term $\rho\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|^2$ in the cost provides strong convexity. This point admits a unique solution and ensures the algorithm convergence when the cost is independent of parts of $\boldsymbol{u}$. For notational convenience, we define the solution operator $\mathcal{S} : \mathbb{R}^{3\times K} \to \mathbb{R}^{3\times K}$ as

$$\mathcal{S}(\hat{\boldsymbol{u}}) \triangleq \arg\min_{\boldsymbol{u}} \mathcal{Q}^K(\hat{\boldsymbol{u}}).$$

This operator $\mathcal{S}$ possesses the following nontrivial properties.

Theorem 1 (Properties of $\mathcal{S}$) *The operator* $\mathcal{S}$ *satisfies the following properties:*

1) *(Stationary Point Preservation) Any fixed point of $\mathcal{S}$, i.e., any $\boldsymbol{u}^*$ satisfying $\boldsymbol{u}^* = \mathcal{S}(\boldsymbol{u}^*)$, is a stationary point of $\mathcal{P}^K$.*
2) *(descent Inequality) $\rho\|\hat{\boldsymbol{u}} - \mathcal{S}(\hat{\boldsymbol{u}})\|^2 \leq g(\hat{\boldsymbol{u}}) - g(\mathcal{S}(\hat{\boldsymbol{u}}))$.*
3) *(Sustainability) For all $\hat{\boldsymbol{u}} \in ri(\mathcal{D}(\mathcal{P}^K))$, if $\hat{\boldsymbol{u}}$ is not a fixed point of $\mathcal{S}$, then $\mathcal{S}(\hat{\boldsymbol{u}}) \in ri(\mathcal{D}(\mathcal{P}^K))$.*

Proof: See appendix B.

Stationary point preservation in Theorem 1 shifts the objective from directly solving $\mathcal{P}^K$ to finding fixed points of the operator $\mathcal{S}$. However, solving the fixed-point equation explicitly remains difficult. The descent inequality implies that $\mathcal{S}$ produces an improved solution from any given $\hat{\boldsymbol{u}}$. This observation naturally motivates the use of operator iteration to approximate a fixed point of $\mathcal{S}$.

C. Algorithm Design and Analysis

The sustainability property in Theorem 1 allows convex subproblems to be constructed repeatedly from an constant $\hat{\boldsymbol{u}} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$. This leads to a sequence of progressively improved solutions. However, the descent inequality alone does not provide convergence guarantees. Therefore, this subsection examines the convergence behavior of the resulting sequence.

Given an initial parameter $\boldsymbol{u}_{(0)} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$, define the sequence $\{\boldsymbol{u}_{(m)}\}_{m=0}^{+\infty}$ by

$$\boldsymbol{u}_{(m+1)} = \mathcal{S}(\boldsymbol{u}_{(m)}).$$

This sequence satisfies the following properties:

Theorem 2 (Properties of $\{\boldsymbol{u}_{(m)}\}$) *The sequence $\{\boldsymbol{u}_{(m)}\}$ exhibits the following properties:*

1) *(Iterative Convergence) $\{\boldsymbol{u}_{(m)}\}$ converges to stationary points of $\mathcal{P}^K$.*
2) *(Monotonic Improvement) Before reaching a stationary point, the sequence $\{g(\boldsymbol{u}_{(m)})\}$ decreases monotonically.*

Proof: See Appendix C.

The feasibility of recursively constructing convex subproblems to solve $\mathcal{P}^K$ has been established. Moreover, the second part of Theorem 2 establishes an anytime property: each solved convex subproblem yields a feasible solution, which is no worse than its predecessor. Therefore, even under stringent computational budgets, one should take as many iterations as possible, and every iteration guarantees improvement.

Note that finding an initial solution $\boldsymbol{u}_{(0)} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$ constitutes the final algorithmic step. However, constraint sets $\mathcal{U}_j$ exhibit considerable diversity. This makes universal initialization methods elusive. We propose the following two initialization strategies that both satisfy bilinear constraints:

- **Static initialization**: set $\boldsymbol{u}_{(0)} = \boldsymbol{0}$.
- **Warm startup**: if $\boldsymbol{u}^-$ is feasible for $\mathcal{P}^{K-1}$, construct $\boldsymbol{u}_{(0)} = (\boldsymbol{u}^-, \boldsymbol{0})$.

Finally, based on the above analysis, we summarize the whole solving procedures as Algorithm 1. Since each subproblem is convex and belongs to the class P, the output of the algorithm can be computed in polynomial time. Moreover, Algorithm 1 possesses anytime feasibility, which is well suited for real-time applications.

Algorithm 1 Trajectory optimizer for SWMRs

Input: Current velocity ν^0.
Initialize $\boldsymbol{u}_{(0)} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$.
Set $m = 0$.
while $m <$ max iterations **do**
  Construct convex subproblem $\mathcal{Q}^K(\boldsymbol{u}_{(m)})$ via (17).
  Solve $\boldsymbol{u}_{(m+1)} = \mathcal{S}(\boldsymbol{u}_{(m)})$ with a convex solver.
  if $\|\boldsymbol{u}_{(m)} - \boldsymbol{u}_{(m-1)}\| <$ certain threshold **then**
    Terminate iteration.
  end if
  Update $m \leftarrow m + 1$.
end while
Compute desired velocity $\nu^1 = \nu^0 + u^1$.
Compute wheel speeds z_n via (4).
Output: Desired wheel velocities $z_1, \ldots, z_n$.

IV. Numerical Validations

The proposed approach consists of two components: the formulation and the solution algorithm. In this section, we use numerical experiments to comprehensively demonstrate the advantages of our algorithm over alternative approaches.

A. Experimental Setup

We evaluate the proposed approach in MATLAB on a trajectory tracking task, which provides clear and interpretable results. Wheel velocities and steering rates are appropriately bounded to test the handling of actuation constraints.

The reference trajectory is a planar Bézier curve defined by five control points $(0, 0)$, $(0.75, 0.25)$, $(0.25, 0.75)$, $(1.25, -1)$, and $(1, 0)$. The trajectory duration is 1 s, and reference states are sampled uniformly with respect to the Bézier parameter. A desired orientation profile is imposed to increase tracking difficulty, where the heading angle increases from 0 to 2π proportionally to the square of the normalized arc length.

The discretization interval is 0.01 s, and replanning is performed at each time step. The prediction horizon for each approach is set to $K = 6$. Four wheels are mounted at the vertices of a 0.655 m × 0.335 m rectangle centered at the chassis, consistent with the *iMoro* platform parameters in [18]. The steering rate and wheel velocity are limited to 5π rad/s and 5 m/s, respectively.

B. Application Details of Algorithm 1

1) Cost Function: Let t_0 denote the start time of the algorithm. The reference state at step k is $\xi_r^k \triangleq \xi_w(t_0 + k\tau)$, and the corresponding tracking error is defined as $e^k \triangleq \xi_w(t_0 + k\tau) - \xi_r^k$. The tracking cost $g(\boldsymbol{u})$ is defined as

$$g(\boldsymbol{u}) = \sum_{k=1}^{K} (e^k)^\top Q e^k + (u^k)^\top R u^k, \tag{18}$$

with $Q = \mathrm{diag}(30, 30, 1)$ and $R = \mathrm{diag}(0.3, 0.3, 0.3)$. Due to the rotational relationship (1), the cost function is nonconvex.

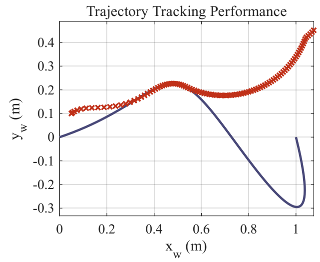
Applying a first-order expansion on $\mathbb{SO}(2)$ [24] yields

$$e^k = \xi_w(t_0) + \begin{bmatrix} \mathcal{R}(\psi_w(t_0)) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \sum_{l=0}^{k-1} \nu^l \tau - \xi_r^k + \mathcal{O}(\tau^2). \quad (19)$$

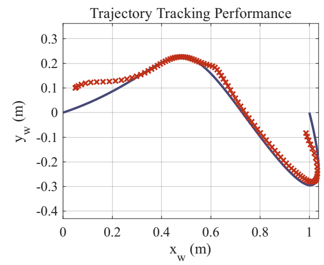
2) The Optimization Problem: In addition to the steering-rate constraint, a maximum wheel speed constraint (20b) is imposed on each wheel at every time step. The optimization problem used in the simulation is:

$$\begin{aligned} \mathcal{P}^{\text{sim}}: \quad & \min_{\boldsymbol{u} \in \mathbb{R}^{3\times K}} \; g(\boldsymbol{u}) \text{ in (18)} \\ \text{s.t.} \quad & {\nu^{k-1}}^\top H_n^\top \mathcal{R}_i^\top H_n(\nu^{k-1} + u^k) \geq 0, && \text{(20a)} \\ & \|H_n \nu^k\| \leq 5\,\text{m/s}, && \text{(20b)} \\ & \nu^{k+1} = \nu^k + u^{k+1},\; k < K. && \text{(20c)} \end{aligned}$$

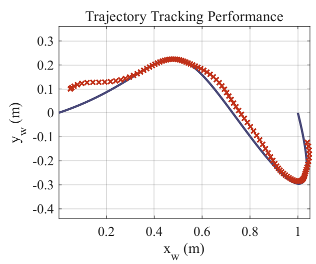
3) Solver: Equation (19) shows that the cost $g(\boldsymbol{u})$ is a quadratic form. The constraints $C_{i,n}^k$ are quadratic inequalities. In addition, the output velocity constraint (20b) can also be written as a quadratic inequality. Therefore, the resulting problem is a quadratic constrained quadratic programming problem. We use MATLAB CVX [16] as a modeling interface and solve the problem with the Embedded Conic Solver (ECOS) [12]. ECOS is used with its default parameters, and the maximum number of iterations in Algorithm 1 is set to 3.



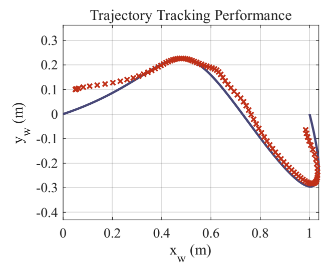
(a) Active-set Algorithm



(b) Interior-point Algorithm



(c) e-LMPC



(d) Proposed Approach with 3 Iterations

Fig. 3: Trajectory tracking results of the evaluated approaches. In subsequent trajectory plots, the reference trajectory is denoted by —, and the simulation results are denoted by ×.

C. Baseline Approaches

Although we aim to compare with various MPC-based approaches to demonstrate the advantages of the proposed approach, there are few mature planning approaches in the SWMR domain. We therefore select two general-purpose NLP solvers and one SWMR-tailored approach as baselines:

1) $\mathcal{P}^K$ solved by an active-set algorithm;
2) $\mathcal{P}^K$ solved by an interior-point algorithm;
3) An extended version of LMPC proposed by Liu et al. [17], hereafter referred to as e-LMPC.

The first two are available in MATLAB's `fmincon` and can be directly invoked. After extending LMPC to support constraint handling, e-LMPC can be implemented as a warm-started Sequential Quadratic Programming algorithm with at most one iteration; we also implement it using `fmincon`.

D. Results

The trajectory tracking results for all four approaches are illustrated in Fig. 3. Except for the active-set algorithm, all compared approaches are able to guide the SWMR to track the given trajectory. We next systematically evaluate the performance of these feasible approaches.

1) Constraint Satisfaction: As shown in Fig. 5, although the interior-point algorithm correctly handles the convex constraint in (20b), it fails to satisfy the constraint in (20a). In contrast, the two SWMR-tailored approaches exhibit good constraint satisfaction across the whole trajectory.

Note that such violations destroy the fundamental perfect-alignment condition defined in (4). Consequently, the complex nonlinear vehicle dynamics [3] will dominate and drive the robot in unpredictable directions. Thus, the seemingly smooth predicted trajectories generated by the interior-point algorithm are essentially not executable on physical hardware.

2) Computational Cost: For robotic planning algorithms, real-time capability is a key metric. In this section, we compare the proposed approach with the interior-point algorithm and e-LMPC based on their characteristics. Notably, the proposed approach allows iterative refinement, and the number of iterations affects the computational cost. Therefore, we report timings using a single iteration.

All simulations are conducted on a laptop equipped with an AMD Ryzen 7940HS processor. To avoid initialization overhead of the solver, a few initial outliers are excluded from the reported statistics.

Fig. 4 presents box plots of the computation time for each approach. The interior-point algorithm requires significantly longer time to solve $\mathcal{P}^K$ (median: 0.4802 s), which exceeds the acceptable range for real-time applications. In contrast, e-LMPC (median: 8.486 ms) and the proposed approach (median: 8.306 ms) have comparable computational cost, both

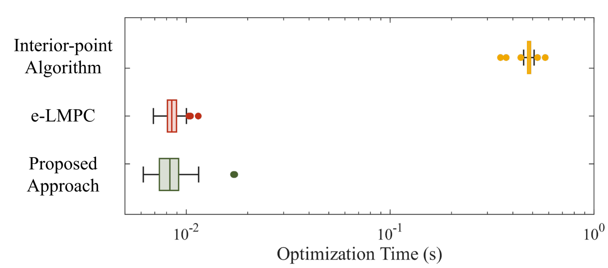
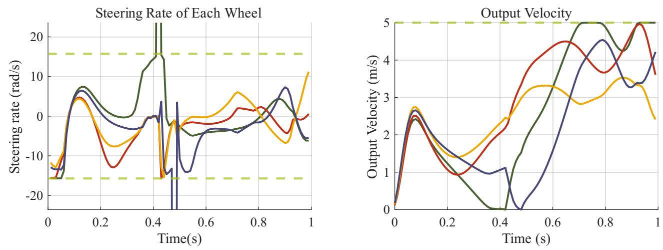
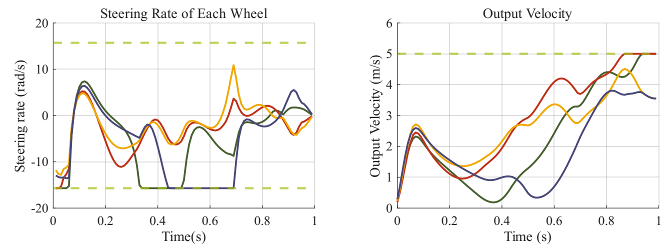


Fig. 4: Box plots of the computation time for each algorithm.

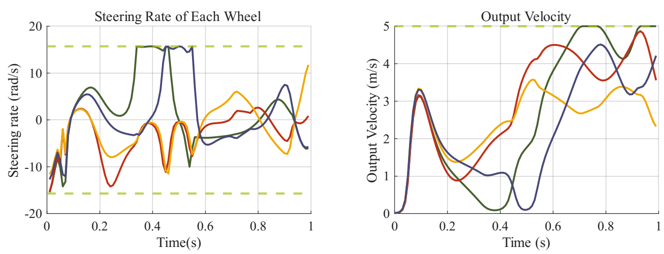
enabling over 100 solves per second. This indicates that both approaches are suitable for real-time deployment.



(a) Interior-point Algorithm



(b) e-LMPC



(c) Proposed Approach

Fig. 5: Constraint satisfaction comparison between the evaluated approaches. In subsequent constraint plots, the wheel–color correspondence is: — front-left, — rear-left, — front-right, — rear-right. The constraint is denoted by - -.

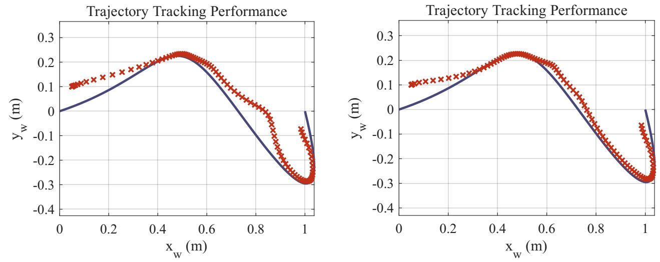
3) Trajectory Quality: For the constraint-satisfying approaches (e-LMPC and the proposed approach), the resulting trajectories are shown in Fig. 3c and Fig. 6a. When the two approaches appear visually comparable, a quantitative analysis is necessary rather than relying on qualitative inspection. For fairness, the proposed approach is evaluated without iterative refinement, consistent with the computational cost comparison.

We adopt the total trajectory cost as the metric:

$$\sum_{k=1}^{100} (e^k)^\top Q e^k + (u^k)^\top R u^k, \tag{21}$$

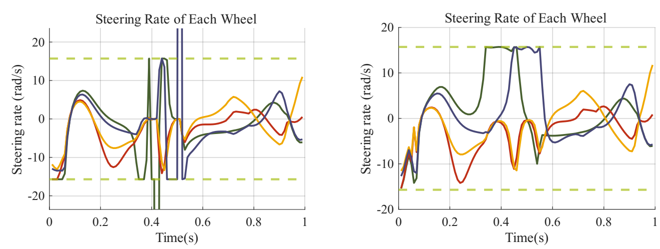
where Q and R are identical to those in (18). This metric evaluates multi-step planning performance from the perspective of trajectory optimization.

Under this metric, e-LMPC yields a cost of 42.90. In contrast, the proposed approach achieves a lower cost of 25.16 without refinement. This demonstrates that, under comparable computational cost, the proposed approach provides fundamentally superior tracking accuracy compared to e-LMPC.



(a) Single iteration (b) Three iterations

Fig. 6: Refinement of the proposed approach over multiple iterations in the trajectory tracking task.



(a) LMPC for 2 iterations (b) Proposed approach

Fig. 7: Severe constraint violations observed when running LMPC for 2 iterations.

4) Refinability: Theorem 2 endows the proposed approach with the ability for iterative refinement. With additional computational resources, the algorithm can progressively improve the result. As shown in the figure, after two additional refinement iterations, the proposed approach reduces the metric in (21) from 25.16 to 13.36, yielding a higher-quality trajectory.

In contrast, e-LMPC lacks such rigorous theoretical guarantees. As a result, even allowing one additional iteration may lead to violations of the constraint in (20a), as demonstrated in Fig. 7. This makes its solution effectively one-shot; even with more computational resources, it is difficult to obtain improved results with additional iterations.

V. Conclusions

In this paper, we developed a versatile and fast optimization approach for the motion planning of SWMRs. To enable real-time computation, we proposed a tailored fast solution algorithm that iteratively contracts the non-convex feasible region while preserving feasibility at every iteration. Theoretical analysis established descent, convergence to stationary points, and anytime feasibility of the proposed approach. Simulation results on trajectory tracking tasks demonstrated that the proposed approach achieves strict constraint satisfaction with significantly reduced computational cost compared to general-purpose NLP solvers. These results indicate that the proposed approach provides a practical and scalable solution for real-time agile motion planning of SWMRs.

Future work includes establishing the convergence rate of the sequence $\{\boldsymbol{u}_{(m)}\}$ and developing more efficient solvers for the resulting subproblems.

TABLE II: Comprehensive Comparison of Evaluated Approaches

Approach	**Feasibility**	**Median Computation Cost**	**Constraints Satisfaction**	**Cost Metric** (21)	**Refinability**
Active-set	×	–	–	–	–
Interior-point	✓	0.4802 s	×	–	–
e-LMPC	✓	8.486 ms	✓	42.90	×
Proposed	✓	**8.306 ms**	✓	**25.16**	✓

Appendix

A. Proof of Proposition 1

Proof: First, based on the definitions of $A_{i,n}^k(\cdot)$, $B_{i,n}^k(\cdot)$ in (16), the convex-concave decomposition of the original constraint (13) is given by

$$A_{i,n}^k(\boldsymbol{u}) - B_{i,n}^k(\boldsymbol{u}) \leq 0, \quad i \in \{1,2\}. \tag{22}$$

Next, we construct a convex tightening of such constraints. For a concave matrix-valued function, we employ first-order expansion on (22) and obtain an upper approximation version, given by

$$\begin{aligned} A_{i,n}^k(\boldsymbol{u}) - B_{i,n}^k(\boldsymbol{u}) \leq A_{i,n}^k(\boldsymbol{u}) &- B_{i,n}^k(\hat{\boldsymbol{u}}) \\ &- \underbrace{\left\langle \nabla_{\boldsymbol{u}} B_{i,n}^k(\boldsymbol{u})|_{\boldsymbol{u}=\hat{\boldsymbol{u}}}, \boldsymbol{u}-\hat{\boldsymbol{u}} \right\rangle}_{L_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}})}, \end{aligned} \tag{23}$$

Since $B_{i,n}^k$ is in a quadratic form, hence $L_{i,n}^k$ is linear in $\boldsymbol{u}$. By the Frobenius inner product definition, this linear term can be computed in block form:

$$L_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}}) = \sum_{l=1}^{k} (\nabla_{u^l} B_{i,n}^k|_{\boldsymbol{u}=\hat{\boldsymbol{u}}})^\top (u^l - \hat{u}^l). \tag{24}$$

Consider the denominator layout for matrix derivatives, and by the chain rule, the gradient of $B_{i,n}^k$ with respect to u^l is:

$$\nabla_{u^l} B_{i,n}^k = \frac{\partial B_{i,n}^k}{\partial b_{i,n}^k} \frac{\partial b_{i,n}^k}{\partial u^l} = b_{i,n}^k (\nabla_{u^l} b_{i,n}^k), \tag{25}$$

where the gradient element of $b_{i,n}^k$ is given by

$$\nabla_{u^l} b_{i,n}^k = \begin{cases} (I+R_i)H_n & l<k \\ R_i H_n & l=k \\ 0 & l>k \end{cases}.$$

Through the above calculations, the linear term $L_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}})$ can be written as

$$\begin{aligned} L_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}}) &= \left((I+R_i)H_n \hat{\nu}^{k-1} + R_i H_n \hat{u}^k\right)^\top \cdot \\ &\left(\sum_{l=1}^{k-1} (I+R_i)H_n \left(u^l - \hat{u}^l\right) + R_i H_n (u^k - \hat{u}^k) \right), \end{aligned}$$

where $\{\hat{\nu}^k\}$ is defined by $\hat{\nu}^{k+1} = \hat{\nu}^k + \hat{u}^{k+1}$, $k<K$.

Notice that the right side of (23) is exactly the value of $C_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}})$, and is a convex function with respect to $\boldsymbol{u}$. The proof is completed. ■

B. Proof of Theorem 1

We prove the three claims separately.

1) Stationary Point Preservation: The Karush–Kuhn–Tucker (KKT) conditions of the convex subproblem $\mathcal{Q}^K$ are given by

$$\begin{cases} \nabla_{\boldsymbol{u}} g(\boldsymbol{u}) + \rho(\boldsymbol{u}-\hat{\boldsymbol{u}}) + \sum_{i,n,k} \lambda_{i,n}^k \nabla_{\boldsymbol{u}} C_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}}) = 0 \\ C_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}}) \leq 0 \\ \lambda_{i,n}^k \geq 0, \quad \forall i,n,k \\ \lambda_{i,n}^k C_{i,n}^k(\boldsymbol{u},\hat{\boldsymbol{u}}) = 0 \end{cases}. \tag{26}$$

By definition, $\boldsymbol{u}^*$ is the unique minimizer of $\mathcal{Q}^K(\boldsymbol{u}^*)$. Therefore, the KKT conditions (26) hold at $\boldsymbol{u}=\hat{\boldsymbol{u}}=\boldsymbol{u}^*$.

By construction, the convexified constraints $C_n^k(\boldsymbol{u},\hat{\boldsymbol{u}})$ satisfy

$$C_n^k(\boldsymbol{u},\hat{\boldsymbol{u}})|_{\boldsymbol{u}=\hat{\boldsymbol{u}}} = A_{i,n}^k(\hat{\boldsymbol{u}}) - B_{i,n}^k(\hat{\boldsymbol{u}}).$$

Consequently, when $\boldsymbol{u}=\hat{\boldsymbol{u}}=\boldsymbol{u}^*$, the KKT conditions (26) reduce to

$$\begin{cases} \nabla g(\boldsymbol{u}^*) + \sum_{i,n,k} \lambda_{i,n}^k \left(\nabla A_{i,n}^k(\boldsymbol{u}^*) - \nabla B_{i,n}^k(\boldsymbol{u}^*)\right) = 0 \\ A_{i,n}^k(\boldsymbol{u}^*) - B_{i,n}^k(\boldsymbol{u}^*) \leq 0 \\ \lambda_{i,n}^k \geq 0, \quad \forall i,n,k \\ \lambda_{i,n}^k \left(A_{i,n}^k(\boldsymbol{u}^*) - B_{i,n}^k(\boldsymbol{u}^*)\right) = 0 \end{cases}.$$

The above system coincides exactly with the KKT conditions of the original problem $\mathcal{P}^K$. Hence, $\boldsymbol{u}^*$ satisfies the first-order necessary optimality conditions of $\mathcal{P}^K$, and is therefore a stationary point of $\mathcal{P}^K$. This completes the proof of the first part of the theorem.

2) Descent Inequality: Let $\hat{\boldsymbol{u}} \in \mathrm{ri}(\mathcal{D}(\mathcal{P}^K))$ be arbitrary and denote $s \triangleq \mathcal{S}(\hat{\boldsymbol{u}})$. Since $\boldsymbol{s}$ minimizes the strongly convex objective of $\mathcal{Q}^K(\hat{\boldsymbol{u}})$ and g is convex, the following inequality holds:

$$g(\hat{\boldsymbol{u}}) - g(\boldsymbol{s}) \geq \nabla g(\boldsymbol{s})^\top (\hat{\boldsymbol{u}} - \boldsymbol{s}).$$

Next, taking the first KKT condition of (26) evaluated at $\boldsymbol{u}=\boldsymbol{s}$, and multiplying both sides by $(\hat{\boldsymbol{u}}-\boldsymbol{s})$, we obtain

$$g(\hat{\boldsymbol{u}}) - g(\boldsymbol{s}) + E(\hat{\boldsymbol{u}},\boldsymbol{s}) \geq \rho \|\hat{\boldsymbol{u}}-\boldsymbol{s}\|^2, \tag{27}$$

where

$$E(\hat{\boldsymbol{u}},\boldsymbol{s}) = \Big(\sum_{i,n,k} \lambda_{i,n}^k \left(\nabla A_{i,n}^k(\boldsymbol{s}) - \nabla B_{i,n}^k(\hat{\boldsymbol{u}})\right) \Big)^\top (\hat{\boldsymbol{u}}-\boldsymbol{s}). \tag{28}$$

The remainder of the proof hinges on proving $E(\hat{\boldsymbol{u}},\boldsymbol{s}) \leq 0$.

By the convexity of the functions ${}^iA_n^k$ and ${}^iB_n^k$, the following inequalities hold:

$$\begin{cases} A_{i,n}^k(\hat{\boldsymbol{u}}) - A_{i,n}^k(\boldsymbol{s}) \geq & \nabla A_{i,n}^k(\boldsymbol{s})^\top(\hat{\boldsymbol{u}} - \boldsymbol{s}) \\ B_{i,n}^k(\hat{\boldsymbol{u}}) - B_{i,n}^k(\boldsymbol{s}) \geq & \nabla B_{i,n}^k(\hat{\boldsymbol{u}})^\top(\boldsymbol{s} - \hat{\boldsymbol{u}}) \end{cases}. \tag{29}$$

Since ${}^i\lambda_n^k \geq 0$, it follows that

$$E(\hat{\boldsymbol{u}}, \boldsymbol{s}) \leq \sum_{i,n,k} \lambda_{i,n}^k \left(A_{i,n}^k(\hat{\boldsymbol{u}}) - B_{i,n}^k(\hat{\boldsymbol{u}}) - \left(A_{i,n}^k(\boldsymbol{s}) - B_{i,n}^k(\boldsymbol{s}) \right) \right).$$

By the complementary slackness condition of the KKT system ($\lambda_{i,n}^k \left(A_{i,n}^k(\boldsymbol{s}) - B_{i,n}^k(\boldsymbol{s}) \right) = 0$), and the feasibility of $\hat{\boldsymbol{u}}$ ($A_{i,n}^k(\hat{\boldsymbol{u}}) - B_{i,n}^k(\hat{\boldsymbol{u}}) \leq 0$), we conclude that $E(\hat{\boldsymbol{u}}, \boldsymbol{s}) \leq 0$. Therefore, the descent inequality holds:

$$g(\hat{\boldsymbol{u}}) - g(\boldsymbol{s}) \geq \rho \|\hat{\boldsymbol{u}} - \boldsymbol{s}\|^2.$$

3) Sustainability: Since $B_{i,n}^k$ is strongly convex, the equality in (23) holds iff $\hat{\boldsymbol{u}} = \mathcal{S}(\hat{\boldsymbol{u}})$. Therefore, when $\hat{\boldsymbol{u}}$ is not a fixed point, the inequality in (23) is strict. This implies that any feasible solution of $\mathcal{Q}^K$ lies in the relative interior of $\mathcal{D}(\mathcal{P}^K)$. The proof is completed. ■

C. Proof of Theorem 2

Proof: We prove the two claims separately.

1) Convergence of Stationary Points: First, Theorem 1 yields the descent inequality

$$\rho \|\boldsymbol{u}_{(m)} - \boldsymbol{u}_{(m+1)}\|^2 \leq g(\boldsymbol{u}_{(m)}) - g(\boldsymbol{u}_{(m+1)}).$$

Since g is bounded from below, summing over m gives

$$\sum_{m=0}^{\infty} \|\boldsymbol{u}_{(m)} - \boldsymbol{u}_{(m+1)}\|^2 \leq \frac{g(\boldsymbol{u}_{(0)}) - \min g}{\rho} < +\infty,$$

which implies $\|\boldsymbol{u}_{(m)} - \boldsymbol{u}_{(m+1)}\| \to 0$. This means the descent step size converges.

Second, since $\boldsymbol{u}_{(m+1)} = \mathcal{S}(\boldsymbol{u}_{(m)})$, we have

$$f(\boldsymbol{u}_{(m+1)}, \boldsymbol{u}_{(m)}) \leq f(\boldsymbol{u}_{(m)}, \boldsymbol{u}_{(m)}).$$

The function f is strongly convex, thus its $f(\boldsymbol{u}_{(m)}, \boldsymbol{u}_{(m)})$-lower level sets are bounded. Therefore $\boldsymbol{u}_{(m+1)}$ is bounded. By the Bolzano-Weierstrass theorem [21], the sequence $\{\boldsymbol{u}_{(m)}\}$ has at least one accumulation point.

Finally, an accumulation point $\boldsymbol{u}^*$ satisfies that there exists a subsequence $\{\boldsymbol{u}_{(m_s)}\} \to \boldsymbol{u}^*$, where the index of subsequence $\{m_s\} \in \mathbb{N}$ is strictly increasing [13]. Using the triangle inequality

$$\|\boldsymbol{u}_{(m_s+1)} - \boldsymbol{u}^*\| \leq \|\boldsymbol{u}_{(m_s+1)} - \boldsymbol{u}_{(m_s)}\| + \|\boldsymbol{u}_{(m_s)} - \boldsymbol{u}^*\|$$

and the fact that both terms on the right side approach zero, we obtain another converged subsequence $\{\boldsymbol{u}_{(m_s+1)}\} \to \boldsymbol{u}^*$.

We then introduce the required mathematical tool. This is a tailored version of Berge's theorem [1, 17.31], retaining only essential components.

Lemma 1 (Berge's Maximum Theorem) *Consider the following parametric problem with variable $\boldsymbol{p} \in \mathbb{R}^p$ and parameter $\boldsymbol{q} \in \mathbb{R}^q$:*

$$\arg\min_{\boldsymbol{p} \in \mathcal{C}(\boldsymbol{q})} F(\boldsymbol{p}, \boldsymbol{q}),$$

where the sequences $\boldsymbol{q}_{(m)} \to \boldsymbol{q}^*$ and $\boldsymbol{p}_{(m)} \to \boldsymbol{p}^*$ hold. If the following hold:

1) *F is continuous in $(\boldsymbol{p}, \boldsymbol{q})$;*
2) *The feasible set $\mathcal{C}(\cdot)$ is nonempty and compact-valued;*
3) *$\mathcal{C}$ is upper semicontinuous: $\boldsymbol{p}_{(m)} \in \mathcal{C}(\boldsymbol{q}_{(m)})$ implies $\boldsymbol{p}^* \in \mathcal{C}(\boldsymbol{q}^*)$,*

then $\boldsymbol{p}^* \in \arg\min_{\boldsymbol{p} \in \mathcal{C}(\boldsymbol{q}^*)} F(\boldsymbol{p}, \boldsymbol{q}^*)$.

Now we prove that Lemma 1 is applicable to problem $\mathcal{Q}^K$: i) The cost $f(\boldsymbol{u}, \hat{\boldsymbol{u}})$ is continuous in both variable and parameter. ii) Each convex component of the constraints is closed. Some components are compact. Hence the overall feasible set is compact. iii) The parameter-dependent terms $C_{i,n}^k$ also satisfy upper semicontinuity. Therefore, Lemma 1 applies.

Let $\boldsymbol{p}_{(m)}$ and $\boldsymbol{q}_{(m)}$ in Lemma 1 be $\boldsymbol{u}_{(m_s+1)}$ and $\boldsymbol{u}_{(m_s)}$ respectively. By applying the Lemma, every accumulation point of $\{\boldsymbol{u}_{(m)}\}$ satisfies $\boldsymbol{u}^* \in \mathcal{S}(\boldsymbol{u}^*)$. Then by the Stationary Point Preservation property in Theorem 1, each accumulation point of $\{\boldsymbol{u}_{(m)}\}$ is a stationary point of $\mathcal{P}^K$. This concludes the first part of the proof.

2) Monotonic Improvement: Since $\boldsymbol{u}_{(m+1)}$ is the optimal solution to the subproblem $\mathcal{Q}^K(\boldsymbol{u}_{(m)})$, we have

$$\begin{aligned} g(\boldsymbol{u}_{(m)}) &= g(\boldsymbol{u}_{(m)}) + \rho \|\boldsymbol{u}_{(m)} - \boldsymbol{u}_{(m)}\|^2 \\ &= f(\boldsymbol{u}_{(m)}, \boldsymbol{u}_{(m)}) \\ &> f(\boldsymbol{u}_{(m+1)}, \boldsymbol{u}_{(m)}) \\ &= g(\boldsymbol{u}_{(m+1)}) + \rho \|\boldsymbol{u}_{(m+1)} - \boldsymbol{u}_{(m)}\|^2 \\ &> g(\boldsymbol{u}_{(m+1)}), \end{aligned}$$

whenever $\boldsymbol{u}_{(m+1)} \neq \boldsymbol{u}_{(m)}$. This implies that the sequence $\{g(\boldsymbol{u}_{(m)})\}$ is strictly decreasing before convergence, which establishes the second part of the theorem. ■

D. Supplementary Simulation Results

Due to space limitations, the rationale behind the numerical setup in (20) is provided in the supplementary material. In addition, we further evaluate the proposed approach under different wheel numbers and configurations, as well as different types of cost functions.

References

[1] Charalambos D Aliprantis and Kim C Border. *Infinite dimensional analysis: a hitchhiker's guide*. Springer, 2006.

[2] Runjiao Bao, Yongkang Xu, Lin Zhang, Haoyu Yuan, Jinge Si, Shoukun Wang, and Tianwei Niu. Deep reinforcement learning-based trajectory tracking framework for 4ws robots considering switch of steering modes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[3] Karl Berntorp, Björn Olofsson, and Anders Robertsson. Path tracking with obstacle avoidance for pseudo-omnidirectional mobile robots using convex optimization. In *American Control Conference (ACC)*, 2014.

[4] Stephen Boyd. *Convex optimization*. Cambridge UP, 2004.

[5] Roger W Brockett et al. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1):181–191, 1983.

[6] Simon Chamberland, Éric Beaudry, Lionel Clavien, Froduald Kabanza, François Michaud, and Michel Lauriay. Motion planning for an omnidirectional robot with steering constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[7] Ziyu Chen, Yang Liu, Wei He, Hong Qiao, and Haibo Ji. Adaptive-neural-network-based trajectory tracking control for a nonholonomic wheeled mobile robot with velocity constraints. *IEEE Transactions on Industrial Electronics*, 68(6):5057–5067, 2021.

[8] Lionel Clavien, Michel Lauria, and François Michaud. Instantaneous centre of rotation estimation of an omnidirectional mobile robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[9] Christian P Connette, Andreas Pott, Martin Hagele, and Alexander Verl. Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an icm representation in spherical coordinates. In *IEEE Conference on Decision and Control (CDC)*, 2008.

[10] Christian P Connette, Andreas Pott, Martin Hägele, and Alexander Verl. Addressing input saturation and kinematic constraints of overactuated undercarriages by predictive potential fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[11] Quoc Tran Dinh, Suat Gumussoy, Wim Michiels, and Moritz Diehl. Combining convex–concave decompositions and linearization approaches for solving bmis, with application to static output feedback. *IEEE Transactions on Automatic Control*, 57(6):1377–1390, 2011.

[12] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, 2013.

[13] James Dugundji. Topology allyn and bacon inc. *Boston, Mass*, 1966.

[14] François Ferland, Lionel Clavien, Julien Frémy, Dominic Létourneau, François Michaud, and Michel Lauria. Teleoperation of azimut-3, an omnidirectional non-holonomic platform with steerable wheels. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[15] Matthias Fuchs, Ch Borst, P Robuffo Giordano, Andreas Baumann, Erich Kraemer, Jörg Langwald, Robin Gruber, Nikolaus Seitz, Georg Plank, Klaus Kunze, et al. Rollin'justin-design considerations and realization of a mobile platform for a humanoid upper body. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[16] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. https://cvxr.com/cvx, 2014.

[17] Xinxin Liu, Wei Wang, Xuelong Li, Fusheng Liu, Zhihang He, Yanzhang Yao, Huaping Ruan, and Tong Zhang. Mpc-based high-speed trajectory tracking for 4wis robot. *ISA transactions*, 123:413–424, 2022.

[18] Reza Oftadeh, Reza Ghabcheloo, and Jouni Mattila. A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[19] Reza Oftadeh, Reza Ghabcheloo, and Jouni Mattila. Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[20] Hans B Pacejka and Egbert Bakker. The magic formula tyre model. *Vehicle System Dynamics*, 21(S1):1–18, 1992.

[21] John Petrovic. *Advanced calculus: theory and practice*. Chapman and Hall/CRC, 2020.

[22] Patrick Scheffe, Theodor Mario Henneken, Maximilian Kloock, and Bassam Alrifaee. Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing. *IEEE Transactions on Intelligent Vehicles*, 8(1):661–672, 2022.

[23] Ulrich Schwesinger, Cedric Pradalier, and Roland Siegwart. A novel approach for steering wheel synchronization with velocity/acceleration limits and mechanical constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[24] Junjie Shen and Dennis Hong. Convex model predictive control of single rigid body model on so (3) for versatile dynamic legged motions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[25] Mohamed Sorour, Andrea Cherubini, Philippe Fraisse, and Robin Passama. Motion discontinuity-robust controller for steerable mobile robots. *IEEE Robotics and Automation Letters*, 2(2):452–459, 2016.

[26] Christoph Stöger, Hubert Gattringer, and Andreas Müller. The virtual wheel concept for the singularity-free kinematics and dynamics modeling and control of pseudo-omnidirectional vehicles. *IEEE Robotics and Automation Letters*, 6(3):4798–4804, 2021.

[27] Luigi Tagliavini, Giovanni Colucci, Andrea Botta, Paride Cavallone, Lorenzo Baglieri, and Giuseppe Quaglia. Wheeled mobile robots: state of the art overview and kinematic comparison among three omnidirectional locomotion strategies. *Journal of intelligent & robotic systems*, 106(3):57, 2022.

[28] Benoit Thuilot, B d'Aandrea Novel, and Alain Micaelli. Modeling and feedback control of mobile robots equipped with several steering wheels. *IEEE Transactions on Robotics and Automation*, 12(3):375–390, 2002.

[29] Spyros G Tzafestas. *Introduction to mobile robot control*. Elsevier, 2013.

[30] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm

for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[31] Yizhi Wang, Degang Xu, and Yongfang Xie. Universal multi-mode kinematic controller for four-wheel independent steering and driving mobile robots. In *China Automation Congress (CAC)*, 2025.

[32] Yizhi Wang, Degang Xu, Yongfang Xie, Shuzhong Tan, Xianan Zhou, and Peng Chen. Hierarchical decision-making for autonomous navigation: Integrating deep reinforcement learning and fuzzy logic in four-wheel independent steering and driving systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[33] Wang Xi, Jiaming Guo, Chenyang Wang, Shukun Wu, and Jianping He. Trajectory planning for pseudo-omnidirectional vehicles: A near-optimal approach. In *IEEE Conference on Decision and Control (CDC)*, 2025.

[34] Lijun Xu, Yankun Yang, Qinhan Chen, Fengcheng Fu, Bihang Yang, and Lijian Yao. Path tracking of a 4wis-4wid agricultural machinery based on variable look-ahead distance. *Applied sciences*, 12(17):8651, 2022.

[35] Joseph Young. Optizelle: An open source software library designed to solve general purpose nonlinear optimization problems. https://www.optimojoe.com, 2016.

[36] Xiaolong Zhang, Yuanlong Xie, Liquan Jiang, Gen Li, Jie Meng, and Yu Huang. Trajectory tracking of a 4wis4wid robot using adaptive receding horizon control based on neurodynamics optimization. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019.